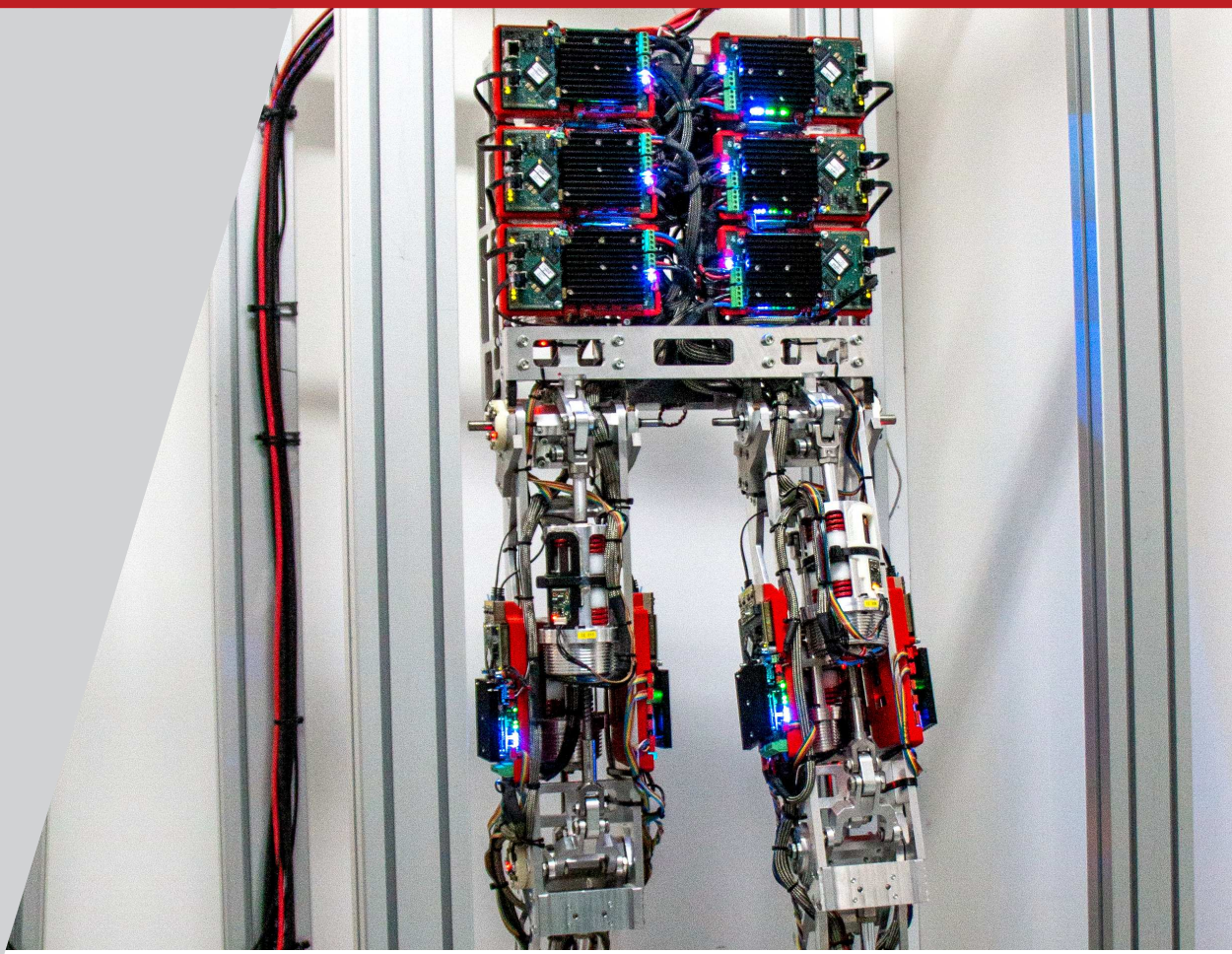
Rheinland-Pfälzische
Technische Universität
Kaiserslautern
Landau



Patrick Michael Vonwirth

Natural Control

Compliant Robot Control Featuring Neuromuscular Characteristics

RRLAB
THE ROBOTICS RESEARCH LAB

Natural Control

—

Compliant Robot Control Featuring Neuromuscular Characteristics

Thesis approved by
the Department of Computer Science

RPTU University Kaiserslautern-Landau

for the award of the Doctoral Degree

Doctor of Engineering (Dr.-Ing.)

to

Patrick Michael Vonwirth

Date of Defense:	December 11, 2025
Dean:	Prof. Dr. Christoph Garth
Head of Committee:	Prof. Dr. Stefanie Roos
Reviewer:	Prof. Dr. Dr. h.c. Karsten Berns
Reviewer:	Prof. Dr. Arne Rönnau
Reviewer:	Prof. Dr. Steffen Schütz

DE-386

To

the Reader

Preface

In their book *"Edison, His Life and Inventions"* Frank L. Dyer and Thomas C. Martin report the following about the research of Thomas A. Edison:

> *"In view of this immense amount of thought and labor, my sympathy got the better of my judgment, and I said: 'Isn't it a shame that with the tremendous amount of work you have done you haven't been able to get any results?' Edison turned on me like a flash, and with a smile replied:*
>
> ***'Results! Why, man, I have gotten a lot of results! I know several thousand things that won't work.'***
>
> *At that time he sent me out West on a special mission. On my return, a few weeks later, his experiments had run up to over ten thousand, but he had discovered the missing link in the combination sought for. "*
>
> [Dyer 10]

Research is neither a straightforward process nor is it reducible to its eventually visible results. Instead, it is a continuous repetition of failure, learning, retrials, small progress, and failure again. This thesis has been written, though it is meant to be read, in the above positive spirit of Edison.

In our modern world, problems have grown beyond the level of complexity that a single person could dare to solve alone. Thus, the work at hand offers only a glimpse into its underlying efforts and only a fraction of the *"ten thousand"* experiments that pave the path behind and in front of this thesis toward biological insight and natural robotics.

Though, in full acknowledgment of the *"immense amount of thought and labor"* that had been carried out by all my former colleagues, referenced and unreferenced authors in the bibliography, myself, as well as the ones who will then follow, this thesis honestly describes its results as they are and especially also does not spare with self-criticism and uninvestigated thoughts on future improvements.

Kaiserslautern, June, 2025
Patrick M. Vonwirth

Acknowledgements

I want to express my genuine gratitude to everyone who has been a part of my journey throughout this work's research. Whether we shared our moments on the ground, in the water, or high up in the air, your kindness, help, support, and encouragement became the silent backbone of this work. I honestly couldn't have done it without you!

A special shout-out goes to those who took the time to proofread this thesis. I'm incredibly grateful for your advice that you will find realized throughout this document in spellings, phrases, formulas, data, graphics, whole pages, and even entire chapter structures. Your contributions have not only elevated the readability and structure but have also added tremendous value to this work.

Very special thank you to all my incredible colleagues who accompanied and supported me throughout my unforgettable time at the RRLab!

First and foremost, a big shoutout to Atabak Nezhadfard and Steffen Schütz for pulling me into this amazing topic and passion for bio-inspired robot control, and for never letting me out of it again. I also want to extend this gratitude to my irreplaceable humanoid teammates, Sascha Steffens and Krzysztof Mianowski. Your technical expertise, unwavering support, and continuous advice have significantly formed the robotic systems we designed and built together throughout this work. None of them would have been possible without you.

As no robot runs without low-level embedded control, I'm glad to give a huge thank you to Alexander Köpper, not only for our numerous shared evenings at the dinner mensa, but also for your advice and for introducing me to logic programming. Under your professional instructions, I hope not to have raped my FPGAs.

As all research continues, I want to give a special thank you to Oleksandr Sivak. It has been a pleasure to introduce you to this exciting topic and to have had endless discussions about it, as well as not exactly about it. I wish you all the best on your own journey through bio-inspired bipedal robotics. May this work offer you helpful guidance and advice for your current and future research.

Keeping an entire research group digitally up and running is neither easy nor quickly done. However, it offers a glimpse into a world of IT hardware and services that usually remains hidden from most computer scientists. A shoutout to Steffen Hemer and Tobias Groll for introducing me to the unique secrets of the RRLab's IT infrastructure, and a special thanks to Dennis Meckel, who shared this administration with me. Dennis, it was truly an honor to work alongside you, especially during those moments when electricity failed, and only prayers kept our systems alive. To Jannis Egelhof, I wish you all the best as you take over those duties.

I also want to thank Ashita Ashok for our many years of shared lecture exercises, and Qazi Hamza Jan, not only for sharing our very first student project supervision but also for over half a decade of joint office. Hamza, it had been a fantastic time, and I don't want to have missed it. I wish you all the best, and may your future be filled with many more Easter eggs!

Special shout-outs go to Axel Vierling and Eike Gassen. Despite your small offices, you always cleared a chair for me and welcomed lengthy discussions, including but not limited to pure technical topics. Those chairs know I have used them, and I genuinely appreciate the time you dedicated to our chats.

Acknowledgement also goes to Maximilian Kunz and Christian Heupel, the initiators of the RRLab*ar*, as well as to all their best customers, whose names I respectfully spare herein. Those fun-filled evenings, breakfasts, quizzes, and countless other events we shared created fantastic memories and will always bring a smile to my face.

A big thank you goes to my Ph.D. supervisor, Karsten Berns. You offered me the opportunity to embrace all these experiences. While our research styles and philosophies didn't always align perfectly, your vast trust always allowed me the freedom to explore my own methodologies.

Lastly, I owe a great thanks to Mrs. Broschart. Your administrative, bureaucratic, and mental support accompanied me throughout my entire stay at the RRLab, literally, from the first contract to the final champagne.

Thanks to all of you!

Kaiserslautern, January, 2026
Patrick M. Vonwirth

In Memory of my Niece

Charlotte Aurelia

It seems to me you lived your life
like a candle in the wind.
Ever knowing who to cling to
when the rain set in.
And I would have liked to have known you
but you were just a kid.
Your candle burned out long before
your memory never will.

Abstract

Bipedal locomotion is a fascinating and versatile research area in both biology and robotics. Thereby, legged systems offer unique advantages in traversing unstructured terrain and integrating into human-made environments. However, even under undisturbed, ideal laboratory conditions, artificial natural walking is challenging to control.

Although recent developments have achieved remarkable results in stability and motion, today's artificial bipedal locomotion still falls short of natural capabilities. Deficits can be observed in agility, versatility, adaptability, and environmental integration, enabled by the compliant biological morphology. Modern control approaches, such as AI-enhanced Model Predictive Control (MPC), not only exceed the computational limits of today's CPUs but also struggle to stabilize compliant constructions. At the same time, biology demonstrates these outstanding control and motion capabilities while requiring only a fraction of the energy of artificial systems.

Thus, human gait serves as a perfect blueprint for studying natural control characteristics and for deriving novel robot control methodologies aimed at more natural robotics, specifically natural compliant control. At the hand of the human-shaped, and with Series Elastic Actuators (SEAs) fully compliant over-actuated bipedal robot Carl, this thesis defines and evaluates such a novel natural control philosophy.

Inspired by the human neuromuscular system, the concept of antagonistic muscular control is developed by implementing a low-level hybrid impedance control strategy on SEA hardware. By exploiting SEA hardware properties, the novel strategy is specifically designed to handle natural, fully compliant, over-actuated morphologies.

Beyond this bio-inspired SEA control strategy, the probability-based concept of Probabilistic Behavior Networks (PBNs) defines a versatile and inherently uncertain Behavior-Based (BB) high-level robot control architecture. The novel concept unifies control-relevant data representations into a single semantic calculus, thus enabling control decomposition and abstraction while maintaining general commutativity and associativity of individual control skills. Furthermore, the probabilistic semantics introduce control tolerance by design, thereby enabling intensional environmental adaptation.

The real-world functionality of all developed technologies is experimentally demonstrated. Thus, this thesis presents, for the first time, a novel and fully functional natural control system capable of reproducing natural control characteristics on the anthropomorphic bipedal robot Carl.

Thorough experimental evaluations highlight both novel control achievements and undisclosed soft- and hardware limitations. Detailed discussions provide recommendations for the consecutive development of the presented natural control philosophy.

Contents

1 Introduction **1**
- 1.1 Musculoskeletal Robot Control in the Context of Bio-Inspired Engineering 2
- 1.2 Natural Control – A Neuromuscular Control Hypothesis 5
- 1.3 Objectives and Thesis Outline 10

2 Natural Locomotion – A Robot's Characteristic Control Requirements **13**
- 2.1 Characteristic Natural Control Requirements of a Human Gait 16
  - 2.1.1 Hybrid Dynamics – Discrete Support Transitions 20
  - 2.1.2 Singular Positions – Locked Knees 22
  - 2.1.3 Variable Compliance – Motive, Passive, and Resistive Motions . . 25
  - 2.1.4 Nonlinear Relations – Linear Spring Leg Behavior 27
  - 2.1.5 Hierarchical Control – Neuromuscular Feedback Loops 28
  - 2.1.6 Model Reduction – SLIP-based Locomotion 34
- 2.2 Review of the Interdisciplinarity of Natural Control Engineering 37
- 2.3 Review of Bipedal Robots and Control Approaches 40
  - 2.3.1 Rigid Robot Control Approaches 41
  - 2.3.2 Compliant Robot Control Approaches 44
  - 2.3.3 Muscular Robot Control Approaches 46
- 2.4 A Robot Control System in the Context of Natural Control Requirements 48
- 2.5 Conclusion 50

3 Hybrid Impedance – A Robot's Musculoskeletal Control System **51**
- 3.1 Construction, Actuation, and Legged Locomotion Control Technologies 53
  - 3.1.1 Development of Carl– From Simulation to a Planar Lower Body 55
  - 3.1.2 The RRLabSEA – A Transparent Series Elastic Actuator 56
  - 3.1.3 The Human-like Morphology of Carl 60
- 3.2 Natural Impedance Control 63
  - 3.2.1 Centralized – Whole Body Dynamics Torque Control 66
  - 3.2.2 Distributed – Whole Body Actuator Impedance 67
  - 3.2.3 Hybrid – Antagonistic Muscular Control 71
  - 3.2.4 Interpretation of the Natural Musculoskeletal System 77
  - 3.2.5 Reduced Series Elastic Actuator Design 77
  - 3.2.6 Dynamics of Antagonistic Muscular Control 79
- 3.3 Distributed Impedance Generation 84

3.3.1 Work Space, Configuration Space, Actuation Space 84
3.3.2 Actuator-Individual Work Space Impedance 87
3.4 Optimization-Based Muscle Recruitment Strategies 89
3.4.1 Wrench Distribution 90
3.4.2 Damping Distribution 92
3.4.3 Capabilities and Limits of Distributed Damping 95
3.5 Experimental Results of Antagonistic Muscular Control on Carl 104
3.6 Conclusion 109

4 Probabilistic Behavior Networks – A Robot's Central Nervous System 111
4.1 Biologically and Technically Motivated Requirements 114
4.2 Behavior-Based Robot Control Decomposition 116
4.3 Definition of a Probabilistic Behavior Network as a Data Flow Computer 120
4.3.1 Commutativity and Associativity 125
4.3.2 Feedback, Loops, and Internal States 127
4.4 Unified Uncertain Data Representation 128
4.4.1 Cartesian Vectors 128
4.4.2 Rotor Orientations 129
4.4.3 Cartesian Symmetric Matrices 134
4.4.4 Combined Data Representation 139
4.4.5 Differentiation, Integration, and Abstraction 141
4.4.6 Data Series 143
4.5 Interfacing the Environment 146
4.5.1 Exponential Sensor Signal Precision Annotation 148
4.5.2 Probabilistic Integration of Control Data Series 149
4.6 Data Composition 155
4.6.1 Cartesian Vectors 155
4.6.2 Rotor Orientations 158
4.6.3 Combined Orientation and Cartesian Data 161
4.6.4 Derivatives and Data Series 161
4.7 Data Abstraction 162
4.7.1 Improper Sensory Data Projection 163
4.7.2 Proper Control Data Back Projection 166
4.8 Virtual 1D Position Control Example 170
4.9 Conclusion 173

5 Natural Squats – PBN-Based Hybrid Impedance Control on Carl 175
5.1 Antagonistic Muscular Control via Probabilistic Behavior Networks . . . 176
5.1.1 Responsiveness to Damping Approximation 178
5.1.2 Velocity to Force Optimization 182
5.1.3 Compliant Pose Control 185
5.1.4 Technical and Theoretical Limits of the Deployed System 187
5.2 Upright Balancing 188
5.2.1 Support Work Space 190
5.2.2 Ankle Strategy 192
5.2.3 Experimental Results of Ankle Strategy Balancing on Carl 195
5.3 Squat Motion via Linear Spring Leg Behavior 203
5.3.1 Linear Spring Legs 204
5.3.2 Experimental Results of Compliant Natural Squats on Carl . . . 206
5.4 Conclusion 214

6 Summary and Outlook **215**
6.1 Objective-Specific Contributions 216
6.2 Technical Support Work 218
6.3 Achieved and Demonstrated Natural Control Requirements on CARL . . 219
6.4 Discussion and Future Work Recommendations 222
6.5 Thesis Conclusion 223

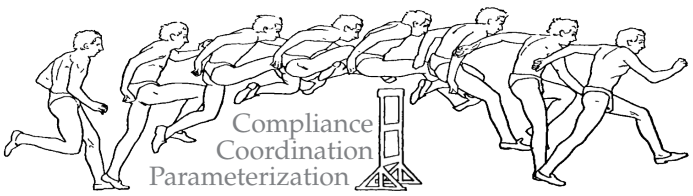
Bibliography **225**
Literature With Contribution of the Author 225
Regular Literature 228

Nomenclature **251**
Abbreviations and Names 251
Symbols and Functions 257

Index **259**

Curriculum Vitae **267**

1. Introduction



[1]

Bio-inspired robotics can look back on about 75 years of history and continuous scientific progression. Until the 1990s, *bio-inspiration* had already influenced robotic engineering in material, sensor- and actuation technologies, as well as robot morphology and foundational, mostly reflex-inspired control. In a 1994 summary paper, Bekey and Lewis discussed *"Biologically Inspired Robot Control"*, referencing back to the 1950s [Bekey 94]. In their work, they state that

> *"[. . .] in a number of practical cases it is possible to use inspiration from biology to design robot control systems. [. . .] Artificial neural networks and genetic algorithms have been used to enable these systems to learn [. . .]."*
>
> [Bekey 94, p. 382]

Despite being 30 years old, the above citation is still valid. However, significant progress has been achieved since then. A small collection of recent bio-inspired robots is, e.g., given in [Chellapurath 23]. Although Chellapurath et al. focus on environment-friendly robots, their studies are representative of the entire domain. They conclude that

> *"[. . .] advancements in design, materials, intelligence, and energy harvesting is leading to the rapid evolution of bioinspired robots. Specifically, improved locomotory capabilities allow them to overcome operational challenges in terrestrial, aquatic, aerial, and arctic environments. [. . .]"*
>
> [Chellapurath 23, p. 08]

Following that same trend, this thesis further contributes to these bio-inspired *"improved locomotory capabilities"*. Specifically, a novel musculoskeletal control is presented, capable of stably controlling a human-like, fully compliant, over-actuated robot.

[1] Graphic altered from the official report of the Olympic Games in Paris in the year 1900 [Ministere du commerce, de l'industrue des postes et des telegraphes 01, p. 393] (Public Domain).

Due to the entirely distributed and interconnected inbuilt elasticities, well-known and technically capable state-of-the-art control approaches such as Optimal Control (OC) or Model Predictive Control (MPC), so far, have failed to stabilize such a natural kinematic layout. Thus, all modern, yet commercially available, humanoids and other legged systems feature unnatural leg designs, including their inbuilt actuation principles. In order to overcome this state-of-the-art limitation, the herein-developed musculoskeletal control principle uses a Central Nervous System (CNS)- and muscle-inspired hierarchical hybrid impedance control architecture. Bio-inspired robotic engineering is a challenging interdisciplinary task. Thus, in order to benefit from all individually involved domains, the herein novel-presented concepts are both bio-inspired and, at the same time, technically reasoned.

Introduction Outline

In the following, Section 1.1 introduces the general field of bio-inspired robotic engineering. The section briefly introduces the herein-used robotic platform, highlights the control problems arising from its entirely compliant human-like morphology, and references the scope of this thesis within the overall development of that robot. Section 1.2, afterward, develops and defines this thesis' overall aim of so-called natural control, hypothesized in Theorem 1.1. The stated control goal is embedded in a description of the therefore required engineering process and references the herein-claimed novel natural control hypothesis against state-of-the-art approaches. Finally, Section 1.3 summarizes and outlines the achieved and herein-presented primary objectives and contributions of this work toward its overall aim of natural robot control.

1.1 Musculoskeletal Robot Control in the Context of Bio-Inspired Engineering

Bio-inspired engineering is spread over various engineering disciplines across all technical domains. Independent of their origin and initially intended interpretation, terms like *biomimetics* or *bionics* have crossed multiple domains, disciplines, and languages throughout the past centuries. During this process, their probably once-intended slight separation of definition has washed out, reinterpreted, mixed, and converged. Existing literature declares multiple origins as well as multiple interpretations for those terms, depending on the context and the language. However, independent of their domain, both terms consistently relate to *bio-inspiration* and bio-inspired engineering.

In their 2007 book *"Bioinspired Legged Locomotion: Models, Concepts, Control and Applications"*, Sharbafi and Seyfarth give, in the context of bipedal legged locomotion, the following interpretation of the term *bio-inspiration*:

> *"[. . .] bio-inspiration does not just mean copying structures or controller[s] from nature but instead describes the concepts behind the design and control of legged systems. Since our understanding of legged locomotion in biological systems is not complete, here, bio-inspiration refers to the insight obtained from biology that can be adapted to the needs and capabilities of engineered systems [. . .]."*
>
> [Sharbafi 17a, p. 2]

Instead of copying nature blindfolded, they particularly focus on *understanding* underlying natural principles. With this interpretation, *bio-inspiration* goes beyond the visible and emphasizes the assumed intellectual labor behind the observed. In order to avoid misinterpretation, *bio-inspiration* and bio-inspired engineering are used according to the above definition within this work. Consequently, bio-inspired robotics describes robot technology and development emerging from such a bio-inspired engineering process.

As defined above, bio-inspired engineering is the process of gathering knowledge *bio-inspiration* and transferring this knowledge into artificial technical systems. Individual actions within this process can be partitioned into the following two distinct categories of knowledge transfer:

1. Physical Transfer – A Robot's Hardware:
   Usually, this is a straightforward copy of visible material structures, actors, receptors, construction dimensions, or interconnection principles.
   E.g., hollow bones, compliant tendons, muscles, eyes, limb segment lengths, ligament attachment positions, nervous routing, . . .

2. Intellectual Transfer – A Robot's Software:
   Usually, this is the attempt to imitate invisible natural function hypotheses, behavior patterns, or information processing.
   E.g., stereo-3D-vision, synergistic muscle coordination, reflex motion stimulation, . . .

In general, the bio-inspired physical transfer covers everything visible or directly measurable. The bio-inspired engineering process in this category is relatively straightforward. In contrast, the bio-inspired intellectual transfer covers everything that is neither visible nor directly measurable, thus leading to a laborious, time-consuming, and iterative reverse engineering process.

The primary contribution of this thesis, i.e., the design and evaluation of a novel natural control philosophy, is entirely embedded within the intellectual transfer category. However, it is well-known that high-quality bio-inspired robotics does not emerge from individual solutions but from the consistent combination and integration of all involved components, including hardware and software. In 1987, Asada et al. postulated that the match between hardware and control software is crucial for a system's overall performance [Asada 87, Preface]. Recent studies on various robot designs [*Vonwirth 20b*][Lee 17, Ghansah 23] and preliminary studies to this thesis strengthen this statement [*Nejadfard 19*, *Vonwirth 20a*, *Vonwirth 23a*].

Hence, for the required robotic hardware underlying the herein-developed control, this work relies on the bipedal robotic platform Carl[2] [Schütz 17][Schütz 20a]. Carl is a bio-inspired planar bipedal lower body whose single-leg construction was designed following the above-defined physical transfer. Thus, Carl's legs closely replicate the planar musculoskeletal morphology of a human [*Nejadfard 18b*, *Nejadfard 19*]. Both legs are entirely compliant and with muscle-imitating Series Elastic Actuators (SEAs) over-actuated.

[2] Carl (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU. URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025)

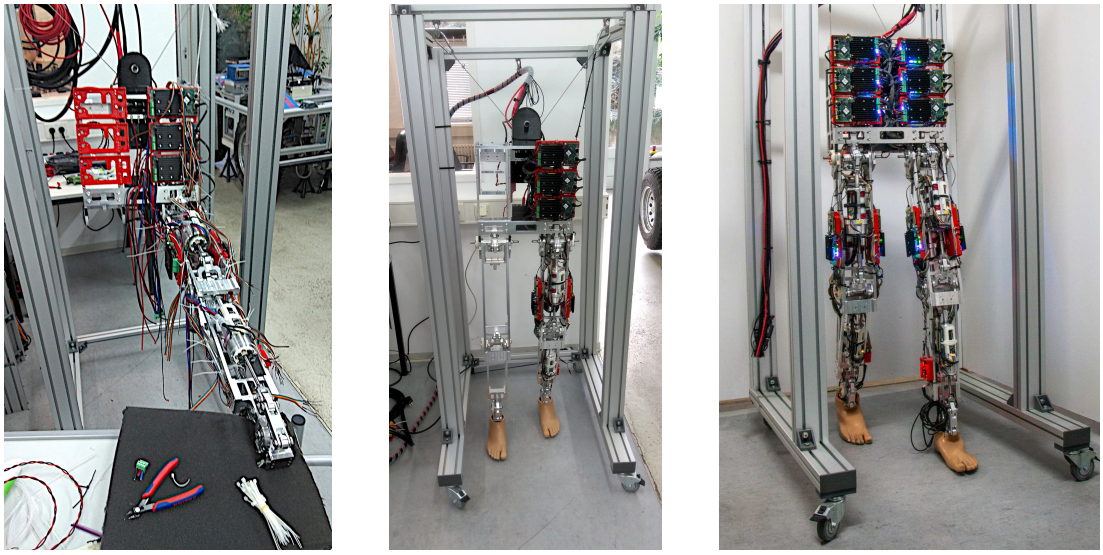


Figure 1.1: Assembly illustration of the bipedal version of CARL. After the individual construction, calibration, and programming of all 10 built-in SEAs, their assembly into the final robot denotes the start of the control-related work of this thesis.

Pictures of CARL's bipedal assembly, denoting the start of this work, are shown in Figure 1.1. More detailed descriptions of CARL's development and human-like morphology are presented later in Sections 3.1.1 and 3.1.3, respectively. Ongoing research on the robot's skeletal structure, material, musculoskeletal system, and low-level muscular reflex control integration can be found in [*Mianowski 24*, *Sivak 24a*, *Sivak 24b*].

In the special case of bio-inspired humanoid robotics, the pool of *bio-inspirations* can be split into five most significant domains. Each domain is individually mappable from its biological origin to an appropriate technical counterpart in robotics.

	Biology	⇒	**Robotics**	
1.	Receptors		Sensors	
2.	Material		Material	
3.	Morphology		Kinematic	
4.	Central Nervous System		Control System	Scope of
5.	Muscles		Actuators	this Thesis

While initial tests with the single-leg robot CARL successfully demonstrated its mechanical functionality, these early experiments failed to reliably stabilize the robot's over-actuated and entirely compliant mechanics [*Vonwirth 17*]. Thus, with the development and integration of the first consistent and fully functional natural control system, this work continues with the preliminary bio-inspired development of the robot CARL. The herein-deployed control system can handle both the built-in mechanical compliances and the natural over-actuation of CARL's human-like morphology. As highlighted, the scope of this thesis, therefore, lies within the intellectual transfer of natural control principles of a human's interconnected neural and muscular systems. Figure 1.2 illustrates the overall picture of CARL's bio-inspired development. Similar to the table above, the scope of this thesis is highlighted and framed in red within the figure.

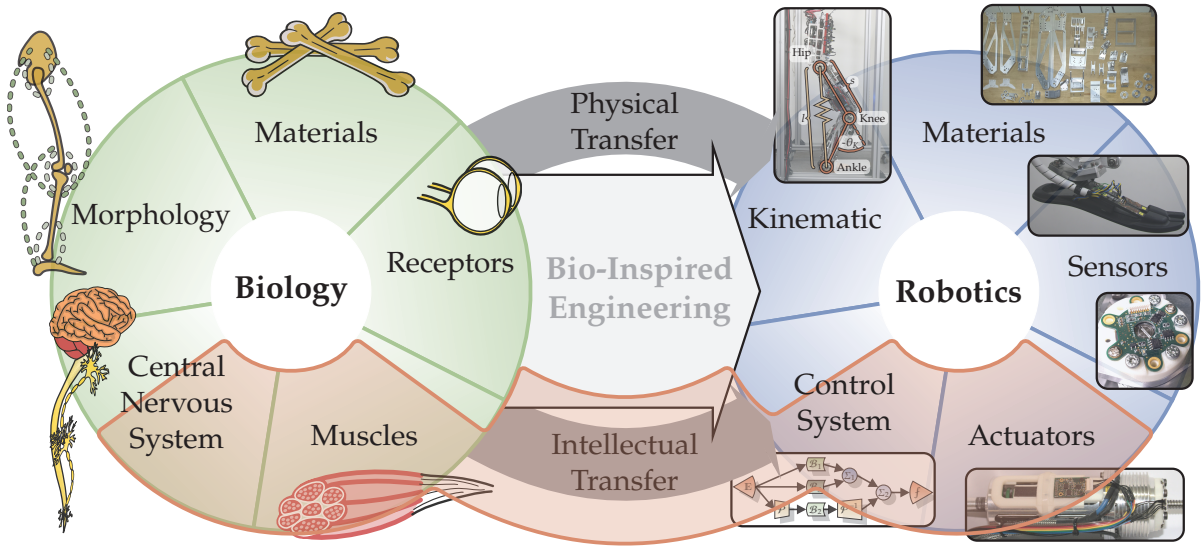


Figure 1.2: Illustration of the bio-inspired engineering process of the bipedal robot Carl. The biology and robotics disciplines are partitioned into five distinct domains, each linked by the bio-inspired engineering process. From left to right, this process aims to transfer inspiration and knowledge from biology to artificial technical robotic counterparts. Property-individual knowledge engineering is partitioned into physical and intellectual transfer. While physical transfer primarily covers all visible mechanical design features, intellectual transfer focuses on the invisible, such as control concepts. The scope of this work, as framed in red, lies within this intellectual knowledge transfer of biological neuromuscular control principles.

1.2 Natural Control – A Neuromuscular Control Hypothesis

Modern, legged robots perform exceptionally well while usually relying on MPC-based control approaches [Katayama 23]. Mixed-criteria OC strategies can generate undisturbed walking simulations close to a human gait [Felis 16]. However, with the increasing complexity, Degrees of Freedom (DoFs), and expected capabilities of modern robots, such types of control nowadays start to push against the technical and physical limits of today's available CPUs.

On the other hand, a human brain consists of approximately 86 billion neurons, interconnected with about 100 trillion synapses, and a total power consumption of approximately 20 W [Gebicke-Haerter 23]. The neural processing power of the human brain is not comparable to the electric signal processing in today's computers. However, the neural computational capacity can be upper-limited by estimations of minimal energy consumption per processed operation [Attwell 01, Sengupta 14, Gebicke-Haerter 23].

Unsurprisingly, modern computers outperform human brains w.r.t. pure computational capabilities. Nevertheless, a human's CNS and neuromuscular interface easily outperform modern humanoid robots w.r.t. locomotion capabilities.

Thus, it is not hypothetical to assume a different, yet unknown, control strategy being embedded in nature. This thesis utilizes *bio-inspiration* to develop a novel natural control philosophy that combines biological neuromuscular principles with technical requirements and suitability for integrating advanced existing solutions, such as MPC.

As introduced beforehand and visualized in Figure 1.2, bio-inspired engineering is categorized into physical and intellectual inspiration and knowledge transformation. A physical transfer usually starts with a natural observation, followed by a technical replication, and eventually aims to achieve a technical explanation for the observed. It is a relatively straightforward process.

In contrast, intellectual transfer deals with neither visible nor directly measurable properties. Modern electrodes allow the measurement of electrical signals of individual neurons. Electroencephalography (EEG) systems are capable of measuring aerial brain activities. Latest Electromyography (EMG) sensors can detect muscle activities outside the human body. High-precision motion tracking systems can recognize body reactions with extreme accuracy. However, despite the availability of numerous such body-sensing equipment, the underlying functionality embedded within the CNS's Neural Network (NN) remains unobservable. Moreover, even for today's human-made, artificial NNs, reverse engineering and explainability are a young, active, and open research field [Samek 19, Holzinger 20, Holzinger 22].

Without the possibility of an initial biological observation, reverse-engineering of bio-inspired control is a highly speculative task. The development process is inverted into a continuous, iterative, reverse engineering procedure. At first, natural characteristics of interest must be defined and converted into technical requirements. Then, intellectual transfer is applied to reverse-engineer an overall control hypothesis that can technically replicate the initially defined natural characteristics.

If the desired characteristics cannot be achieved, the control hypothesis is respectively reworked for the next consecutive engineering iteration. The process succeeds when the hypothesis satisfactorily reproduces the desired characteristics. A visualization of both categories' bio-inspired engineering processes aside from each other is sketched in Figure 1.3. As in the previous section, the scope of this thesis is highlighted and framed in red within the figure. The two distinct bio-inspired transformation processes are visualized as circular flows surrounding a central bio-inspired engineering sketch.

Physical transfer, visualized as the clockwise blue inner process, starts on the left with an observation from biology. Then, this initial observation is copied, understood, and eventually abstracted toward similar technically applicable principles in a straightforward flow. This development process has been followed by the preceding design of the human-like robotic platform Carl, replicating a human leg's entirely compliant musculoskeletal morphology [Schütz 20a]. The associated required first control software and proof of mechanical functionality are given by [*Vonwirth 17*]. Subsequent investigations on muscle coordination and the influence of the robot's kinematic structure on its control capabilities derive technical explanations to the observed biological musculoskeletal system [*Nejadfard 18b*, *Nejadfard 18a*, *Nejadfard 20*, *Vonwirth 20a*, *Vonwirth 23a*, *Vonwirth 23b*][Nezhadfard 20].

On the other hand, intellectual transfer is visualized by the green outer circular anticlockwise process flow in Figure 1.3. As highlighted in red, the bio-inspired natural control engineering described herein is embedded within this category. The process starts on the lower left of the figure, defining the overall natural control hypothesis and the desired characteristics of interest.

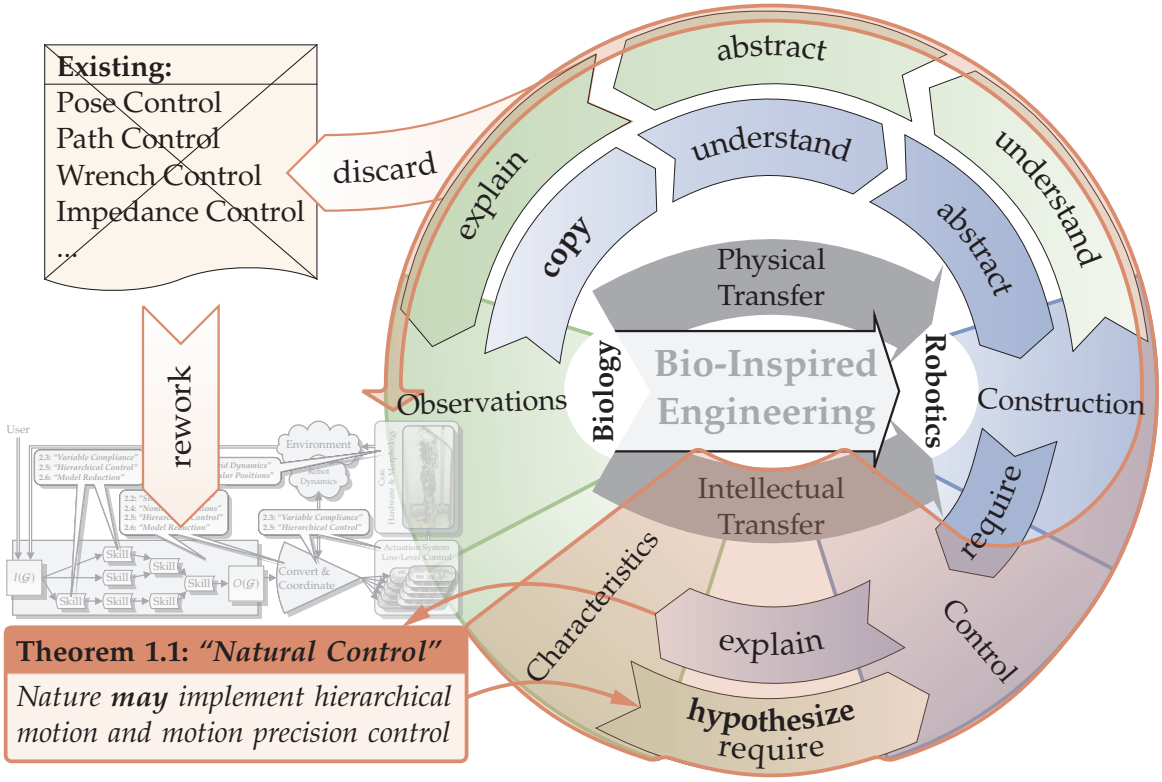


Figure 1.3: Illustration of the two distinct categories of bio-inspired engineering processes, as specified in Section 1.1. The clockwise, blue inner circle visualizes the process flow of the physical transfer. Anticlockwise in green, the outer circle represents the herein-relevant reverse-engineering process of intellectual transfer. The scope of this thesis is highlighted and framed in red. Following this work's final control hypothesis, formulated in the upcoming Theorem 1.1: *"Natural Control"*, it develops a novel neuromuscular control strategy. As visualized, this iterative process may, and in practice often does, end in discarding the chosen control hypothesis. In this case, the knowledge gathered from this failure is used to rework the hypothesis and restart the next engineering iteration. Reasoning about the herein-listed discarded approaches is sketched throughout this section and Chapter 3.

Although bio-inspired control is well-known and has successfully been applied to several robot control problems, such as balancing [Zhao 14] or walking [Yazdani 18], biological and robotic research are still mostly separated today. Thus, both communities largely ignore and miss the opportunity to learn and benefit from each other's findings. Biological and physiological studies continue to uncover motion-related natural body properties, from intrinsic muscle fiber physics [Groote 17] to estimations of areas of responsibility within the human brain [Thach 14]. Simulation studies, e.g., [Izzi 23], reproof such biological observations, as described in [Mussa-Ivaldi 85] or [Brown 00], from the robotics perspective.

With recent significant improvements in mechanical actuation, light and durable materials, and energy-efficient processing, natural neural and musculoskeletal capabilities have become closer in reach for robotics than ever before. To eventually overcome the yet-existing gap between biology and robotics, the natural control characteristics desired herein are purely derived from human gait and neuromuscular control analyses,

independent of any consequent technical and mathematical challenges. Six such characteristics are identified and converted into respective technical requirements throughout the upcoming Sections 2.1.1 to 2.1.6.

As introduced in Section 1.1, choosing and evaluating an appropriate control hypothesis is a non-straightforward, iterative, and highly speculative process. Hence, a chosen control may, and in practice often does, fail its technical and physical real-world robotic evaluation. Within this work, the biggest hurdle is the entirely compliant, human-like leg morphology of the bipedal robot Carl used herein. The most significant are:

1. Compliant Stabilization:
   The mechanical robot and digital control stabilization of Carl's entirely compliant musculoskeletal morphology.

2. Actuator Coordination:
   A cooperative rather than contradictive utilization of Carl's human-like natural over-actuation with muscle-like SEAs.

3. Parameter Limitation:
   The limitation of the number of required control parameters and commands below a computationally handleable threshold despite Carl's built-in over-actuation.

Numerous well-known and capable existing control approaches were tested and discarded throughout the development of this thesis. A representative set of those are listed in Figure 1.3. In detail, these are:

1. Pose Control:
   A combined position and orientation control that is well-known from industry robots, e.g., from pick-and-place operations. However, this approach is too stiff and impact-sensitive to render the dynamic, partially passive, and highly reactive walking observed in nature.

2. Path Control:
   A combined pose and twist[3] control, well-known from industry applications such as welding or painting. Due to their similar background, the same argument that invalidated pose control also applies here.

3. Wrench Control & Centralized, Whole-Body Impedance Control:
   Both wrench[4] and centralized impedance control require control frequencies exceeding the bandwidth of the natural CNS by orders of magnitude. It is, hence, clear that such a control cannot replicate what is implemented in nature. Impedance control renders wrench vectors according to a virtual ideal source (constant term), a spring (position term), a damper (velocity term), and sometimes even a virtual mass (acceleration term).

[3] Originating from *screw theory*, a twist vector is interpreted as a combined linear and rotational velocity throughout this thesis. Consequently, twist control combines linear and angular velocity control. [Ball 76]

[4] Originating from *screw theory*, a wrench vector is interpreted as a combined force and torque vector throughout this thesis. Consequently, wrench control combines force and torque control. [Ball 76]

4. Hierarchical, Distributed Impedance Control:
   Distributed impedance control, unfortunately, results in an enormous amount of required parameters and control values. Even for a relatively simple planar bipedal system like CARL, this number is too large to be handled suitably. However, due to the excellent natural properties of impedance control in general, a simplified, distributed impedance strategy seems promising.

Despite partial functionality, all the above-listed control approaches fail to meet some specific individual required control criteria. They are either unsuitable for the impedance requirements of natural walking, unstable when faced with an entirely compliant system, or incomputable for systems with a significant amount of actuators to control. Thus, none of these approaches alone can offer reliable control for an entirely compliant, human-like musculoskeletal leg morphology, such as in CARL.

To overcome all these approach-individual limitations, the herein-proposed novel and eventually functional natural control philosophy forms a compromise between multiple of these existing techniques. The following theorem defines the novel bio-inspired natural control hypothesis.

Theorem 1.1: Natural Control

The natural Central Nervous System (CNS) **may** implement a hierarchical control framework, commanding both, and only both, motion and motion precision to its underlying musculoskeletal complex.

thus

Natural robot control **may** be achieved by implementing a hierarchical control architecture, commanding both, and only both, motion and motion precision to underlying, muscle-imitating actuators.

A short version of this theorem has already been formulated in Figure 1.3.

The hypothesized natural control in Theorem 1.1 combines twist control (motion) with responsiveness[5] control. Both are inverted into respective force and damping controls at the actuator level.

Utilizing twist control compromises the dynamically overrestricted pose control with the underrestricted wrench control. This way, small intrinsic and extrinsic disturbances are still compensated on the distributed, concurrent actuator level while at the same time allowing for slight uncontrolled, flexible, and dynamic control diversions. Commanding responsiveness, rather than controlling it, relocates the high-frequency-requiring damping control to the low-latency actuator-local control processing. Thus, the natural control strategy implements a neuromuscular-inspired, hybrid impedance control, derived and explained in detail within Section 3.2.3.

The deployed hybrid impedance control strategy entirely complies with the technical design recommendations in [Zhao 19]. However, at the same time, the distributed,

[5] Following the definition of Seely et al., a responsiveness matrix G represents the mechanical equivalent of electrical conductance [Seely 70, p. 200]. I.e., G is the real part of a complex mechanical admittance or the inverse of mechanical damping.

actuator-intrinsic damping control also replicates the most significant control principle of a natural muscle group: antagonistic co-contraction. Hence, the above-proposed natural control approach is not only fully functional under real-world conditions but additionally offers a technically reasonable explanation for observable musculoskeletal behavior patterns in nature.

A consistent Behavior-Based (BB) high-level control architecture is developed on top of the underlying natural control principle. Multiple real-world experiments demonstrate their combined functionality, compliant control stability, and over-actuated cooperative actuator coordination.

1.3 Objectives and Thesis Outline

Throughout this thesis, multiple individual objectives are achieved, all of which aim to replicate natural control, as hypothesized in Theorem 1.1. The individual contributions are based on *bio-inspiration* and, at the same time, technically reasoned and evaluated. Furthermore, all utilized and developed methodologies are concluded with critical discussions and, where necessary, recommendations for future research and technical improvements. As a notable side effect, possible explanations of biological circumstances and observable natural behavior patterns are offered, further strengthening the validity of the presented artificial natural control design decisions.

The following paragraphs name and briefly outline the most significant objectives that are achieved throughout this thesis. The objectives are ordered by chapter.

Natural Control – A Neuromuscular Control Hypothesis **Chapter 1**

Reverse engineering technical system implementations is a challenging task. Reverse engineering biology and its embedded neuromuscular control structure is impossible. As visualized in Figure 1.2, bio-inspired robotic engineering is an iterative process. With the hypothesis of natural control, as stated in Theorem 1.1, this thesis presents the result of a couple of such development iteration steps. This novel control philosophy is able to stably replicates biological observations while simultaneously satisfying technical suitability. Thus, natural control is not only a novel technical control method to handle entirely compliant, natural morphologies, but also a possible and meaningful interpretation of the biological neuromuscular complex. The following objectives all contribute toward this basic overall aim of coherent natural control.

Characteristic Natural Control Requirements **Chapter 2**

To understand and replicate biological locomotion, the overwhelming complexity of arbitrary motion control must be broken down into smaller, manageable control goals. Through planar human gait and neuromuscular analyses, this work identifies six essential natural control characteristics that are derived into the respective technical Requirements: *"Hybrid Dynamics"*, *"Singular Positions"*, *"Variable Compliance"*, *"Nonlinear Relations"*, *"Hierarchical Control"*, and *"Model Reduction"*. Although limited, these essential requirements define the entire development and bio-inspired engineering process toward natural control. The six identified characteristics establish a solid foundation of bio-inspired requirements for natural robot control. Thus, they can be applied across all robotic domains beyond the scope of this thesis.

Representative Review of Existing Robot Control Approaches **Chapter 2**

In recent years, commercial humanoid robots have become available and affordable in the open market, featuring a wide range of impressive locomotion skills. However, even the most capable robots today cannot perform natural, human-like locomotion. Representative selections of *rigid*, *compliant*, and *muscular* control approaches are presented and evaluated against the prior derived natural control requirements. The reviews highlight that all characteristic requirements have individually already been addressed by some control methodology. However, none of them combines all requirements at once. Note that the described analyses address only the six characteristic natural control requirements but not the approaches' general capabilities or real-world performances.

Antagonistic Muscular Control – A Hybrid Impedance Approach **Chapter 3**

Non-destructive robot-environment interaction requires mechanical compliance, such as in SEAs. To control their intrinsic elasticity, SEAs usually rely on complex internal controllers that significantly reduce their hardware's principal capability. With the introduction of the so-called *antagonistic muscular control*, this work contributes a novel bio-inspired hybrid impedance control strategy that overcomes the aforementioned limitations. For stability, the proposed concept relies on state-of-the-art hardware in a *quasi*-direct drive configuration. Mechanical impact tolerance is achieved by the SEA's combined elasticity and transparency. Antagonistic muscular control allows for significant simplifications while simultaneously amplifying the stably controllable range of its underlying hardware. Experimental evaluation using the bipedal robot Carl demonstrates the real-world functionality of antagonistic muscular control.

Interpretation of the Natural Musculoskeletal System **Chapter 3**

Biological research on human arm stability indicates that muscular co-contraction largely controls the hand's impedance scale while its shape and orientation mainly depend on the arm's pose. A distributed, muscle-intrinsic impedance control, such as the herein-proposed antagonistic muscular control, replicates such a behavior. However, in this case, the natural musculoskeletal system becomes under-actuated for the entire multidimensional impedance generation. Additional muscles reduce this under-actuation. Thus, by replicating the observed natural control characteristics, antagonistic muscular control additionally offers a coherent and technically suitable reason for the biological design of a natural musculoskeletal system. Technical reasoning on delay and other significant control properties further strengthen this theory.

Probabilistic Behavior Networks – A Control Decomposition **Chapter 4**

Like a robot's morphology must match its actuation system, a high-level control must support its underlying low-level actuation philosophy. Thus, the herein-proposed high-level control architecture builds upon two main demands: the low-level antagonistic muscular control and the six prior-defined natural control requirements. With the definition and development of the so-called *Probabilistic Behavior Networks* (PBNs), a novel data flow architecture is introduced that satisfies all these requirements. The PBN architecture is based on the probability theory of normal distributions, semantically unifying all sense and control calculations relevant herein. Thus, PBNs significantly contribute to the development of explainable BB robot control decomposition.

Tolerance by Design – Unified Uncertain Data Representation **Chapter 4**

Robot control approaches are traditionally designed to aim for absolute exact control goals. However, in reality, every task only needs to satisfy a certain amount of acceptable tolerances. In mechanical engineering, such tolerances have been a self-evident, essential part of every construction design and manufacturing process for centuries. Interestingly, this reality awareness is only rarely present in computer science, although digital, floating-point computer arithmetic, by definition, is mathematically wrong. With the above contribution of *probabilistic control*, this work introduces the well-known concept of task tolerances into robot control. Since high precision is usually unnecessary in the herein-addressed special use case of bipedal locomotion, motion precision can be traded against the more important reaction speed or energy efficiency.

Real-World Demonstration at the Hand of Natural Squats **Chapter 5**

The previous contributions defined all necessary individual building blocks required for a natural robot control architecture. Through the iterative development, implementation, and deployment of an experimental natural squat control for the bipedal robot Carl, all those individual components are eventually integrated into one consistent control system. Three different experiments are performed, visualized, and critically interpreted. Thus, these successful experiments provide proof of concept for all developed, integrated, novel natural control principles under real-world conditions. The experiments include *upright balancing* during various body and ground disturbances, as well as entirely unsupported, autonomous, compliant *squat motions*.

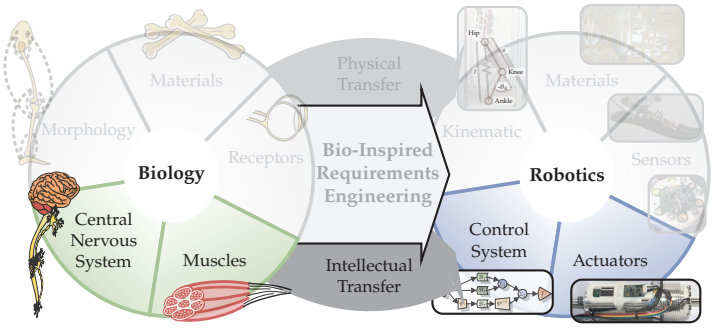
Analysis and Interpretation of Experimental Results on Carl **Chapter 5**

The novel bio-inspired control philosophy developed within this work aims to replicate the natural control characteristics of a human gait. Due to the planarity of the bipedal robot Carl and the lack of external support hardware, planar walking is dynamically impossible at the current stage of hardware development. However, with the execution of disturbed upright balancing and compliant squat motions, the best possible approximation to the individual walking subfunctions is achieved. In-depth analyses of the experiment executions provide detailed insights into the individual involved components. The analyses include the robot's prior-developed mechanical construction, the low-level antagonistic muscular control, the hardware interface, and the high-level PBN architecture. Undisclosed issues are highlighted and discussed.

Discussion and Future Work Recommendations **Chapter 6**

Lastly, this thesis is summarized and evaluated against the initially formulated aim of Theorem 1.1: *"Natural Control"*. The most significant scientific objectives achieved, as well as unmentioned but crucial technical support work, are outlined. Furthermore, the novel natural control system is evaluated against the prior objective of six identified *"Characteristic Natural Control Requirements "* and compared against the state-of-the-art. However, as pictured in Figure 1.3, bio-inspired engineering is an iterative, circular process. While this thesis makes important contributions toward more natural robotics, the overall development must continue beyond the results presented herein. Thus, a topic-specific discussion and reasoning on the most significant open issues, along with respective promising future research recommendations, is provided.

2. Natural Locomotion – A Robot's Characteristic Control Requirements

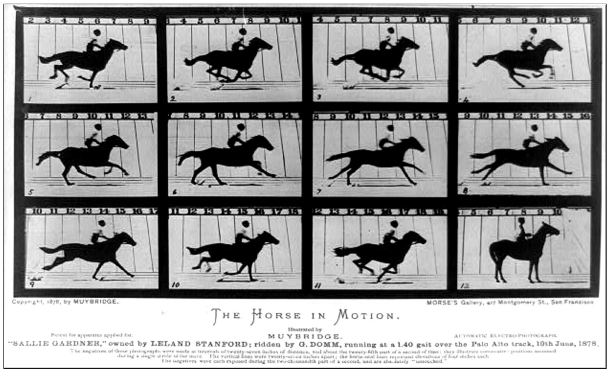


Legged locomotion is a fascinating and versatile research area in biology and robotics. Unlike wheeled locomotion, legged systems offer unique advantages in traversing unstructured terrain and overcoming obstacles. With, e.g., the ability to swim, legs even offer inter-domain propulsion capabilities. However, legged propulsion is a challenging, highly dynamic, nonlinear, and, already on a mechanical level, complex process, even in the undisturbed case. The control complexity then increases with environmental complexity and intrinsic and extrinsic uncertainties. Hence, studying legged locomotion, specifically human-like bipedal locomotion, aims to understand natural biomechanics and the interoperability of the natural Central Nervous System (CNS) with its attached musculoskeletal apparatus.

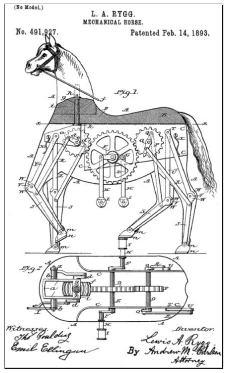
The start of bio-inspired, artificial, legged locomotion studies dates back to the 19th century, to the invention of *chronophotography*[1]. At about the same time, public interest in human-like, bipedal locomotion rose equally.

In 1878, Eadweard Muybridge published a trotting horse's motion capture photography sequence, shown in Figure 2.1a [Muybridge 78]. Back then, Muybridge's art-photography enabled scientific, kinematic studies on the horse's trotting gait and answered the big, open question of whether a horse trotting gait contains a pure flight phase or does not. Shortly afterward, technical systems were constructed, replicating

[1] Chronophotography is a unique technique of rapid photography that captures motion phases over time. Chronophotography can be considered the predecessor of films.

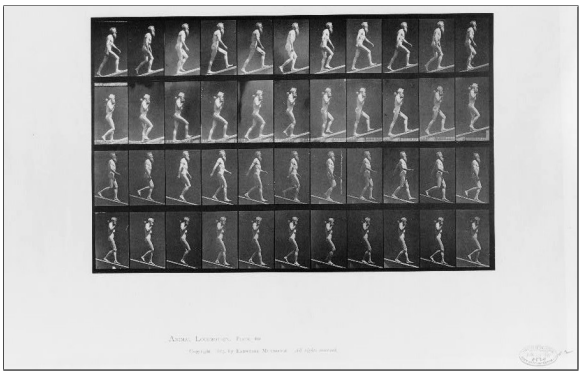


(a) The Horse in Motion

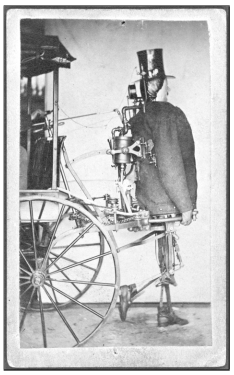


(b) Mechanical Horse

Figure 2.1: Historical studies of legged locomotion from the 19th Century. Figure (a) shows the photographic study of a horse's trotting gait by Eadweard Muybridge [Muybridge 78]. On the right, Figure (b) pictures a mechanical horse design sketch of Lewis A. Rygg [Rygg 93].



(a) Human Up- and Down Slope Walking



(b) Bipedal Walker

Figure 2.2: Historical studies of bipedal locomotion from the 19th Century. Figure (a) shows the photographic study of a human performing up- and down slope walking by Eadweard Muybridge [Muybridge 87, excerpt]. On the right, Figure (b) pictures a steam-powered, bipedal walker in grayscale patented by Dederick and Grass in 1868 [Dederick 68].

the captured animal gaits. The *"Mechanical Horse"*, patented by Lewis. A. Rygg, is one of the most famous early mechanical, quadrupedal walking constructions [Rygg 93]. Its patented construction drawing is given in Figure 2.1b.

Next to his animal studies, Muybridge also captured many versatile human motions on camera [Muybridge 87]. An exemplary *chronophotography* of an up and down-slope walking human is given in Figure 2.2a. The patented construction of a steam-powered, bipedal, human-like walker even dates back another decade before the photographic horse study of Muybridge. In 1868, Dederick and Grass designed an upright, human-shaped, bipedal walker pulling a cart behind him [Dederick 68]. Of course, the cart was necessary these days as it carried the bulky and heavy steam engine. However, the cart also offered additional ground contact, thus stabilizing the walker at its front. Hence, this early bipedal walker, honestly, is bipedal-bi-wheeled.

A more modern but still pure mechanical walking construction is the so-called *Jansen's linkage*, an eight-bar linkage developed by Theo Janson in 1990. The complex mechanical

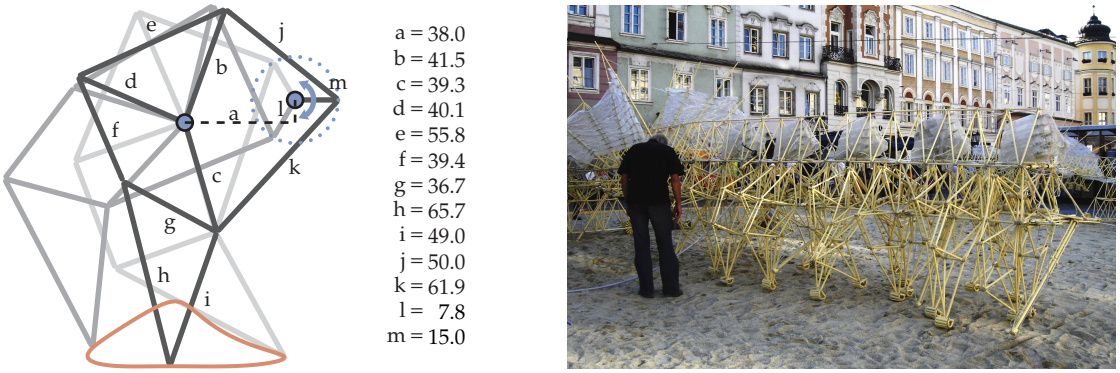


Figure 2.3: Mechanical leg construction of Theo Janson's Strandbeests. To the left, the construction of the planar, eight-bar *Jansen's linkage* is sketched together with the computer-optimized link dimensions. The leg's only propulsion is the rotation of the 'm' segment as sketched, which results in the red foot trajectory. The picture[2] on the right shows a wind-driven exemplar of Janson's Strandbeests.

linkage converts the rotational motion of a single drive shaft into a continuous walking cycle with almost no vertical foot displacement during stance. In contrast to the earlier walkers, Janson's unique design does not require additional supporting structures if multiple such legs are coupled. The resulting walking machines, the so-called Strandbeests[3], feature an arbitrary number of legs. Figure 2.3 pictures the mechanical linkage, its parameters, the foot trajectory, and a photo of one of Janson's Strandbeests.

Natural, legged locomotion attracted human interest for centuries. Due to its naturalness and its combined simplicity and complexity, it inspired technology, research, and art. All the above-presented works are related to bio-inspired engineering.

Muybridge unintentionally founded the field of motion capture studies. In the hope of replicating natural gait characteristics, Rygg, Dederick, and Grass simply copied biological morphologies. One century later, Janson utilized a more sophisticated technique. He abstracted from exact replication and adapted the basic principle of walking, consecutive stepping, to his technical needs. To determine the 13 parameters of his mechanical linkage, Janson used a bio-inspired, computer-aided, evolutionary optimization strategy. Since this optimized mechanical linkage is solely responsible for the whole motion coordination, in an abstract sense, the Strandbeests can be seen as one of the first technical walking systems that use Optimal Control (OC).

Today, most early mechanical solutions have been replaced by computer-aided controllers. Nevertheless, artificial bipedal locomotion still encapsulates many unsolved control problems. Among others, these problems include hybrid dynamics, passive Degrees of Freedom (DoFs), or leg singularities. Due to the latter, modern OC strategies classify human-like walking as *uncontrollable*. However, it is undeniable that human-like, bipedal walking is controllable and that a widely capable solution exists in nature, implemented in our own bodies. Its existence is guaranteed even though state-of-the-art research has

[2] Photo by *"Erik Möller"*, donated to the Public Domain at `https://en.wikipedia.org/wiki/File:Jansen-Strandbeest_crop.jpg` (2024), unaltered

[3] Strandbeests by the artist Theo Janson are a mechanical art engaged to create a *"new form of life"*. URL: `https://www.strandbeest.com` (2025)

not yet discovered this solution. Thus, human-like bipedal locomotion is the perfect domain for novel, bio-inspired, natural robot control studies.

Following the continuous, iterative bio-inspired engineering process introduced in Section 1.1 and illustrated in Figure 1.3, the very first required step before engineering is to identify a small set of representative biological characteristics that define the technical requirements of natural control as declared in Theorem 1.1.

Chapter Outline

Section 2.1 identifies six fundamental natural control characteristics from an undisturbed human gait and derives respective technical requirements. In Section 2.2, these requirements are related to all the engineering disciplines that are involved in developing bio-inspired robotics. A review of state-of-the-art technology regarding these natural control requirements is presented in Section 2.3. The review includes distinct analyses of a representative subset of existing control strategies, thereby focusing solely on the prior-defined characteristic requirements. Section 2.4 provides an overview of a general robot control system, assigning the most relevant requirements to each involved subsystem. This overview serves as a reference throughout the subsequent chapters, which consecutively address those subsystems. Section 2.5 concludes this chapter.

2.1 Characteristic Natural Control Requirements of a Human Gait

As if today, modern bipedal robots cannot replicate natural walking: neither stable nor reliable [Sharbafi 17a, p. 158]. Of course, this fact has well-known, good reasons.

Sophisticated and highly capable Model Predictive Control (MPC) approaches are widely used within state-of-the-art humanoids [Katayama 23]. Unfortunately, due to the naturally utilized singular knee, these techniques declare natural walking *uncontrollable* by their mathematical definition. However, numerous other approaches that do not suffer this particular theoretical limitation also failed to handle the overall complexity and the dynamics of natural walking [Torricelli 16].

Furthermore, natural walking is a very versatile gait that combines several important dynamics and control properties of other fundamental locomotion skills such as standing, jumping, or running. Thus, the yet unsolved problem of natural walking provides a perfect foundation for developing the novel, bio-inspired natural control philosophy, as hypothesized in Theorem 1.1 within Section 1.2. As highlighted and described in Figure 1.3, this work aims to replicate natural control characteristics within a technical robot control architecture through bio-inspired intellectual transfer. Thus, these characteristics are herein identified from a wide range of physiological, biological, and technical human gait studies. An overview of the most significant human gait analysis in the sagittal plane referenced in this work is given in Figure 2.4.

Legged locomotion is a repetitive, cyclic process characterized by a single gait cycle that covers two strides, one per leg. This gait cycle is further partitioned into multiple phases separated by observable events and control changes. While the number, duration, and

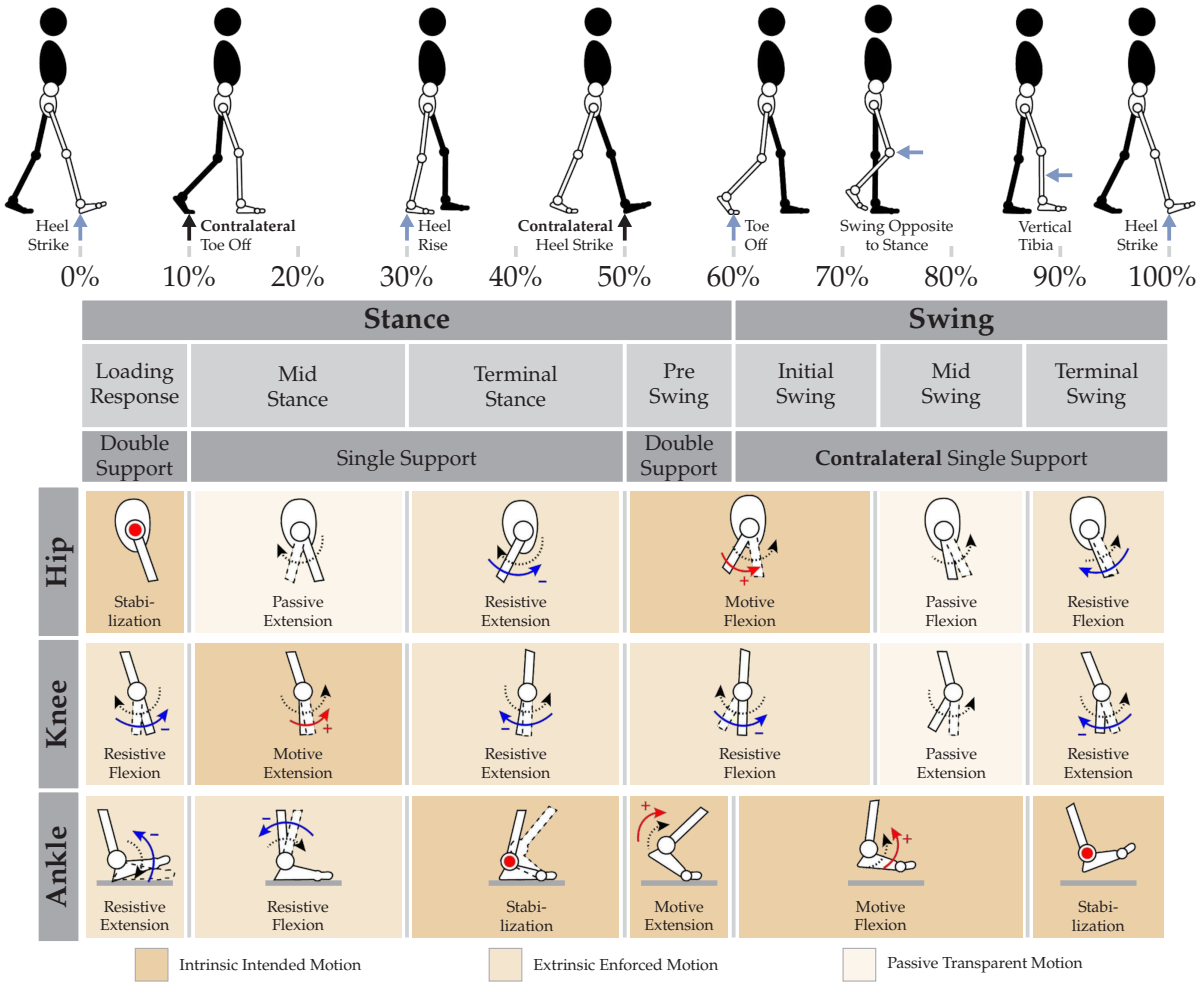


Figure 2.4: Diagram of the planar hip, knee, and ankle joint motor subfunctions within the seven gait cycle phases by [Torricelli 16, Perry 92]. Subfunctions with an orange background are actively intended and executed motions or no-motions (stabilizations). The ones backed in light orange are motions enforced by external influences, not intrinsically commanded. All others are pure passive movements due to the undisturbed and uncontrolled dynamics. Black arrows visualize the actual joint motions. The red and blue arrows indicate the applied control torques leading to positive or negative work, respectively[4].

placement of gait phases vary in the literature, the definition of the start of each gait cycle has emerged mostly common as the well-defined event of one leg's heel strike, i.e., one leg's initial ground contact [Perry 92, Rose 06, Winter 09, Perry 10, Hamill 14]. However, alternative stride definitions are also existing. In [Sharbafi 17a], David Lee utilizes a gait cycle definition that lasts between consecutive mid-stance of one leg, i.e., shifted by about a quarter of the above standard stride definition. Luksch refers to the standard gait cycle in his PhD thesis for analysis purposes but defines a different control cycle, starting at toes-down, i.e., complete foot contact [Luksch 10a].

[4] Based on *"Human-like compliant locomotion: state of the art of robotic implementations"* [Torricelli 16], published under Creative Commons Attribution License (CC BY 3.0), doi:10.1088/1748-3190/11/5/051002

A similar picture can be drawn w.r.t. the gait phases of a gait cycle. For simplicity reasons, some robot control approaches only distinguish the two phases, stance and swing, e.g., [Geyer 10, Geijtenbeek 13]. The B4LC[5] [Luksch 10a, Zhao 16] bipedal simulation control architecture distinguishes five phases of walking [Luksch 10a, Luksch 10b]. Hamill et al. also distinguish five phases in their biomechanical studies, but five slightly different ones [Hamill 14]. Finally, Perry defines eight individual gait phases in her biological studies of human walking [Perry 92, Perry 10]. However, the first of her phases, called *Initial Contact*, is nothing more than *"the moment when the foot just touches the floor"* [Perry 92, p. 11]. With this phase being reduced to the event that initiates the gait cycle, a seven-phase standard gait partitioning is achieved, which is widely referenced in the literature.

An extensive summary of the walking-phase-individual motor subfunctions of humans has been collected in the article [Torricelli 16]. The studies highlight the control-relevant active and passive motions throughout each of the seven gait cycle phases defined by Perry [Perry 92]. Projecting a real-world, three-dimensional gait to the sagittal plane significantly simplifies the gait process. However, it preserves all control-relevant biological characteristics. Figure 2.4 visualizes a planar gait on top of a collection of the respective planar joint subfunctions. The collection is limited to the planar hip, knee, and ankle joints relevant to this thesis.

Natural control characteristics can be partitioned into several categories. The categories define the derivation efforts of their containing characteristics, their level of abstraction, and, thus, their uncertainty. Within this work, the following three categories, ordered by their level of abstraction, are distinguished:

Definition 2.1: Categories of Natural Control Characteristics

Natural control characteristics are organized into the following three categories of increasing levels of abstraction and uncertainty:

Direct, i.e., Visible Characteristics

Characteristics with features that can directly, thus primarily visually, be observed.

E.g., dimensions, links, positions, motions, velocities, . . .

Indirect, i.e., Measurable Characteristics

Characteristics that cannot be observed directly but require indirect measurements, such as motion responses to induced force disturbances.

E.g., forces, torques, tensions, impedances, . . .

Model-based, i.e., Hypothetical Characteristics

Control model hypotheses that are engineered from both quantitative analyses and logical reasoning. They aim to explain the underlying causes of direct and indirect observable characteristic behaviors. Hypothetical characteristics come without any chance of formal or experimental proof. They remain valid as long as their claimed predictions are as accurate as required.

E.g., information processing, reasoning, model-building, behaviors, . . .

[5] B4LC is a scientific humanoid robot simulation by the RPTU developed to study compliant, natural dynamics-exploiting gait.

Visible characteristics are primarily based on direct kinematic observations. Figure 2.4 illustrates a kinematic human gait sequence, divided into the seven gait phases by [Perry 92]. Snapshots of important moments throughout the gait cycle visualize the respective kinematic configurations. The gait phases are matched to the overall gait periods, i.e., stance and swing periods or *double support* and *single support* periods. The pictograms underneath explain the joint-individual contributions to the gait.

Joint-individual motion and motion stabilization are pictured for all seven gait phases. Black arrows represent the physical motion of a joint. Colored arrows aside indicate joint torques being applied by the neuromuscular system. Propulsive torques are given in red, while resistive torques are colored in blue.

The three distinct backgrounds highlight the underlying source of motion. An orange background indicates intrinsically motivated motions, intended and actively induced. Extrinsically enforced motions are backed in light orange. Those are the resistive and actively compliant motions throughout the gait cycle. All others are pure passive motions without active forces, neither intrinsic nor extrinsic. Studies of muscular activities primarily contribute to the measurable characteristics category.

In contrast to both previous categories, hypothetical characteristics cannot be backed by observations or measurements. These characteristics generally define an assumption-based model that approximately describes and explains an entire complex behavior, such as upright balancing. The complexity of such models is generally unlimited and may go far beyond the scope of planar human walking in future studies.

Unfortunately, it is impossible to reverse engineer brain-internal processing, algorithms, or information representations. Thus, the correctness of hypothetical characteristics cannot be proven. Instead, it is feasible and successful practice in all natural science disciplines to model the world as observed and use those models for as long as they work. In the case of future hypothetical model abstraction, the well-known principle of *divide and conquer* is worth following.

In the following, Sections 2.1.1 to 2.1.6, identify six fundamental characteristics of natural gait and derive respective technical requirements from the identified characteristics. Following Definition 2.1, the first two sections identify visible characteristics. They are followed by three measurable characteristics in the middle sections and, lastly, by a single hypothetical characteristic. The derived technical requirements of natural robot control are concluded individually for each biological characteristic. All of these bio-inspired requirements collectively form a small but powerful set of design and control principles for both the hardware and software of natural robot control.

All of the identified fundamental natural control characteristics are present in a planar projection of a standard, undisturbed, three-dimensional human gait. Thus, realizing and validating a thereof developed natural robot control system in a planar hardware setup is feasible. All characteristics, requirements, and conclusions drawn herein remain valid within other arbitrary (reduced or extended) dimensionalities. Moreover, the planar limitation not only simplifies the underlying math but also allows for better focus and highlighting of the essential biological control properties. Thus, the control foundations developed herein can be extended to full-featured, three-dimensional, real-world robot control in future work.

2.1.1 Hybrid Dynamics – Discrete Support Transitions

Legged robots are among the most challenging types of mobile floating-base ground systems. First, their legs are complex, multi-link, high-dimensional manipulators. Second, a legged gait blends continuous locomotion with discrete footsteps.

A floating-base system is characterized by not having an origin fixed within the surrounding operational environment. In contrast, a fixed base system has a clearly defined origin. For example, a typical industry robot arm has a base fixed to its working cell. This fixed robot origin enables the usage of classical *Newton-Euler* rigid-body dynamics (see, e.g., [Featherstone 08, p. 92]) and classical control theory based on single-input single-output systems (see, e.g., [Franklin 18, p. 27]). Without such a reference, this all changes and immediately becomes significantly more complex.

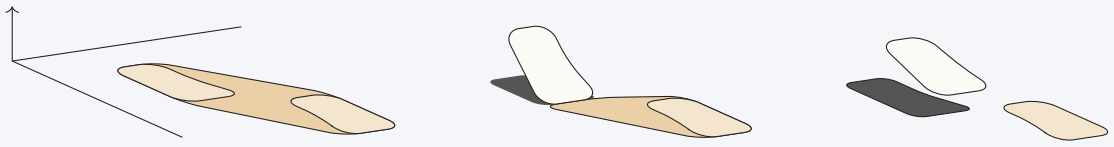
In order to enforce fixed base kinematics, the floating-base is usually modeled as a virtual, infinite, passive 6-DoF linkage against an arbitrary but fixed world reference frame (see, e.g., [Featherstone 08, p. 179]). With this trick, every floating-base system can again be modeled as a fixed base system. However, it also highlights the significant difficulty inherent to any such system, i.e., *passive* DoFs. Classical control theory is insufficient to control passive DoFs, thus requiring more sophisticated techniques that can exploit dynamic prediction models, such as MPC.

Throughout continuous bipedal walking, the two legs alternatingly support the system's body. The discrete contact changes when stepping cause immediate transitions between respective support polygons and, thus, emerge in a hybrid dynamical setup. Hybrid dynamics are characterized by a set of continuous system dynamics connected via discrete, instant state transitions. Theoretical foundations on Lyapunov stability[6] criteria in hybrid dynamical systems can be found in [Ye 98]. Since these state transitions correlate with the feet observably losing or gaining ground contact, the discrete support transitions of a human gait are categorized as a visible characteristic.

The support polygon is the convex hull of the ground projections of all contact points [McGhee 68]. It is defined, illustrated, and explained as follows:

Definition 2.2: Support Polygon

The support polygon is the *convex hull* of all contact points, projected onto a plane, normal to the environmental gravity direction.



Exemplary footprints of a two-legged system at three distinct foot configurations with their respective orange colored support polygon shapes. From left to right, *double support* with both feet on the ground, *double support* with only the back foot's toes in ground contact, and *single support* with just one foot on the ground.

based on [McGhee 68]

[6]: If a dynamic system that is (infinitesimal) close to one of its equilibrium states forever stays that close, it is called *Lyapunov stable* [Lyapunov 92]

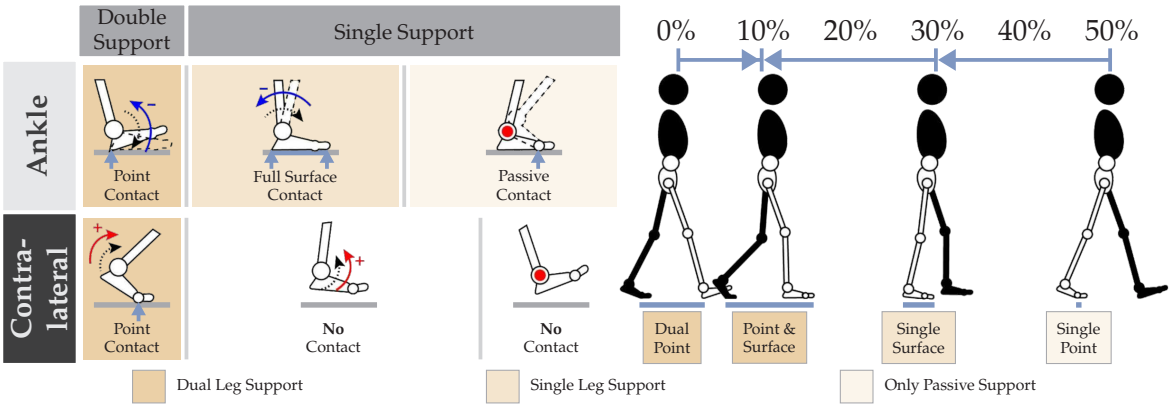


Figure 2.5: The distinct support polygons and, hence, hybrid dynamics throughout the planar, natural gait cycle as introduced in Figure 2.4. On the left, both ankles' actions and ground contacts are pictured aside each other throughout a single stride. The respective, relevant kinematic sketches are shown to the right. Each snapshot visualizes one of the four distinct support states throughout one stride. Both contact and support points and surfaces are marked in blue both left and right. The background colors indicate whether the support polygon is supported by both legs, only a single leg, or degenerates to just a single point.

In the planar case, the two-dimensional surface shape of the support polygon reduces to a one-dimensional line lasting from the most backward contact point to the most frontal one. The distinct support polygons and hence hybrid dynamics of a planar, two-legged system throughout the natural gait cycle is described in Figure 2.5.

On the left, the gait phases of the stance-supporting ankle and its contralateral one are pictured on top of each other throughout a single stride, i.e., half a gait cycle. Both feet' contact points or surfaces are described and marked in blue for each phase. The resulting four distinct support polygon configurations of the biped are visualized on the right. Note that the toe joint within the foot is considered passive for two reasons. First, most modern bipedal robots do not feature that particular joint at the toes. Second, its strength and support surface are generally not big enough to significantly prevent a system from falling. When humans balance on their toes, they mainly utilize the ankle joint to lift the heel off the ground and perform dynamic balance with the body inertia to counteract bigger than just gentle disturbances.

During *double support*, both feet are in point contact with the ground. The resulting support polygon stretches from the toes of the back foot to the heel of the frontal one. Throughout this phase, the frontal toes are lowered while the heel on the back is raised. *Double support* ends with the toes-off event of the back foot. This event coincides with the touchdown of the frontal toes, initiating *single support*. At this particular moment, the support polygon extends forward to the frontal toes before losing the backward contact point. It is visualized at the 10% position on the right side of Figure 2.5. The dual leg support offers a wide stretched support polygon with an equally wide range of control possibilities. However, the two ground contact points close a loop in the system's kinematic chain that has to be considered for control.

During mid-stance, the first half of the *single support*, the contact surface of the single supporting foot equals the support polygon. The contralateral leg is in its swing phase

and will not touch the ground till the next stride. In this phase, the system features an open, tree-like kinematic structure with the supporting foot acting as its base. It is the kinematically most straightforward configuration, allowing the application of classical, well-known robot arm control strategies. However, in the case of a human-dimensioned foot, this robot base is quite narrow, as are the respective control limits.

After the heel rise of the single support foot, the gait is in the terminal stance phase. This phase is characterized by the passive toe point contact of the supporting foot while the contralateral one is still in the air. Hence, the support polygon degenerates to a single point. Kept together by gravity, this point implements the functionality of a passive hinge joint. Consequently, as argued initially, modern model-based approaches like MPC are required to control the gait within that phase.

In conclusion, the human gait inherent hybrid dynamics imply the following technical requirements on natural robot control systems and underlying hardware:

Requirement 2.1: Hybrid Dynamics

Natural gait is characterized by discrete transitions between alternating support polygons. The resulting hybrid dynamics, thus, feature alternating open-loop and closed-loop kinematic topologies with interchanging active and passive environmental anchoring.

implications

Natural control, as declared by Theorem 1.1, must support:

Impact Tolerance
Stability under instantaneous sensor changes and mechanically impact-resistant, compliant hardware must be supported (impulse and step response).

Alternating Open- & Closed-Loop Kinematics
Kinematic calculus must offer runtime constraint modeling and adaptation.

Interchanging Active & Passive Environmental Anchoring
Model-based control approaches must be supported due to temporary passive DoF and a floating-base.

Finally, it is important to note that the situation with both feet flat on the ground, as visualized on the left within Definition 2.2, does not occur in a natural gait at all. Nevertheless, it is the default configuration for upright standing and, thus, significantly important.

2.1.2 Singular Positions – Locked Knees

The design of artificial legs is a crucial part of the overall development of legged robotics. First, the legs are the later connection components between the robot and its environment. Second, the legs will be the most moving parts of the robot during locomotion and hence embody the parts most challenging to control. The following two historical approaches in robotic leg design are distinguished in [Rezazadeh 18]:

1. Biological:
   Using a natural leg morphology, fitting technical actuation and control systems.

2. Mathematical:
   Using a technically optimized design, applying initially unconsidered controls.

Rezazadeh et al. conclude their analysis that the first ones often suffer from control malfunction, while the second ones are often limited to their single-designed task.

To overcome that, a template modeling process is recommended in [Rezazadeh 18]. Starting with some desired natural characteristics, Rezazadeh et al. propose iterative specification and optimization of a technical design progressing from abstract kinematic and dynamic models to actual hardware and construction decisions. It is the same principle process as proposed and performed within this work. However, due to their sole focus on mechanics rather than control, Rezazadeh et al., in brackets, limit their approach to the usual constraints of all state-of-the-art MPC approaches:

"Dynamically, all tasks can be mapped to the actuators using the mechanism's Jacobian matrix, regardless of its form (so long as it is not singular)" [Rezazadeh 18, p. 54374]

Unfortunately, entirely stretched, thus singular, knees are a fundamental and characteristic posture of both stance and gait [Perry 92, Perry 10]. Figure 2.6 visualizes the singularity, i.e., the configuration in which the knee's Jacobian becomes singular, of the human knee during natural gait. Multiple research groups have investigated solutions to overcome the singularity problem, primarily by adding additional joints that stay non-singular in series to the knee [Kurazume 05, Ogura 06b, Ogura 06a, Handharu 08, Yoon 11, Miura 11]. However, compensating singularity by adding non-singular serial joints must be treated as a *workaround* rather than a solution.

As shown in the left-hand-side plot of Figure 2.6, the human knee motion twice traverses through singular and close-to-singular[7] configurations during a normal walking gait cycle. The singular position at 0° knee flexion is highlighted as blue horizontal line. It is surrounded by a light orange 5° environment visualizing the close-to-singular configuration range. The first, close-to-singular configuration is located within the terminal stance phase. Second, the singular knee configuration occurs before the heel strike at the end of the swing phase. Both events surround a leg's single support phase. In this phase, the leg acts like a spring suspension between the ground and the main body [Geyer 06]. Due to the knee's operating point close to the singular position, the gear ratio between the (knee) muscle forces and the effective resulting ground force tends toward infinity exactly within the singularity. Consequently, the muscles can operate at a very low activation level and energy consumption.

Within the singularity, the whole Ground Reaction Force (GRF), i.e., the whole body weight, is offloaded from the (knee) muscles onto the (knee) joint. During stance, this offers an upright, stretched leg position at zero energetic cost. However, during walking, at heel strike, the exact singularity is to be avoided to prevent structural damage to the skeletal system on environmental impact.

[7] Close-to-singular describes a configuration that is not yet entirely singular but within a narrow region around a singularity that may already cause numerical instability when digitally processed with finite-resolution representations, such as floating-point numbers.

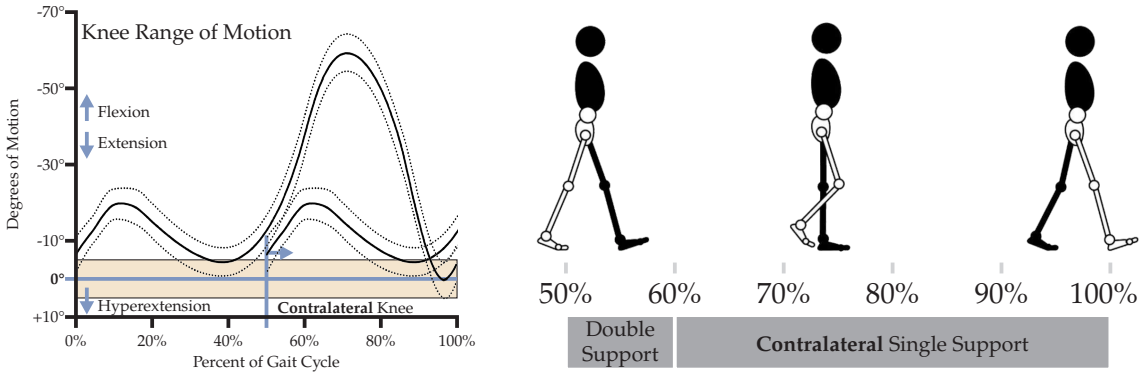


Figure 2.6: Sketch of the characteristic singular and close-to-singular knee configurations during a human walking gait. The knee angle throughout a normal undisturbed gait cycle is shown on the left. The motion data are taken from [Perry 92]. On the right, planar walking sketches of the second half of the gait cycle are provided, as introduced in Figure 2.4. Ideally, the human knee only flexes backward and is locked in its singular position at 0°. However, some humans can hyperextend their knees forward, while others maintain a permanent flexion of up to 5° [Perry 92]. A 5° close-to-singular environment around the singular position is visualized with a light orange background on the left-hand-side plot. At the second half of the gait cycle, the contralateral knee motion (the first half plot) is added to the knee range of motion plot.

Technically, a singular leg is not needed to achieve low-energy locomotion. Proper springs can enable a passive upright stance with bent knees. The effect of such springs has been studied at the hand of bipedal walking with the robot ERNIE[8] [Yang 08]. The studies conclude that stiffer springs better meet the requirements of faster walking and vice versa. This observation reinforces the common assumption that optimal energy efficiency is achieved when walking at a natural frequency. Peak performance of low-energy walking exploiting natural dynamic oscillations is achieved in entirely passive systems [McGeer 90, Collins 05, Hobbelen 08]. However, exploiting natural dynamics always comes at the cost of limited control and possible control tasks. Thus, pure passive dynamics are not suitable to meet the versatility of a human leg.

In conclusion, the observable and characteristic natural locked knee configuration implies the following technical requirements for natural robot control systems:

Requirement 2.2: Singular Positions

Natural human gait is characterized by singular or close-to-singular knee angles.

implication

Natural control, as declared by Theorem 1.1, must allow a robot to operate in:

Singular and Close-to-Singular Configuration

Natural control algorithms and their underlying arithmetic must either stably handle singular and close-to-singular kinematic Jacobian matrices or preventively prohibit all potentially singular coordinate transformations.

[8] ERNIE is a bipedal robot prototype designed and housed at the *Ohio State University*.

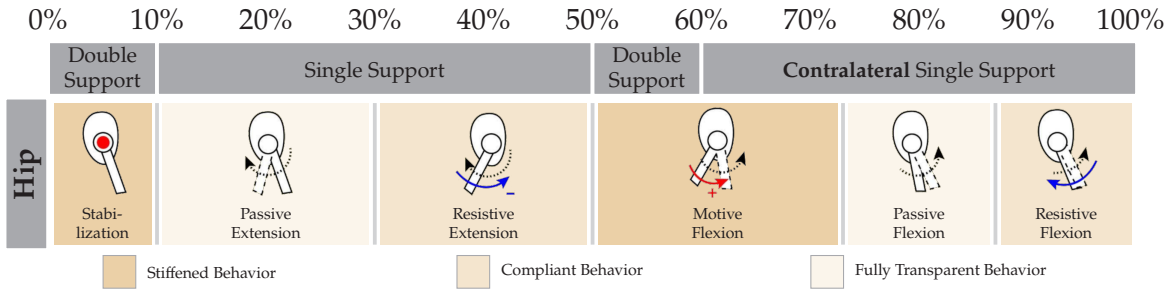


Figure 2.7: Diagram of the characteristic planar hip joint motor subfunctions throughout a regular human gait cycle by [Torricelli 16, Perry 92]. The background coloring of the individual subfunctions indicates the variable level of compliance being applied during each subfunction. Compliances range from highly stiffened behavior colored in orange, via medium compliance levels colored in light orange, to fully transparent, passive motion.

Note that the motion data by [Perry 92], as illustrated in Figure 2.6, do not explicitly require an entirely locked, thus singular knee during walking. However, locked knees are the default configuration for natural, upright standing, and at least a close-to-singular angle must be achieved during regular gait. Depending on the exact processing algorithm and the underlying computational (floating-point) arithmetic, those close-to-singular Jacobian matrices usually already start to become numerically unstable.

2.1.3 Variable Compliance – Motive, Passive, and Resistive Motions

A human gait is characterized by dynamic motions that integrate environmental effects. The resulting control features all motive, passive, and resistive motions carried out at respectively proper levels of compliance. Figure 2.4 addresses these motions for each subfunction throughout the walking phases and joints. The hip's subfunctions separated in Figure 2.7, however, provide an entirely representative excerpt of those.

At the start of the gait cycle, the hip actively stabilizes both the leg and main body motion. As indicated by the orange background color in Figure 2.7, the joint is highly stiffened to withstand the impact forces and mechanical impulses at the heel strike. However, due to the mechanical impact, a small amount of compliance is required and kept. A second stiff hip action is performed within and after the second *double support* period. This time, the upper leg intensionally moves forward, thus off-loading the body weight to the contralateral limb and initiating the leg swing phase.

The hip joint is entirely relaxed after both of those highly stiff actions. In this context, entirely relaxed equals zero impedance or infinite admittance. The joint acts passively, driven only by the intrinsic and the extrinsic dynamics. In reality, however, achieving zero impedance is impossible. The mechanical characteristics of the attached ligaments, tendons, and muscles always induce small portions of intrinsic impedance and friction resistance. Equally, this implication also applies to any technical system, independent of whether it tries to mimic these natural characteristics or not.

The rest of the gait cycle, the terminal stance and terminal swing phases, are resistive motions. A resistive motion is initiated not by the actuation system but by extrinsic or intrinsic dynamics. The actuation system controls, guides, and dampens such enforced

movements. By definition, this is neither a rigid, i.e., maximal stiff behavior mentioned at first, nor a passive, fully transparent one. As indicated by the light orange background in Figure 2.7, this involves compliant behavior somewhere between stiff and fully transparent. Thus, the whole range of compliance is covered throughout a gait cycle.

Early studies on disturbed human arm movement experimentally support this hypothesis of variable compliance [Mussa-Ivaldi 85]. Their results suggest that humans modulate their compliance level according to task-induced motion precision requirements. Hence, technical solutions like clutches are unsuitable for mimicking natural control, although they have been designed and proposed for robotic and prosthetic usage, e.g., [Haeufle 12]. Moreover, clutches embody non-continuous behavior that massively complicates control, risking smoothness and exponential stability. A broad overview of clutched elastic actuators has been collected and classified by Plooij et al. in [Plooij 17].

In conclusion, the following technical requirements can be derived from the characteristic versatility of controlled motions throughout natural gait:

Requirement 2.3: Variable Compliance

Natural gait is characterized by a variety of motive, passive, and resistive motions controlled at variable levels of precision and compliance.

implications

Natural control, as declared by Theorem 1.1, and its underlying hardware must support continuous, variable-precision control and mechanical impedance:

Mechanical Impedance

Natural compliance ranges from fully transparent to highly stiffened, nearly rigid. Hence, robots that aim to replicate natural control characteristics must be able to render identical or at least similar physical behavior.

Task and Control Tolerances

Motion control at variable impedance inherently implies variable precision of those motions. Consequently, control tasks and control commands should be further annotated with appropriate execution precision information, i.e., tolerances.

Note that precision annotations are not only related to control. Humanoid robots are embedded in a dynamically changing, imperfect environment, recognizing this same via noisy sensors. I.e., all the data fed into a humanoid robot come at an inherent uncertainty, i.e., a limited amount of precision. Furthermore, the robots' computed reactions emerge from mathematically wrong digital floating-point arithmetic and are applied to, again, noisy and imperfect actors.

Considering uncertainty is not a new idea. In mechanical engineering, construction tolerances are common practice. The well-known *Kalman* filter has been used to exploit data uncertainty modeling in sensor data processing since the 1960s [Kalman 60]. However, annotating tasks and control commands with tolerance information in robot control is, to the best of the author's knowledge, a novel strategy that has not yet been applied. Further information about its purpose and usage follows in Section 2.1.5.

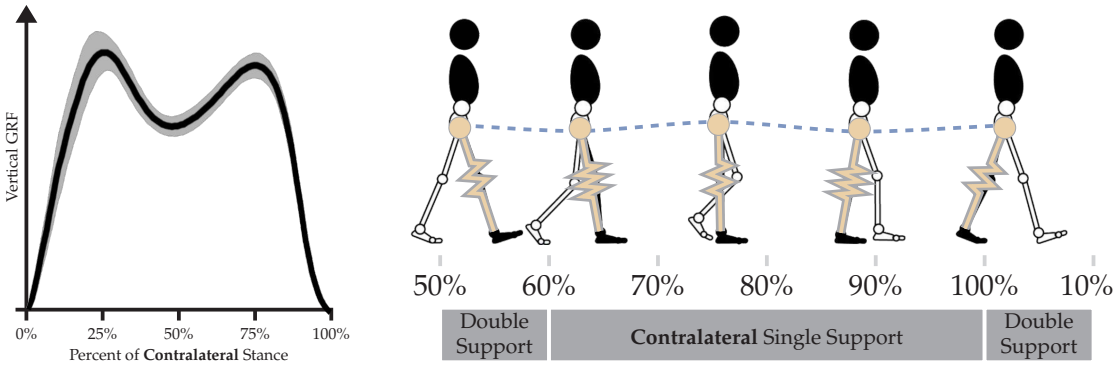


Figure 2.8: The SLIP model applied to the contralateral stance period of a human walking gait as introduced in Figure 2.4. To the left, the vertical GRF throughout a complete stance period is sketched as experimentally captured in [Geyer 06]. The same gait period, illustrated with skeletal sketches, is shown on the right. Each skeleton's contralateral leg is overlaid by the technical drawing of the SLIP's spring-mass model. An amplified hip trajectory is underlaid as dashed, blue line.

2.1.4 Nonlinear Relations – Linear Spring Leg Behavior

Simple, undisturbed walking can be considered one of the basic types of human, legged locomotion. Two others are hopping (single-legged or bipedal, with or without forward motion) and running. All three share the common natural characteristic of being modelable via a simple spring-mass system, known as the Spring Loaded Inverted Pendulum (SLIP) model [Blickhan 89, Geyer 06]. While hopping and running have been modeled as SLIP early on [Blickhan 89, McMahon 90], bipedal walking has long been considered independent and modeled as a *rigid* inverted pendulum model [Alexander 76, Mochon 80]. The authors of [Geyer 06] have traced this distinction between rigid walking and compliant running dynamics back to the seventeenth century to a Latin publication titled *"De motu animalium"* [Borelli 85][9].

However, modern research found this distinction unnecessary. In the state-of-the-art literature, running and walking are just two distinct solutions to identical SLIP dynamics [Geyer 06, Antoniak 19]. While the spring leg in the running gait compresses only once per step, it undergoes two compression cycles during walking. Higher numbers of compression cycles per step are theoretically possible but do not correlate to any existing, meaningful natural gait pattern. The spring leg model applied to the natural walking gait and the resulting vertical GRF is visualized in Figure 2.8. More details about the SLIP model will be presented in Section 2.1.6.

All three types of natural locomotion, walking, running, and hopping, are characterizable by an abstract model featuring linear spring legs. Thus, an appropriate natural robot controller must be able to render identical behavior. Unfortunately, natural legs are not linear but segmented with rotational joints. The mapping between intrinsic rotational joint motions and an extrinsic linear spring leg behavior is highly nonlinear.

[9] The content of this publication has not been verified and is used in trust of having been verified by the authors Geyer et al. within their original work [Geyer 06]

This nonlinearity is highlighted by mapping a linear stiffness K_{lin} between the ends of a leg's thigh and shank to a rotational stiffness K_{knee} at the connecting knee joint:

$$K_{knee} = l_{thigh}\, l_{shank} \sin(\theta_{knee})\, K_{lin} \tag{2.1}$$

with l_{thigh} and l_{shank} being the thigh and shank segment lengths, respectively.

The nonlinearity is, hence, two-folded. First, the stiffness itself is non-constant w.r.t. to the angle of the affected joint θ_{knee}. Second, this dependency is correlated by the highly non-const and nonlinear sine function. It has to be noted that the sine function causes non-zero linear stiffnesses to be projected onto zero rotational ones within singular knee configurations. Hence, the mapping is not injective and, consequently, not invertible. Since singular and close-to-singular configurations are specifically demanded by Requirement 2.2: *"Singular Positions"*, these cases must not be neglected.

In conclusion, the following technical requirements can be formulated to enable replication of the characteristic natural spring leg behavior on segmented limbs:

Requirement 2.4: Nonlinear Relations

Natural gait is characterized by a linear spring leg behavior of a segmented leg.

implications

Natural control, as declared by Theorem 1.1, must support robot state space transformations featuring:

Nonlinearity
Transformations between state spaces must not be limited to linear projections.

Non-Injectivity
Non-injective, hence non-invertible projections between state spaces of different dimensionalities must be supported.

Due to the unconstrained definition space and the image of the above example when mapping linear forces to rotational torques, it is obvious that non-surjective control transformations must be considered, too. This property is covered within Section 2.1.6.

2.1.5 Hierarchical Control – Neuromuscular Feedback Loops

It is long-known that robot control systems have very different and unique requirements compared to other standard desktop software [Siciliano 16, p. 284].

From an architectural perspective, real-time capability at multiple, parallel, varying temporal scopes is one of the most essential *non-functional* requirements. On the other hand, robot systems often operate within uncertain, dynamic environments, reducing the importance of the absolute correctness of calculated controls. Hence, within different hierarchical levels of complexity, different tasks can be executed at varying precision and frequency. Lower-level subtasks can independently compensate for disturbances without escalating the detected deviation to the more complex and computationally more expensive higher-level tasks operating at lower frequencies.

Such a hierarchical layout is physically built into the human CNS and the attached muscles. On the left of Figure 2.9, the four most characteristic hierarchical layers of natural neuromuscular control are outlined at the hand of a human muscle and CNS sketch. From top to bottom, these characteristic layers are:

1. Intelligence:
   The central part of the human brain, i.e., the cerebrum and its surroundings. Being the largest part of the human brain, the cerebrum is the primary location of intelligence. Thus, it houses complex mental processing and voluntary control [Moini 20, chap. 6, p. 177].

2. Intuition:
   The cerebellum, assumed to house the so-called *muscle memory* and *muscle synergies* [Thach 14]. It offers complex intuitive motions that can can execute without active thinking about them.

   The cerebrum itself only includes about 10% of the overall brain mass. However, within these, it contains more than half of the total number of neurons of the entire brain [Moini 20, chap. 16, p. 497]. Thus, intuitive muscle or automatic actuator coordination should be addressed with an equivalent importance in robotics.

3. Reflexes:
   The spinal cord, housing the relatively primitive, hard-coded neural reflex circuits. Reflex circuits at this level range from the most primitive, direct feedback circuit possible, i.e., the stretch reflex, to more advanced circuitry incorporating various complex data processing through *interneuron*s[10], such as the withdrawal reflex responding to pain stimulus [Moini 20, chap. 19, p. 604].

4. *Preflexes*:
   The natural muscles themselves, offering the so-called *preflexes*. These are tunable, mechanical properties of the muscle fibers, implementing a physical, zero-delay reactive layer [Brown 00, Groote 17].

In the following, the general functional control responsibilities of all four natural hierarchy levels are discussed. Technical requirement implications for enabling such a control hierarchy are derived for each layer. Since neural studies are far from exact, the described functions are simplified and, most likely, also not separated from each other as sharply as written. Thus, the mentioned features may be further distributed or even relocated by future studies. However, this simplified view of the current state-of-the-art is sufficient to derive technical guidelines for designing bio-inspired, natural robot control, as hypothesized in Theorem 1.1. Please note that the following discussion focuses on neural control structures rather than sensory information processing.

The central part of the human brain, the cerebrum, embodies the highest hierarchical control level, processing the most abstract and complex tasks. It is assumed to house a human's *intelligence*. Thus, active, voluntary, and self-aware mental processing and

[10] In contrast to *sensor neurons* and *motor neurons*, *interneurons* only connect between other neurons. Thus, *interneurons* implement pure processing purposes. [Moini 20]

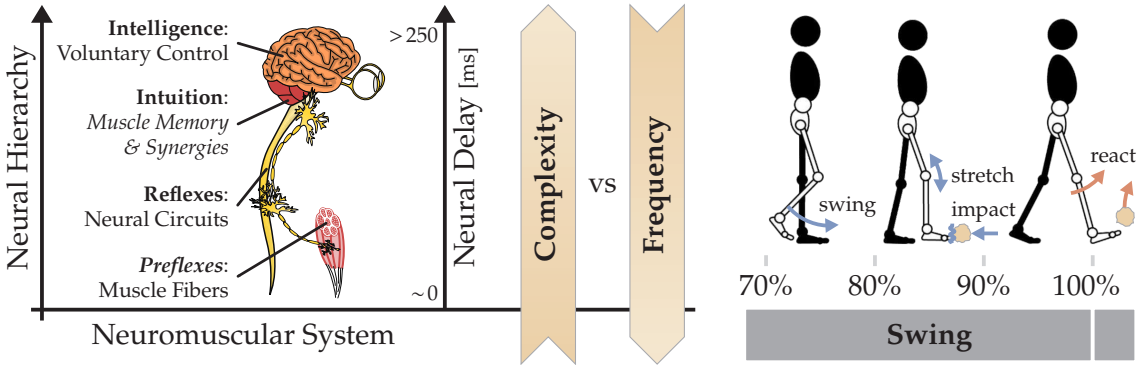


Figure 2.9: The hierarchical layout of the human neuromuscular system and an exemplary reflex reaction during leg swing while walking under disturbances. Four characteristic control hierarchies are distinguished from the tunable mechanical properties of the muscle fibers to the cerebrum, the primary location of intelligence in humans. Respective reaction delays last from zero at the *preflex* level to over a quarter of a second for a cerebrum reaction[11]. As an example of a reflex loop, one action possibility of the well-known *patellar reflex* during normal walking is sketched on the right. The patella tendon is unintendedly stretched on impact during the passive leg swing. Immediately, the so-called muscle stretch reflex shortens the affected muscles, creating additional knee torque that self-stabilizes the disturbed swing faster than the disturbance is signaled to the brain via neural transmission.

control is located within the cerebrum [Moini 20, chap. 6, p. 177]. The cerebrum must implement all the functionality that either cannot or has not yet been learned to be processed in hierarchically lower control layers, such as the cerebellum.

In the context of natural control, the cerebrum is mapped to the highest control layers, implementing the most complex skills. At this level, control problems are abstracted from the underlying body. Complex dynamic maneuvers are preplanned and mentally evaluated before being executed. Technical equivalents of the cerebrum's duties are:

1. Information abstraction, compression, and storage
2. Reasoning and scheduling
3. Multi-task coordination and goal fusion
4. Advanced, preplanned, and predictive control, such as in MPC

Still part of the brain but physically and functionally separated from the cerebrum, the cerebellum implements the next lower hierarchical level, the *intuition* layer. Experimental studies on the cerebellum strongly suggest this organ to be responsible for so-called *intuitive* reactions and movements [Thach 14]. Intuitive motions spread from simple motion primitives like hand, foot, or Tool Center Point (TCP) repositioning to highly complex movement sequences such as acrobatic acts or playing an instrument. The more complex, intuitive skills are commonly referred to as *muscle memory*. Equal to the intrinsic reflexes, intuitive motions are executed without self-aware thinking about the

[11] Since the exact amount of delay is irrelevant for the context of this analysis, the cerebrum reaction delay is the author's personal estimation out of multiple varying literature and other available sources.

tasks themselves. However, intuition can be learned and trained, and the resulting skills go far beyond the scope of natural reflexes. The motion primitives encoded within the cerebellum incorporate essential muscle recruitment, coordination, and basic motion abstraction. They implement the control interface between virtual, abstract command data from the cerebrum and the physical hardware, the muscles. Investigations on *muscle synergies* lead to the so-called *Minimum Transition Hypothesis*[Karniel 13]. This hypothesis postulates two essential properties of the humans' neural control architecture:

1. Control is hierarchically organized, with decreasing data dimensionality and control frequency in higher, thus more abstract, control layers.
2. *Learning* relocates control effort from mentally complex and exhausting higher control layers to more intuitive and mentally cheaper lower layers.

The latter is, e.g., being realized by establishing new *muscle memories* and synergies. Due to its task variety, the cerebellum embodies most of the data-related, functional, and architectural requirements for biologically characterized robot control:

1. Control hierarchy with the task complexity inverse to the control frequency Figure 2.9 visualizes these principles.
2. Multi-task and multi-control fusion
3. Muscle-, body-, or workspace abstraction and feed-forward control, i.e., forward and inverse kinematics and dynamics via *muscle synergies* and memory

Underneath the human brain, the spinal cord embodies the next lower layer of control. Its functionality is two-fold. Technically, the spinal cord implements the major data bus of the human CNS, connecting the neural CPU, the brain, with its peripherals. Functionally, it encodes the intrinsic, inborn neural feedback circuits, the reflexes. Hence, they can achieve very low latency between stimulus and respective reflex reaction initiation. Reflex circuits exist in a wide variety of complexity [Moini 20, chap. 19, p. 606]. The most primitive possible reflex, the stretch reflex, basically only involves two directly linked neurons: a *sensor neuron*[12] coupled to a *motor neuron*[13]. All other reflexes include at least one *interneuron* and may reach high levels of circuit complexity.

It is well-known that reflexes significantly contribute to stabilizing human motion, such as walking or hopping [Lance 81, Perry 92, Rose 06, Liu 18]. The most popular reflex is the *patellar reflex*, also known as the *knee-jerk* reflex. It is the muscle stretch reflex of the quadriceps. In medical investigations, this reflex is stimulated by tapping the patellar tendon to stretch the quadriceps muscle impulsively, consequently stimulating the reflex circuit. During walking, this stimulus is initiated by the external impact on the foot. A sketch of the *patellar reflex* response to an unexpected external impact during normal walking is shown on the right of Figure 2.9. The intrinsic reflex reaction stimulates additional knee torque, counteracting the disturbance [Lance 81, Garrett 83]. In the best case, the higher control layer can stay unaffected. If the disturbance is too strong, the

[12] *Sensor neurons* transport information from peripheral receptors toward the CNS. [Moini 20]
[13] *Motor neurons* carry information from the CNS toward peripheral effectors, such as muscles. [Moini 20]

reflex action overcomes the neural delay that the higher control layers require to react. A technical implementation of that reflex is realized in [*Sivak 24b*].

With the housing of the intrinsic reflexes, the spinal cord strengthens the technical derivation of a hierarchical control architecture, already drawn from the duties of the cerebellum. In contrast to the cerebellum, reflexes operate more or less at the level of unprocessed row data. Hence, the intrinsic reflexes can be seen to implement nothing but another lower layer of control with less abstract data but operating at a higher response frequency.

An analog argumentation can be made up for the lowest of the four mentioned hierarchical control layers in a human body. It comprises the so-called muscular *preflexes* [Brown 00]. These are tuneable physical properties, i.e., complex impedance profiles, built into the muscle fibers. Thus, these controllers are located outside the neural CNS. Studies on disturbed, voluntary human movements show an increased impedance to counteract those disturbances [Milner 98]. The more precision is required, the higher the impedance needs to be generated. It has been shown that the damping part of the impedance is the most relevant [Brown 00] and that the shape of the TCP impedance strongly depends on the limb configuration [Mussa-Ivaldi 85]. From the latter, it can be concluded that the impedance/damping is generated at the muscles' level rather than being rendered in a higher, actively controlled abstract workspace.

Via muscular co-contraction, a mechanical response to disturbances is (pre-) programmed into the musculoskeletal complex, thus implementing *zero-delay* reactions. From a technical point of view, this concept is inherently logical. Rendering damping requires a much higher control frequency and accuracy than stiffness but is also crucial for stabilization [Zhao 19]. However, the bandwidth of the neural system is quite limited. Mathematical analysis and simulative studies on the effect of such biological muscle characteristics validate that the existence of *preflexes* can reduce the required neural control load and can be tuned into proper movements [Haeufle 20, Izzi 23]. Hence, outsourcing damping from active neural control into mechanics is a consequent decision.

Technically, actuator-level *preflexes* might be approximated by so-called *smart actuators* [*Vonwirth 23a*], or be achieved by using actuation concepts with similar properties. Most popular such concepts exploit antagonistic mechanical springs, either coil-based, as present in the BioBiped3[14] [Sharbafi 16, Sharbafi 17b], or air-based, as is the case in the EPA-Hopper II[15] [Mohseni 22, Silva 24]. Other designs rely on antagonistic mechanical linkage constructions to operate on only a single mechanical spring [Wolf 11]. Please note that either nonlinear springs or nonlinear mechanical linkages are required to achieve a mechanical variable stiffness.

In contrast to the other two examples, the hardware of BioBiped3 cannot offer such a mechanical variable stiffness. Biologically, it is quite likely that the *preflex* realizing mechanism in nature is the *short-range stiffness* of the individual muscle fibers [Hufschmidt 87, Campbell 98, Groote 17]. Unfortunately, due to the limited existence of interdisciplinary medical-robotic research, it is hard to make a definitive statement.

[14] BioBiped3 is a planar scientific lower body constructed at the *Technical University of Darmstadt* featuring series elastic actuation.

[15] EPA-Hopper II is a planar, 3-DoF robotic research leg designed at the *Technical University of Darmstadt*.

The technical advantages of distributed robot control systems over centralized and distributed approaches are theoretically studied in detail in [Zhao 19]. In their work, they conclude that a partially distributed control strategy, decentralizing only time-critical control properties, can improve stability against disturbances while maintaining the advantages of centralized control computation.

From the characteristic layout of the natural neuromuscular system and its response times, the following technical requirements for natural robot control can be drawn:

Requirement 2.5: Hierarchical Control

The human's Central Nervous System (CNS) and its attached muscular complex implement a characteristic control hierarchy. Within this architecture, information abstraction and control frequency occur inversely to each other throughout all hierarchical layers, from muscle fibers to the cerebrum.

implications

Robot control architectures in favor of natural control, as demanded in Theorem 1.1, must support similar hierarchical control characteristics:

Hierarchical Software and Hardware Architecture

To realize the hierarchical response character of the natural neuromuscular system, an equivalent hierarchy must be reproducible by natural control software and its underlying hardware. Distributed and concurrent computations should be supported to process all the hierarchical control layers in parallel.

Control Fusion

Both cooperative and competitive control data fusion are required to support the natural multitasking capability.

Transformation Invariance

Data must be abstractable to operate in higher control layers.

Due to the special case of transformation singularity, Requirement 2.4 already demands support for non-injective transformations. In Section 2.1.6, this reason will be generalized to support arbitrary state transformations. I.e., sensor data are being compacted, while control data are being extended.

Frequency Invariance

Due to the multilayer control loops, all data must be super or subsampleable by different control hierarchies and control frequencies, especially hierarchically underlying control layers.

The above implications mainly derive from and focus on the actual robot control. However, the identical requirements apply to sensor data processing. Therefore, both control and sensor computation can and should be embedded into a single combined data calculus. For example, data precision, as introduced in Requirement 2.3, is equally suitable for fusing multitasking control or information from multiple sensors.

2.1.6 Model Reduction – SLIP-based Locomotion

In recent decades, general control theory evolved from simple Single-Input Single-Output (SISO) systems to high-dimensional Multiple-Input Multiple-Output (MIMO) ones. Today's fully-featured humanoid robots embody DoF in the high, double-digit range. For example, the humanoid robot Valkyrie[16] [Paine 15, Radford 15] even offers an enormous amount of 44 DoF.

In parallel with the systems' complexities, stability analysis has evolved from simple, *Laplace-Transformation*-based approaches to eigenvalue analyses of high-dimensional state space-based Linear Time-Invariant System (LTS) models, which are now suitable for verifying Lyapunov stability [Franklin 18]. At the same time, control methodologies evolved from simple PID controllers for SISO systems to OC approaches, which are also suitable for MIMO systems. State-of-the-art MPC approaches are necessary to control passive DoFs, as present in all floating-base systems, including legged robots [Felis 16, Schwenzer 21]. A collection of more than 30 legged robots known to utilize MPC between the years 2002 and 2022 can be found in [Katayama 23].

Unsurprisingly, the computational power required by those modern control techniques has evolved, too. With today's control demands, these approaches have reached the state of exceeding modern CPUs' capabilities. Consequently, investigations have started to decompose such complex systems into computationally cheaper, interconnected subsystems [Khadke 21]. Thus, these subsystems operate at individual, reduced models.

As argued, model reduction reflects the problems and limitations of classical modern robot control. However, it is also a biological model and, thus, a significant natural control characteristic. The most prominent and probably oldest simplified model is the so-called Inverted Pendulum (IP) model [Alexander 76, Mochon 80]. It has been used for over 300 years to describe the major motion characteristics of natural, legged locomotion [Borelli 85]. Thus, this model has extensively been studied, modified, and extended throughout the past centuries.

In their book *"Bioinspired Legged Lomotion"*, Sharbafi and Seyfarth have listed and categorized a collection of more than 15 individual variations and extensions of the IP model [Sharbafi 17a, chap. 3, p. 110]. However, the most common, versatile, and powerful one is one of the more primitive extensions of the IP model, the so-called SLIP model, e.g., [Blickhan 89, McMahon 90].

In their work [Schumacher 20], Schumacher et al. not only map the SLIP to the whole set of locomotion subfunctions spanning over balance, stance, and leg swing [Song 16] but also correlate those models to the human musculoskeletal apparatus. Although simple, the SLIP model in its dual-leg configuration is capable of representing all types of planar, bipedal locomotion considered within this work. Figure 2.10 shows a set of SLIP interpretations of the different types of locomotion.

Walking and running modeled as SLIP are *"just two out of the many stable solutions to legged locomotion of the same mechanical system"* [Geyer 06, p. 2865] [Antoniak 19]. The plots on the left of Figure 2.10 show multiple distinct areas of individually stable dynamics.

[16] Valkyrie is a humanoid research robot developed and build by the *NASA's Johnson Space Center*. URL: `https://www.nasa.gov/technology/r5` (2025)

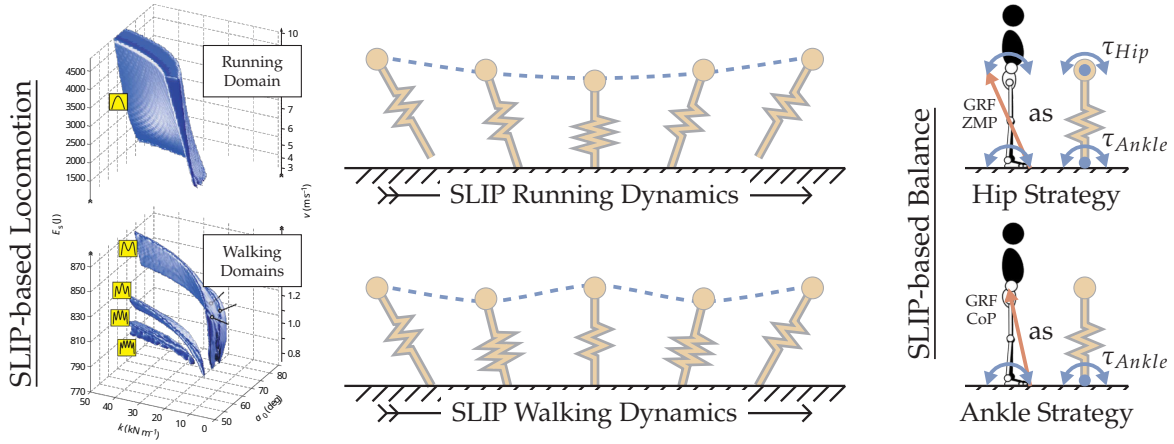


Figure 2.10: Visualizations of the SLIP model characterizing four types of bipedal motion: running, walking, and balancing via hip and ankle strategy. The stable dynamic domains for running and walking motions as individual solutions to the same system are shown on the left. Both left plots originate from [Geyer 06, p. 2864]. Right next to these plots, in the middle, both respective temporal behaviors of the SLIP abstraction are sketched. Note that the lower one, the standard walking dynamics, has already been introduced as a skeletal walking gait overlay in Figure 2.8. The two natural balancing strategies, the hip and ankle strategies, are sketched on the right, along with their respective SLIP abstractions.

Stable running (the upper plot) is achieved within the domain featuring only a single GRF peak. All others (the lower plot) are considered within the walking domain. The double-peak domain represents normal walking. Continuing the series of multi-peak dynamic solutions to the SLIP model, it can be hypothesized that the eventual result at infinite GRF peaks per gait cycle approximates in-place hopping as a simple, upright spring-mass oscillator.

Right next to these graphs, the respective temporal SLIP behaviors of running and walking are sketched. Note that the SLIP walking dynamics have already been introduced in Figure 2.8 within Section 2.1.4. The dashed blue curves visualize the amplified Center of Mass (CoM) motions, highlighting the single and double GRF peaks for running and walking, respectively.

In the case of planar, upright standing, both legs are usually assumed to be entirely aside from each other. Hence, w.l.o.g., the dual-leg configuration, in this particular case, can be degenerated to a single-leg SLIP. The identical argumentation holds for standard, in-place hopping [Blickhan 89, McMahon 90, Tamaddoni 10, Sharbafi 13]. However, this reduced representation still models both natural upright balancing approaches, the so-called *ankle strategy* and the *hip strategy* [Creath 05, Stephens 07, Zhao 14]. Both strategies are sketched on the right side of Figure 2.10.

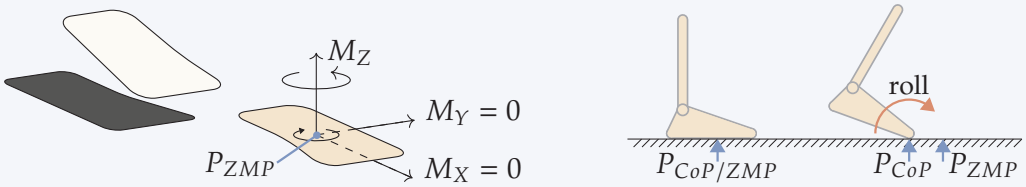
The so-called *ankle strategy* is a quasi-static approach. By only applying torque on the ankle, the Center of Pressure (CoP) is actively moved within the support polygon, generating horizontal GRF components to balance the CoM. Hence, the *ankle strategy* is also technically named CoP-balancing. Since the CoP is restricted within the support polygon area, the *ankle strategy* is equally limited. Definitions of the Zero Moment Point (ZMP), the CoP, and their relation to each other are given in the following:

Definition 2.3: Zero Moment Point

The so-called *Zero Moment Point (ZMP)* for legged robots is the unique point on the ground that does not experience non-vertical moments.

Assuming that the vertical moment is fully compensated by friction, the ZMP is the only point on the ground where the whole robot's dynamic moments sum to zero. I.e., the point where the robot could be pivoted in the upright equilibrium as an inverse pendulum.

The ZMP does not need to be inside its robot's support polygon. However, if inside, it always coincides with the robot's respective CoP (see Definition 2.4). If the ZMP is outside the support polygon the robot tilts around the support polygon edge towards the ZMP location.



On the left, the three-dimensional definition of the ZMP inside the single-support foot contact surface is sketched. The non-vertical moments are required zero, and M_Z is inherently compensated by an identical, frictional reverse moment within the contact plane.

On the right, the planar version of the ZMP is sketched. It is important to mention that the only unrestricted moment around the Z-axsis does not exist in the planar projection.

based on [Vukobratovic 69, Vukobratović 04]

Definition 2.4: Center of Pressure and Ground Reaction Force

1. The so-called *Center of Pressure (CoP)* is the unique point of attack of the GRF.
2. The so-called *Ground Reaction Force (GRF)* is the sum of all forces acting on the robot's contact areas with the environment (the ground).

relation to the ZMP

Unlike the ZMP, the CoP is strictly bound within the robot's support polygon. In fact, the CoP is the unique point inside the robot's support polygon that is closest to the ZMP.

The ZMP coincides with the CoP as long as both are located within the robot's support polygon. Whenever the ZMP is outside the support polygon, the CoP sticks to the closest edge. In such a situation, the robot dynamically tilts around that particular edge toward the location of the ZMP.

See Definition 2.3 for a planar illustration of those two cases.

The *hip strategy* is more dynamic than the *ankle strategy*, embodying a double IP rather than a single IP. By applying additional torque on the CoM, the robot's intrinsic inertia is exploited to achieve additional horizontal forces on the CoM. Hence, within the literature, the *hip strategy* is also technically called ZMP-balancing. Note that natural joints generally do not support infinite rotation. Therefore, except for circling the arms, any inertially initiated CoM motion must be stopped again after a short time by applying the opposite torque. Theoretical investigations and proofs of their respective stability regions for both ankle and hip strategy can be found in [Stephens 07].

Among all existing SLIP variations, the Trunk-SLIP is the most interesting extension [Poulakakis 09]. This extended model aims for a more realistic upper body abstraction and, thus, does not reduce the robot's trunk to a point mass at the hip joint. The most significant benefit of a modeled trunk is the support of upper-body balancing strategies, such as the Virtual Pivot Point (VPP), which is suitable for all motions: standing, hopping, walking, and running [Lee 17, Drama 19, Sharbafi 13]. However, no SLIP extension uncovers natural control characteristics beyond the scope of the original model.

In conclusion, the following technical requirement for natural robot control architectures can be derived from the characteristic SLIP model of human motion:

Requirement 2.6: Model Reduction

Natural, bipedal locomotion can be characterized by simplified dynamic models, such as the SLIP model.

implication

Natural control, as declared by Theorem 1.1, must support:

Arbitrary Control Abstraction

Simplified, abstract control models, such as the SLIP model, should be executed within their individual state spaces in potentially lower dimensionalities. Consequently, these simplified, abstract controls must be out-rolled while propagating through the control software to be eventually applied adequately and in full dimensionality to the robot's actuation system.

Note that Requirement 2.6 is related to the demanded transformation invariance in Requirement 2.5: *"Hierarchical Control"* within Section 2.1.5. With both Requirements combined, complex robot behavior is desired to originate from simplified subspaces and intuitively project into higher-dimensional operation and actuation spaces. Hence, control data must stay invariant against dimensionality-increasing transformations, while sensory data must be projectable into these subspaces.

2.2 Review of the Interdisciplinarity of Natural Control Engineering

Bio-inspired engineering of a robot in general, and a legged one in particular, is a highly complex and multi-disciplinary task. As usual, the weakest component limits the overall system performance. Thus, each discipline is crucial for achieving consistent, functional,

and stable robot control. In other words, the best hardware is worthless without a suitable controller, just as a perfect controller is useless without compatible underlying hardware. Thus, to successfully engineer a bio-inspired robot from design to operation, all individual building blocks must supportively intertwine with each other and cannot be addressed separately.

Throughout the development of a novel natural control system that satisfies all requirements derived from the six identified fundamental human gait characteristics, this thesis addresses and combines all of these individual engineering domains and relevant scientific disciplines. The individual disciplines covered underneath the herein-performed bio-inspired engineering are:

1. Mechanical Engineering:
   To construct hardware suitable to mechanically meet the desired task requirements and withstand the expected physical stress during the desired operations.

2. Electrical Engineering:
   To design an electrical layout covering the demands and interconnection of the robot's sensors, actuators, information communication, data processing, control, energy supply, and management.

3. Control Engineering:
   To enroll a multi-level, most likely multi-abstraction control system suitable to run on, drive, and stabilize the combined electromechanical construction.

4. Computer Science:
   To offer the necessary digital infrastructure of data encodings, communication protocols, or information processing. All follow the objective of meeting the controllers' requirements, e.g., data precision, data integrity, information delay, control frequency, real-time guarantees, or processing parallelization.

5. Physics:
   To model, predict, and eventually execute the robot's body behavior, desired motions, and tasks within the real physical world.

6. Mathematics:
   To efficiently and reliably model and implement all of the above problems throughout all layers of abstraction.

7. Biology, Physiology, and Neurology:
   When aiming for bio-inspired robot engineering, all the non-engineering but biology-related disciplines must additionally be involved. As introduced in Section 1.1, observations and insights from these fields serve as blueprints, objectives, and characteristic desires introduced and discussed in Section 2.1.

Since the six natural gait characteristics are identified independently of technical constraints, their derived requirements are spread across all the above-mentioned disciplines. Thus, it is essential to emphasize that achieving each of the identified natural control requirements is a challenging task that requires novel interdisciplinary

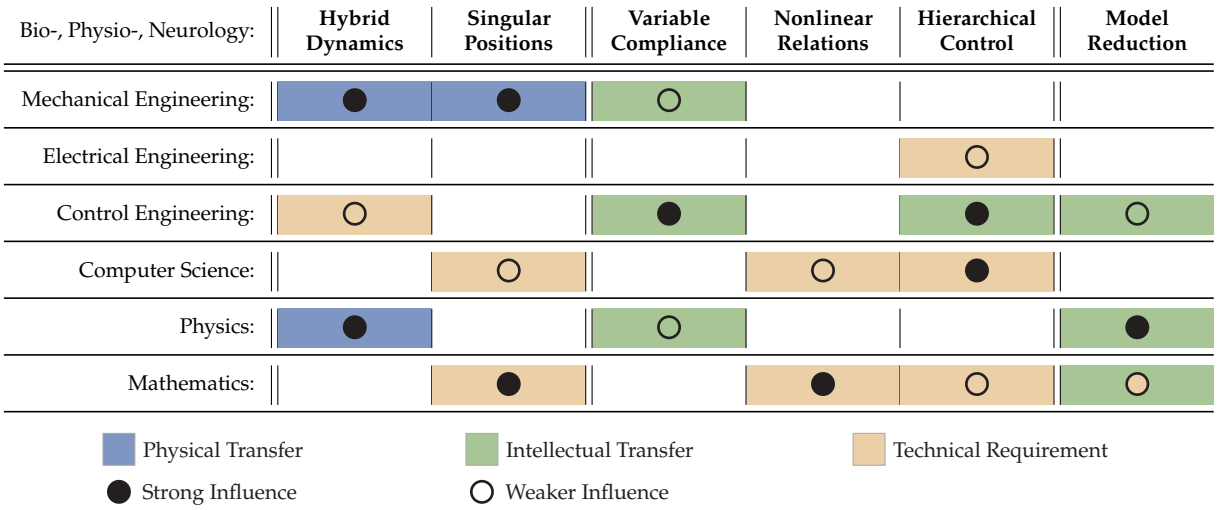


Table 2.1: Overview of the affected bio-inspired robotics-relevant engineering disciplines and their correlation to the six fundamental natural control requirements derived throughout Section 2.1. The requirements are ordered by increasing levels of abstraction according to their underlying natural gait characteristics, following the classifications in Definition 2.1: *"Categories of Natural Control Characteristics"*. Discipline and characteristic-individual bio-inspired engineering categorization, as defined in Section 1.1, are indicated by respective background colors. Each filled circle in a table cell represents a strong influence of an engineering discipline on the satisfaction of the respective requirement. Unfilled circles denote weaker influence. Empty cells represent no significant correlation between the requirement and engineering discipline.

methodologies and solution approaches. Table 2.1 provides an overview of the scientific disciplines and the impact of the six fundamental requirements on each of them.

The two left-most columns represent the requirements of visible characteristics. These requirements are the only ones whose underlying natural characteristics can be partially achieved through the straightforward physical transfer process, as described in Section 1.1 and visualized in Figure 1.3. A blue background color indicates such a property.

Subsequently, the requirements of the three measurable characteristics follow. As their underlying characteristic properties are no longer copyable, these requirements already require the highly speculative reverse engineering process of intellectual transfer. Similar to Figure 1.3, these properties are shown with a green background.

The right-most column of Table 2.1 contains Requirement 2.6: *"Model Reduction"*, derived from the only herein-identified hypothetical characteristic. Since the outcome of this characteristic is neither observable nor measurable, all its interdisciplinary influences are marked as speculative intellectual transfer. However, the mathematical formulations used to model and achieve this particular requirement are still a technical issue. The orange infill within the green background indicates this special double role.

In addition to the functional aspects, further technical requirements inherently arise from the methods and technologies utilized to achieve the six basic requirements. They include concrete needs for particular methodologies, specific modeling techniques, or dedicated algorithms. An orange background indicates that such technical requirements are disclosed throughout this work.

2.3 Review of Bipedal Robots and Control Approaches

None of the biological characteristics identified throughout Section 2.1 and their related natural control requirements are entirely new to the robotics community. Throughout the past decades, researchers worldwide have addressed all of these requirements within all individual disciplines listed in Section 2.2. Today, those efforts have already achieved robots with impressive legged locomotion skills. However, the most advanced approaches are purely technical, generating partially nature-like solutions only randomly undesired, such as [Agility Robotics 22]. Natural locomotion, especially natural gait, remains yet unsolved in modern robotics.

Therefore, a large variety of both individual robot designs and appropriate control approaches can be found in the literature. Extensive but non-exhaustive overviews of hardware designs and control philosophies throughout recent history have been collected by Ficht and Behnke in 2021 [Ficht 21] and by Mikolajczyk et al. in 2022 [Mikolajczyk 22]. The following examples should only offer a rough overview of this existing variety. A representative set of different methodologies is discussed in detail according to their compliance with the six prior-derived natural control requirements throughout Subsections 2.3.1 to 2.3.3.

One example of a different hardware design can be found in the leg morphology of bipedal robots. The vast majority of humanoids are equipped with an anthropomorphic leg design that features human-shaped, two-segmented limbs with forward-bending knees. An example of that design is the planar robotic platform Carl[17] [Schütz 17][Schütz 20a]. It is specially designed to follow natural, human-like properties. Contrastingly, the bipedal robot Digit[18] features bird-like-looking, backward-folding legs. However, Agility Robotics themselves state that their design is neither meant to mimic bird legs nor is biologically inspired [Agility Robotics 22]. It is a result of physical and task-oriented hardware optimization. To their understanding, the fact that it looks natural is just a good indicator of a properly chosen design decision.

Two examples of different actuation principles are realized in the two humanoid robots, Lola[19] [Lohmeier 09, Buschmann 09] and Nadia[20]. Lola is classically equipped with electric motors and relatively high gear ratios [Lohmeier 09, Buschmann 09]. Nadia, in contrast, features additional hydraulic, so-called *smart actuators* [Barasuol 18]. It is clear that the question of an appropriate actuator-level control must be raised together with the choice of the mechanical propulsion solution. Unfortunately, this question is still an open research topic.

The implementation of the absence of one definite answer to that question can be seen within the humanoid robot Valkyrie. Its actuators are equipped with even three contradicting variants of actuator-level control opportunities to choose from. These are torque, impedance, and position control [Radford 15, Paine 15]. However, although the

[17] Carl (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU.
URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025)

[18] Digit is a commercial humanoid developed and produced by the company *Agility Robotics*.
URL: `https://agilityrobotics.com` (2025)

[19] Lola is a scientific humanoid robot developed and built at the *Technical University of Munich*.

[20] Nadia is a humanoid commercially developed and produced by the company *Boardwalk Robotics*.
URL: `https://boardwalkrobotics.com` (2025)

availability of such a variety of control approaches within a single robot is perfect for research, it must also be claimed that such a system must not be the eventual solution. Consequently, all those approaches appear on the list of herein-discarded strategies within Figure 1.3.

Continuing from lower to higher control levels, the variety of existing strategies spreads even more. Depending on the hardware below, different control philosophies, hierarchies, decomposition frameworks, models, and other techniques have been developed, applied, and tested on robot hardware platforms and simulations. Their span starts at centralized, whole-body dynamics controllers like, e.g., [Hosokawa 18], or lower-dimensional control projection models like, e.g., [Hyon 09]. It continues via optimization-based approaches that primarily use a model for parameter tuning but later may operate without one, e.g., [Geijtenbeek 13]. Please note that modern Artificial Intelligence (AI)-based approaches, e.g., [Li 21], are also part of this optimization-base class of control strategies. Last, model-free strategies are based on, e.g., pattern generators and reflexes [Luksch 10a].

A good overview and review of early bipedal robots and their respective walking control approaches has been collected by Al-Shuka et al. in 2014 [Al-Shuka 14]. In their work, they also order the different control approaches by defining different classes and subclasses. Those are model-based gaits, motion-capture-based gaits, trajectory-based gaits, CoM-based gaits, optimization-based gaits, biological mechanism-based gaits, Central Pattern Generator (CPG)-based gaits, AI-based gaits, or natural dynamics exploiting gaits. Although those classes still do give an overview of the underlying technologies, most modern gait approaches are hybrids of multiple classes. Pressing the existing control strategies into such a strict classification is, hence, hard to argue and may vary with the context and purpose of the classification.

Following the bio-inspired background and aim of this work, the following sections group a representative but non-exhaustive set of different control strategies according to the naturalness of their actuation systems. The list is limited to bipedal robots and bipedal simulation control. Hence, solutions for quadruped systems such as Bert[21] [Lakatos 18, Pollayil 22] or ANYmal[22] [Hutter 16, Hutter 17, Fankhauser 18, Shi 21] are not included. Their primary focus is on the principal evaluation of the described technologies against the six previously derived natural control requirements, not their actual capabilities or real-world performances.

2.3.1 Rigid Robot Control Approaches

Two popular and successful approaches that closest embody the so-called classical robot control are the Divergent Component of Motion (DCM) gait generation principle and the Hybrid Zero Dynamics (HZD) trajectory optimization strategy. The two approaches are experimentally evaluated by executing stable walking on the two bipedal robots TORO[23] [Englsberger 13, Englsberger 14] and DURUS[24] [Hereid 16, Reher 16] respectively.

[21] Bert is a compliant quadruped robotic research platform developed by the *German Aerospace Center*.

[22] ANYmal is a previously scientific, yet commercial quadruped robot by the company *ANYbotics*. URL: `https://www.anybotics.com` (2025)

[23] TORO is a torque-driven humanoid robot developed and operated at the *German Aerospace Center*.

[24] DURUS is a bipedal robot developed and designed at the *Caltech California Institute of Technology*.

The gait generation concept of DCM [Englsberger 13, Englsberger 15, Hosokawa 18] is a three-dimensional extension of the two preliminary, two-dimensional concepts named *Capture Point* [Pratt 06, Englsberger 11] and Extrapolated Center of Mass (XCoM) [Hof 08]. All three focus on upright balancing via pre-planned footstep placement adaptation during locomotion. Equal to the XCoM, the DCM is a virtual point in cartesian 3D space derived from the CoM and its velocity. Once a stable DCM trajectory is calculated, the robot's CoM will smoothly follow that trajectory.

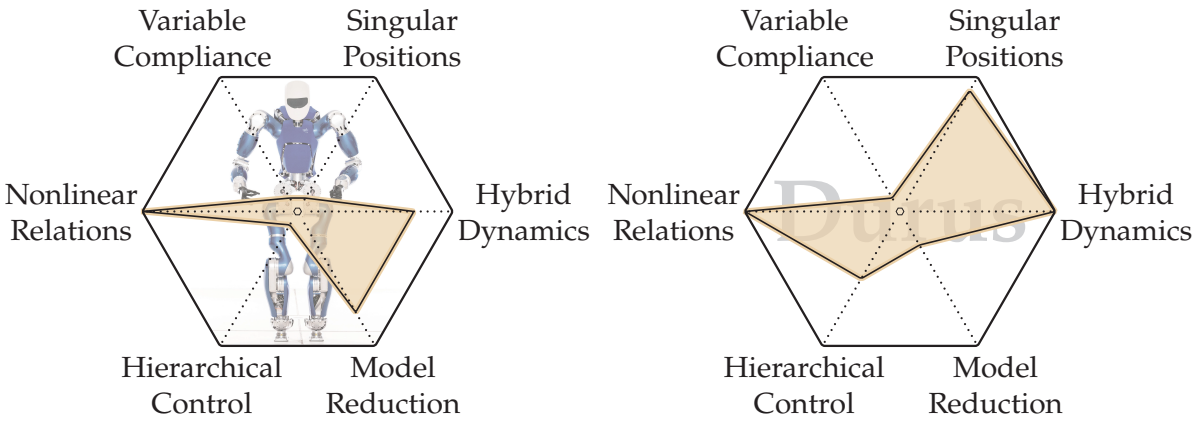
A DCM trajectory is stabilized by so-called *virtual repellent points* pushing the DCM trajectory away from them. These virtual points are complete representations of the robot's later footsteps during the pre-planned walking gait. With alternating *virtual repellent points* to the left and right side of the desired DCM trajectory, the robot's CoM and, therefore, the overall walking gait can be stabilized in time between the alternating footsteps. Moving from one footstep to the next is then executed by, e.g., a standard whole-robot rigid body dynamics controller.

In contrast to the DCM, both preliminary approaches, *Capture Point* and XCoM, are restricted to the mathematically ideal linear inverted pendulum model. It is important to note that the DCM strategy solely aims for gait stability. Typical other control goals like energy consumption or naturalness of the gait are not addressed.

A completely different motivation than pure mechanical stability is behind the gait trajectory generation framework of HZD. Assuming the human walking gait has been optimized by evolution, it seems a valid approach to copy and imitate the trajectories of human gait cycles from, e.g., motion tracking experiments. Similar to the DCM approach, a standard, whole-robot rigid-body dynamics controller may track these natural trajectories. Of course, to be mechanically able to track such trajectories, the robot's skeletal morphology must be similar to a human body shape. A simple proof of functionality of that assumption has been given on a single, pneumatically-driven, compliant leg in [Lin 15, Shen 15]. However, simultaneously, it becomes clear that this pure trajectory tracking approach is neither energy efficient on the deployed hardware nor stable against disturbances or noise.

To address these problems, the captured motion trajectories are offline optimized for, e.g., minimal cost of transport on the dynamics of the bipedal robot DURUS [Cousineau 15, Hereid 15, Hereid 16, Reher 16]. The nonlinear optimization is formulated using precisely modeled HZD [Westervelt 07, Grizzle 14, Ames 14] and hand-chosen virtual constraints, such as ensuring the conservation of smooth forward walking during optimization [Hereid 16]. Extended with additional, reflex-like feedback modulation components, the HZD framework can generate stable, real-world walking on flat ground, as demonstrated with the bipedal robot DURUS [Reher 16]. Non-empirical evaluations of both approaches and their respective robots regarding the six initially determined fundamental natural control requirements are plotted in Figure 2.11.

Hybrid dynamics, required by the first and most essential natural control requirement, are completely modeled within HZD but only partially supported by the DCM approach. While alternating footsteps are supported, the passive ground contact phase between the stance leg's heel-off and the swing leg's touch-down is not explicitly designed. See Figure 2.5 for the illustration of the natural passive support phase. However, such behavior could be integrated into the footstep tracking controller when allowing the



(a) Divergent Component of Motion (DCM) **(b)** Hybrid Zero Dynamics (HZD)

Figure 2.11: Non-empirical illustration of the approach-individual compliance with the six identified fundamental natural control requirements for the two rigid robot control approaches (a) DCM and (b) HZD applied to the two bipedal robots TORO[25] and DURUS[26], respectively. The diagrams do not represent the fact that both approaches have been demonstrated to generate stable and reliable real-world gait.

robot to become temporarily unstable. Requirement 2.2: *"Singular Positions"* is no problem for the HZD approach because its optimized trajectories are tracked within the robot's singular-free Configuration Space (CS).

Since both above approaches operate on nearly rigid underlying hardware platforms, Requirement 2.3: *"Variable Compliance"* is not considered. In contrast, Requirement 2.4: *"Nonlinear Relations"* no longer excludes modern classical control approaches. While the DCM strategy does not feature a control hierarchy, the transformation from trajectory control to footstep planning can be acknowledged as a significant Work Space (WS) simplification, thus, partially satisfying Requirement 2.6: *"Model Reduction"*. Opposite, the HZD framework implements Requirement 2.5: *"Hierarchical Control"*. HZD tracks the optimal trajectory while using local feedback modulation as a secondary, reflex-like control hierarchy in parallel. On the other hand, model reduction only barely exists within the initial zero-dynamics model used for offline trajectory optimization.

Finally, it is essential to note again that the above analyses and their illustrated results in Figure 2.11 only address the controls' compliance with the six natural control requirements defined herein. Neither plot represents the fact that both approaches have been proven to generate stable, three-dimensional walking on real hardware. Furthermore, ongoing research [Pollayil 22] aims to efficiently apply the DCM trajectory planning to compliant robots, as demonstrated on the simple quadruped robot Bert. With this strategy, the DCM approach will be embedded in a motion generation framework similar to the HZD.

[25] Photo of TORO by the *German Aerospace Center*, published under Creative Commons Attribution License (CC BY 3.0) at https://www.dlr.de/en/images/2014/2/toro-walking-robot_14503, cropped

[26] Non of the publicly available photos of DURUS grants permission to reprint. However, a collection of pictures and videos of DURUS can be found in a 2016 IEEE Spectrum article available at `https://spectrum.ieee.org/durus-brings-humanlike-gait-and-fancy-shoes-to-hyperefficient-robots` (2025)

2.3.2 Compliant Robot Control Approaches

The bipedal robot Cassie[27] and the humanoid robot simulation B4LC represent two entirely distinct control approaches. Nevertheless, both systems embody compliant robots with respective control systems. Both robots feature mechanical, passive compliances within their actuated joints. Within Cassie, these compliances are blade springs that embody soft lever arms driving the joints. In contrast, the B4LC system simulates programmable, tunable rotational impedances directly built into its links. Note that both approaches, although introducing compliance, preserve the classical robotic layout characterized by actuated rotational joints and rigid body segments.

The Cassie platform is the commercialized successor of the research platform ATRIAS[28] [Hubicki 16, Rezazadeh 18] and the predecessor of the complete humanoid robot Digit. Due to its commercial availability, Cassie has been used by many research groups across various institutes with numerous control approaches. Strategies applied to Cassie include model-based control [Reher 19], force-base control [White 20], and Reinforcement Learning (RL)-based control [Xie 19, Li 21]. The model-based control relies on a fast, online optimization strategy that uses the whole robot's dynamics, including its compliant elements.

Regarding the model-based approach, things become a lot simpler. The robot is abstracted as a kind of a single IP model supported by two massless legs. A simple PD-Controller with the Jacobian transpose is then applied to distribute a virtual workspace spring-damper system to the joint motors to keep the robot upright and balanced. This setup has the advantage of not caring about the foot placement or movement.

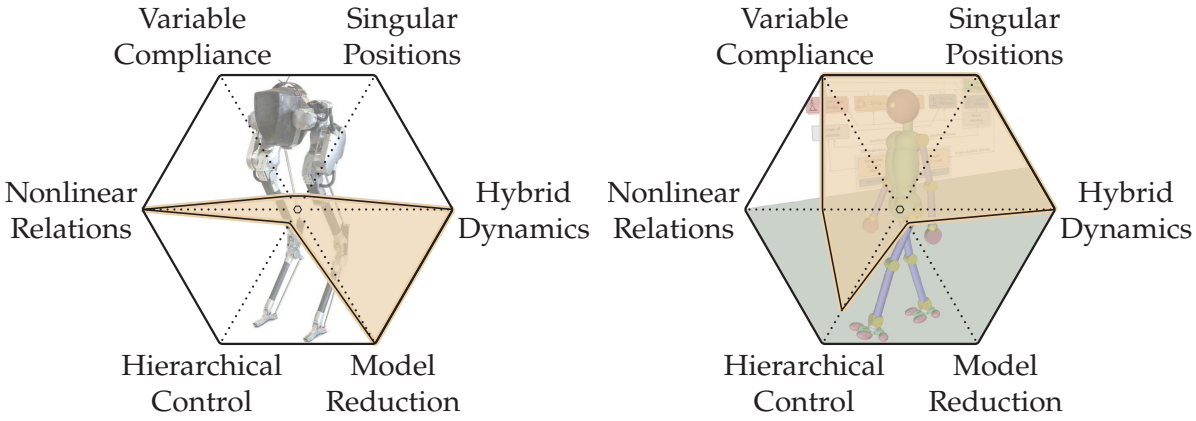
Opposite to that, the last approach is again based on multiple technologies and precise dynamic models. In their work [Li 21], Li et al. use the previously introduced HZD framework to generate training data in a simulation setup. Then, they used model-free RL and parameter randomization to train robust control policies. Those policies are then transferred from the simulation onto the actual hardware.

Different, but again similar to the HZD approach, is the control of the B4LC robot simulation. Within this approach, a CPG serves feed-forward joint torque and impedance profiles and activates a set of local reflexes for stabilization against disturbances and noise [Luksch 10a, Luksch 10b]. The system's modularity allows Zhao et al. to extend the original control structure with more advanced features like slope walking and push recovery during gait [Zhao 13, Zhao 15]. Eventually, optimization and learning-based approaches have been used by Liu et al. to tweak the system's parameters for robustness against uneven ground or stepping over obstacles [Liu 15, Liu 16]. With this final optimization step, the B4LC control philosophy matches the basic principles of the HZD framework. These are, first, to provide an optimal feed-forward control exploiting the system dynamics and, second, stabilizing those with arbitrary, additional, parallel reflex-like components.

As is visualized in Figure 2.12, the two robotic systems, Cassie and B4LC, are almost opposite. Cassie is explicitly not designed to mimic some biological counterpart

[27] Cassie is the bipedal predecessor robot of Digit, both produced by the company *Agility Robotics*. URL: `https://agilityrobotics.com/` (2025)

[28] ATRIAS is a compliant bipedal robotic research platform at the *Oregon State University* to exploit SLIP-based walking.



(a) Cassie – Various Control Approaches **(b)** B4LC Biped Simulation

Figure 2.12: Non-empirical illustration of the robot-individual compliance with the natural control requirements derived from the six fundamental human gait characteristics for the two compliant robots (a) Cassie[29] and (b) B4LC. The plots do not reflect that Cassie has been evaluated with various control strategies and successfully demonstrated its capabilities under numerous real-world conditions. In contrast, the B4LC system is a pure virtual simulation model.

[Agility Robotics 22]. It results from pure physical, functional, and rational design optimization. Figure 2.12a precisely reflects that motivation.

From the six natural control requirements derived from the human gait characteristics, the robot Cassie satisfies only those that are required for operation within natural environments. Thus, the two *naturalness* Requirements 2.2: *"Singular Positions"* and 2.3: *"Variable Compliance"* are not addressed. However, the latter might have changed within Digit, Cassie's complete humanoid successor robot, due to the new requirement of handling unknown objects with its added arms and upper body. Unfortunately, Digit is a commercial product, and its internal control is no longer publicly available. Requirement 2.6: *"Model Reduction"* is achieved using the well-known abstract SLIP model, which is the simplest, stable, natural-dynamic walking controller. SLIP-based walking control is already known from Cassie's predecessor, the bipedal robot ATRIAS.

Figure 2.12b pictures the B4LC simulation system design. Except for the essential hybrid dynamics, the B4LC system addresses mostly the opposite requirements than Cassie.

The B4LC's research intention was specifically to study human-like locomotion and appropriate control. Thus, its primary focus is on the more biological control Requirements 2.2: *"Singular Positions"*, 2.3: *"Variable Compliance"*, and 2.5: *"Hierarchical Control"*. On the other hand, the more technically oriented Requirements 2.4: *"Nonlinear Relations"* and 2.6: *"Model Reduction"* are only barely present. The B4LC control operates on hand- and optimization-tuned joint-level impedance control, i.e., within the robot's CS. Consequently, the emerging gait does not require care or avoidance of knee singularities and appears human-like. However, its functional capabilities, as represented in Figure 2.12, are limited in comparison to the Cassie robot.

[29] Photo of the robot Cassie taken from *"NU-Biped-4.5: A Lightweight and Low-Prototyping-Cost Full-Size Bipedal Robot"* [Folgheraiter 23], published under Creative Commons Attribution License (CC BY 4.0), doi:10.3390/robotics13010009, cropped

These functional limitations also emerge from the B4LC system being a pure virtual simulation. To overcome this issue, the bipedal robot CARL has been designed and constructed as a real-world platform, offering all the necessary hardware and control features to run a B4LC-similar control concept on it [Schütz 17][Schütz 20a]. However, with the first practical experiments on CARL, issues related to real-world physics, control complexity, reliability under changing environmental conditions, and stability have been uncovered [*Vonwirth 17*]. Hence, further research toward natural control, as declared by Theorem 1.1: *"Natural Control"*, is continued by the work at hand.

2.3.3 Muscular Robot Control Approaches

Control and underlying hardware design are closely coupled and should be considered a single unit representing a robot's whole actuation system. Hence, the most natural-like control approaches are the ones that already build upon muscular actuation principles. The probably closest mechanical construction featuring the most significant muscle properties is an air-inflatable Pneumatic Artificial Muscle (PAM). Invented as so-called McKibben muscles in the 1950th, air muscles are still under development [Na 23].

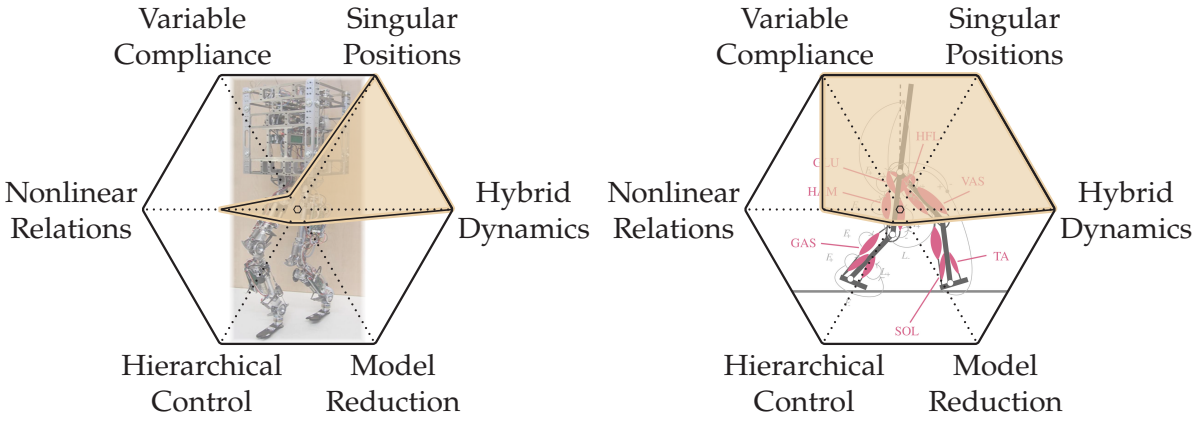
One example of a bipedal lower-body robot featuring PAM actuation is the PNEUMAT-BH[30] [Shimizu 12, Kawakami 15]. Its McKibben muscles are placed according to the human leg physiology, including mono- and biarticular muscles. Similar to the previously introduced B4LC simulation, the control of PNEUMAT-BH relies on a feed-forward pattern combined with event-based switching [Kawakami 15]. However, due to the limited usable range of motion, the slow controllable bandwidth, and the low energy efficiency of pneumatic drive trains, those types of robots have disappeared nowadays. Thus, muscular robot control is represented by two other representatives.

One example is the planar BIOBIPED3 lower-body robot featuring Series Elastic Actuator (SEA)-emulated muscles in a human-like morphology. The second representative is a pure simulation model controlled by reflex-based activation of so-called *Hill-type* muscles [Song 15, Hill 38]. As with the PNEUMAT-BH, both robotic research platforms implement control philosophies similar to the B4LC simulation. However, within those two systems, these philosophies are applied to muscle-based actuation systems or simulations rather than traditional joint-based actuation.

BIOBIPED3 is the result of a continuous bio-inspired engineering process. Former versions of this robot can be found in [Scholz 11, Radkhah 12]. BIOBIPED3 is equipped with cable-driven SEAs. Hence, each muscle is realized by a motor tensioning or releasing a cable with an integrated linear spring. Such a cable design restricts the involved motor to pull-only operations. Consequently, similar to nature, at least a pair of two antagonistic motors is required to control a single joint of BIOBIPED3.

Due to the linearity of the springs used in BIOBIPED3's hardware setup, it is not possible to alter a joint's mechanical stiffness by muscular co-contraction. Hence, the control relies solely on the suitability of the predetermined spring constants and active control of the spring rest lengths. The robot's motion eventually emerges from its intrinsic dynamics [Sharbafi 16, Sharbafi 17b].

[30] PNEUMAT-BH is a scientific, pneumatically-powered, human-like lower body built at the *Osaka University*.



(a) Muscular Rest Length Control **(b)** Muscular Reflex Control

Figure 2.13: Non-empirical illustration of the two muscular control approaches (a) and (b) and their individual compliance with the natural control requirements derived from the six initially identified human gait characteristics. Both approaches exploit and rely on the robots' natural dynamics to generate motion from muscle stimulation. The muscular rest length control has been tested on the physical robotic research platform BioBiped3[31]. In contrast, the muscular reflex control is, as pictured[32], a virtual simulation approach. An early stage of this concept had been mapped to the kinematics of the ATRIAS robot and successfully demonstrated in [Batts 15].

Creating motion from intrinsic dynamics is also present within the reflex-based muscular control simulation by Song and Geyer, extending the earlier planar version by Geyer and Herr [Song 15, Geyer 10]. Recent research by Bunz et al. improves the system's robustness against perturbations [Bunz 23, Bunz 24]. Their control system is built upon two types of muscular reflexes: force and length feedback reflexes. Force feedback reflexes create muscle stimulation based on a linear transformation of a delayed muscle force. Analog to that, length feedback reflexes emit muscle stimulation based on a linear transformation of a delayed muscle length.

It is important to note that the muscle observed by a reflex does not need to be the controlled one. The overall muscular stimulation is the sum of all reflexes. From an abstract point of view, the whole set of reflexes implements non-zero elements of a big matrix, transforming observed muscular inputs into muscular simulation outputs. Off-diagonal elements for both force and length inputs realize muscular interplay.

Finally, a dynamic *Hill-type* muscle model generates forces from the muscular stimulations [Hill 38]. A similar approach can also be found in the work of Geijtenbeek et al., who optimizes a muscular feedback controller for many morphologically individual, bipedal creature simulations [Geijtenbeek 13]. Among their creature simulations, there are also different styles of human-like bodies.

[31] Photo of BioBiped3 taken from *"A new Biarticular Actuator Design Facilitates Control of Leg Function in BioBiped3"* [Sharbafi 16], published under Creative Commons Attribution License (CC BY 3.0), doi:10.1088/1748-3190/11/4/046003, faded

[32] Graphic of the muscular reflex control simulation taken from *"Bioinspired preactivation reflex increases robustness of walking on rough terrain"* [Bunz 23], published under Creative Commons Attribution License (CC BY 4.0), doi:10.1038/s41598-023-39364-3, elongated ground

Figure 2.13 notably highlights the primary focus of both approaches on the more natural control requirements rather than the more technical ones. Satisfying Requirement 2.1: *"Hybrid Dynamics"* is task-inherent. Support for Requirement 2.2: *"Singular Positions"* is achieved in both approaches as both operate the raw muscles, i.e., within their respective Actuation Spaces (ASs). However, the pure AS control simultaneously causes the absence of the two technical Requirements 2.5: *"Hierarchical Control"* and 2.6: *"Model Reduction"* for both approaches equally.

As is the case in nature, muscular co-contraction is used to modulate impedance within the muscle reflex framework, thereby satisfying Requirement 2.3: *"Variable Compliance"*. Due to the pure linear springs within BioBiped3, this method does not apply to the herein-referenced muscular rest length control.

Last, both approaches partially support Requirement 2.4: *"Nonlinear Relations"*, built into the nonlinear muscular attachments and intrinsic muscle and control properties. However, neither approach explicitly aims to achieve such a behavior. Instead, this is implicitly caused by the pure AS control, which allows the robot dynamics to initiate and define the eventual real-world motion.

2.4 A Robot Control System in the Context of Natural Control Requirements

As analyzed in the previous section, all of the fundamental natural control requirements have already been applied to robotic hardware and control systems. However, this analysis also reveals that no yet existing control framework aims to cover all of them. Motivated by the fact that biology yet remains far beyond the locomotion skills of today's robots, this open problem in modern bio-inspired robotic research is worth evaluating, shrinking, and eventually solving one day.

Similar to the individual engineering disciplines presented in Section 2.2, the six herein-defined characteristic natural control requirements can also be assigned to the different subsystems of a generalized robot control system. A sketch of such a system architecture is given in Figure 2.14. Throughout the following chapters of this work, this particular control system architecture is consecutively developed, deployed, and evaluated on the bipedal robot Carl.

The figure illustrates all essential components and the general closed-loop data flow that underlies every robot control system. Due to the versatility of robotic hardware, software, and control methodologies, the actual realization of each component may differ significantly between different technologies and controlled robots. Each component is annotated with a list of their most significant affecting natural control requirements. Similar to Table 2.1, the background colors indicate the nature of their application: either one of the bio-inspired engineering processes or a purely technical requirement.

The actual control components of the generalized sketch in Figure 2.14 are located at the bottom. They build three elements with data flowing from left to right. The high-level control, embodying all types of control architectures, is placed on the left. Its consecutive component to the right is the hardware interface, which converts any abstract control data emitted by the high-level framework into coordinated hardware-specific control

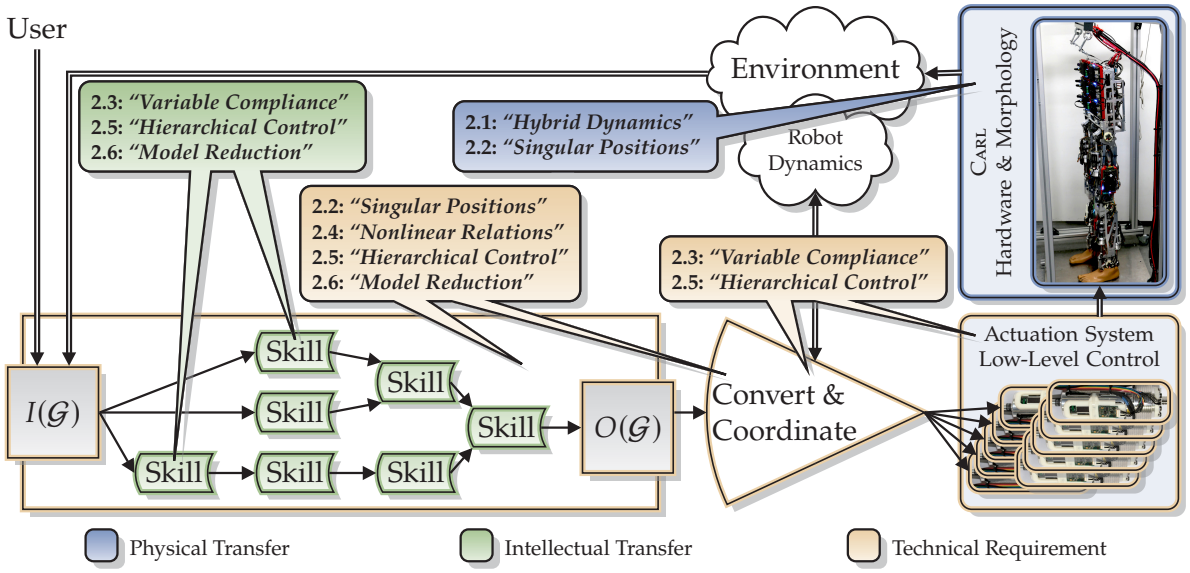


Figure 2.14: Sketch of the generalized control system of CARL featuring the same abstract components all robot control systems are composed of. Each component is individually annotated with the most significant related natural control requirements, as derived from the identified human gait characteristics throughout Section 2.1. The data flow in the sketched control loop is oriented anticlockwise. All novel control approaches developed herein are embodied within the bottom components. At the top, the robot's morphology, environment, and dynamics are closing the control's environmental feedback loop. Similar to Table 2.1, the background coloring of the requirement annotations classifies their primary achievement procedures as either one of the bio-inspired engineering processes, as defined in Section 1.1, or as purely technical.

commands. These control commands are then physically executed by the actuation hardware, whose abstract component is located at the figure's bottom right.

In classical robot control, the left-hand-side control component, for example, could contain a path controller. This controller would then be followed by an inverse dynamics model and, eventually, electric current-controlled joint actuators. In the context of legged robots, the path controller might be replaced by more advanced methodologies, such as an HZD-based trajectory optimization. Within this work, a novel Behavior-Based (BB) natural control architecture is developed for that purpose in the upcoming chapters.

On the top of Figure 2.14, the robot's morphology and environment close the robot's physical feedback loop via mechanical interaction with its environment. For simplicity, the robot itself is treated as part of the environment throughout this work. Furthermore, internal states, intrinsic and extrinsic models, are separated from the control and, without loss of generality, are also treated as part of the generalized environment. Thus, ideal virtual sensors and actors grant access to these model data. Additional user input is provided in the top-left corner but is irrelevant to the overall control design.

The robot's morphology, i.e., its mechanical construction, is characterized by mostly physical transfer of natural properties. Since the biological musculoskeletal apparatus has been optimized evolutionary throughout the past million years, this correlation seems intuitive. Hence, a geometrical copy of the human skeletal structure and muscle

placement probably results in a humanoid robot design that satisfies the natural control requirements and enables the respective identified gait characteristics. As this work relies on the previously built robotic platform Carl, its bio-inspired robot morphology will be described and discussed later. Detailed studies on the morphological musculoskeletal design of Carl are published in [*Nejadfard 18b*, *Nejadfard 19*][Nezhadfard 20].

Driven by *bio-inspiration*, intellectual transfer is used to engineer all the individual hierarchical control layers. These include the low-level control of the robot's mechanical actuation system, which is not illustrated in Figure 2.14, as well as arbitrary high-level control skills. Independent of the concrete realization, it is essential that all components, from the lowest to the highest abstraction level, fit and build upon each other. I.e., lower layers must offer the functionality required by the upper layers, while higher layers, in return, must provide the control commands required by the lower ones.

Due to the inherent complexity of robotic systems, a mathematical, technical framework is inevitable to consistently bind all these controls throughout all the hierarchical layers together. Such a framework must integrate robot-typical sense and control processing while keeping the overall control software manageable, understandable, and extendable. Figure 2.14 highlights these technical requirements by the orange frames of the left-hand-side high-level control component and the adjacent *"Convert & Coordinate"* hardware interface. It is easy to understand that this particular engineering part is a pure technical necessity to address the technical requirements that describe each of the initially identified, desired natural gait characteristics.

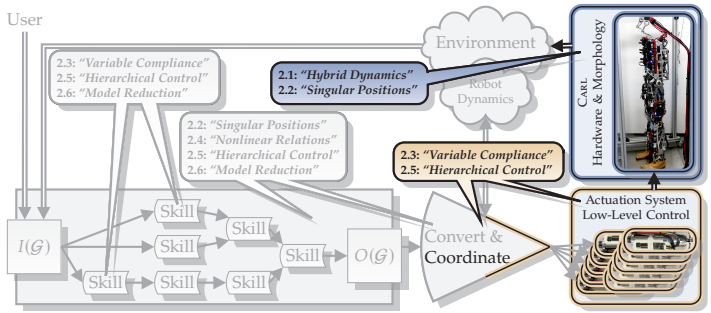
2.5 Conclusion

This chapter has identified six fundamental human gait characteristics and derived respective technical requirements intended to lead toward more natural robotics, specifically natural robot control, as declared in Theorem 1.1. All six characteristics are thoroughly derived from comprehensive analyses of a normal undisturbed human gait and its biological supporting structures. The identified human gait characteristics establish a solid foundation of derived bio-inspired technical requirements for natural robot control. Thus, these fundamental requirements can be applied across all robotic domains beyond the scope of this thesis.

The derived requirements have been attributed to all robotic-relevant engineering disciplines and evaluated against a representative variety of existing state-of-the-art methodologies. Eventually, these requirements are outlined on the individual subsystems of a general robot control system, as sketched in Figure 2.14.

In the following chapters, these different subsystems are individually addressed and developed according to their related natural control requirements.

3. Hybrid Impedance – A Robot's Musculoskeletal Control System



Modern robotic actuation systems can perfectly control industrial robot arms on extraordinary high-precision trajectories. In order to achieve this high performance, they often rely on high gear ratios within their joints and significantly bigger intrinsic inertia than their manipulated mass. Consequently, these classical robot designs are heavy, slow, stiff, and neither back-drivable nor adaptive to external disturbances. Unfortunately, all those properties are nearly opposite to what can be observed in nature, especially in the human body. However, a good reason for this difference is that natural locomotion features entirely different requirements than most modern industrial applications.

One of the most significant differences is that there is no need for perfect precision or repetitiveness in a typical natural bipedal walking scenario. Instead, impact tolerance (see Requirement 2.1), a fast reaction time, and energy efficiency are of much more relevance. Thus, it is intuitive to hypothesize that a suitable natural, legged locomotion control strategy should also differ from the approaches deployed in industrial automatization.

To develop such a locomotion-suitable control strategy, this work intentionally distances itself from the classical, well-known robot control approaches. Instead, the human musculoskeletal system serves as a blueprint for developing a novel control philosophy that satisfies the natural control requirements derived throughout Section 2.1: *"Characteristic Natural Control Requirements of a Human Gait"*.

Modern robot control requires suitable, well-matched components on all control-related disciplines, techniques, and hierarchical levels. Thus, a robot's morphology, mechanical

construction, electrical circuitry, and implemented communication protocols must support and integrate into the mechanical and electrical requirements of the actuation system and control strategy. At the same time, these low-level control demands must be matched and properly fed by high-level control infrastructure and abstract, semantic control policies.

All control-relevant components must cooperate to achieve the overall aim of achieving natural control, as declared in Theorem 1.1. The relevant natural control requirements, as illustrated in Figure 2.14, must be satisfied across the entire hardware and software system, respectively.

Highlighted by the initial top-left graphic, this chapter covers the development and discussion of the low-level robotic components. In biological terms, this is the technical realization of the natural musculoskeletal system. A musculoskeletal system, in general, consists of two closely coupled subsystems: the skeleton and the muscles with attached tendons. However, the contribution of this chapter focuses on the muscular control system, only. For the mechanical construction, this work relies on the morphology of the previously developed robotic platform Carl[1] [Schütz 17][Schütz 20a].

Based on the well-known mechanical hardware design of electrical Series Elastic Actuators (SEAs), a novel control strategy is biologically and technically motivated, specified, and validated. The proposed control is integrated into the existing hardware of Carl and experimentally evaluated.

The described approach idea has first been published in [*Vonwirth 23a*] and extensively analyzed and discussed within [*Vonwirth 23b*]. A particular focus is spent on the analysis of the benefits of a massively over-actuated, muscular actuation system as present in natural bodies. After thorough research, no other approach has been found in literature trying to achieve natural, compliant motion control with an equivalent strategy.

Chapter Outline

Throughout the following sections, this chapter will first introduce state-of-the-art robot actuation and control technologies in Section 3.1. This section also introduces the planar bipedal robot Carl, which is used as the experimental robotic hardware platform throughout this work. Section 3.2 outlines the benefits and drawbacks of different impedance control layouts. Out of those, the bio-inspired, novel antagonistic muscular control strategy is developed and analyzed. This strategy is applied as the low-level control layer of the overall aim of natural control, as stated in Theorem 1.1. Antagonistic muscular control is a hybrid impedance control layout. Thus, Section 3.3 covers the concept and issues of distributed impedance. Mathematical and kinematic notations, which will be used later, are introduced therein. The last two Sections, 3.4 and 3.5, introduce, analyze, and evaluate the low-level approach's connectivity to higher control goals. The capabilities and limitations of antagonistic muscular control are highlighted, and its functionality is demonstrated on real-world hardware via the bipedal robot Carl. Section 3.6 finally concludes this chapter.

[1] Carl (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU. URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025)

3.1 Construction, Actuation, and Legged Locomotion Control Technologies

Artificial cognition capabilities increase with the development of better-suited and more sensitive sensors. Artificial reasoning capabilities expand with the availability of more and faster computational power. Artificial, legged locomotion develops with the improvement of drive technologies, enabling novel construction and control designs. However, in all cases, it is not the pure hardware that enables the technological progression. Instead, the improving hardware engages the research on novel algorithms, control, and design strategies.

In earlier days, the lack of lightweight, powerful actuators forced humanoid robots like ASIMO[2] [Sakagami 02] to be built from weak, high-speed electric motors. In order to generate enough torque to operate the robot's weight, those motors had to be equipped with high-transmission gears, such as harmonic drives. Unfortunately, with the torque increase, these gears equivalently reduce the maximum velocity and cancel out transparency, i.e., back-drivability, in total.

Early humanoid robots, such as ASIMO, have been limited technologically to a relatively heavy, slow, and rigid body. Thus, their movement capabilities were limited to mechanically stiff and relatively static motions. This limitation, however, is in total conflict with Requirement 2.1: *"Hybrid Dynamics"*. As a result, legged locomotion, especially bipedal locomotion, played only a minor role in recent mobile robotics research.

Nevertheless, actuation technology and, thus, bipedal locomotion control have been researched and developed. Next to the classical, stiff pose control and path control techniques, softer and more dynamic control philosophies have been introduced, such as wrench control, i.e., combined force and torque control, impedance control, or admittance control.

In parallel, increasing computational capacities enables the real-time calculation and simulation of high-dimensional, complex robot dynamics models. Classically oriented bipedal locomotion concepts like the so-called Divergent Component of Motion (DCM) [Pratt 06, Englsberger 13, Englsberger 15, Hosokawa 18] or the so-called Hybrid Zero Dynamics (HZD) [Sreenath 11, Hereid 15, Hereid 16, Nguyen 16, Ghansah 23] have successfully been implemented by exploiting those technical developments. Additionally, nature-oriented strategies such as pure *Muscle-Reflex Strategies* [Geyer 10, Song 15, Batts 15, Bunz 23, Bunz 24], Machine Learning (ML)-based approaches [Geijtenbeek 13, Li 21, Shi 21], or the B4LC[3] [Luksch 10a, Zhao 16] simulation [Luksch 10a, Luksch 10b, Liu 15, Liu 16, Zhao 15, Zhao 16] have been developed and successfully demonstrated in simulation and real robotic hardware.

With the recent availability of impact-tolerant, direct, and *quasi*-direct drive actuation systems, research on legged locomotion has returned to interest. Soft drive trains with low to no gear ratio are robust against environmental impacts, thus satisfying the hardware-related part of Requirement 2.1: *"Hybrid Dynamics"*. Recent research significantly

[2] ASIMO (Advanced Step in Innovative Mobility) is a child-size, full-featured humanoid robot designed and built by the company *Honda* in 2000 and 2005 (version 2).

[3] B4LC is a scientific humanoid robot simulation by the RPTU developed to study compliant, natural dynamics-exploiting gait.

progressed in quadruped robots as, e.g., the MIT Cheetah[4] [Seok 12, Seok 13, Seok 15], HyQ[5] [Semini 11, Semini 15, Semini 16, Aceituno-Cabezas 18], ANYmal[6] [Hutter 16, Hutter 17, Fankhauser 18, Shi 21], or Spot[7] (see e.g. [Bouman 20]).

Despite the fact that all the robots and their respective controls open a large variety of legged locomotion approaches, all of them share one crucial property: All the control approaches are developed upon, and hence rely on, a concrete underlying actuation system offering special hardware features required and exploited by the individual controllers. These are, e.g., position or torque controllability or stiff or compliant mechanical structures. Consequently, the herein-introduced, actuator-level, muscular control concept is fundamental to this work's novel, overall natural control philosophy.

Ficht and Behnke give a short overview of the recent progress in humanoid robot technology in their work [Ficht 21]. They highlight the modern trend of moving away from static and rigid robots toward more compliant, thus, more dynamic designs in both robotic hardware and respective control approaches. Among the widening of the robots' capabilities in general, this trend also directly addresses the above-mentioned impact tolerance required for dynamic locomotion. Aside from powerful artificial muscles designed for robot locomotion, various big- and small-scale technologies exist that are individually suitable for all types of motions. An overview of those technologies can be found in the work of Mirvakili and Hunter [Mirvakili 18].

A robot's compliance is often achieved by implementing artificial muscles of various types and scales. The most significant characteristic of an artificial muscle compared to standard actuators is its intrinsic compliance. Artificial muscles that are powerful enough for legged locomotion most commonly feature a pneumatic, hydraulic, or electric drive train or a combination of two. Compliance is then either attached as an individual series elasticity or inherently present within the actuation principle's physical properties (air spring) and transparency (direct drive). Nevertheless, unusual, innovative construction concepts, such as Best et al.'s fully inflatable humanoid upper body presented in [Best 15], also exist.

The so-called *Integrated Smart Actuator*, developed and presented by Barasuol et al., is one of the latest hydraulic drive train developments [Barasuol 18]. It is meant to be integrated into the next generation of the above-introduced quadruped robot HyQ to allow for high-bandwidth, precise force control.

Modern electric artificial muscles are usually realized as *quasi*-direct drive units [Ficht 21]. Those systems' low gear ratio compromises the necessary gear reduction to achieve high enough torques with small enough reflected inertia to keep the overall drivetrain transparent and easily back-drivable. A rotational example of such an elastic, electric actuator is introduced by Seok et al., which was meant to be built into the above-mentioned quadruped robot MIT Cheetah [Seok 12]. In contrast, the bipedal robot

[4] MIT Cheetah is a lightweight, four-legged robotic research platform developed at the *Massachusetts Institute of Technology*.

[5] HyQ is a hydraulic-electric actuated quadruped robot by the *Italian Institute of Technology*.

[6] ANYmal is a previously scientific, yet commercial quadruped robot by the company *ANYbotics*. URL: `https://www.anybotics.com` (2025)

[7] Spot is a commercial quadrupedal robotic platform offered by the company *Boston Dynamics*. URL: `https://bostondynamics.com/` (2025)

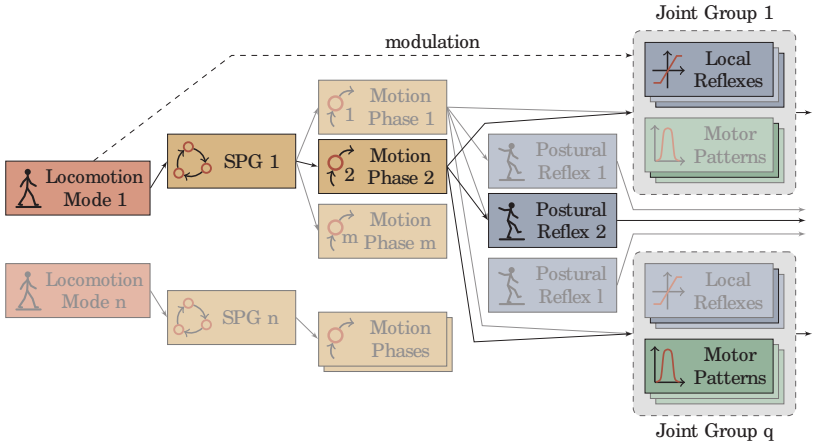


Figure 3.1: Schematic illustration of the hierarchical B4LC control architecture from [Luksch 10a, p. 73]. From left to right, a high-level locomotion mode is chosen, scheduling a respective Spinal Pattern Generator (SPG) [8] for execution. The SPG's motion phases, then, individually activate subsets of predefined reflexes and execute feed-forward motor patterns.

CARL, the hardware platform used within this work, is equipped solely with linear SEAs, the so-called RRLabSEAs [Schütz 16a].

A hybrid electric-pneumatic actuation concept, powering the compliant, single-leg robot EPA-HOPPER II[9] [Mohseni 22, Silva 24], is proposed in [Mohseni 22]. As intended by the work at hand, this hybrid actuator is meant to reproduce, study, and eventually exploit natural leg construction and bio-inspired natural control.

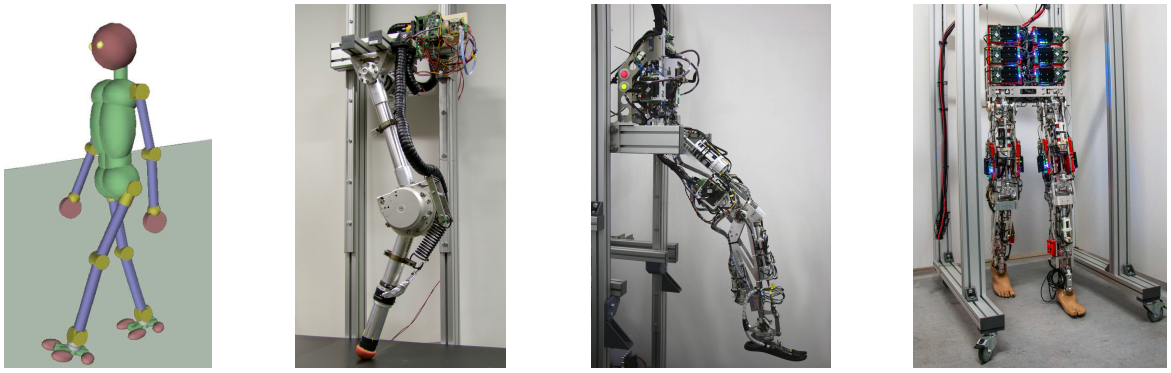
Although significant progress in robotic actuation technology has already been achieved, there is still a wide gap to close. Compared to the human musculoskeletal apparatus, much may still be learned from biology, especially w.r.t. the computationally limited neural processing capabilities.

3.1.1 Development of CARL– From Simulation to a Planar Lower Body

In 2010, the bio-inspired B4LC humanoid robot control simulation was developed and demonstrated virtually functional [Luksch 10a, Luksch 10b]. This concept's fundamental idea underlying both hard- and software design is combining relatively simple, torque-based feed-forward control patterns with reactive, impedance-based reflexes. Reflexes are implemented across multiple hierarchical layers down to distributed, isolated, in-joint impedance rendering. The joints within the B4LC control implement compliant, smart actuators forming a classical, joint-based propulsion layout. Both reflexes and joint torque patterns are activated and inhibited in sets by arbitrary, higher-level control strategies. Thus, the B4LC architecture already reflects the natural robot control layout, as sketched in Figure 2.14 within Section 2.4. Figure 3.1 shows an overview of the hierarchical control structure of the B4LC system, originally published in [Luksch 10a].

[8] The SPGs within the B4LC control sketches are predefined Finite State Machines (FSMs). *"[S]tate transitions are triggered by sensory events. [...] the periodicity emerges from the robot[']s interaction with the environment."* [Luksch 10a, p. 70]

[9] EPA-HOPPER II is a planar, 3-DoF robotic research leg designed at the *Technical University of Darmstadt*.



(a) B4LC Simulation **(b)** Early Monopod **(c)** Single-Legged Carl **(d)** Bipedal Carl

Figure 3.2: The development process of the bio-inspired, compliant, robotic lower-body Carl. From left to right, the four stages of evolution are (a) the B4LC robot simulation [Luksch 10b], (b) the very first single-leg hardware prototype [Blank 09], (c) the single-legged first version of Carl [Schütz 17], and (d) the current, full-featured, planar bipedal lower body [*Vonwirth 23b*].

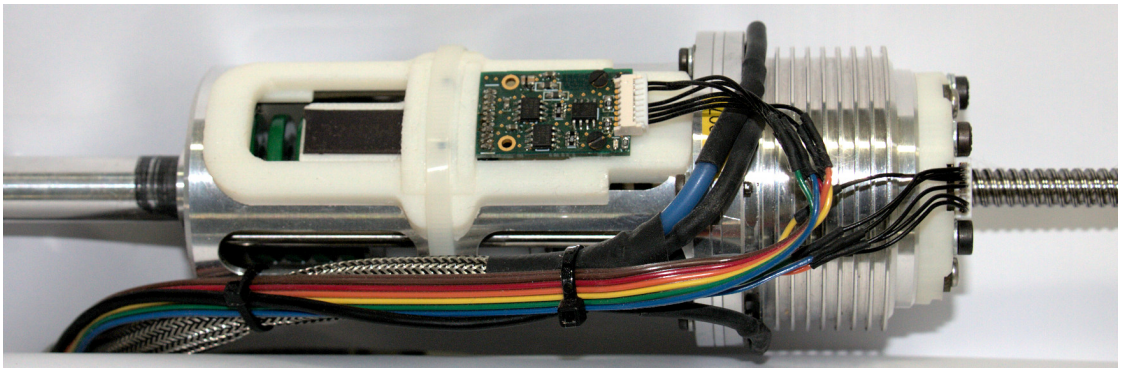
After its initial proof of concept, the original B4LC approach has been extensively developed by plugging in more advanced locomotion skills. Such advanced skills are, e.g., push recovery during walking [Zhao 13], upslope walking [Zhao 15], walking on rough terrain [Liu 15], stepping over obstacles [Liu 16], or varying walking speed [Zhao 16]. Consequently, the general idea of the B4LC control seemed suitable for real-world applications and eventual evaluation on real-world hardware.

The very first hardware development to reproduce the bio-inspired control principles of the B4LC simulation started simultaneously with its digital blueprint. In 2009, Blank et al. published the design of a single leg featuring a compliant knee actuator suitable to generate variable physical impedance locally within the joint [Blank 09]. A picture of this prototype leg is given in Figure 3.2b. Overall, Figure 3.2 shows the evolutionary development process of Carl. It starts with the B4LC simulation to the left and lasts till today's full-featured, planar bipedal lower body on the right.

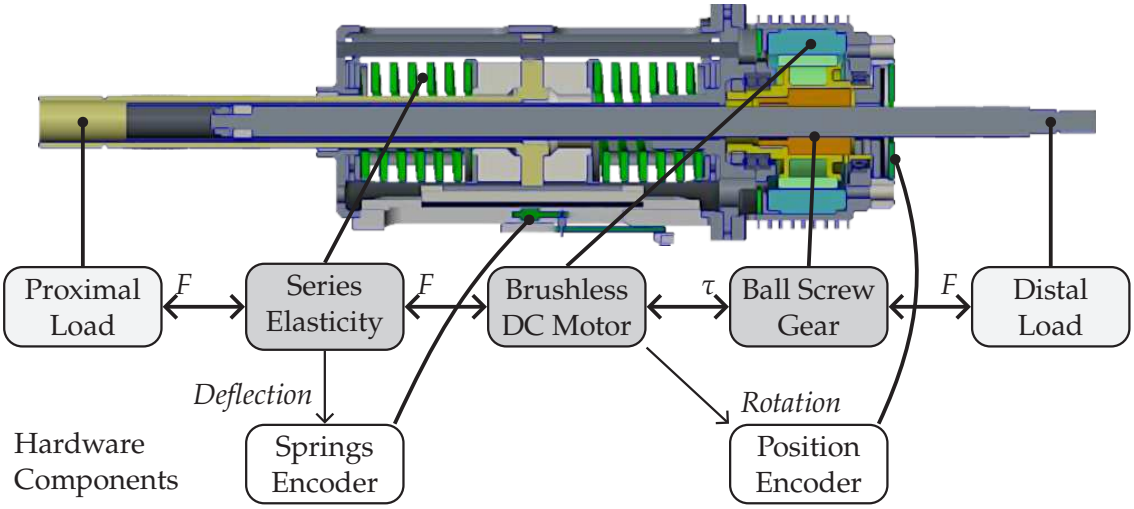
Due to the high demands on the actuation system, the single-legged version of Carl (Figure 3.2c) was the first successful prototype capable of replicating the control principles of the B4LC system [*Vonwirth 17*][Schütz 20a]. With the availability of modern, powerful electric drives, Carl's construction meets both mechanical weight and space requirements. At the same time, Carl satisfies all hardware-relevant natural control Requirements, 2.1: *"Hybrid Dynamics"*, 2.2: *"Singular Positions"*, 2.3: *"Variable Compliance"*, and 2.5: *"Hierarchical Control"*, as listed in Figure 2.14 within Section 2.4.

3.1.2 The RRLab SEA – A Transparent Series Elastic Actuator

This section briefly introduces the most relevant parts of the electromechanical design of the RRLabSEA, developed and initially published in [Schütz 16a, Schütz 20a]. Detailed information about the actuator's design criteria, dynamic modeling, and analysis are available in those works and in [Schütz 16b, Schütz 16c]. Within Carl, the RRLabSEA is available in two sizes with multiple individual compliance and gear configurations. However, their relevant properties and overall design principles are identical so that



(a) The RRLabSEA actuating the bipedal robot Carl



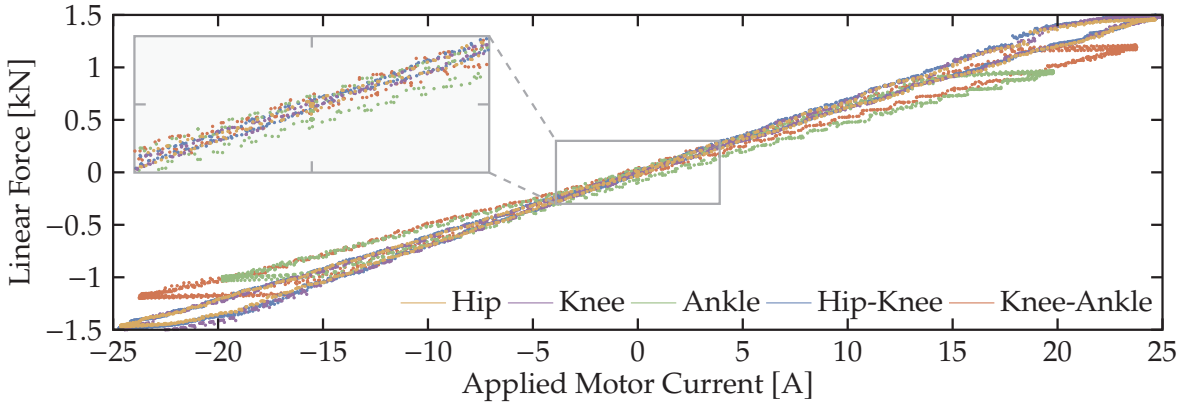
(b) Cross-section through the RRLabSEA's CAD model with its most relevant components

Figure 3.3: The RRLabSEA hardware design used in the bipedal robot Carl. Pictured is (a) a photo of one of the operational actuators in Carl and (b) a scheme of its mechanical construction. The actuator's most relevant components and their interactions are sketched and referenced below the CAD cross-section.

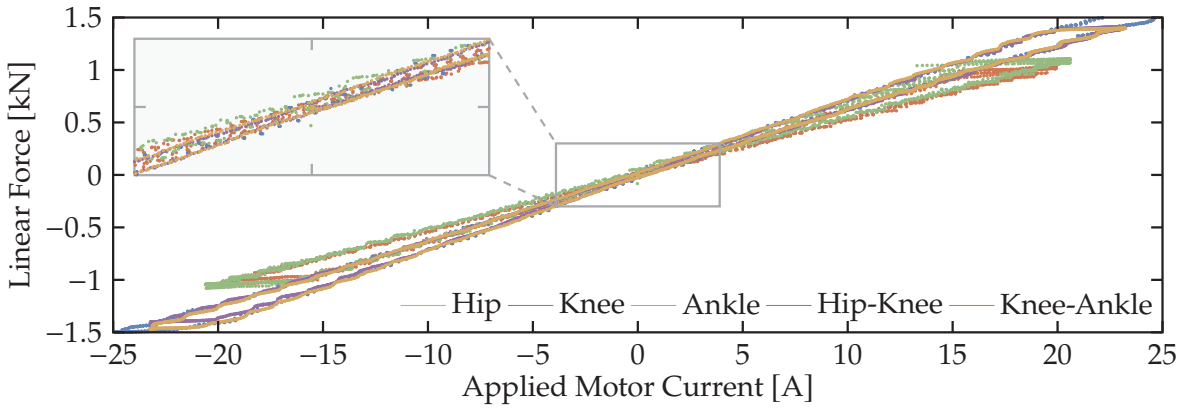
in the following, only the smaller variant is presented on behalf of both. Figure 3.3 pictures the RRLabSEA and a sketch of its mechanical construction. Further details on the hardware design of both RRLabSEA variants are available in [Schütz 20a].

The RRLabSEA is a linear *quasi*-direct drive actuator. Its mechanical drivetrain consists of three main components, assembled in series. These are a relatively soft elastic element, a powerful brushless DC motor, and a ball screw with a low gear ratio.

The brushless DC motor is operated in current control to provide a torque control interface for overlaying impedance control. With the gear ratio of the ball screw taken into account, this motor torque control is converted into a linear force control between two loads on each side of the RRLabSEA, as illustrated in Figure 3.3. All force-current correlations of each of Carl's actuators are experimentally determined and linearized into an actuator-individual force-current constant. The data of these force identification experiments are plotted in Figure 3.4.



(a) Right Leg RRLabSEAs Linear Force Identification of CARL



(b) Left Leg RRLabSEAs Linear Force Identification of CARL

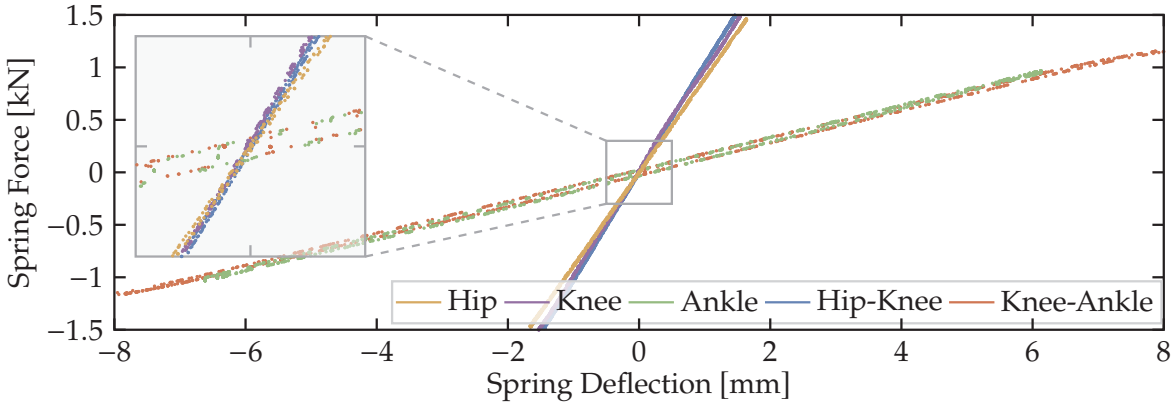
Figure 3.4: Experimental identification of the force-current correlation for all RRLabSEAs within the bipedal robot CARL. The actuators' identified mean force constants in [N/A] are about:

		Hip	Knee	Ankle	Hip-Knee	Knee-Ankle
(a)	Right Leg	62.23	62.84	53.02	62.83	53.69
(b)	Left Leg	64.32	63.40	55.45	63.40	54.47

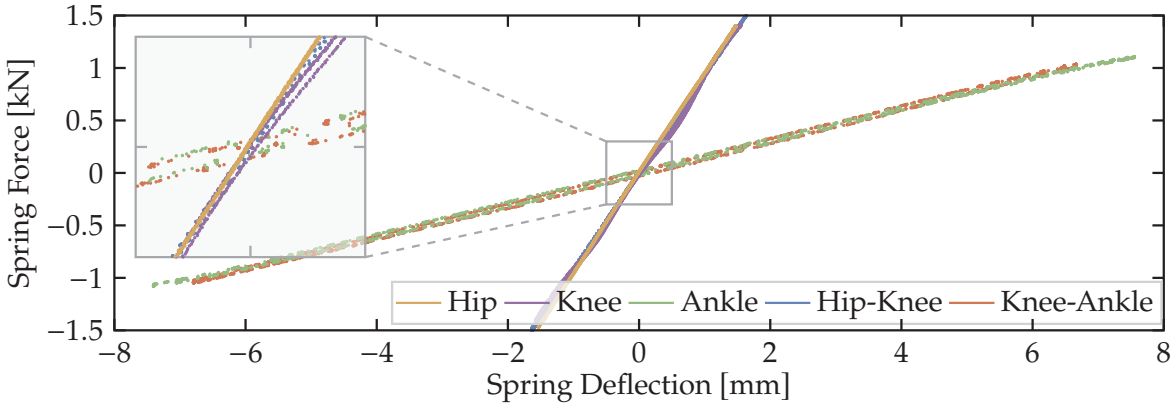
All plots are sets of individual measurements. It can be seen that the actual force-current correlations of the actuators feature friction-induced hystereses with not perfectly linear flanks. A zoomed view of the captured data highlights the force interval of $[-0.3,\ 0.3]$ kN. The later performed experiments with CARL, as described in Chapter 5, do not exceed this interval.

On average, the larger versions of the RRLabSEAs, located around the hip joints and the thigh segments, provide approximately 63 N/A. The smaller variants, placed within the shanks, provide about 54 N/A. Both plots disclose friction-based hystereses with step-shaped flanks. The latter may be due to the discrete rotor and stator magnets within the motor itself. However, both effects remain relatively small within the zoomed regions embedded within each plot. These regions highlight the force interval of $[-0.3,\ 0.3]$ kN, which is not exceeded by the experiments conducted later in Chapter 5.

The series spring system of the electromechanical design of the RRLabSEA serves two purposes. First, the spring deflection is measured and used as force feedback for the control. Second, the springs protect the other, more fragile mechanical components



(a) Right Leg RRLabSEAs Spring Identification of Carl



(b) Left Leg RRLabSEAs Spring Identification of Carl

Figure 3.5: Experimental identification of the ten series springs within all RRLabSEAs within the bipedal robot Carl. The actuators' identified linearized mean stiffnesses in [kN/mm] are about:

		Hip	Knee	Ankle	Hip-Knee	Knee-Ankle
(a)	Right Leg	0.890	0.976	0.156	1.025	0.146
(b)	Left Leg	0.954	0.909	0.149	0.920	0.154

All plots are sets of individual measurements. It can be seen that the springs feature friction-induced hystereses and are not perfectly linear. To highlight these hystereses, a zoomed region of interest is embedded into each plot. This region covers the spring force interval fo $[-0.3,\ 0.3]$ kN, which later in Chapter 5 described experiments with Carl do not exceed.

from external impacts. Together, the series elastic element and the small reflected inertia ensure the actuator's overall transparency and back-drivability. Therefore, both the gear and spring systems significantly contribute to the mechanical realization of Requirement 2.1: *"Hybrid Dynamics"*, as requested for a natural robot morphology in Figure 2.14 within Section 2.4.

Spring stiffnesses of all RRLabSEAs built into the bipedal robot Carl are experimentally identified and presented in Figure 3.5. The placements of all individual actuators within the robot's leg morphology are pictured in Figure 3.6 and described in the subsequent section. As pictured and described in Figure 3.5, Carl's momentary spring setup features two different spring stiffness strengths.

The softer stiffnesses of about 150 $^{N}/_{mm}$ are within the shank actuators, mainly defining the ankle joint's passive compliance. Hardware stiffnesses of about 900 $^{N}/_{mm}$ and higher are used within all other actuators, defining the hip and knee joints. Since softer springs will store more impact energy at equal impact force than harder ones, this setup intends to compensate for external impacts at their expected occurrence, i.e., at the most distal joint to the robot's center. Although this intention seems reasonable, an equally reasonable argumentation can also be given for the exact opposite configuration.

The natural frequency of an ideal spring-mass oscillator is given by $f_0 = \frac{1}{2\pi}\sqrt{\frac{K}{M}}$, where K is the spring stiffness, and M is the load mass[10]. Thus, two possibilities exist to achieve a high natural frequency as preferred for better control: a high stiffness K and a small mass M. If a leg is in the air, the latter one is generally given by design with the load mass significantly reducing from the robot's center toward its feet. However, this relation inverts if the leg is on the ground and has to support the whole robot's body weight. Since, in this case, much higher masses are involved, they have to be counteracted by respectively larger stiffnesses to preserve overall high natural frequencies. Consequently, following this argumentation, stiffness should increase from the robot's center toward its feet. Additionally, such a setup shifts the impact compensation damping work inward toward the naturally stronger hip actuators.

It must be concluded that future research on this topic. Extensive modeling and experimental studies are required to evaluate all the interconnected benefits and drawbacks and substantially determine a best-practice solution. As always in bio-inspired robotics, physiological studies of the actual tendon stiffnesses within the human body may significantly contribute to such studies.

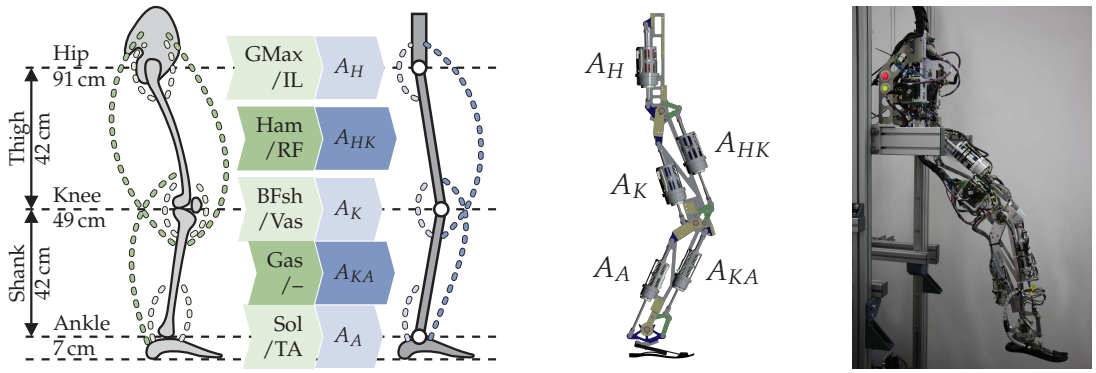
3.1.3 The Human-like Morphology of Carl

This section briefly introduces the robot Carl's planar morphology and references it to the requested morphological requirements, as given in Figure 2.14 within Section 2.4. Detailed information about the development of the Carl, its design criteria, and mechanical insights are available in [Schütz 17, *Nejadfard 19*][Schütz 20a]. Extensive kinematic analyses focusing on the biarticular actuation integration can be found in [*Nejadfard 18b*, *Nejadfard 18a*][Nezhadfard 20].

These studies are primarily motivated by the goal of exploiting combined mono- and biarticular over-actuation to reduce the robot's overall power consumption during wrench generation. The works of Vonwirth and Berns study the identical setup from the point of view of under-actuated, distributed impedance generation [*Vonwirth 23a*, *Vonwirth 23b*]. A sketch of the bio-inspired mechanical morphology realization of the single-leg Carl is visualized in Figure 3.6.

The average dimensions, DoFs, and Ranges of Motion (RoMs) of the human body are well-researched [Winter 09, Terlemez 14, Mandery 16]. Following those studies, the

[10] For readers unfamiliar with the basics of mechanics and rigid-body physics, the author recommends the lecture notes *"Mechanics and Thermodynamics"* [Demtröder 17] for preliminary studies on this topic. For preliminary studies to familiarize with the general topic of control theory and stability, the author recommends the textbook *"Feedback Control of Dynamic Systems"* [Franklin 18].



(a) Nature-Inspired Morphological Layout **(b)** CAD Design **(c)** Single-Leg CARL

Figure 3.6: Bio-inspired development of the planar robotic leg CARL. To the left, (a) sketches the principle musculoskeletal morphology of an average human leg and its transformation into a natural, robotic kinematic layout. All nine relevant muscles in the sagittal-projected human leg are visualized and paired with their direct antagonists. The posterior muscles are written on top of their anterior antagonists. Monoarticular muscles are colored in light green, while biarticular ones are visualized in regular green. As pictured, each natural pair of antagonistic muscles is mapped to a single, bi-directional robot actuator colored in light blue and blue for mono- and biarticular actuators, respectively. A textual mapping between the full-named muscles and actuators is given in Table 3.1. Note that the biarticular Gas muscle spanning over the shank is missing a direct antagonist in nature. On the right, (b) and (c) show the thereout-constructed mechanical CAD model and single-leg build of CARL, respectively.

starting points of CARL's leg design are the natural segment dimensions of a 1.72 m tall, average human lower body. As shown in Figure 3.6a, this results in a 42 cm long thigh, a shank of identical length, and an ankle height of about 7 cm above ground.

The mechanical design of CARL rebuilds those dimensions except for the foot, which is a commercial prosthesis[11]. With a design height of 10 cm, this foot prosthesis is slightly higher than desired. However, due to its compliant construction, it compresses under load, thus reducing its *operational* height toward the desired dimension. The exact value changes dynamically and may only be approximated by a constant. This mechanically built-in compliance within the prosthetic foot already addresses the demanded Requirement 2.1: *"Hybrid Dynamics"*. However, as analyzed in detail later in Chapter 5, this additional passive DoF comes at the cost of degraded controllability.

The overall ankle-foot complex is inspired by the active prosthesis described in [Eilenberg 10]. A detailed argumentation about its design is available in the original works on the mechanical design of the initial, single-legged version of CARL [Schütz 17][Schütz 20a]. The second morphological requirement, Requirement 2.2: *"Singular Positions"*, is achieved by allowing the knee joint to extend to the full stretched position both mechanically and in the later control.

To complete the natural musculoskeletal design of CARL, the previously introduced RRLabSEAs are added to the mechanical skeleton in a natural configuration. This natural muscle configuration, reduced to the nine relevant for planar motion in the sagittal

[11] *Triton Low Profile 1C63* from the company *Ottobock* (`https://www.ottobock.com`) Prosthesis-URL: `https://www.ottobock.com/en-us/product/1C63` (2025)

Human Antagonistic Muscle Pairs		Spanned	CARL
Posterior	Anterior	Joints	SEA
Gluteus Maximus (GMax)	Iliacus Psoas (IL)	Hip	A_H
Hamstrings Group (Ham)	Rectus Femoris (RF)	Hip & Knee	A_{HK}
Biceps Femoris short-head (BFsh)	Vastus Group (Vas)	Knee	A_K
Gastrocnemius (Gas)	–	Knee & Ankle	A_{KA}
Soleus (Sol)	Tibialis Anterior (TA)	Ankle	A_A

Table 3.1: Correlations between the human lower body's sagittal muscular system and its imitating technical setup in the robot CARL. The five relevant pairs of antagonistic human leg muscles are mapped to their spanned joints and respective bidirectional SEAs within the kinematic leg design of the robot CARL. Note that the biarticular Gastrocnemius (Gas) muscle does not have a direct antagonist in the natural musculoskeletal sagittal leg setup. A graphical representation of the entire musculoskeletal setup of a human leg and its technical replication by the single-leg version of CARL is given in Figure 3.6.

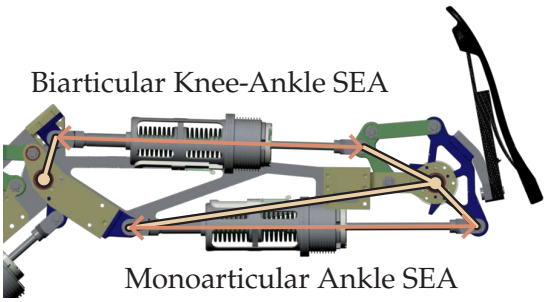
plane, is integrated into the skeleton sketch to the left of Figure 3.6a. As visualized in the figure, those muscles are represented by five bi-directional electric SEAs, each replacing an antagonistic muscle pair. The only exception is the biarticular Gastrocnemius (Gas) muscle, which is missing a direct antagonist in nature. Table 3.1 summarizes the overall muscular representation of CARL.

A detailed view of the construction and integration of both mono- and biarticular actuators is given in the two Figures 3.7a and 3.7b. In the close-up of the CAD model's shank, the two linear actuators and their respective lever arms are highlighted in red and orange, respectively. It can be seen that the biarticular knee-ankle actuator operates on both knee and ankle joints without direct connection to the shank skeleton.

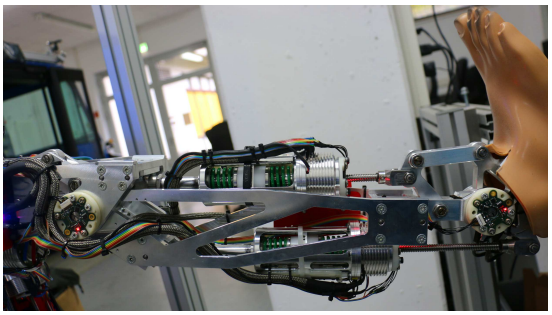
Furthermore, it can be seen that the actuator is connected to the ankle via a four-bar linkage, functioning as an additional gear stage. Due to the construction using linear actuators connected with lever arms or four-bar linkages, the gear ratios between actuators and joints are position-dependent and nonlinear. Having an additional four-bar linkage within an actuator's drivetrain offers the ability to shape its nonlinear transmission curve mechanically. Linkages may be designed to offer higher torque or velocity output of the same actuator within joint configurations as needed.

In the case of the biarticular actuators, shaping the gear ratio between the two spanned joints actually does change the effective direction of action of that particular actuator within the Cartesian world space. Therefore, the mechanical design of such linkages eventually has a significant impact on the final robot's overall control capabilities and efficiency. Hence, ongoing research on the design of bio-inspired, multiarticular actuated robotic legs is taking them into account from scratch [*Sivak 24a*].

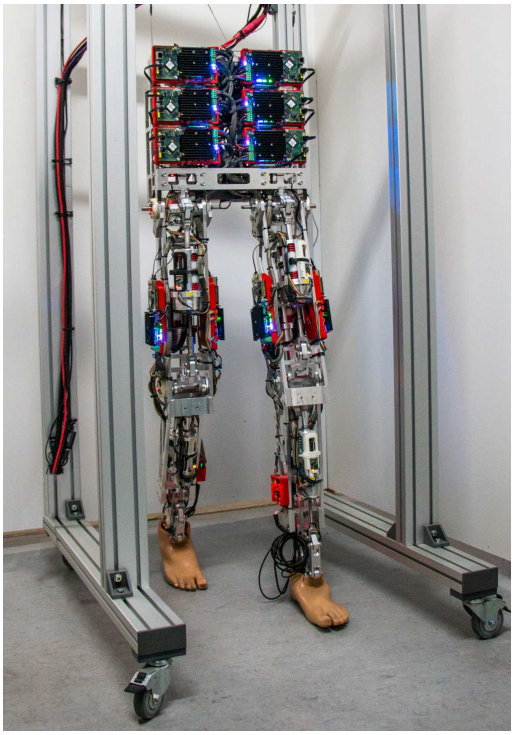
Further detailed analyses of the directional effects of the gear ratios of biarticular muscles dependent on their control purpose and placement within the leg skeleton are available in [*Nejadfard 18b*, *Nejadfard 19*]. All concrete lever arms and gear ratios of the herein-utilized planar bipedal robot CARL are given in preliminary studies of the single-leg version of CARL [Schütz 17][Schütz 20a, Nezhadfard 20].



(a) CAD Design of Carl's Shank



(b) RRLabSEAs, Built into Carl's Shank



(c) Bipedal Version of Carl

Figure 3.7: On the left, there is a close-up view of (a) the CAD design of Carl's shank and (b) its respective assembled hardware. Within the CAD close-up, the two actuators' operational linear forces and lever arms w.r.t. their spanned joints are highlighted in red and orange, respectively. On the right, the (c) complete bipedal version of Carl is pictured. It consists of two identical planar legs, as described in Figure 3.6, that are connected by a short trunk.

3.2 Natural Impedance Control

Low-level robot impedance control can be organized in three hierarchical layouts:

1. Centralized Control
2. DistributedControl
3. Hybrid Control, a combination of both.

In their work [Zhao 19], Zhao and Sentis discuss these three impedance control layouts in detail, including formal modeling, analysis, and verification. Their analysis focuses on methodology potentials, communication delay, and control frequency. Figure 3.8 sketches all these layouts, their relations, typical control strategies, and expected signal delays w.r.t. the above-mentioned criteria.

Centralized control, represented on the upper left, is the classical and the most straightforward robot control layout. Each hardware actuator is directly linked to one central high-level controller. Due to its centralization, this controller can consider a robot's entire state and its modeled dynamics to calculate and optimize all actuated joint torques in a single control algorithm.

Robots that benefit from such a centralized control are, e.g., the torque control modes

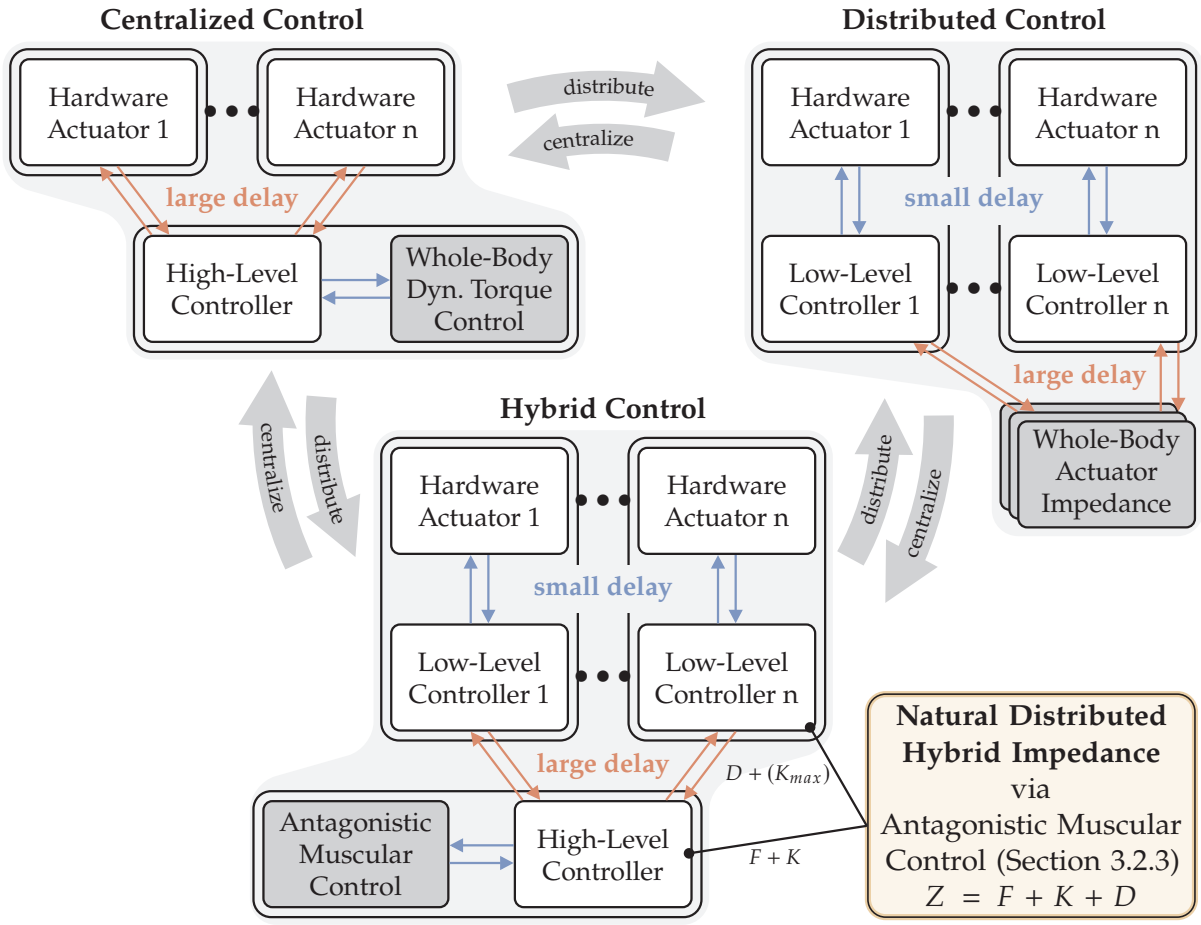


Figure 3.8: Sketch of the three different hierarchical low-level impedance control layouts: Centralized, Distributed, and Hybrid Control. In a centralized setup, a single high-level torque controller determines and controls the entire body impedance, forwarding only the raw torque commands to the hardware actuators. An example of centralized control is a classic *Whole-Body Dynamics Torque Control*, discussed in Section 3.2.1. The opposite of centralized control is an entirely distributed approach, as discussed in Section 3.2.2. In such a setup, individual actuator impedances are distributed and locally rendered. A hybrid control layout implements a combination of both. The herein-proposed novel antagonistic muscular control, as developed in Section 3.2.3, realizes such a hybrid setup. As pictured, this layout distributes the most time-critical impedance component, the damping, while the others are determined on a centralized higher level. This picture is based on [Zhao 19, Fig. 1].

of the two exemplary robots TORO[12] [Englsberger 13, Englsberger 14] and DURUS[13] [Hereid 16, Reher 16], introduced in Section 2.3.1: *"Rigid Robot Control Approaches"* (see Figure 2.11). Note that actuator-intrinsic torque control is not considered a local controller as this functionality is non-complex and usually embedded within a motor driver's analog current control circuits.

By design, the communication between the CPU and the actuators is physical, i.e., operating on wiring hardware. Thus, communication with the robot hardware suffers

[12] TORO is a torque-driven humanoid robot developed and operated at the *German Aerospace Center*.
[13] DURUS is a bipedal robot developed and designed at the *Caltech California Institute of Technology*.

from significant delays due to the required signal preprocessing, encoding, transmission, decoding, and postprocessing. Unfortunately, especially for torque control operation, this delay highly limits the bandwidth of the central whole-body controller and, consequently, the robots' capabilities and robustness against disturbances. An example of such a centralized control approach is a classic *Whole-Body Dynamics Torque Control*. A more detailed discussion on the benefits and drawbacks of a centralized control layout is given in the upcoming Section 3.2.1.

Alternatively, robot impedance control can also be addressed entirely distributed. In such a setup, impedance is not calculated for the whole body but instead rendered locally on each actuator individually and in parallel. With a distributed architecture, the musculoskeletal complex becomes much more similar to the natural human body setup, featuring compliant muscle-tendon actuation throughout the whole body.

Examples of distributed setups are the position control modes of the two previously-mentioned robots, TORO and DURUS, or the impedance control of the single-leg version of the robot Carl [*Vonwirth 17*][Schütz 20a]. A discussion about the distributed impedance generation on the single-leg version of Carl using the RRLabSEAs as muscle-imitating *smart actuators* is given in Section 3.2.2. Two other examples of distributed control are the bio-inspired muscular control approaches, muscular rest length control and muscular reflex control, as described and analyzed in Figure 2.13 within Section 2.3.3. Respectively, these two approaches implement actuator-level mechanical stiffness regulation and hill-type muscle simulation.

Both examples show that distributed actuator-level control is not limited to technical, linear impedance generation but may be realized arbitrarily in hardware and software. The latter approach, the hill-type muscle simulation, is the only approach implementing pure distributed, distributed high-level control. Within this approach, a predetermined set of independent reflexes drives the muscles' stimulations merged actuator-individually [Song 15]. Motion coordination arises only from the dynamic interaction of the system's motion and the reflex implementations.

As just outlined, both of the above layouts suffer from critical disadvantages. Centralized control suffers from limited bandwidth due to the significant delays of low-level control commands. Distributedcontrol lacks crucial synchronization mechanisms or requires a centralized control on top. However, hybrid control is designed to overcome both those problems by combining the advantages of both approaches while avoiding their respective drawbacks. As pictured in the bottom middle of Figure 3.8, the hybrid control layout combines the characteristic properties of both others. A centralized, high-level controller is placed on top of distributed, individual low-level controllers for each actuator.

The big difference between hybrid control and the two other setups is the hierarchical separation of the individual impedance components, described as distributed impedance. While damping is crucial for stability and is significantly sensitive to signal delay, neither is the case for stiffness, force, or torque control. Torque control, however, requires precise knowledge of the whole body dynamics to be optimally distributed and applied to the individual actuators. Hence, in their technical analysis, Zhao and Sentis recommend locating damping generation at the actuator level, distributed to low-level controllers. At the same time, force/torque and stiffness control should be kept centralized.

Although motivated by natural muscle observations [Brown 00, Winter 09, Haeufle 20, Izzi 23], the novel antagonistic muscular control for SEA hardware, developed in Section 3.2.3, unintendedly follows these recommendations. Note that Figure 3.8 also lists a K_{max} component being distributed to the low-level controller. This component represents the uncontrolled mechanical series elasticity of the SEA hardware that upper-limits the stably generable local actuator stiffness. Its biological analog would be the muscle-intrinsic *short-range stiffness* and the muscle-extrinsic tendon's elasticity.

3.2.1 Centralized – Whole Body Dynamics Torque Control

Compliance is necessary to prevent damage to both the robot and its environment whenever there is an interaction between them. In an ideal case, the involved robot counteracts the environmental compliance. Interaction with soft environments prefers a stiff robot, while interaction with stiff environments requires a soft robot. This way, the stiff contact partner is precisely controlled, and the soft one adapts. If both the robot and the environment are inelastic, the mechanically weakest material involved will take over the compliant role and bend or break. If both parties are relatively soft, their combined natural frequency is appropriately low. Consequently, the overall system becomes sensitive to parasitic oscillations making it difficult to control.

Two main contradicting options are available to implement compliance in robotics:

1. Impedance Control: $F = Z[p, \dot{p}]$
   Relies on precise inner force control F and, hence, on relatively soft hardware, e.g., a transparent direct or *quasi*-direct drive SEA.
   Results in:
   + Good performance in soft compliances below the built-in hardware
   – Bad tracking accuracy due to the low hardware bandwidth

2. Admittance Control: $[p, \dot{p}] = Z^{-1}F$
   Relies on precise inner path control $[p, \dot{p}]$ and, hence, on relatively stiff hardware, e.g., a high-geared motor with a loadcell.
   Results in:
   + Good performance in stiff compliances and tracking accuracy
   – Bad on impact-response due to the rigid internal mechanics

Due to Requirement 2.1: *"Hybrid Dynamics"* and specifically its demanded impact tolerance, admittance control seems technically unsuitable for humanoid robots. Furthermore, the technical decision to favor impedance control and compliant hardware, i.e., SEA hardware, is supported by biological principles. The mechanical design of a SEA, consisting of an actuation unit in series with an elastic element, replicates the construction of a biological muscle, which consists of an actuated contractile element in series with a passive elastic tendon, respectively.

However, to profit from the benefits of well-known rigid body dynamic control, Semini et al. proposed 2015 the principle of *"stiff inside, soft outside"* [Semini 15], a guideline for designing impact-tolerant robotic limbs. In favor of high-precision controllability, the principle suggests building the active parts of robotic limbs as stiff as possible with

passive elasticities in series with the whole leg at its expected contact points to the environment. Those dedicated, compliant contact structures protect the internal rigid structures by absorbing external mechanical impacts. Examples of such compliant contact points are spring-loaded feet or rubber feet. In other words, when following the principle of Semini et al., the leg as a whole becomes one combined SEA with an uncontrolled elasticity.

The most significant advantage of this approach is its proximity to traditional, stiff, but well-known robot construction and control principles. By definition, this principle allows for out-of-the-box rigid trajectory tracking. Thus, it is well-known in legged robotics and has been demonstrated successfully on several robots, e.g., the bipedal robot DURUS or the quadrupedal robot HyQ. However, the advantage of having only a single, passive elasticity close to a limb's end also forms the biggest drawback of this approach. By design, such a robotic leg is strictly limited to environmental interactions occurring at only the designed positions and only in the designed elastic directions.

To overcome this limitation, a whole-body impedance or admittance control must be provided. Both can be realized by mechanical compliance, virtual compliance, or a combination of both. Mechanical compliance offers impact tolerance at the cost of controllability, whereas virtual compliance provides maximum controllability at the cost of impact sensitivity. By focusing only on controllability, Semini et al. suggest using pure active impedance control since *"no springs are needed to protect the actuation system"* [Semini 15]. However, their principle of *stiff inside, soft outside* within the same work may already be considered a violation of their own suggestion.

A probably good compromise between all the previous arguments on impedance, admittance, mechanical, and virtual compliance seems to be a setup as built into the quadruped robot MIT Cheetah [Seok 12]. The robot relies on mechanical elasticity at its feet to protect the system against external impulses. At the same time, *quasi*-direct drive actuators offer the mechanical transparency required for impact-tolerant virtual impedance generation.

3.2.2 Distributed – Whole Body Actuator Impedance

The total opposite of centralized control is an entirely distributed layout, as pictured in the upper right of Figure 3.8. With such a layout, the primary impedance control is executed locally on the individual actuators. Thus, all the low-level control loops run in parallel and benefit from the locally low delays. A distributed layout has three significant advantages over its centralized counterpart.

First, the locally lower delay enables higher control gains. Second, the control commands sent to the actuators are more abstract, which typically allows for lower control frequencies, as required by the higher delay in physical communication with the actuation units. Due to their intrinsic, complex control, such actuation units are often referred to as *smart actuators*. Third, the control distribution among those *smart actuators* also enables distributed, distributed, higher-level control that is accepted and handled by the individual actuation units. Furthermore, following the two demanded actuator Requirements 2.3: *"Variable Compliance"* and 2.5: *"Hierarchical Control"*, a distributed, compliant setup is much more similar to the human body than a centralized layout.

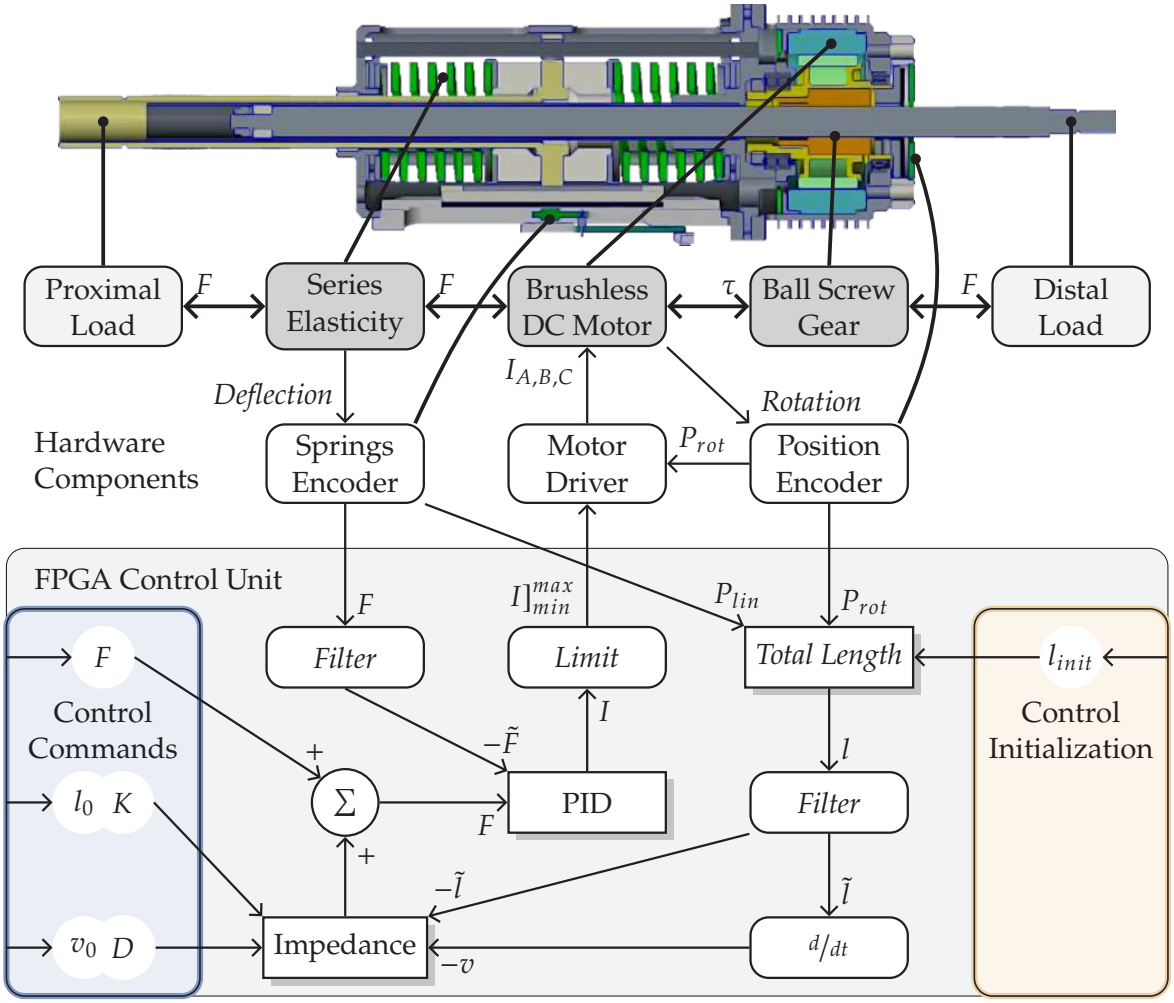


Figure 3.9: Cross-section of the RRLabSEA CAD model and its complete impedance control block diagram according to [Schütz 20a, p. 136]. Below the cross-section, the most relevant electromechanical elements are listed. The block diagram sketches digital hard- and software components implementing a three-staged impedance control. All five impedance control commands are input values inside a light blue area. Similarly, the length initialization required on every system startup is highlighted by the light orange background on the right. The three sharp-edged blocks, *Total Length*, PID, and *Impedance*, implement non-trivial functions.

The most significant disadvantage of total decentralization is the absence of actuation and whole-body synchronization. While simple tasks are perfectly fine with unsynchronized actuation, advanced operations, such as bipedal walking, require both actuation synchronization and whole-body dynamics. Consequently, centralized high-level controllers overlaying distributed low-level controllers become mandatory to compensate for dynamic disturbances induced by the independent low-level control loops.

With Carl's bio-inspired musculoskeletal design, Schütz et al. introduced a planar bipedal robot with entirely distributed impedance control. Carl is equipped only with compliant *smart actuators*, the so-called RRLabSEAs [Schütz 16b, Schütz 16c, Schütz 16a], introduced in Section 3.1.2. Their respective local, FPGA-based impedance control is sketched in Figure 3.9 according to [Schütz 20a, p. 136].

The RRLabSEA impedance control is realized by a stack of three consecutive control loops. As sketched in Figure 3.9, the inner-most motor current control loop is part of the DC motor driver. On top of this, an intermediate force control loop based on the measured spring deflection is meant to remove the small but existing reflected motor inertia. This second stage should simulate the whole actuator as an ideal force source.

The outmost control eventually implements the full-featured impedance control offering virtual force F, stiffness K, and damping D rendering w.r.t. an equilibrium length l_0 and velocity v_0. Since the system lacks a dedicated, absolute encoder of the total length, the overall length is calculated as the sum of the spring deflection and the integrated motor rotation according to the gear ratio. In order to correctly bias the gear integrator, a length initialization value is passed once at startup. Unfortunately, this initialization requires a time-consuming setup routine to avoid injecting erroneous kinematic data already at startup.

Due to the sensory data and control complexity, the two outer control rules cannot be loaded into the motor driver's internal control computer. Hence, as pictured in Figure 3.9, the force and impedance control loops and their corresponding sensor data processing are located on a separate external compute unit. In the case of the RRLab SEA, an FPGA is used to house and execute those sense and control rules.

The FPGA offers the ability to handle all needs in parallel, at various individual frequencies, and with individual real-time guarantees. Among others, those guarantees include communication between the sensor and actor hardware, the actual control calculation, and other tasks such as thermal monitoring or an STO watchdog.

The single-leg version of Carl was successfully tested to replicate B4LC-inspired walking patterns over a treadmill via entirely distributed impedance control [*Vonwirth 17*]. Unfortunately, equally controlled experiments on the stand-alone bipedal Carl have failed. However, from these initial experiments, three key problems have been identified as the most-responsible failure sources:

1. Numerous robot-intrinsic oscillations with pure actuator-intrinsic impedance control bandwidth

2. High overall robot control command complexity

3. Torque pattern volatility against real-world noise

First, the amount of intrinsically embedded oscillation-capable subsystems in Carl is significant. Mechanical and digital oscillation circuits, ordered from low-level to high-level hardware and control, are:

1. The mechanical spring-mass oscillator of the electric motor's rotor inertia and the SEA's serial elasticity, including the actuator-individual spring-deflection controller (see Figure 3.9)

2. The actuator-local outer closed-loop impedance control of the whole RRLabSEA on top of the inner spring-deflection controller

3. The virtual and mechanical spring-mass system of several SEAs, mechanically coupled via shared joints and multiarticular over-actuation (see Figure 3.6a)

4. The centralized force distribution strategy, applied over all actuators within each leg of Carl (see [*Nejadfard 18a*])

5. The physical coupling of both individual legs in the bipedal version of Carl, embodying a system of two interconnected pendulums

The RRLab SEA's spring is designed soft to support precise deflection measurement and high external impact energy compensation. Consequently, the natural frequency of the actuator-internal spring-mass oscillator is, by design, appropriately low. Thus, the actuator's overall force control bandwidth is limited. This limitation is carried on to the impedance control loop, i.e., the outer one of the three consecutive control loops.

As shown in Figure 3.9, the RRLab SEA impedance is rendered over its whole drive train, including its series spring. Thus, this control suffers from the additive noise of two sensors, the spring deflection and the rotor position encoder. Internal friction, nonlinearities, and hystereses within the spring and motor constants further disturb the correctness of the controlled property. As a result, the controllable impedance rendering capabilities of all SEAs in Carl had to be limited far below their theoretical and physical possibilities to stabilize the robot.

Second, when summed over all robot actuators, the number of required control values becomes enormous. As shown in Figure 3.9, each RRLab SEA must be continuously supplied with the five control commands previously defined: $[F, l_0, K, v_0, D]$. For Carl's bipedal version, with each leg equipped with five actuators, the number of required control commands already accumulates to 50. At this point, it is important to mention that Carl is still a limited, planar-only, humanoid lower body with only 6 DoF.

It must be concluded that the enormous amount of 50 control commands necessary to control only 6 DoF does not scale for a full-featured, 3-dimensional humanoid robot. Thus, a simple, more straightforward low-level control strategy with fewer control parameters is required. Interestingly, this aim for simplicity is entirely backed by the human Central Nervous System (CNS). In nature, a low-level interface of no more than a single *motor neuron* is required to activate each muscle.

The third reason causing the B4LC control concept to fail on real-world hardware is the usage of feed-forward torque patterns. Applying feed-forward torque patterns requires massive parameter tuning on every system startup due to continuously changing environmental conditions, such as temperature or dust-dependent friction. This necessity is evident because the predefined shapes of such feed-forward patterns have to encode the robot's whole-body dynamics.

In such a setup, no built-in control loop is involved between high-level pattern generation and low-level torque generation. Thus, an improved low-level SEA control should aim for a control compromise between a centralized static pose control and a distributed dynamic wrench control. E.g., a velocity, i.e., motion-based control as demanded by the overall control goal in Theorem 1.1: *"Natural Control"*.

3.2.3 Hybrid – Antagonistic Muscular Control

A natural muscle is a composition of two major components. The so-called contractile element embodies a muscle's actuation unit. A muscle's tendons build elastic attachment constructions of those contractile elements and the rest of the body, herein, the bones. Every contractile element is embedded in tendons, thus entirely decoupling the contractile elements from potential external impacts.

In nature, muscle-tendon complexes occur in arbitrary topologies, including serial and parallel contractile elements, as well as forking and merging tendons. The so-called *Golgi Tendon Organ* is located between a contractile element and an attached tendon, observing changes in muscle-tendon tension. It is used to control muscular tension and to protect the contractile element against overstressing. For example, the commonly well-known muscle stretch reflex is triggered by the *Golgi Tendon Organ* [Mihailoff 18].

In a simplified manner, a single, cut-out muscle reduces to a contractile element coupled in series with a passive elasticity. A dynamic model of this simplified structure was already published in 1938 by A.V. Hill [Hill 38]. Under the later name *Hill-type* muscle model, these dynamics became the de facto standard in today's research literature.

The raw *Hill-type* muscle model consists of three components: the contractile element, embedded in a damped series elastic element, and a damped parallel elastic element. However, the parallel elastic element can be merged with the contractile element into a single component, a complex, controllable, first-order impedance. With this simplification, it is easy to recognize the mechanical construction of a SEA as a perfect platform to emulate muscular control.

However, when using an electrical SEA realization, such as the RRLab SEA, there is one big difference left compared to the ideal *Hill-type* muscle model. The reflected inertia of the rotating electric motor creates a parasitic second-order[14] impedance component inside the contractile element that is not present in nature. Actuator-intrinsic, intermediate force and torque control loops are widely implemented to eliminate this undesired inertial impedance term [Paine 14, Paine 15, Schütz 20a]. This inertial-cancelling intermediate force control loop is also present in the previously described impedance control of the RRLab SEA, as pictured in Figure 3.9. However, the previous Section 3.2.2 also points out the significant problems arising from such an intermediate force control loop.

Consequently, as first published in [*Vonwirth 23b*, *Vonwirth 23a*], this work proposes a novel, muscle-inspired control philosophy for SEA hardware, compromising between the necessary technical controllability and the desired natural muscle emulation accuracy.

In order to simplify the actuator-intrinsic control complexity, the novel muscular SEA control waives the reflected motor inertia compensation. Appropriate SEA hardware should feature small reflected inertia to keep the resulting parasitic second-order impedance term small, as is the case for the RRLab SEA [Schütz 16a, Schütz 20a]. Consequently, this aim prefers the usage of multiple, smaller SEAs in parallel over a single, bigger one with higher reflected inertia. Note that the reflected inertia scales linearly with the number of in-built actuators but quadratic with the in-built gear ratio.

[14] A second-order impedance is a force, depending on the second input derivative, i.e., $F_z = M_z a$

However, the overall robot weight also scales linearly with the number of actuators but not with the actuators' in-built gear ratios.

With the motor inertia being ignored, the actuator's impedance control can now follow the compacted *Hill-type* muscle model. This way, impedance control can be limited to the actual actuator, i.e., the contractile element only, while the series elastic element remains passive and uncontrolled. This design decision not only implements the *Hill-type* muscle model but also follows the design principle of *stiff inside, soft outside*, introduced in [Semini 15].

However, instead of applying this principle to the entire leg as proposed by Semini et al., it is herein applied to each SEA individually. By doing so, the entire limb remains compliant and thus impact tolerant over its entire structure. Thus, the desired morphological Requirement 2.1: *"Hybrid Dynamics"* is satisfied, even outside the focused use case of bipedal walking.

Without the need for high-precision force sensing and control of the elastic element, both higher or lower stiffnesses, or event non-linear, progressive stiffnesses may be used within some SEA hardware. Furthermore, excluding the spring from the SEA control opens the ability to exploit the springs' dynamics, as, e.g., researched with the quadrupedal robot Bert[15] [Lakatos 18, Pollayil 22]. However, without dedicated force control within a SEA, its elasticity only protects the motor hardware against small and short impacts. For lasting external impacts, the actuator must rely on its transparency, i.e., back-drivability and reflected inertia. Thus, back-drivability and low reflected inertia become mandatory mechanical SEA designs with uncontrolled elasticities.

With this simplified SEA control layout, the amount of impedance control commands can be reduced to a minimum. Following the process of bio-inspired engineering, this reduction is based on both biological studies on the intrinsic dynamics of a natural muscle and validated against technical reasonableness.

In 1985, Mussa-Ivaldi et al. investigated the *short-range stiffness* of a human arm and derived three significant impedance characteristics of the musculoskeletal apparatus [Mussa-Ivaldi 85]:

1. Experimentally recorded stiffness matrices at a human arm's Tool Center Point (TCP) can be treated symmetric, i.e., elliptical.

2. The general shape and orientation of the measured stiffness ellipse primarily depend on the arm configuration rather than its muscles' activities.

3. Muscular co-contraction strongly increases the overall size of the measured stiffness ellipse while only marginally influencing its shape and orientation.

All three properties support the hypothesis that the human arm's *short-range* TCP stiffness is generated distributed on a muscular level. Using muscular co-contraction, the human CNS controls its body's overall impedance characteristic, exploiting the intrinsic dynamics of both the muscles' contractile elements and tendons. However, the overall impedance shape is primarily defined by the mechanical mounting positions of

[15] Bert is a compliant quadruped robotic research platform developed by the *German Aerospace Center*.

the individual muscles within the skeleton. It can directly be followed that the more muscles are available within the musculoskeletal system, the more control the CNS has over the body's impedance shape.

State-of-the-art studies on the effect of actuator-intrinsic muscle-like dynamics support this hypothesis. It is known that the pure existence of muscle-like force characteristics within robot actuators can significantly reduce the amount of required control information to be provided by a central intelligence [Haeufle 20]. The type and shape of such characteristics are well-studied and known to implement a distributed, zero-delay feedback system on the muscular level, the so-called *preflexes* [Brown 00, Winter 09]. Technical equivalents of muscular co-contraction can be used to alter the damping characteristics of muscular *preflexes* and significantly contribute to the overall system stability if embedded within the low-level control for maximum effectiveness at minimal delay [Izzi 23].

Muscular force generation is primarily influenced by four values: Activation a, Time t, Length l, and Length Change $v = \dot{l}$. First of all, the time dependency, i.e., muscle fatigue, is treated as an unnecessary limitation similar to technical motor overheating and neglected within the following. However, it is also possible to interpret this automatic force reduction as intrinsic motivation to relax muscles as much as possible to save energy. All other characteristics represent the three components of interest of a mechanical first-order impedance. The emitted force of a muscle w.r.t. these impedance properties can be estimated as follows:

$$F(a) \sim F_{Max}\, a \tag{3.1}$$

$$F(l) \sim F_{Max} - K(l)(l - l_0) \tag{3.2}$$

$$F(v) \sim F_{Max} - D(v)(v - v_0) \tag{3.3}$$

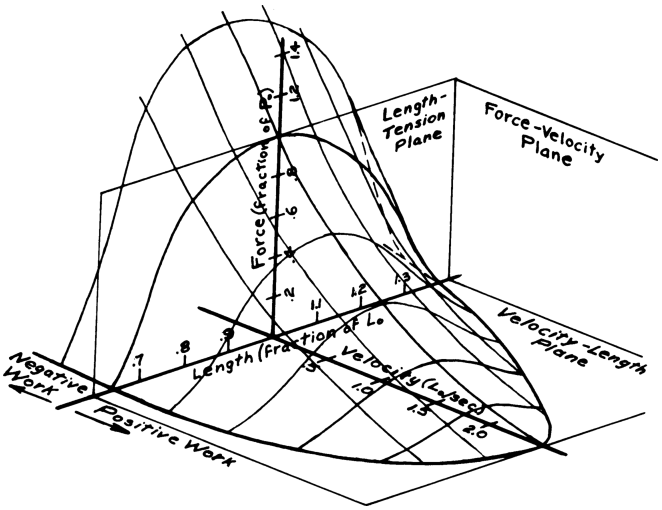
with F_{Max} being the maximum muscle force at rest and K and D representing stiffness and damping, respectively.

While the first relation against general muscle activity is trivial, the latter two are more interesting. Both together define a complex, state-dependent, intrinsic muscular impedance, which is well-studied in the book *"Biomechanics and Motor Control of Human Movement"* by A.D. Winter [Winter 09]. Figure 3.10 sketches the general shapes of both those force characteristics and their respective derived impedance properties.

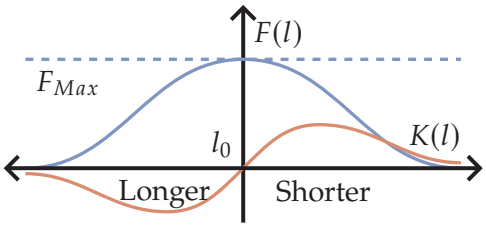
The force-length characteristic, sketched in Figure 3.10b, has a symmetric shape w.r.t. the muscle's rest length l_0. Hence, its derived stiffness characteristic becomes point symmetric w.r.t. the origin at l_0. This symmetry means that the intrinsic muscular impedance contains partially positive and negative stiffness. Hence, from a technical point of view, the local, in-built muscular stiffness is partially unstable.

Furthermore, an actuator-local stiffness always requires an additional equilibrium position l_0 as a secondary control. In the case of the RRLabSEA, an additional absolute reference value l_{init} is required for length initialization, as pictured in Figure 3.9. Disqualifying local stiffness from the actuator's impedance control eliminates not only two control commands but also the need for this additional actuator initialization.

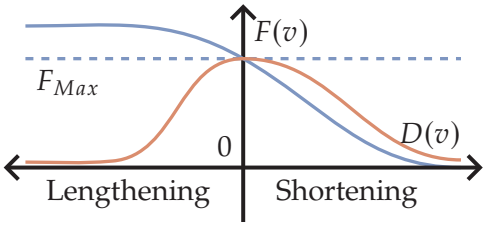
On the other hand, the muscle-intrinsic damping characteristic stays positive over the whole range of muscle motion, as shown in Figure 3.10c. Hence, natural muscular



(a) Contractile Element Impedance by [Winter 09]



(b) Force-Length Characteristic



(c) Force-Velocity Characteristic

Figure 3.10: Sketches of the intrinsic impedance characteristics of a natural muscle $F(l)$ and $F(v)$. On the left, Figure (a) shows the original three-dimensional impedance characteristic plot of a natural muscle's contractile element at maximum muscular activation. The plot is sourced from [Winter 09, p. 239], remaining unchanged. Hence, $F_{Max} = P_0$ on the vertical axis of this plot. On the right, the two plots sketch the surface's general shape for both horizontal axes, (b) length and (c) velocity, individually. The two force characteristics, the left-hand force surface intersected by the respective axis planes, are drawn in blue. Derived from those force profiles, the two red graphs sketch the shapes of their corresponding stiffness $K(l)$ and damping $D(v)$ properties. The dashed blue lines indicate the muscles' maximum force at rest F_{Max}.

damping is technically stable. The damping's reference velocity v_0 can be fixed to the natural constant zero without loss of generality as this value can be reconstructed with the activation-based additive force $F(a)$.

From a technical point of view, the distributed SEA-local impedance control should be reduced to cover only the muscle's activation force and intrinsic damping characteristic. Interestingly, this reduction of distributed controllers matches the hybrid impedance control proposed by Zhao and Sentis, as initially introduced [Zhao 19]. Due to its bio-inspired background, this reduced hybrid impedance control approach will be referenced as so-called *antagonistic muscular control* throughout the rest of this work. Figure 3.11 sketches a simplified block diagram of an antagonistic muscular control loop deployed on the RRLabSEA.

The reduced actuator-local impedance significantly simplifies the respectively required attached higher-level control system. Limiting the actuator's local impedance generation to only contractile force F and contractile damping D reduces the continuously required control commands by 60% from the original 5 to only 2 values. In the case of the bipedal robot Carl, the total number of low-level controls is reduced from 50 to 20. Assuming the control values are determined by a quadratic optimization, a theoretical maximum computational effort reduction of 84% can be achieved using the proposed antagonistic muscular control strategy instead of a complete distributed impedance control. Thus, the proposed hybrid impedance strongly contributes to algorithmic simplicity, faster

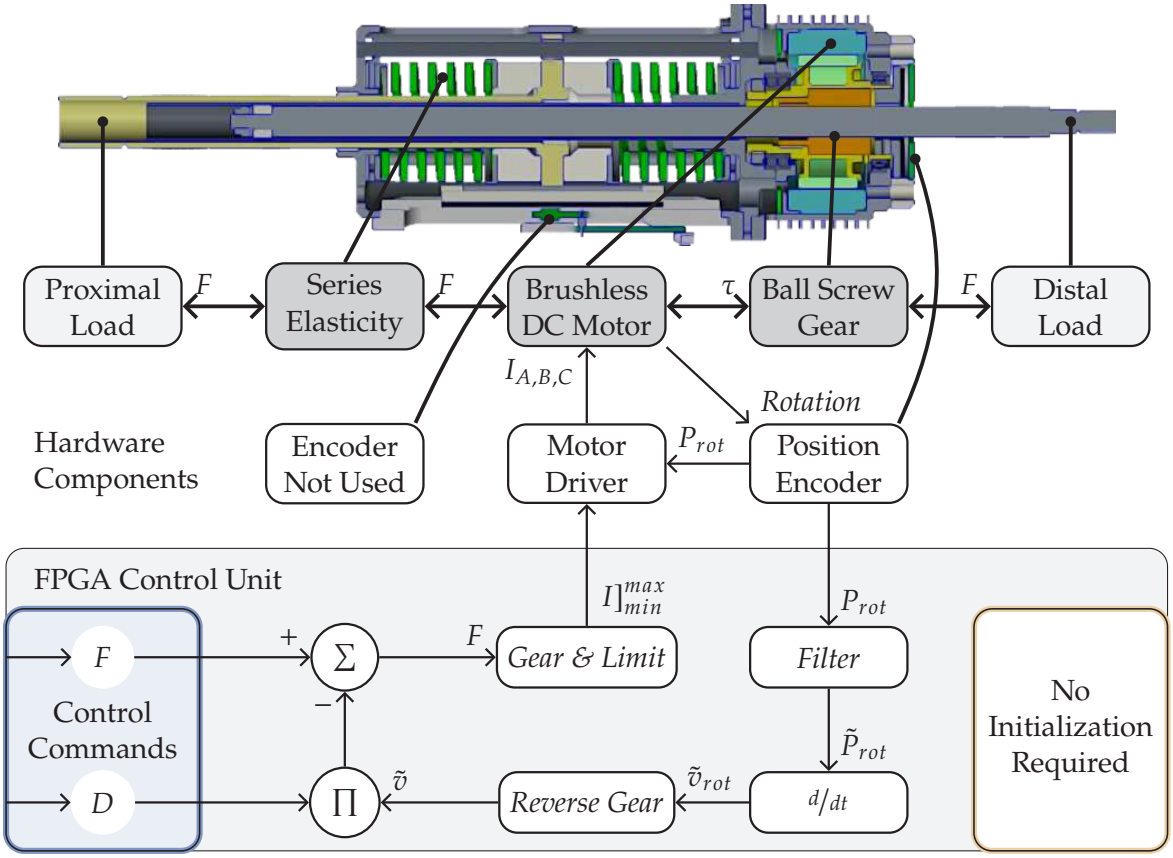


Figure 3.11: Cross-section of the RRLabSEA CAD model and its antagonistic muscular control block diagram according to [*Vonwirth 23b*, *Vonwirth 23a*]. Analog to the complete impedance control pictured in Figure 3.9, the gray blocks below the cross-section represent the most relevant mechanical elements. The blocks below those refer to the digital hard- and software components realizing the antagonistic muscular control. Note that the springs encoder has become unused in this hybrid impedance setup. The two impedance control commands, contractile force F and contractile damping D are drawn on the left as input values inside a blue background area. Due to the removal of actuator length-related impedance terms, no initialization is required.

response times, and less energy consumption.

Furthermore, following the principle of *stiff inside, soft outside*, as proposed by Semini et al., the novel antagonistic muscular control is executed on an entirely stiff subsystem of the overall SEA [Semini 15]. The reduced delay and the rigidness of the controlled subsystem enable the rendering of significantly higher damping gains compared to the full-featured impedance control applied on the identical RRLabSEA hardware. Due to the removed need for the actuator's overall length, no initialization routine is required by the antagonistic muscular control.

Of course, the suitability of this reduced actuator-local impedance generation is also biologically motivated. Investigations on muscular *preflexes* by Brown and Loeb [Brown 00] strongly support the decision to prefer damping over stiffness. In their work, they explicitly conclude that their studies on natural muscles and their intrinsic dynamics have shown the effect of the force-length relation to be neglectable compared to the effect of the force-velocity relation w.r.t. perturbation resistance and system stabil-

ity. Furthermore, force and damping are directional and non-directional properties, respectively.

Antagonistic muscles may counteract force, but antagonistic damping is always summed. Hence, muscular co-contraction can be used to increase damping independent of force. However, vice versa, increasing the force without damping is not possible. This way, the muscular force defines a lower bound to its damping, which hypothetically may ensure overall stability throughout the whole range of muscle activity.

The proposed reduced impedance control on SEA hardware defines a consistent, antagonistic muscular control principle. A single, bidirectional SEA represents a pair of two antagonistic, unidirectional natural muscles. The two independent, antagonistic muscle activities, $a_1, a_2 \in [0, 1]$, are bijectively mapped to two independent control commands, the force F and the non-negative damping $D \geq 0$, of a single SEA.

$$\begin{pmatrix} F \\ D \end{pmatrix} = \begin{bmatrix} F_{Max} & -F_{Max} \\ \frac{D_{Max}}{2} & \frac{D_{Max}}{2} \end{bmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \tag{3.4}$$

$F_{Max} > 0$ and $D_{Max} > 0$ describe the antagonistic muscle pair's maximum possible, absolute force and damping capabilities, respectively. Individual muscle activities are mapped to either a positive or negative force but always to a positive damping. Thus, as observed from nature and pictured in Figure 3.10, muscular force generation without damping cannot be generated. On the other hand, simultaneous, identical muscle activation $a_1 = a_2$, i.e., pure muscular co-contraction, cancels out the two antagonistic forces and results in pure damping.

Since there is no active stiffness within the antagonistic muscular control, the SEA's passive stiffness requires a brief discussion. It is clear that in an ideal, fully transparent, and massless motor unit with no damping, the whole SEA will compress and extend under load with no spring deflection. I.e., the whole SEA will show zero-stiffness behavior.

In the other extreme case, i.e., infinite damping of the massless motor, the overall sea will behave just like its internal spring. Between those damping extremes, the observable stiffness continuously rises from zero, approximating the in-built mechanical stiffness. Hence, with increasing damping of the contractile element, the in-built tendon stiffness also increases to dominate the outside observable overall muscle-tendon stiffness characteristic.

The observable end-point stiffness of a natural limb can, thus, be stiffened via muscular co-contraction up to the hard upper limit of its in-built mechanical compliance. With this effect, the novel antagonistic muscular control strategy satisfies the desired Requirement 2.3: *"Variable Compliance"*. Nevertheless, the end effector's general impedance shape stays primarily related to the limb's posture, as stated in [Mussa-Ivaldi 85].

Various other studies on disturbed natural motions support this theory [Milner 98, Granata 01, Milner 02, Gribble 03, Reeves 08, Cashaback 15]. All those biological studies observe a rise of muscular co-contraction to counteract unexpected disturbances or to increase motion accuracy. The only but important drawback of co-contraction is that increased damping always comes at the cost of higher energy consumption.

3.2.4 Interpretation of the Natural Musculoskeletal System

It can be concluded that a hybrid low-level impedance control layout is best suited to achieve natural control, as stated in Theorem 1.1. The match of derivations from biological observations and technical reasoning strengthens this hypothesis. As an interesting side effect, the above argumentation provides an apparent technical reason and a semantic to the natural musculoskeletal control system and its observable control characteristics. Thus, the hypothesis that nature follows the just-derived principle of antagonistic muscular control might be an interesting starting point for future physiological and neural investigations on human behavior and potentially reverse-engineering the CNS' neural control mechanisms.

An other interesting biological question can be raised due to the technical design of the antagonistic muscular control. The *Golgi Tendon Organ* measures a natural muscle's force between its contractile element and a tendon. It is, thus, the biological analog of the current feedback usually embedded within modern brushless DC motor drivers. If now an external impact (positive or negative) hits a SEA, its spring will deflect, generating a respective torque on the motor's rotor via the back-drivable gear. Consequently, the rotor will decelerate or accelerate, which, in turn, will decrease or increase the complex AC impedance inside the rotor's windings. Eventually, this change in complex resistance will lead to an increased or decreased current flow and, thus, a motor torque counteracting the initial impact.

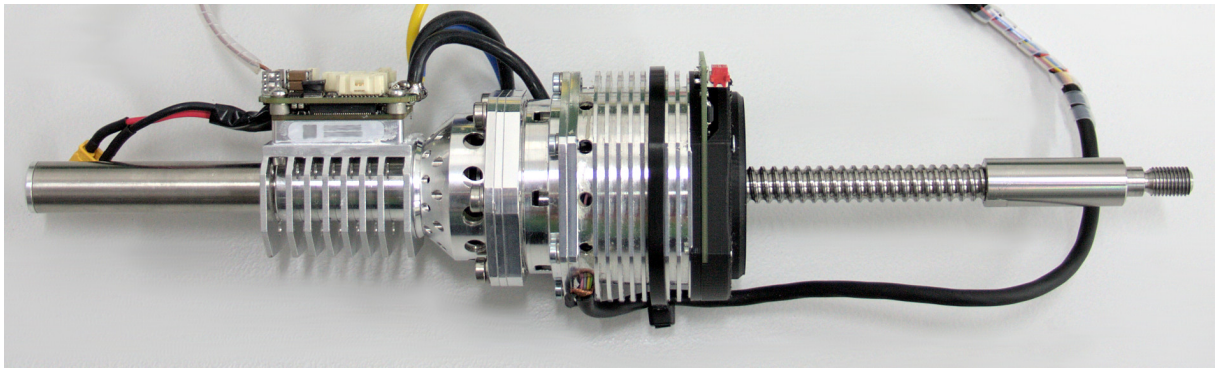
Of course, a modern motor drive will rapidly neutralize this sudden change in current flow. However, if this compensation is delayed, a behavior similar to the muscle stretch reflex emerges. Hence, the question must be raised whether the natural muscle stretch reflex , the only reflex without *interneurons*, is really present on purpose. It is absolutely possible that the neural feedback loop of this reflex only implements a SEA's inner current control loop. In this case, the muscle stretch reflex is nothing more than a consequence of bad long-delay hardware, i.e., neurons.

Future investigations into this question and a potential further simplification of the just-proposed antagonistic muscular control approach seem interesting and valuable for future research.

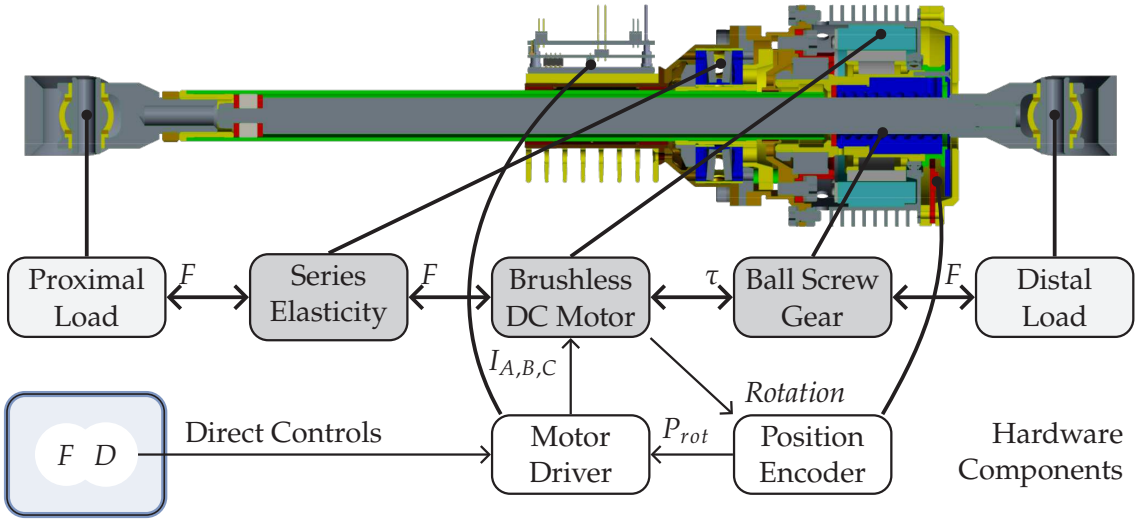
3.2.5 Reduced Series Elastic Actuator Design

With the rigorously simplified, antagonistic muscular control introduced in the previous Section 3.2.3, a simplified actuator hardware design becomes possible. While the general mechanical setup stays identical to the classical SEA design, the required control hardware can be drastically reduced due to the elimination of the active control of the actuator's elastic element. A scheme of such a reduced, second-generation RRLab SEA is given in Figure 3.12. The radical reduction of required hardware components in the novel design is uncovered by a simple visual comparison of the novel design against the current one pictured in Figure 3.3.

The first simplification shows up at the hardware level by removing the spring deflection sensor. This decision significantly relaxes the mechanical design on integrating the series elasticity into the actuator's hardware. Due to the absence of sensing hardware,



(a) First functional prototype of the second-generation RRLabSEA including its motor driver



(b) Cross-section of the RRLabSEA's novel CAD model and its hardware components

Figure 3.12: The novel, second-generation RRLabSEA hardware designed to support the novel antagonistic muscular control principle with lower weight and volume compared to the hardware currently in use. Pictured is (a) a photo of one of the first function prototypes and (b) a scheme of its mechanical construction. Due to the simplicity of the antagonistic muscular control, no external compute unit is required. The two control inputs, the contractile force F and Damping D are directly injected into the brushless DC motor driver.

the overall spring system can be fully encapsulated and optimized in both volume and weight, as proposed in the novel CAD design. Moreover, the positioning of this passive elasticity is generally unrestricted and may also be entirely decoupled from the actuator and integrated into a robot's skeletal structure. However, the decision on which stiffnesses are best for each actuator location within a robot is still an open research topic beyond the scope of this work. Hence, this morphological robot design decision is left open for future work.

Second, a further simplification with a significant effect on the actuator's complexity and performance is embedded within the simplified actuator-local impedance control. The antagonistic muscular control implements nothing more than a standard proportional velocity control with the proportional gain D. In modern, integrated brushless DC motor drivers, such a basic control functionality can be configured on-board. Consequently,

the need for a dedicated, separate processing unit, such as the FPGA in the case of the original RRLabSEA design, vanishes. As shown in the block diagram of Figure 3.12b, the two antagonistic muscular control commands are directly injected into the motor driver. Removing that extra computing unit significantly reduces the required hardware components and wiring efforts. For example, the communication delay between the motor driver and its peripheral impedance control is entirely dropped.

However, this new hardware design is still in its prototyping stage. Furthermore, equipping the bipedal robot CARL with those new actuators, the respective electronics, and appropriate communication protocols is beyond the scope of this thesis and, therefore, left open for consecutive works. This work's primary focus is on the development, implementation, and validation of a novel, natural control philosophy rather than on the reconstruction of robot hardware. Hence, throughout this work, the original RRLab SEA hardware is kept in use but equipped with the novel hybrid impedance control, as sketched in Figure 3.11.

3.2.6 Dynamics of Antagonistic Muscular Control

With the novel proposed antagonistic muscular control, the dynamic behavior of the overall actuator significantly changes compared to its previous impedance control design. Introducing the uncontrolled series elastic element significantly affects the actuator's stability. To provide an overall impression of the dynamic behavior of muscular-controlled SEAs, several views of a simplified dynamic model are presented and interpreted in the following. In the simplified model, all sources of disturbances, such as control delays, sensor noise, friction, or spring-intrinsic damping coefficients, are neglected. This way, the general shape and characteristics of the antagonistic muscular control dynamics are highlighted without parasitic effects. A block diagram of the simplified actuator dynamics, modeled in *MATLAB Simulink*[16], is shown in Figure 3.13.

The actuator's mechanical dynamics are modeled as a so-called *physical system* using mechanical component blocks, visualized in green. From left to right, the model starts with a mechanical ground attached to an ideal damper D_{in} and an ideal force source F_{in} in parallel. These two components represent the two commanded control inputs of the antagonistic muscular control, as pictured in Figure 3.11. Both controls are connected to the mass m_J modeling the actuator's reflected rotor inertia. Continuing rightward, attached to the reflected inertia, is the series elastic element K, followed by the two masses m_C and m_{ext}. Those masses respectively model the actuator's chassis and some arbitrary external load.

Since the antagonistic muscular control principle aims to control the robot's motion, the system's output on the right is defined as its overall velocity. This velocity is the summed velocity of the damper, i.e., the ball screw spindle, and the deflection change of the passive spring. With the help of the technical block $f(x)=0$, the mechanical model is automatically converted into its transfer function. The red connectors visualize the input and output of the converted system. In the setup, as visualized in Figure 3.13, the mechanical model is embedded between a standard step input followed by an additional

[16] *MATLAB Simulink* is a commercial software for system modeling offered by the company *MathWorks*. Further information is available at `https://de.mathworks.com/products/simulink.html`.

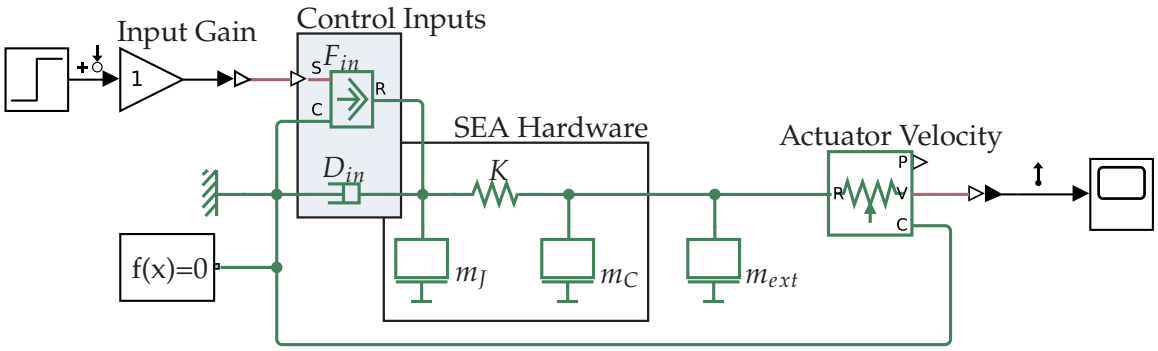


Figure 3.13: Scheme of the simplified dynamics of SEA hardware equipped with antagonistic muscular control. The mechanical model is drawn in green with so-called *physical system* components. From left to right, these components are a mechanical ground, an ideal force source F_{in} in parallel to an ideal damper D_{in}, the reflected inertia m_J, the series elastic elements K, the actuator's chassis mass m_C, some arbitrary, external load m_{ext}, and finally an ideal velocity sensor. The mechanical model is embedded between a step input followed by an additional input gain 1 on the left and a so-called *scope* on the right, used to plot the step response within *MATLAB Simulink*. The technical block $f(x) = 0$ provides necessary software functionality without affecting the model itself. The block diagram is modeled and evaluated in *MATLAB Simulink*.

input gain 1 on the left and a so-called *scope* component on the right. The latter is used to plot the simulated step response within *MATLAB Simulink*.

As pictured in Figure 3.13, the system analysis is performed as an open-loop Single-Input Single-Output (SISO) system with the actuator's force control F_{in} as input and the actuator's overall velocity as output. The three hardware-intrinsic constants K, m_J, and m_C are configured to the experimentally evaluated parameters of the RRLabSEA 50, as published in [Schütz 20a, p. 153,154]:

$$K = 200.8\,\frac{\mathrm{kN}}{\mathrm{m}} \qquad m_J = 8.9\,\mathrm{kg} \qquad m_C = 0.9\,\mathrm{kg} \tag{3.5}$$

The two left-over variables, the control command D_{in} and the external load m_{ext} are kept constant within each analysis. Showing the general effects those parameters have on the open-loop dynamics of the novel proposed SEA control, multiple Bode plots with different damping and load parameterizations are drawn on top of each other in Figures 3.14 and 3.15, respectively.

Figure 3.14 plots the actuator's open-loop dynamics at four increasing, constant damping commands $D_{in} = 10^{0..3}\,\mathrm{kg/s}$ without external load. Three main characteristics can be derived from this plot. First, the four magnitude graphs are stacked equidistant from each other in the lower frequencies range. The top-most orange graph starts at a magnitude of 0 dB at $D_{in} = 0\,\mathrm{kg/s}$, with the others following ordered with increased damping at −20 dB distance each. Hence, by the definition of the magnitude unit, [dB], for low force input frequencies, the actuator strictly follows the input according to the given damping, as desired.

Second, both plots show increasing straight magnitude and zero phase response with increasing damping. It can directly be followed that the actuator damping, the secondary

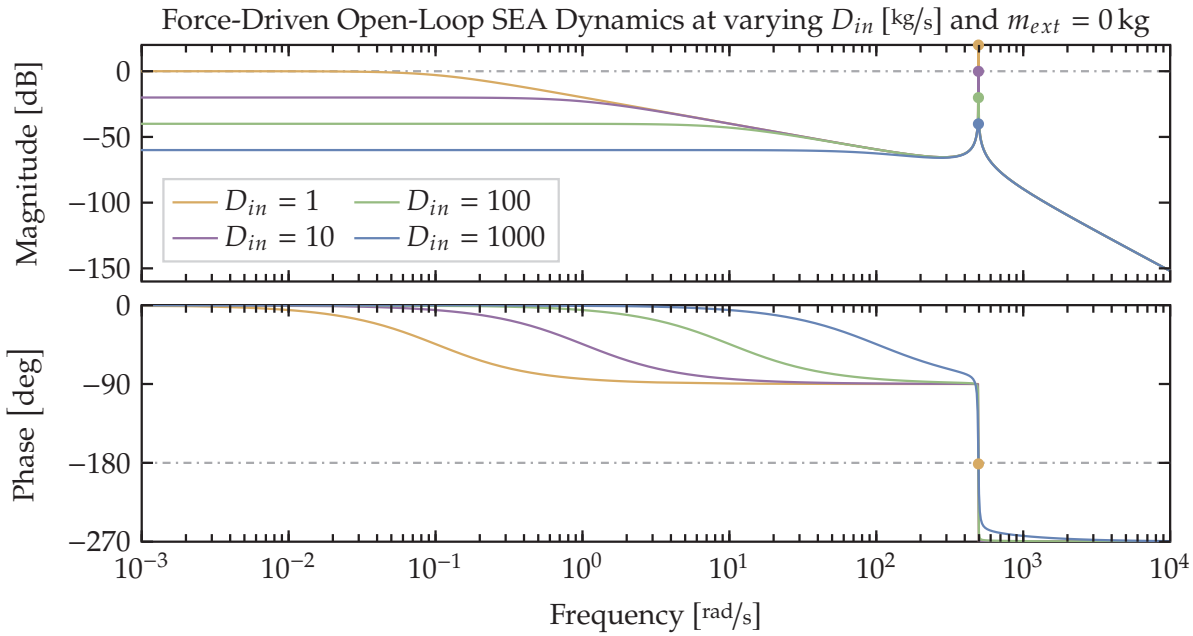


Figure 3.14: Open-loop dynamics of the novel proposed antagonistic muscular control, as sketched in Figure 3.11 and modeled in Figure 3.13. Since this approach aims for motion control according to Theorem 1.1: *"Natural Control"*, the plots show the SEA's velocity response to force input F_{in} at four different damping commands. Within each dynamics plot, their individual damping command D_{in} is kept constant. Although the system's input and output are different properties, the critical closed-loop stability points are marked in the diagrams.

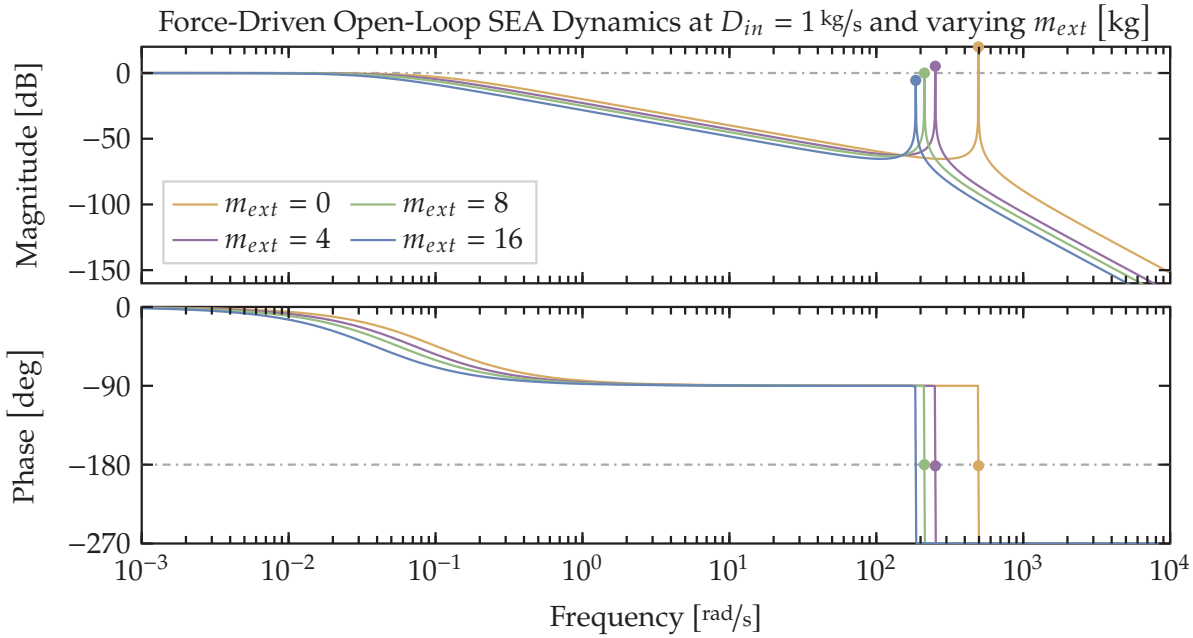


Figure 3.15: Open-loop dynamics of the novel proposed antagonistic muscular control, as sketched in Figure 3.11 and modeled in Figure 3.13. Since this approach aims for motion control according to Theorem 1.1: *"Natural Control"*, the plots show the SEA's velocity response to force input F_{in} at four different external loads m_{ext}. The critical closed-loop stability points of all graphs are marked, although the system's input and output are different.

SEA control command, can realize the secondary, hypothesized control characteristic, i.e., motion precision, as declared in Theorem 1.1: *"Natural Control"*. Hence, at the cost of higher energy consumption caused by higher damping values, the proposed open-loop SEA control can increase its perfect response bandwidth and, consequently, its commanded input tracking precision. As desired, the actuator-intrinsic damping control features the primary characteristic of observed, natural muscular co-contraction, as explained in Section 3.2.3.

Figure 3.14 also uncovers the expected stability criticality due to the uncontrolled series elasticity. As illustrated in the plots, the damping command does not affect the natural frequency of the uncontrolled spring-mass subsystem within the SEA hardware. However, the response's magnitude is still inversely proportional to the damping. An infinite magnitude will only occur at zero damping, which the control software must prohibit. Furthermore, defining a minimal damping control so that the magnitude peak at the system's critical frequency stays below 0 dB might be an even more suitable choice w.r.t system stability. In this case, such a choice would be $D_{min} > 10$.

Note that the system model used for this analysis (Figure 3.13) is highly simplified and idealized. It is intended to gain a first impression of the system's dynamics but should not be utilized for concrete value extraction. A suitable safety margin regarding D_{min} in the real system should be considered. Notably, when the damping value is increased to a higher level than plotted in Figure 3.14, the magnitude peak slightly decreases in frequency before stabilizing again at one particular frequency.

Treating the system's stability, Figure 3.15 plots the SEA's dynamics at constant damping $D_{in} = 1\,\mathrm{kg/s}$ with four increasing external loads m_{ext}. Note that the orange graph equals the orange one in Figure 3.14. As expected, with increasing external load, the natural frequency reduces. However, with the natural frequency, the response magnitude also decreases. In this case, the magnitude peak reaches the critical value of 0 dB at exactly $m_{ext} = 8\,\mathrm{kg}$. This load mass is the exact point where the masses on both sides of the SEA's spring element are equal $m_J = m_C + m_{ext}$. The system's peak magnitude drops below 0 dB for heavier external loads. Since the spring force acts equally on both sides, it is clear that the lighter side will react with higher acceleration and, consequently, higher integrated velocity than the heavier one. Thus, energy is transferred from the heavier mass through the actuator's elastic element into the lighter mass.

The system's general theoretical stability entirely relies on the smallness of the reflected motor inertia $m_J < m_C + m_{ext}$. Hence, small reflected inertia is crucial not only for impact tolerance but also for the stability of the proposed antagonistic muscular control. Note that the perfect system, $m_J = 0\,\mathrm{kg}$, is close to being given by a natural muscle's non-rotating, non-geared contractile element.

Thinking beyond the scope of a single actuator, a similar stabilizing effect may or may not occur on the scale of a whole robot, e.g., when trying to stabilize an un-actuated but relatively heavy trunk using only lightweight, attached limb control.

For a better impression of actual motion control, i.e., velocity control, the force input of the system model in Figure 3.13 is multiplied by an additional gain equal to the damping $G_{in} = D_{in}$. This way, the previous force input becomes a desired velocity $F_{in} = D_{in}\, v_{in}$, and the whole system becomes a velocity-transmission SISO system capable of being

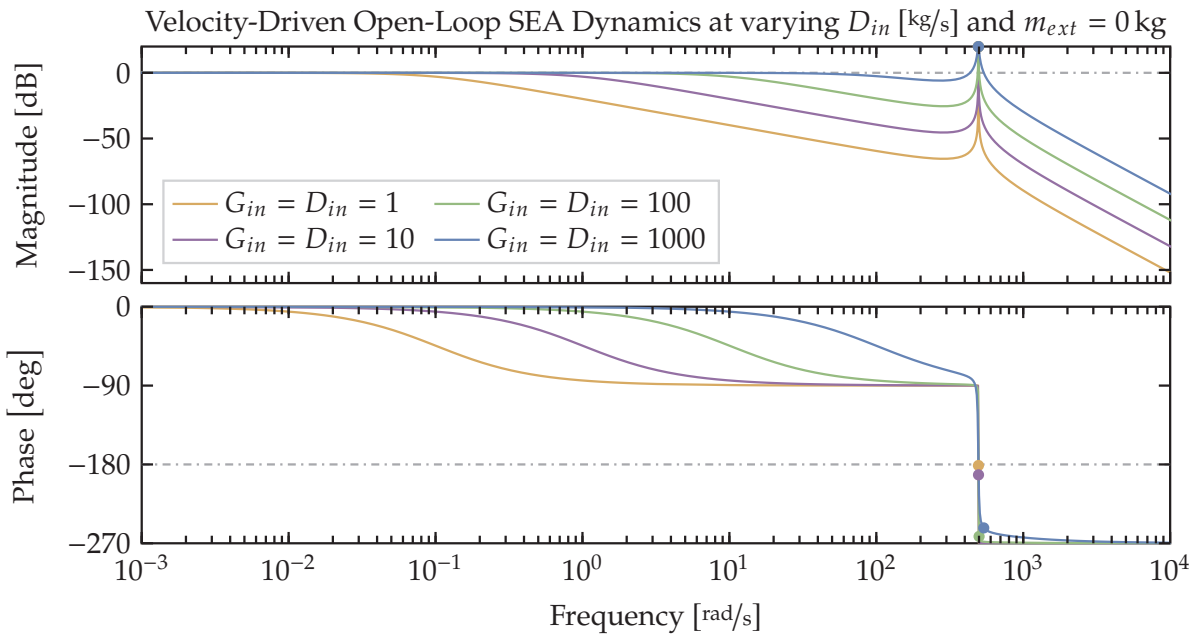


Figure 3.16: Open-loop dynamics of the novel proposed antagonistic muscular control, as sketched in Figure 3.11 and modeled in Figure 3.13. To offer an impression of possible closed-loop control characteristics, unlike the Figures 3.14 and 3.15, the system's input is multiplied with an additional, proportional gain equal to the actuator's damping control input $G_{in} = D_{in}$. This way, the system's dynamics form a closed-loop-capable velocity transfer function. The plots show the actuator's velocity response w.r.t. velocity input $F_{in} = D_{in}\, v_{in}$ at four different damping commands D_{in}. In all graphs, the critical closed-loop stability points are marked.

used in a closed-loop scenario. Equal to the first analysis setup, the external load is set to zero, and the damping command is kept constant within four individual dynamics plots, visualized in Figure 3.16. For better comparison, note that the orange graph, again, is equal to the orange ones of the previous system analyses in Figures 3.14 and 3.15.

The additional input gain increases the magnitude of all four plotted system dynamics graphs to exactly 0 dB, as desired for open-loop control. At the same time, the peak magnitude at the system's critical frequency is raised to an identical point throughout all damping values. Hence, unlike the previous system, stability cannot be achieved by introducing an enforced minimum damping command. Instead, stability entirely relies on the actuator's transparency. The additional input gain does not affect the muscular co-contraction property, which increases the control bandwidth with perfect 0 dB input-to-output transmission magnitude and zero phase shift.

The three analyses above show that antagonistic muscular control can be used for open-loop force and velocity control. The system satisfies all desired natural control requirements and closed-loop stability criteria. However, due to the passive elasticity, the control's closed-loop stability entirely depends on the underlying hardware.

The proposed novel control is strongly coupled with its controlled hardware and no general-purpose solution. Applied on suitable hardware, the novel SEA control promises to be a simple, hybrid muscle-inspired control system with only a few required parameters and control characteristics, similar to biological observations.

3.3 Distributed Impedance Generation

In the previous Sections, the musculoskeletal morphology and the bio-inspired antagonistic muscular control strategy of the bipedal robot CARL and its RRLabSEAs have been introduced. The central aspects of these designs are the robot's natural over-actuation utilizing more muscle-like SEAs than existing skeletal DoFs and the hybrid, distributed impedance generation. One benefit of distributed impedance is a hierarchical feedback structure offering reduced centralized communication efforts and delays. Furthermore, the crucial natural control Requirement 2.1: *"Hybrid Dynamics"* is satisfied at the actuator level by design.

However, there are also challenges arising from impedance distribution. This section examines the implications of a distributed, hybrid impedance control strategy, as developed in Section 3.2.3, on the overall robot control. The mathematical problem description is outlined, and a detailed analysis of the actuator-individual impedance contributions to an arbitrary Work Space (WS) representation is given.

3.3.1 Work Space, Configuration Space, Actuation Space

In an easy and intuitive higher-level robot control interface, an arbitrary TCP might be controlled by its appropriate pose, twist, or wrench description vectors in a Cartesian WS representation. The real-world robot motion, however, is executed in the robot's so-called Configuration Space (CS). This space represents each skeletal DoF within an individual dimension, hence implementing robot-intrinsic complete generalized coordinates. Due to the muscular over-actuation, a third space is required to realize the desired control in hardware. The so-called Actuation Space (AS) describes a vector space representing each available actuator individually. In general, the dimensionality of all three spaces may increase in the order they are named.

$$\dim(WS) \quad \leq \quad \dim(CS) \quad \leq \quad \dim(AS) \tag{3.6}$$

This means that, as usual, a robot must have at least as many DoFs as the world it is operating in and, similarly, require at least as many actuators as its own skeletal DoFs. In the case of the planar robot CARL, this equation holds for each leg individually.

$$\dim_{\text{CARL}}(WS) \quad = 2 \times 3 = \quad \dim_{\text{CARL}}(CS) \quad < \quad \dim_{\text{CARL}}(AS) \; = 2 \times 5 \tag{3.7}$$

Note that both the CS and the AS are Cartesian spaces, although they may contain rotational joint or actuator information. In contrast, the WS is to be defined arbitrarily and might, e.g., contain orientation representations by complex numbers, quaternions, or other higher-dimensional geometric representations. However, for the analysis within this section, the intuitive, two-dimensional, orientation-less, Cartesian subspace of the three-dimensional planar world description is used as exemplary WS. At this point, it is essential to note that the dimensionality property above represents the real, unconstrained DoF, i.e., the number of variables within a description vector minus the number of potentially existing constraints on that vector. For example, the space of quaternions is considered three-dimensional although containing four-dimensional vectors. Figure 3.17 sketches the kinematic coupling of the twist and wrench vectors between these three spaces and their respective interconnection Jacobian matrices.

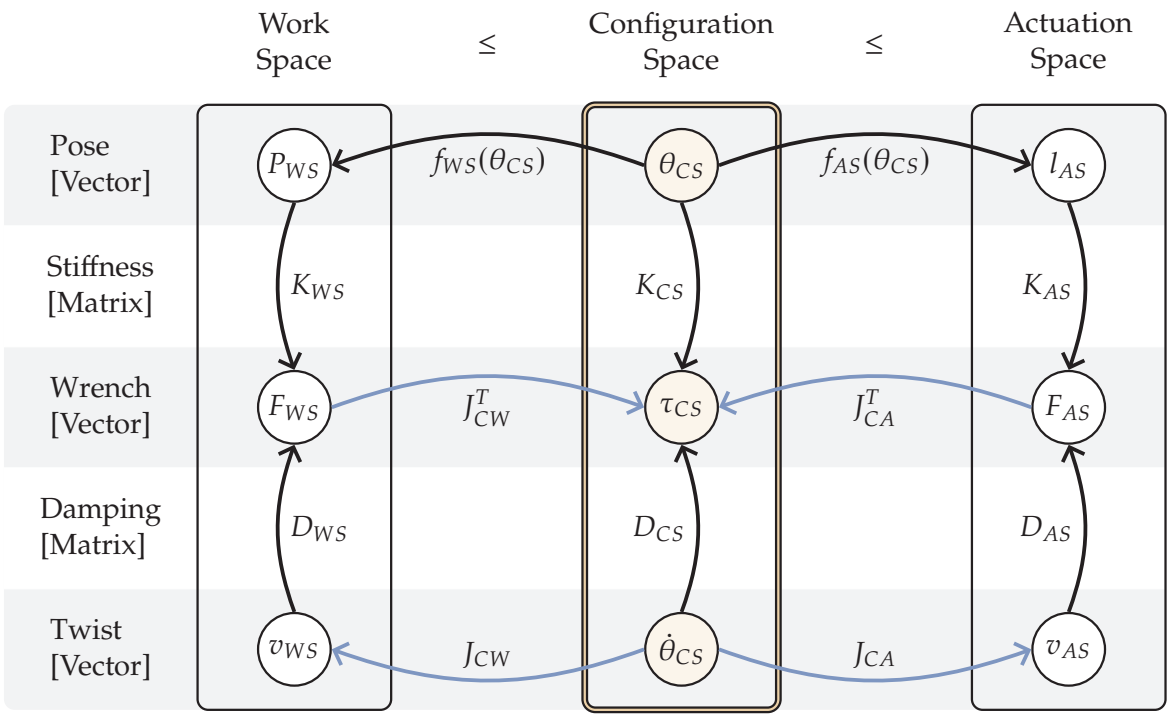


Figure 3.17: Sketch of the kinematic coupling of all five relevant impedance properties: pose, stiffness, wrench, damping, and twist– among each other and across the three representation spaces – Work Space (WS), Configuration Space (CS), and Actuation Space (AS). Each circular node represents a description vector. The arrows represent respective transformations between two such vectors, pointing in the transformations' well-defined directions. Their inverses may be ambiguous. The four Jacobian transformations that are relevant for the antagonistic muscular control, as defined in Section 3.2.3, are highlighted in blue. In the middle, the CS is highlighted in orange. This space always defines a robot's generalized or minimal coordinates. Pose and twist vectors combine into a state description. The wrench vector specifies the robot's action. It can be noticed that both state description nodes feature only outgoing transformations, while the action node has only incoming transformations.

From left to right, the three representation spaces, WS, CS, and AS, are represented in increasing order by their possible dimensionalities, as indicated by the less-or-equal symbols. From top to bottom, the five relevant impedance properties, pose, stiffness, wrench, damping, and twist, are represented. In every space, the two outer description vector nodes, pose and twist, specify the respective system state, while the middle wrench nodes describe the space-respective actions.

The six vertical arrows in Figure 3.17 couple the individual vector nodes within each representation space. They define the first-order impedance terms stiffness and damping. Both are represented by symmetric, positive semidefinite square matrices K and D, respectively. Both matrices can be treated as always invertible by allowing them to become infinite. However, both matrices are constrained by the robot's kinematics.

By definition, actuator-intrinsic impedance, given by the vector z_{AS}, forms a pure diagonal impedance matrix Z_{AS} in the AS:

$$Z_{AS} := \text{diag}(z_{AS}) \tag{3.8}$$

Off-diagonal elements of an AS impedance matrix embody herein-unpermitted impedance correlations between actuators. Such correlations can be achieved by granting actuators access to other actuators' local state information. By doing so, additional virtual actuators are generated, which offer further impedance generation capabilities at the lowest hardware level. However, since such virtual actuators do not contribute additional hardware, their impedance contribution is limited to improved generation precision. General higher impedance values cannot be achieved in this manner.

On the other hand, a low-level realization of such multi-actuator impedances requires additional low-latency inter-actuator wiring and hardware clock and control synchronization. Thus, whether the achievable improvement of low-level impedance precision justifies the required extra hardware is questionable. However, further detailed investigations are necessary to evaluate this topic, thus offering an interesting direction for future research beyond the scope of this work.

Within the CS, each actuator contributes a rank-1 sub-matrix in the dimensions of its spanned joints. These sub-matrices are spread according to their joints' positions within the overall state vector. Monoarticular actuators contribute a 1×1 sub-matrix, biarticular actuators contribute 2×2 sub-matrices, and generally, n-articular actuators contribute $n{\times}n$ sub-matrices, all of rank 1. Thus, CS impedance is the sum over all existing actuators $a \in AS$ of the matrix products of each actuator-respective column vector in J_{CA}^T, scaled by the actuators' individual impedance components $z_{AS}[a]$.

$$Z_{CS} \;=\; J_{CA}^T \,\mathrm{diag}(z_{AS})\; J_{CA} \;=\; \sum_{a \in AS} z_{AS}[a] \left[\mathrm{col}_a\left(J_{CA}^T\right) \mathrm{col}_a^T\left(J_{CA}^T\right) \right] \tag{3.9}$$

The six horizontal arrows in Figure 3.17 represent space transformations. Due to the spaces' unequal dimensionalities, those mappings are generally not invertible. Interestingly, all state mappings originate at the generalized coordinates in the CS, decreasing the data dimensionality toward the WS but increasing the data dimensionality toward the AS. Hence, especially in the AS, pose and twist configurations exist that can be commanded to the distributed SEAs but are mechanically impossible to be tracked by the robot hardware. The two wrench mappings in the middle behave vice versa, summing up multiple actuator forces into lower-dimensional joint torque equivalents. Hence, actuators can cooperatively combine their forces or distribute a desired torque according to some arbitrary optimization criterion. In the reverse direction, from the CS to the WS, those relations and implications reverse. Note that in the case of antagonistic muscular control, as introduced in Section 3.2.3, only the lower Jacobian transformations, highlighted in blue, are relevant.

Both coordinate transformations are entirely defined by their respective pose transformation functions $f()$. Those functions are unrestricted and will usually be highly non-linear. The AS transfer function $f_{AS}(\theta_{CS})$ is given by the robot's mechanical morphology and its concrete actuator mounting positions. In contrast, the WS transfer function $f_{WS}(\theta_{CS})$ is task-specific. Within this context, it is the standard, planar, Cartesian forward kinematic of each leg of the robot Carl, individually. The twist and wrench mappings are the first derivatives of their defining pose transformation functions, i.e., the Jacobian and Jacobian transpose matrices, respectively. Note that due to the derivative-inherent

linearization, the Jacobian matrices represent a linear transformation equal to the impedance matrices. However, according to the divergent dimensionalities of their definition and image spaces, those Jacobian matrices may be non-square. In the case of the robot CARL, the actuation Jacobian J_{CA} is a 5×3 matrix, as introduced in Figure 3.6 within Section 3.1.3. Further details on the actuation Jacobian of CARL are available in [*Nejadfard 18b*, *Nejadfard 20*].

3.3.2 Actuator-Individual Work Space Impedance

Distributed impedance generation within a robot's actuators offers low-delay feedback loops with only limited, uncoupled sensor noise and, hence, potentially high-gain controllers. However, aside from those benefits, this strategy also introduces drawbacks to the effort of actuator coordination and achievable impedance versatility. First of all, the additional impedance control commands have to be generated by some higher-level intelligence and have to be distributed to the actuators. Hence, the interconnection between the robot's actuators and its CPU becomes more complex. Second, the achievable overall CS impedance is limited to a linear combination of the actuator impedances, as introduced in Equation (3.9). The subspace of distributed achievable CS impedances of a robot is entirely defined by its morphology, i.e., the number of available actuators and their placement within the robot's skeleton.

Since the CS still does not offer the most intuitive description of a robot's behavior, a description within the WS is desired. To do so, the CS impedance can be expressed in terms of WS impedance, analogous to the actuator-based CS impedance, introduced in Equation (3.9):

$$\begin{aligned} Z_{CS} &= J_{CW}^T Z_{WS} J_{CW} &&= \sum_{a \in AS} J_{CW}^T Z_{WS[a]} J_{CW} \\ &= J_{CA}^T \operatorname{diag}(z_A)\ J_{CA} &&= \sum_{a \in AS} z_A[a] \left[\operatorname{col}_a\left(J_{CA}^T\right) \operatorname{col}_a^T\left(J_{CA}^T\right) \right] \end{aligned} \tag{3.10}$$

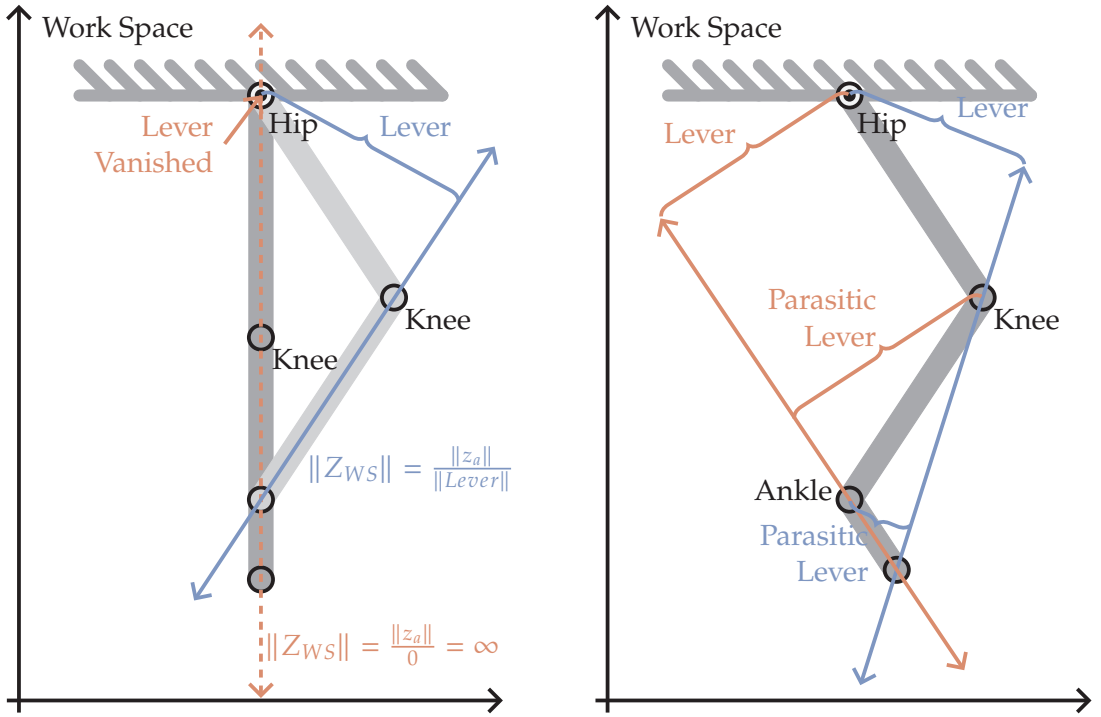
with $\sum_{a \in AS} Z_{WS[a]} = Z_{WS}$ being the WS impedance matrix split into a sum of actuator-related parts $Z_{WS[a]}$. By exploiting this equality's summations on both sides, it is possible to observe Equation (3.10) for every available actuator individually:

$$\forall a \in AS \ : \quad J_{CW}^T Z_{WS[a]} J_{CW} \quad \stackrel{!}{=} \quad z_A[a] \left[\operatorname{col}_a\left(J_{CA}^T\right) \operatorname{col}_a^T\left(J_{CA}^T\right) \right] \tag{3.11}$$

$$\Rightarrow \quad \forall a \in AS \ : \quad J_{CW}^T\, e_a \quad \stackrel{!}{=} \quad \sqrt{z_A[a]}\, \operatorname{col}_a\left(J_{CA}^T\right) \tag{3.12}$$

with e_a being the only existing eigenvector of $Z_{WS[a]}$, scaled by the square root of the eigenvalue, and hence $e_a e_a^T = Z_{WS[a]}$. The decomposition of the actuator-related WS impedance into a single, scaled eigenvector is guaranteed by the fact that each actuator is a one-dimensional object. Therefore, independent of how those actuator impedances are further processed, they must not generate impedances of higher dimensionality.

As can be seen in Equation (3.12), the directions of the vectors e_a represent the directions of action of the actuators with the WS. However, it can be seen that the computability and even the existence of those direction vectors e_a entirely depend on the WS definition,



(a) Singular WS Jacobian J_{CW} **(b)** Non-quadratic WS Jacobian J_{CW}

Figure 3.18: Visualization and mechanical interpretation of the mathematical problems of actuator-individual impedance generation in the WS correlated to the properties of the J_{CW} Jacobian at the example of a monoarticular hip actuator. On the left, Figure (a) sketches the problem of matrix singularity in the example of a planar two-link system. Figure (b) on the right shows the problem of a non-quadratic transformation Jacobian at the example of a planar, three-link system sketch. Visualized is the case of having a higher-dimensional CS than WS. Both examples feature the leg morphology of the robot Carl.

i.e., the Jacobian transformation matrix J_{CW}. Figure 3.18 illustrates the mechanical interpretation of the mathematical problems related to matrix singularity and non-quadratic dimensions.

One case in which J_{CW}^T within Equation (3.12) is not invertible and e_a, the actuator's direction of action vector, cannot be resolved whenever J_{CW} is singular. Figure 3.18a visualizes the problem of singularity at the hand of a monoarticular hip actuator within a planar, two-link robot without TCP orientation. As analyzed in detail in the work of Nejadfard et al., the direction of WS force generated by a hip torque, i.e., a monoarticular hip actuator, is always aligned with the robot's shank segment [*Nejadfard 18a*]. This alignment results from the requirement that all generated forces have to exactly pass through all non-actuated joints in order not to affect them. The distance between these vectors and the actuated joint embodies the effective lever arm coupling the joint torque with the respective WS force.

Without loss of validity, this pure force direction can be generalized to the direction of action of a complete, complex impedance. In Figure 3.18, those directions of action are pictured as blue and red bidirectional arrows. As highlighted in the sketch, the lever arm for the hip actuator vanishes in the robot's singular position. Since the WS

impedance is divided by the lever arm (stiffness and damping are divided by the square of the lever arm), the WS impedance approaches infinity. Hence, in a singular position, the effective direction of actuation of such an actuator in the WS is undefined and may not be used. However, vise versa, when a WS impedance is desired to be generated by an actuator, its correlated local impedance continuously and smoothly approaches zero, with the robot approaching singularity. Hence, impedance distribution algorithms will not suffer from numerical instabilities such as discontinuities or infinite results when applied in this direction.

The second reason for e_a not being calculable within Equation (3.12) is a non-quadratic, and hence non-invertible J_{CW} matrix. J_{CW} becomes non-quadratic when the robot's CS is of a different dimensionality than the defined WS. To visualize this problem, Figure 3.18b extends the planar, two-link robot of Figure 3.18a by an ankle joint. The orientation of the new TCP is again not considered. As visualized, in such a situation, there are two possibilities to orient the hip's direction of action in the workspace. Either it passes through the TCP and knee joint, or it passes through the TCP and ankle joint. In Figure 3.18b, both those variants are sketched in blue and red, respectively.

As stated above, the direction of action must pass through all non-actuated joints, which is impossible in the pictured robot configuration. Thus, as illustrated, both variants, when being applied, cause parasitic torques on the joints they do not pass through. Hence, actuators must not act independently from each other within such a WS. Multiple actuators must be activated in cooperation with each other to achieve valid WS impedance. However, as the stiffness and damping properties must not cancel each other out, such actuator cooperation often comes at the cost of additional parasitic WS impedance.

In conclusion, the following statement can be drawn from the above discussion. For arbitrary robot morphologies and configurations, actuator-individual impedance generation has neither a guaranteed proper nor unique WS representation. Hence, controlling WS impedance via distributed, actuator-intrinsic local impedances is not straightforward. It will always require some approximation strategy to distribute CS impedance best suitably. One possible optimization strategy is introduced in the upcoming Section 3.4.

3.4 Optimization-Based Muscle Recruitment Strategies

Section 3.3.1 introduced the increasing dimensionalities of the operational spaces, namely WS, CS, and AS. The exact dimensions w.r.t. the planar, bipedal robot Carl are given in Equation (3.7). Figure 3.17 shows the interconnection of those spaces by arbitrary but mostly linear, directed transformations. Directed, in this case, means that the transformations between spaces are generally projections and, hence, non-invertible. This limiting property follows directly from the spaces' different dimensionalities.

As explained beforehand, the CS represents the entire physical robot configuration while the AS represents the robots' actual actuation capabilities. This section covers the transformation of control commands given in CS into executable descriptions in the AS. Applying the antagonistic muscular control strategy introduced in Section 3.2.3, the relevant impedance control elements are reduced to only the wrench vector and the damping matrix. Hence, the region of interest in Figure 3.17 is the lower right square

formed by the four elements τ_{CS}, F_{AS}, $\dot{\theta}_{CS}$, and v_{AS}. All involved transformations within this area, D_{CS}, J_{CA}, and D_{AS}, are linear.

Following Definition 2.1, the bipedal version of CARL features three so-called *visible* CS velocity/torque control DoFs per leg, i.e., the skeletal joints. Accordingly, each leg features six additional so-called *measurable* CS symmetric positive semi-definite impedance control matrix DoFs. Thus, the robot's total required CS control DoFs sum up to 18, nine per leg. With the antagonistic muscular control applied to the RRLab SEAs, the available low-level controls in CARL are two control inputs (force and damping) for all five actuators per leg. Hence, the total number of available low-level control DoFs sums to 20. Although these numbers seem as if CARL is still over-actuated, a more detailed view leads to a different result.

The overall control commands are equally half distributed in 10 per leg, containing each five position-related *visible* force controls and impedance-related *measurable* damping controls. However, the robot's desired control DoFs in CS are split into only 6 positional and 12 impedance DoFs. Thus, CARL actually is mechanically over-actuated w.r.t. wrench control but, at the same time, under-actuated w.r.t. impedance control. This contradicting setup is intrinsic to any distributed or hybrid impedance control setup, as pictured in Figure 3.8 within Section 3.2.

The following two equations can be formulated to distribute CS force vectors and damping command matrices to a robot's actuators:

$$\text{Wrench} : \quad \tau_{CS} \overset{!}{=} J_{CA}^T \, F_{AS} \qquad \text{over-actuated } \mathbf{Optimization} \tag{3.13}$$

$$\text{Damping} : \quad D_{CS} \overset{!}{=} J_{CA}^T \, D_{AS} \, J_{CA} \qquad \text{under-actuated } \mathbf{Approximation} \tag{3.14}$$

3.4.1 Wrench Distribution

As introduced and described above in Equation (3.13), distributing a wrench vector in CS to a robot's actuators requires solving an underdetermined system of linear equations. Hence, the existence of a solution may be assumed, and the solution space, in general, may contain infinitely many valid solutions. In order to determine a unique solution, it is a standard procedure to define one or more quadratic, positive-definite optimization criteria $L_n^2(\cdot)$ on top of the actual problem. The minimization is then solved subject to the original equation as a linear constraint.

$$\begin{aligned} \underset{F_{AS}}{\arg\min} \;\; & \frac{1}{2} \sum_n L_n^2 \left(F_{AS} \right) \\ \text{s.t.:} \;\; & J_{CA}^T \, F_{AS} = \tau_{CS} \end{aligned} \tag{3.15}$$

Assuming the existence of a solution satisfying the optimization constraints, by definition of the quadratic optimization, this solution is unique.[17] The following two Actuator Force Distribution (AFD) criteria have been developed, tested, and analyzed on the single-leg version of CARL [*Nejadfard 18a*, *Nejadfard 19*][Nezhadfard 20].

[17] For readers unfamiliar with the topic of constrained optimization, the author recommends the book *"Constrained Optimization and Lagrange Multiplier Methods"* [Ber 14] for preliminary studies.

1. Normalized Thermal AFD:
   The thermal DC power loss P of the motor winding's electrical resistance R is proportional to the square of the winding's current I, $P = I^2R$. Since the current is generally proportional to the motor's torque and force after gear F_a, an electric motor's thermal loss is proportional to its quadratic commanded force after gear, $P \sim F_a^2$. Normalized by the actuator's maximum force $F_{a_{max}}$, the T-AFD criterion is formulated as follows:

$$L_T^2(F_{AS}) := \sum_{a \in AS} \left(\frac{1}{F_{a_{max}}} F_a \right)^2 \tag{3.16}$$

2. Normalized Power AFD:
   The mechanical output power P_a of a motor is its output velocity v_a multiplied by its output force, $P_a = v_a F_a$. Accordingly, an actuator's maximum output power $P_{a_{max}}$ is its maximum velocity $v_{a_{max}}$ times its maximum force, $P_{a_{max}} = v_{a_{max}} F_{a_{max}}$. The respective, normalized P-AFD criterion is formulated as follows:

$$L_P^2(F_{AS}) := \sum_{a \in AS} \left(\frac{P_a}{P_{a_{max}}} \right)^2 = \sum_{a \in AS} \left(\frac{v_a}{P_{a_{max}}} F_a \right)^2 \tag{3.17}$$

Note that the actuators' velocities v_a may become zero, leaving the P-AFD criterion L_P^2 only positive semi-definite. Hence, the P-AFD criterion must not be used independent of other real positive-definite criteria, such as the T-AFD. Weighting constants may be added to the individual minimization criteria to balance the influence of both T-AFD and P-AFD against each other if desired.

In the original, above-mentioned publication [*Nejadfard 18a*], an additional constraint is added to the optimization in Equation (3.15):

$$F_{AS_{max}} \overset{!}{\geq} \|F_{AS}\| \tag{3.18}$$

The original intention of this additional constraint is to restrict the optimization result within physically generable boundary conditions. While this intention seems valid on the first look, two problems arise from these constraints. First, limiting the result space F_{AS} without limiting the input space τ_{CS} may lead to contradicting constraints and, therefore, to an unsolvable optimization problem. Second, within the herein-proposed antagonistic muscular control, the actual generated motor force is not only the above distributed one but also the actuator-internally rendered damping force $F = F_a - D_a v_a$.

Assuming a higher actuator-internal control frequency than the one solving the above minimization, the actuator-internal damping force is non-constant within an optimization cycle. Hence, the best place to limit actuator forces is to limit the applied motor currents distributed within the actuators' internal low-level controllers, as sketched in Figure 3.11. In favor of mathematical and computational stability and maximum possible motor performance, Equation (3.18) is not applied within this work.

As can be seen in Equation (3.16), the T-AFD criterion balance the distributed forces relatively equally among the actuators. In contradiction to that, Equation (3.17) shows the

P-AFD criterion to rigorously prefer non- or less-moving actuators (in favor of biarticular actuators during appropriate robot motions). Its purpose is to redistribute the forces with the goal of reduced energy consumption [*Nejadfard 19*]. Unfortunately, this desired unbalancing may lead to undesired skew force limitations within the actuators' low-level controllers. A possible workaround for this case may be to reformulate Equation (3.18) only to limit the P-AFD contribution. However, since energy-efficient control goes beyond the scope of this thesis, investigations into this topic are left open for future work. Within the work at hand, wrench distribution is applied with only T-AFD-based optimization criteria, such as Equation (3.16).

By thinking beyond the scope of this work, more beneficial reasons can be found for the usage of an actuator force distribution optimization. With the pure antagonistic muscular control setup presented in Figure 3.11, each artificial muscle represents a pair of two antagonistic biological counterparts. Due to the mechanical equivalence of forward and backward motion and force, each actuator particularly implements two identical, antagonistic muscles. However, this usually is not the case in nature.

In biology, e.g., in the human body, every muscle is individually optimized, resulting in antagonists being unequal in size, power, tendon stiffness, and skeletal connection. Technically, such a setup may be approximated by adding additional passive springs parallel to the active actuation system. Such parallel springs produce configuration-based skeletal forces, hence shifting the operational window of the overall propulsion system away from its natural zero position.

Designing and optimizing a robot's musculoskeletal system according to its requirements, needs, and task desires is a non-trivial but highly complex and probably non-convex problem. Thus, future investigations in this research direction, aiming to uncover robot design strategies, seem promising.

3.4.2 Damping Distribution

Similar to the previously discussed wrench distribution strategy, damping distribution also requires an optimization technique to be solved. However, in contrast to wrench distribution, as introduced in Equation (3.14), determining distributed actuator-intrinsic damping requires solving a usually overdetermined system of linear equations. Damping in the CS is a symmetric matrix property. Hence, the number of DoFs within such a matrix is calculated by the following formula:

$$\#\mathrm{DoF}_D(CS) = \tfrac{1}{2}\dim(CS)\,(\dim(CS)+1) \tag{3.19}$$

Generally, there is one damping DoF per mechanical DoF plus one for each pair of two. Within a symmetric matrix representation, these are the diagonal and off-diagonal elements, respectively. Since all actuators are one-dimensional, at least one actuator for every above-mentioned DoF is required to represent all theoretically existing CS damping DoFs in the AS. Unfortunately, building this in hardware is usually neither possible nor suitable. Within the morphological setup of the bipedal robot Carl, this would require an actuator spanning between trunk and foot. Hence, in contrast to the previously discussed wrench distribution, damping distribution is an under-actuated optimization problem. Note that the damping's positive semidefiniteness in the CS only

restricts the valid solutions within the solution space but not the general DoFs. The same holds for the restriction to apply only positive damping to the actuators.

It can be concluded that, in general, a solution to the damping distribution problem does not exist in real-world applications, and an approximation is all that can be achieved. The goal of such an approximation is to find AS damping values D_{AS}, that best possible approximate the desired behavior of a given CS damping matrix:

$$\forall \dot{\theta}_{CS} \in \{\|\dot{\theta}_{CS}\| = 1\} \quad : \quad D_{CS}\, \dot{\theta}_{CS} \stackrel{!}{=} J_{CA}^T\, D_{AS}\, J_{CA}\, \dot{\theta}_{CS} \tag{3.20}$$

To determine a unique approximation solution, a quadratic, positive-definite optimization problem can be formulated, minimizing the integrated, weighted square error between both sides' matrices. The weight, in this case, is the symmetric, positive-definite matrix W.

$$\begin{aligned}
& \underset{D_{AS}}{\arg\min} \int_{\|\dot{\theta}_{CS}\|=1} \left\| W \left(D_{CS} - J_{CA}^T\, D_{AS}\, J_{CA} \right) \dot{\theta}_{CS} \right\|^2 && (3.21)\\
\Leftrightarrow \quad & \underset{D_{AS}}{\arg\min}\ \mathrm{tr}\left(\left(W \left(D_{CS} - J_{CA}^T\, D_{AS}\, J_{CA} \right) \right)^T \left(W \left(D_{CS} - J_{CA}^T\, D_{AS}\, J_{CA} \right) \right) \right) && (3.22)\\
\Leftrightarrow \quad & \underset{D_{AS}}{\arg\min}\ \mathrm{tr}\left(\left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right)^T \left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right) \right) \\
& \qquad - 2\, \mathrm{tr}\left(\left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right)^T (W\, D_{CS}) \right) && (3.23)\\
\Leftrightarrow \quad & \underset{D_{AS}}{\arg\min}\ \frac{1}{2}\, \mathrm{vec}\left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right)^T \mathrm{vec}\left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right) \\
& \qquad - \ \mathrm{vec}\left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right)^T \mathrm{vec}(W\, D_{CS}) && (3.24)
\end{aligned}$$

Using Equations (3.8) and (3.9) with damping-only impedance, the vectorization, including the desired actuator damping matrix D_{AS}, can be rewritten w.r.t. the actual actuator damping description vector d_{AS}.

$$\begin{aligned}
\mathrm{vec}\left(W\, J_{CA}^T\, D_{AS}\, J_{CA} \right) &= \mathrm{vec}\left(W\, J_{CA}^T\, \mathrm{diag}(d_{AS})\ J_{CA} \right) && (3.25)\\
&= \mathrm{vec}\left(W \sum_{a \in AS} \left(d_{AS}[a] \left[\mathrm{col}_a\left(J_{CA}^T \right) \mathrm{col}_a^T\left(J_{CA}^T \right) \right] \right) \right) && (3.26)\\
&= \mathrm{vec}\left(\sum_{a \in AS} \left(W \left[\mathrm{col}_a\left(J_{CA}^T \right) \mathrm{col}_a^T\left(J_{CA}^T \right) \right] d_{AS}[a] \right) \right) && (3.27)\\
&= \sum_{a \in AS} \mathrm{vec}\left(\mathrm{col}_a\left(W\, J_{CA}^T \right) \mathrm{col}_a^T\left(J_{CA}^T \right) \right) d_{AS}[a] && (3.28)\\
&= \underbrace{\left[\left(\mathrm{col}_a\left(W\, J_{CA}^T \right) \otimes \mathrm{col}_a\left(J_{CA}^T \right) \right), \cdots \right]}_{=:L}\ d_{AS} && (3.29)
\end{aligned}$$

Using this reformulation, the overall quadratic minimization can be written in the following standard form:

$$\begin{aligned} \underset{d_{AS}}{\arg\min}\ & \frac{1}{2}\, d_{AS}^T\, L^T\, L\, d_{AS}\ - d_{AS}^T\, L^T \mathrm{vec}(W\, D_{CS}) \\ \text{s.t.}\ & 0 \leq d_{AS_{min}} \leq d_{AS} \leq d_{AS_{max}} \end{aligned} \tag{3.30}$$

The minimization is extended by two inequality constraints on the result damping vector d_{AS}. These constraints restrict the generated damping commands within actuator-individual, non-negative stable boundaries. A good value for the lower bound might be indicated by an actuator dynamics investigation as done in Section 3.2.6 or a more detailed and, hence, more realistic one. By doing so, Figure 3.14 suggests a minimum actuator damping command of $d_{min} > 10\,\mathrm{kg/s}$. The upper bound, i.e., the maximum allowed damping, depends on the involved hardware and requires experimental determination. Among others, the most significant factors limiting the maximum possible, stable damping commands are sensor noise, signal processing delay, and the capability of the motor driver. With the current hardware setup and the implemented FPGA control logic, the actuators embedded in Carl are configured to accept damping commands of up to $100k\,\mathrm{kg/s}$ for the small RRLab SEAs and up to $200k\,\mathrm{kg/s}$ for the big ones. Construction details about both actuators are available in [Schütz 20a].

Furthermore, the optimization's *Hessian* matrix $\left(L^T L\right)$ may only be positive semidefinite, allowing the problem to feature multiple solutions. In order to enforce solution uniqueness, an additional optimization criterion is added, ensuring real positive definiteness of the problem's *Hessian* matrix. Since the eventual purpose of this optimization is to deliver a continuous stream of suitable motor control values, a good choice to enforce positive definiteness is the result's continuity.

$$\begin{aligned} & \underset{d_{AS}}{\arg\min}\ \frac{1}{2}\, \left(d_{AS} - d_{AS_{prev}}\right)^T\, \epsilon\, \left(d_{AS} - d_{AS_{prev}}\right) \\ \Leftrightarrow\quad & \underset{d_{AS}}{\arg\min}\ \frac{1}{2}\, d_{AS}^T\, \mathrm{diag}(\epsilon)\, d_{AS}\ - d_{AS}^T\, \left(\epsilon\, d_{AS_{prev}}\right) \end{aligned} \tag{3.31}$$

Where $d_{AS_{prev}}$ is the previous optimization result vector and $\epsilon > 0$ is an arbitrary but small user-defined constant. The value of ϵ weights the result's continuity, Equation (3.31), against its correctness, Equation (3.30). Both combined implement a so-called Damped Least Square (DLS) approximation strategy offering guaranteed convexity and a guaranteed unique solution.

Over time, the added stability criteria in Equation (3.31) behave as a very high-frequency (assuming ϵ is very small), exponential low-pass filter on the optimization result vectors. Hence, the filter-induced distortion can be neglected as long as this filter's border frequency is higher than the expected control frequencies. Independent of the frequencies, with a small enough ϵ, the distortion should be below the sense and control signal noise. The combined optimization is formulated as follows:

$$\begin{aligned} \underset{d_{AS}}{\arg\min}\ & \frac{1}{2}\, d_{AS}^T \left(L^T\, L + \mathrm{diag}(\epsilon)\right) d_{AS}\ - d_{AS}^T \left(L^T \mathrm{vec}(W\, D_{CS}) + \epsilon\, d_{AS_{prev}}\right) \\ \text{s.t.}\ & 0 \leq d_{AS_{min}} \leq d_{AS} \leq d_{AS_{max}} \end{aligned} \tag{3.32}$$

3.4.3 Capabilities and Limits of Distributed Damping

As derived in the introduction of this section, wrench distribution is represented by an underdetermined system of linear equations, whereas damping distribution usually leads to an overdetermined system. Hence, the desired CS wrench can be assumed to be perfectly tracked in the AS within the available hardware capabilities. At the same time, CS damping matrices are, most likely, not precisely reproducible in the AS, falling back to only an approximation. The approximation strategy, presented in the previous Section 3.4.2, is only one possible solution to this problem. However, independent of the exact approximation formalization, all possible solutions will depend on some weighting mechanism to decide on the approximation direction, as well as the robot's morphology.

Both the weighting parameter and skeletal morphology significantly influence the ultimately distributed actuator damping commands. For a better understanding of these influences, three sets of kinematic experiments are conducted using the above-introduced damping distribution strategy and analyzed in the following. All experiments have first been published in [*Vonwirth 23b*];

The kinematic experiments utilize the actual morphology of the biped robot Carl, reduced to the thigh and shank segments. Both segments have an identical natural length of 42 cm. The hip and knee joints are limited by Carl's hardware range of motion from 0° to −120° and −90°, respectively. Further details of the construction of Carl are available in [Schütz 17][Schütz 20a].

As pictured in Figure 3.6a, the relevant actuators are accordingly limited to the two monoarticular actuators A_H and A_K, and the biarticular one A_{HK}. The WS is defined as the orientation-less, cartesian, planar position of the robot's ankle joint, equal to Figure 3.18a. Using this setup, the individual spaces' dimensionalities, according to Equation (3.6), are:

$$\dim_{Ankle}(WS) \;=\; 2 \;=\; \dim_{Ankle}(CS) \;<\; \dim_{Ankle}(AS) \;=\; 3 \tag{3.33}$$

However, when looking at the number of available DoFs w.r.t. damping, the matrix nature in the WS and the CS increases their dimensionality according to the formula in Equation (3.19).

$$\#\mathrm{DoF}_D(WS) \;=\; 3 \;=\; \#\mathrm{DoF}_D(CS) \;=\; \#\mathrm{DoF}_D(AS) \tag{3.34}$$

Note that the damping in the AS is a vector property rather than a matrix property within the two other ones. Due to the equal number of DoF across all spaces, all inter-space damping transformations are quadratic and theoretically invertible. Hence, the only limitation to perfectly tracking the desired WS damping matrix is the demanded actuator-individual damping non-negativity within the AS.

In all experiments, the ankle of the two-link leg of Carl is kinematically moved in an elliptical motion. Throughout this motion, the damping distribution strategy, as given in Section 3.4.2, is applied, and the actuator-individual resulting damping commands are recorded. The desired WS damping ellipses are the matrix products of four individual, one-dimensional unit-damping vectors d_{WS} with themselves $D_{WS} = d_{WS}\, d_{WS}^T$. The four damping directions are 0°, −65°, −90°, and −115°.

Figure 3.19 illustrates the experimental setup, along with a small set of explicit approximation results extracted from those experiments. The two-link leg setup and the kinematic definitions are sketched in Figure 3.19a. The four desired WS unit damping direction vectors d_{WS} are illustrated at the ankle.

At the same time, Figure 3.19b illustrates the experiment's elliptical ankle motion and the desired and resulting $-115°$ d_{WS} damping ellipses in the CS. The dashed damping ellipses are drawn at three concrete, exemplary positions for each three different distribution approximation parameterizations which are described later within this subsection.

Equation (3.12) is used to convert the WS damping matrix D_{WS} into the CS damping matrix D_{CS}, as required by the damping distribution algorithm (3.32).

$$\begin{aligned} D_{CS} &= J_{CW}^T \, D_{WS} \, J_{CW} \\ &= J_{CW}^T \, d_{WS} \, d_{WS}^T J_{CW} \\ &= d_{CS} \, d_{CS}^T \\ \text{using: } d_{CS} &:= J_{CW}^T \, d_{WS} \end{aligned} \tag{3.35}$$

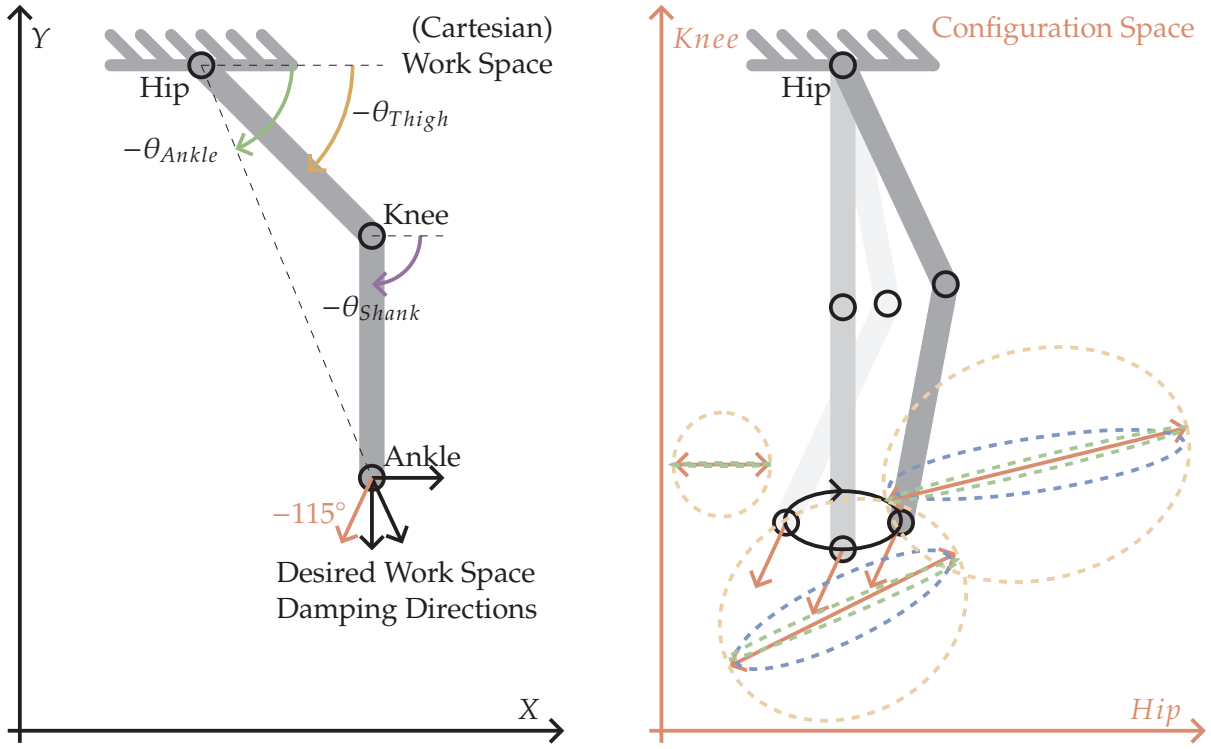
The identical elliptical ankle motion is executed in three individual experimental runs. Starting at the topmost position, the ankle is moved clockwise, as sketched in Figure 3.19b. After half of the ellipse, the robot passes its singular position in a backward direction of motion. The motion ellipse is kept small with a horizontal diagonal of 200 mm and a vertical diagonal of 100 mm, keeping the influence of the changing gear values low.

In all three experiments, the four identical, above-mentioned, one-dimensional desired WS damping matrices $D_{WS} = d_{WS} \, d_{WS}^T$ are approximated. All four damping directions are chosen to highlight particular properties of the applied damping distribution strategy. The horizontal and vertical damping directions, $0°$ and $-90°$, represent the directions of upright balancing and gravity resistance, respectively. Furthermore, the horizontal damping especially highlights the benefits of adequately chosen biarticular actuation, i.e., proper, control-related hardware morphology design.

At the same time, the pure vertical damping highlights the algorithm's resistance against singular configurations, often causing numerical instability. The damping direction angle of $-115°$ highlights the effect of being aligned with the shank. Both the problem of configuration singularity and the actuators' principle directions of action have been discussed in detail in Section 3.3.2 and are sketched in Figure 3.18a.

The damping direction of $-65\,°$ compared to $-115°$ presents the differences between equally forward and backward pointing desired WS damping directions and, hence, once again, the significant influence of the robot's morphology, i.e., its hardware design, on the applied control strategy. In the case of Carl, the forward-bending knee plays a significant role as it defines the segments' orientations in all WS poses.

The direction of action of the monoarticular hip actuator A_H in a two-link leg is always aligned with the leg's shank segment. This property has been introduced in Section 3.3.2 and sketched in WS coordinates in Figure 3.18a. Figure 3.19b now visualizes this phenomenon in CS coordinates at the backward ankle position.



(a) Kinematic definitions of the reduced, two-link leg of CARL in Cartesian coordinates with the desired WS damping directions.

(b) Desired and generated damping ellipses in the CS at three exemplary positions throughout the elliptical ankle trajectory.

Figure 3.19: Definition and three explicit evaluations of the damping distribution experiments, as initially published in [*Vonwirth 23b*]. To the left, Figure (a) sketches the reduced two-link leg of CARL, the kinematic definitions, and the four unit-damping direction vectors d_{WS}. At the system's ankle, the four desired damping directions that are approximated throughout the kinematic experiments are shown. The elliptical motion itself is visualized on the right in Figure (b), together with three exemplary sets of damping ellipses generated by different approximation parameterizations for the desired WS damping direction of −115°. While the desired, one-directional damping ellipses are pictured as solid, red, double-sided arrows, the dashed ellipses around these arrows visualize the generated ellipses. The blue and orange ellipses denote the optimization results using unit-weight $W = I$ and damping-equivalent weight $W = D_{CS}$, respectively. In green, a result ellipse is shown that emerges from a robot morphology featuring a constant biarticular A_{HK} gear ratio (Hip-to-Knee) of $^2/_1$ at unit weight. All actuator-individual damping results over the whole elliptical motion for all three mentioned optimization parameterizations are plotted in Figures 3.20, 3.21, and 3.22.

In this backward pose, the shank features the identical −115° orientation in the WS as the desired damping vector. Consequently, the identical desired damping in CS coordinates becomes a pure horizontal line, as pictured by the solid, red, double-sided arrow. The hip actuator is the only required one to generate the desired Cartesian WS damping direction exactly. In both other two ankle positions, a clear vertical, i.e., knee, component is present and visible.

Furthermore, it is important to note that the singular leg configuration in the middle of the motion, the bottom position, does not cause any numerical artifacts or instabilities.

This algorithmic stability can also later be observed in detail in the damping distribution result plots in Figures 3.20, 3.21, and 3.22. The approximation itself is individually parameterized for all three conducted kinematic experiments as follows:

1. Unit Weight:
   The first experiment run uses the damping distribution approximation with unit matrices as approximation weights $W = I$. This way, both desired damping and desired non-existing damping are considered equally important by the optimization algorithm. The detailed results of this parameterization for all four WS damping directions are plotted in Figure 3.20.

2. Damping-Equal Weight:
   The second experiment run uses the desired damping itself as the according weight matrix within the approximation formula $W = D_{CS}$. With respect to control stability, over-damping a system is always safer to do than under-damping the same. Weighting the damping with itself leads to the situation that higher damping values are preferred over lower ones and directions with zero damping being entirely ignored by the optimization. The detailed results of this parameterization for all four WS damping directions are plotted in Figure 3.21.

3. Biarticular Gear Ratio:
   Next to the free choice of directional weighting matrices, damping distribution also highly depends on the robot's morphology, i.e., the Jacobian actuator gear matrix J_{CA}. While the first two experiments use the real JCA-matrix of the robot Carl, the third experiment is executed with a constant gear ratio of $^2/_1$ for the biarticular actuator A_{HK}. Equal to experiment 1, constant unit matrices are utilized as weights. The detailed results of this parameterization for all four WS damping directions are plotted in Figure 3.22.

For a better comparison between the individual parameterizations, Figure 3.19b visualizes the results of all three parameterizations on top of each other at three concrete ankle positions throughout its elliptical motion in CS coordinates, i.e., Hip and Knee angles. The positions are about one, two, and three-quarters of the overall motion, i.e., at the front, bottom, and back positions of the elliptical ankle trajectory. Their exact locations within the elliptical motion are marked in the detailed result plots in Figures 3.20, 3.21, and 3.22 for all three individual experiment parameterizations, respectively.

For all three positions, the desired $-115°$ WS damping is drawn as solid, red, double-sided arrows. Dashed ellipses surround those arrows, visualizing the actuator-based damping ellipses generated by the approximation strategy, as given in Equation (3.32). The blue and orange ellipses represent optimization results of Experiment 1 (Unit Weight) and 2 (Damping-Equal Weight), respectively. Colored in green, the result ellipse of experiment 3 is sketched.

The herein-proposed damping distribution strategy, in general, is not capable of exactly matching arbitrary desired WS damping matrices. However, this limitation is not due to the approximation strategy but entirely depends on the robot's morphology and its available actuators. In fact, the proposed optimization perfectly fits the desire whenever possible, independent of its concrete parameterization.

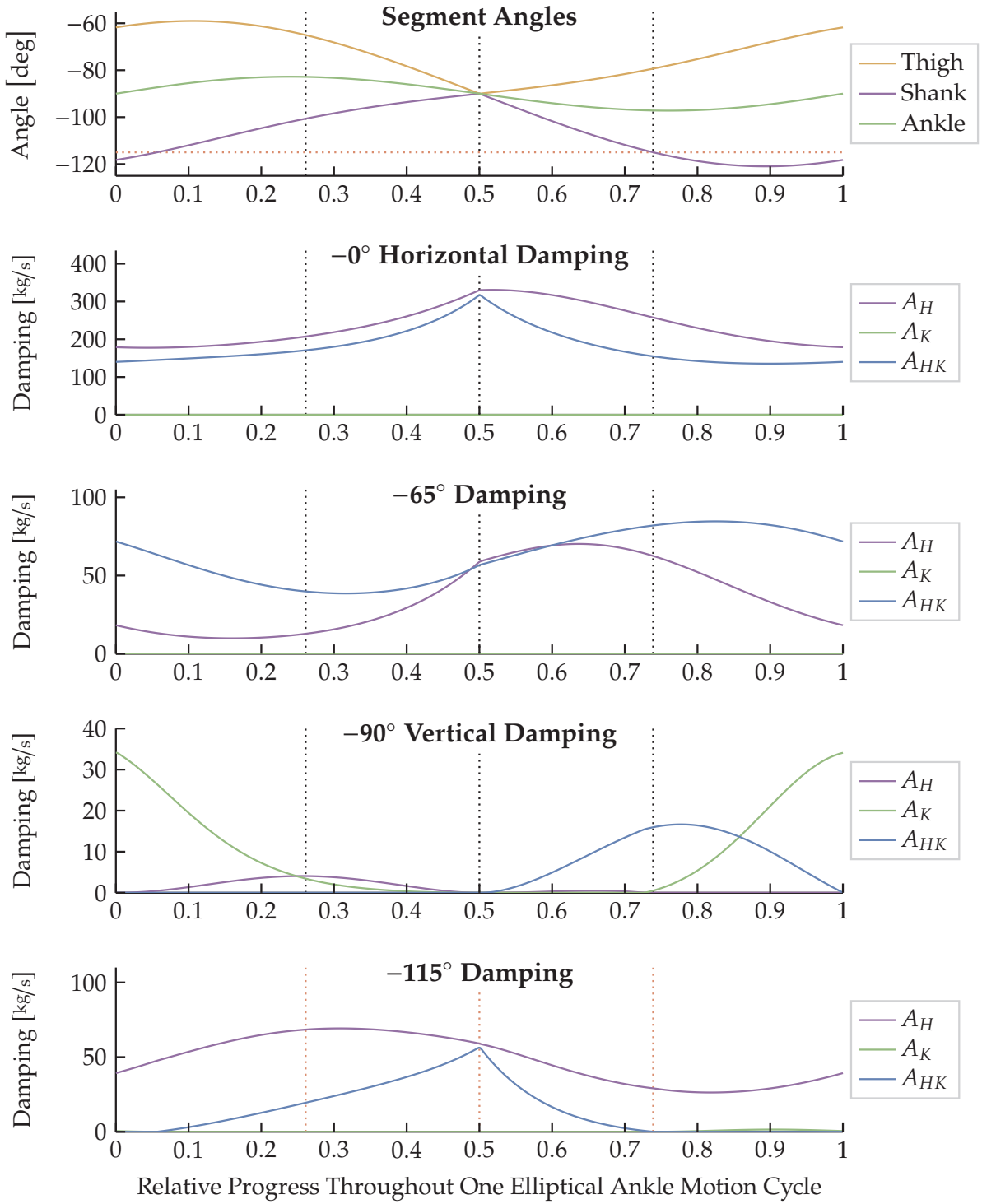


Figure 3.20: Damping distributions of the kinematic experiment 1 for all four WS damping directions, defined in Figure 3.19a. As described in the text, the actuator-individual damping values are calculated using Equation (3.32) with unit weight matrices $W = I$. On top, the angles of the two leg segments and the ankle are plotted as defined in Figure 3.19a. The interesting angle of −115° is drawn as a dotted red line in the angles plot. Furthermore, the three explicit positions, visualized in Figure 3.19b, are marked with vertical, dotted lines in all five plots. However, only the red ones of the bottom-most −115° damping plot are visualized. From left to right, these are the frontal, singular, and backward positions.

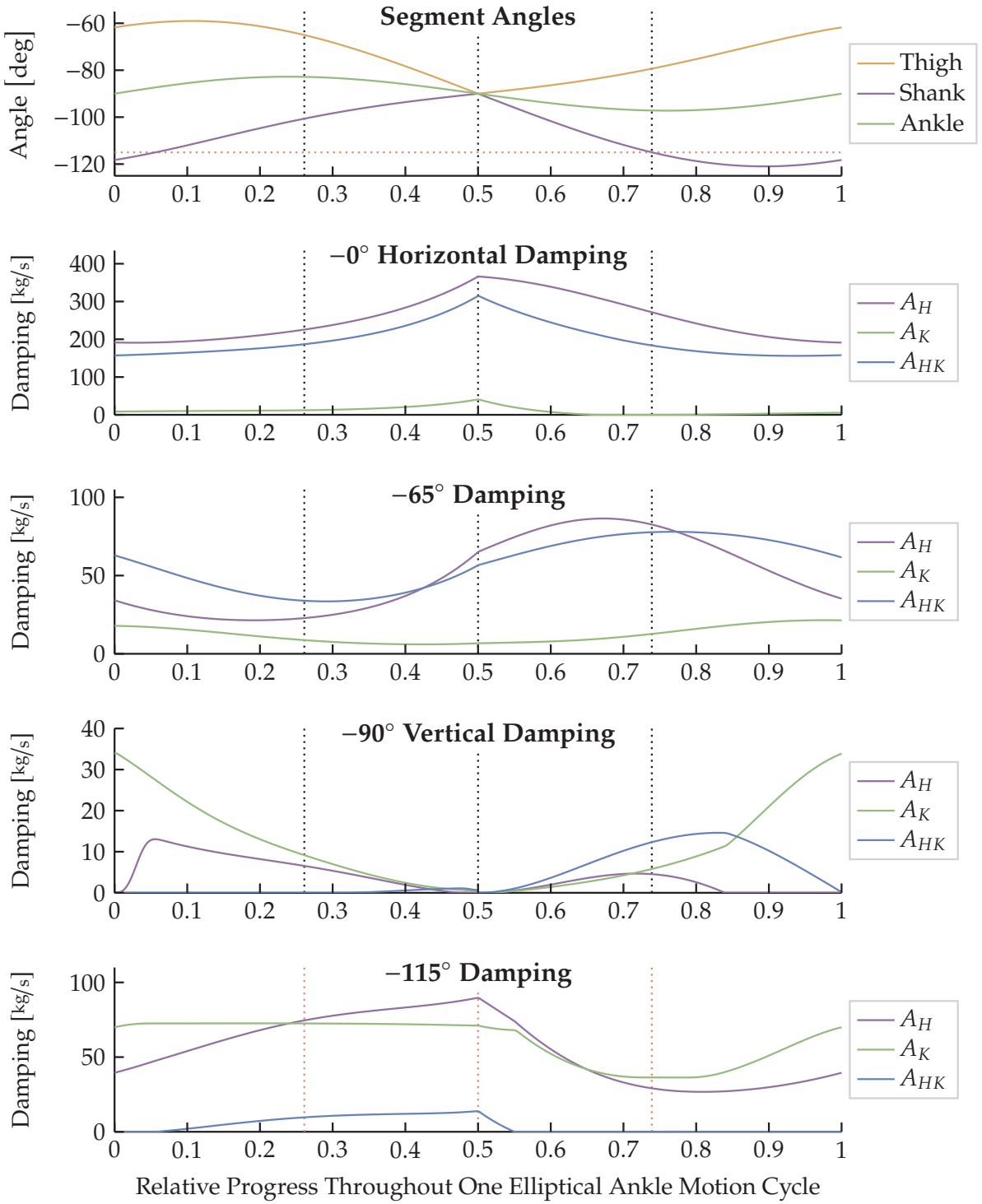


Figure 3.21: Damping distributions of the kinematic experiment 2 for all four WS damping directions, defined in Figure 3.19a. As described in the text, the actuator-individual damping values are calculated using Equation (3.32) with weight matrices equal to the desired damping itself $W = D_{CS}$. On top, the angles of the two leg segments and the ankle are plotted as defined in Figure 3.19a. The interesting angle of −115° is drawn as a dotted red line in the angles plot. Furthermore, the three explicit positions, visualized in Figure 3.19b, are marked with vertical, dotted lines in all five plots. However, only the red ones of the bottom-most −115° damping plot are visualized. From left to right, these are the frontal, singular, and backward positions.

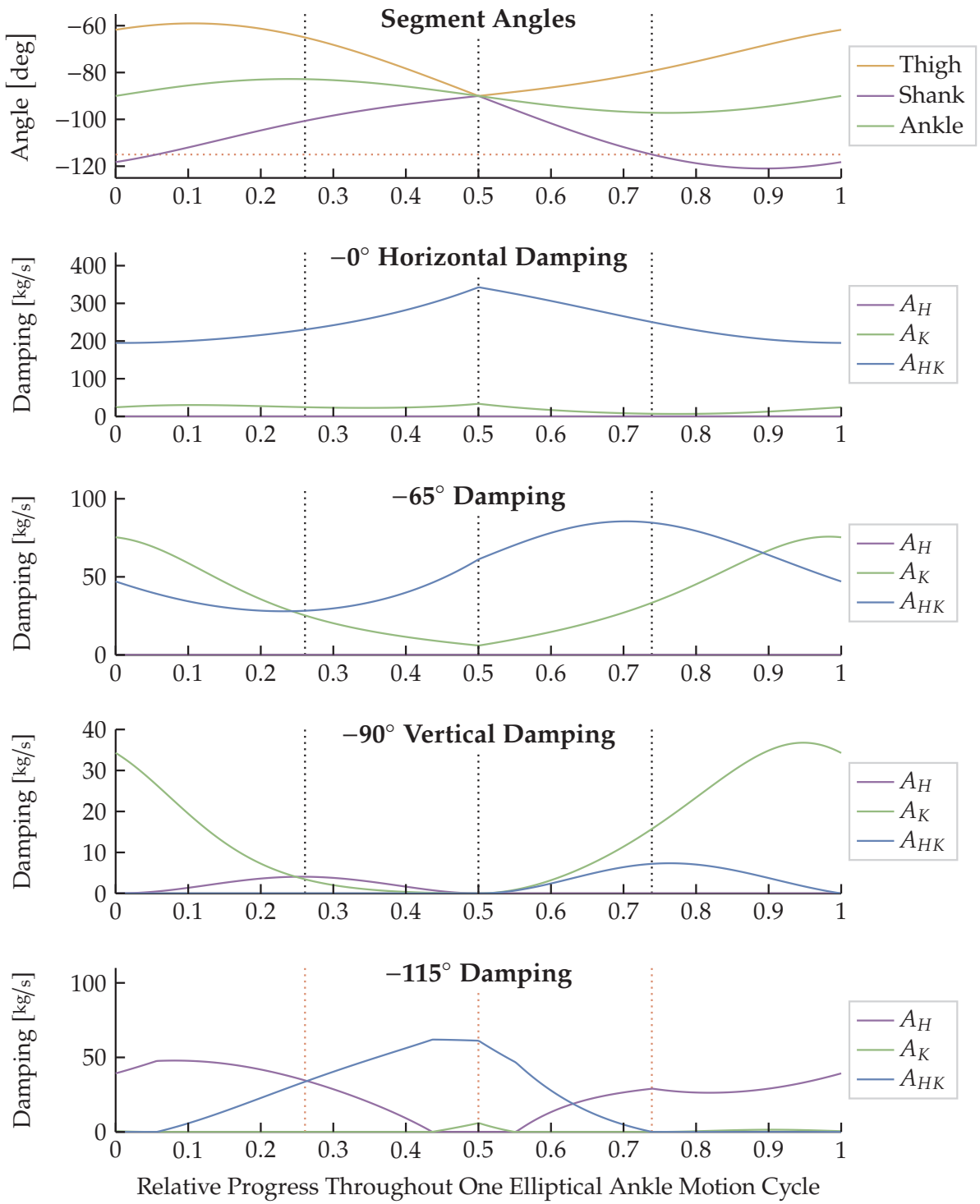


Figure 3.22: Damping distributions of the kinematic experiment 3 for all four WS damping directions, defined in Figure 3.19a. As described in the text, the actuator-individual damping values are calculated using Equation (3.32) with unit weight and a constant gear ratio of $^2/_1$ for the biarticular A_{HK} actuator. On top, the angles of the two leg segments and the ankle are plotted as defined in Figure 3.19a. The interesting angle of −115° is drawn as a dotted red line in the angles plot. Furthermore, the three explicit positions, shown in Figure 3.19b, are marked with vertical, dotted lines in all five plots. However, only the red ones from −115° damping are visualized. From left to right, these are the frontal, singular, and backward positions.

Throughout the elliptical motion conducted herein, this is the case in the backward ankle position, which perfectly aligns the shank with the desired −115° WS damping direction. Figure 3.19b visualizes this perfect match over all three parameterizations in CS coordinates. Note that the blue approximation ellipse at the backward position is identical to the green one, entirely covering its visualization. The orange approximation ellipse features the special case of the knee coordinate being zero-weighted. Hence, as introduced in Equation (3.31), the approximation strategy smoothens unspecified result directions, falling back to return the previous directional result value.

By definition, the directions of action of monoarticular actuators in the CS are its independent axes. Hence, the gear ratios of additional biarticular or even multiarticular actuators are the only possibility to tune the robot's capability of generating rotated, i.e., arbitrary, CS damping matrices. Unfortunately, this is a *mechanical* robot design decision. Figure 3.19b clearly shows the green approximation ellipses, generated with the improved biarticular gear ratio of $^2/_1$ for the biarticular A_{HK} actuator, much closer to the desired ones than both others, using the real morphology of Carl. In-depth analyses and discussions on biarticular gear ratios at the hand of the leg-design of Carl can be found in [*Nejadfard 18b*, *Nejadfard 19*, *Nejadfard 20*].

The choice of the weighting matrices highly influences the actuator damping approximation results, which can easily be seen by comparing the orange ellipses against the blue and green ones. Only the orange result ellipses perfectly match the tips of the desired red arrows in all three positions, drawn in Figure 3.19b. Since the orange results are weighted with the desired damping itself, the desired ellipse's thickness of zero is not taken into account at all. Hence, the approximation only focuses on matching the desired red arrow tips without further interests, preferences, or limitations.

The orange ellipses' widths emerge from the unchanged previous result projected onto the zero-weighted dimension. In both other cases, unit-weight matrices have been utilized, leading the optimization to balance both the length and the width of the resulting ellipses against the desired, flat one. Since a perfect match is mechanically impossible, the blue and green actuator damping ellipses are shorter and thicker than the desired one.

As described in [Schütz 17], the gear ratio of the biarticular A_{HK} actuator in the robot Carl stays significantly smaller than 1 throughout the experimental, elliptical ankle motion. Using the detailed analysis of such a gear ratio by Nejadfard et al., the A_{HK}'s resulting direction of action is permanently very close to the one of the monoarticular A_K [*Nejadfard 18b*]. Hence, the benefit of having both the monoarticular A_K and the biarticular A_{HK} is limited w.r.t. distributed, actuator-level damping generation.

The result plots of Figures 3.20 and 3.21 show the monoarticular knee actuator A_K being unused most of the time. Except for vertical WS damping, the biarticular A_{HK} entirely dominates the knee actuator. The second exception is the −115° desired damping weighted. In this case, the A_K dominates the A_{HK}. However, having both actuators active together is a rare optimization result except for the plots in Figure 3.22. All plots in the figure visualize the strong effect of the robot's morphology on the damping distribution strategy.

The effect of the biarticular A_{HK} gear ratio can be observed by comparing Experiments 1 and 3 in Figures 3.20 and 3.22. Both experiments are executed with constant unit-weight matrices.

The most significant effect can be observed at horizontal damping. It is known that an A_{HK} gear ratio of $^2/_1$ leads to a direction of action orthogonal to the ankle angle, as defined [*Nejadfard 18b*]. Since the ankle angle is permanently close to $-90°$, as shown in the angles plot, the A_{HK} nearly perfectly matches the horizontal damping throughout the entire elliptical ankle motion.

Comparing the $-65°$ WS damping within the same two experiments uncovers a significant change in muscle recruitment. Identical in both setups are the directions of action of the hip and knee actuators. The hip actuator operates aligned with the shank segment, and the knee actuator's direction of action is equal to the ankle angle, as defined. However, the biarticular direction of action differs a lot.

As introduced above, the biarticular A_{HK} operates similarly to its monoarticular A_K in Experiment 1. Hence, the approximation is practically limited to only two distinct directions of action to choose from: A_H and A_K/A_{HK}. At the cost of notable deviations from the desired, there is no other option available than recruiting the unsuitable hip actuator.

In contrast, the A_K and A_{HK} directions of action are orthogonal in Experiment 3. Consequently, both together build a capable (orthogonal) basis for the approximation strategy to operate with, and the less-suitable hip actuator is not needed anymore. Again, these two cases highlight the tight coupling between a robot's morphology and its control.

No relevant change can be observed in the vertical and slightly backward-oriented damping cases. The monoarticular knee actuator remains the best-suited choice for both of those directions.

Two important observations and conclusions can be drawn from the above data and analyses, summarizing this section.

1. A robot's morphology and its control capabilities are tightly coupled. In the special case of hybrid impedance control (see Figure 3.8 in Section 3.2), distributed damping generation is entirely defined by the available actuators and their individual mounting positions within the robot's frame.

2. The higher the variability in the set of a robot's available actuators, the higher the possibility to precisely approximate desired CS damping matrices with distributed, actuator-level damping generation. This variability also includes the simple number of available distinct actuators.

Consequently, when designing a new robot, it is good advice to design both hardware and control together as one single musculoskeletal actuation system rather than independent from each other. Both these conclusions match the state-of-the-art review publication *"Biarticular muscles in light of template models, experiments and robotics: a review"* by Schumacher et al. from 2020:

> *"Thus, the muscle's mechanical function is strongly influenced by the leg architecture. This coupling also affects the neural coordination of the muscles."*
> [Schumacher 20, p. 3]

The herein-proposed novel control philosophy of hybrid impedance with distributed damping offers a novel view of the existence and reasoning of the muscular variety observed in natural bodies. Aside from the known potential to reduce energy consumption [*Nejadfard 18a*], multiarticular, additional actuation offers the possibility to implement relatively precise, distributed, high-gain, and low-delay damping generation.

3.5 Experimental Results of Antagonistic Muscular Control on Carl

As introduced in Section 3.2.3 and analyzed in Section 3.2.6, a robot's desired motion precision, i.e., its TCP twist precision, is specified by the artificial muscles' contractile damping commands. In fact, the damping coefficient implements the gain of a proportional-only velocity controller. Nevertheless, due to the natural, muscle-like over-actuation available in the robot Carl (see Figure 3.6a), a force-based low-level control is to be preferred over a classical, velocity-based P-control scheme. CS wrenches can efficiently be distributed among the available actuators using an optimization strategy introduced in Section 3.4.1. However, the standard velocity-base proportional control scheme using D and v_0 can easily be converted into equivalent wrench and damping commands F and D.

$$F_{Motor} = D(v_0 - v) = Dv_0 - Dv =: F - Dv \tag{3.36}$$

Note that this is the identical conversion already done in the system analysis, plotted in Figure 3.16. Clearly, this velocity-P-controller can neither compensate for steady-state errors, e.g., caused by gravity, nor eliminate significant disturbances, e.g., caused by dynamics. To achieve state-of-the-art robot control that considers all those issues, a capable higher-level control strategy, such as Model Predictive Control (MPC), is required on top of the presented musculoskeletal low-level control.

Another possibility is to reduce the disturbing dynamic effects by simply slowing down the robot's motion. For slower desired motions, the required control frequencies can be reduced and upper-limited below the stability-critical frequency caused by the passive elastic element, as highlighted in the Bode plot in Figure 3.16. Reducing execution speed in favor of precision is very natural and thus preferred behavior. Observations of such natural actions, including human behavior, are, e.g., studied in the following works: [Milner 02, Gribble 03, Cashaback 15].

However, in order to study the real-world behavior of only the hybrid muscular impedance control, no such sophisticated control must be used not to disturb the pure low-level control reactions. Hence, a very primitive, commonly known compliant control approach can be used to verify the herein-proposed natural control strategy, first published in [*Vonwirth 23b*]. For these experiments, a single, planar leg of Carl is reduced to a two-link system featuring only the thigh and shank, as known from the

previous Section 3.4.3. A virtual spring-damper component is attached to the reduced leg's TCP, its ankle. The equilibrium of this virtual impedance is then moved on a fixed, desired, feed-forward trajectory. Dependent on the virtual impedance values, the ankle, and hence the whole leg, is compliantly following this virtual equilibrium trajectory. Figure 3.23a shows a sketch of this experimental setup, including an exemplary compliant trajectory and the virtual spring-damper element.

Since the generation of damping is just an approximation, a correct wrench control vector F may only be derived once the actually applied actuator damping commands are known. This dependency introduces a (strict) partial order of operations enforcing all wrench computing functions being processed after the actuator damping distribution has been resolved. Hence, wrench control is replaced by twist control throughout any hybrid, muscular impedance robot control system. Only in its final stage, in CS coordinates, the desired wrench vector $\dot{\theta}_{CS}$ is converted into the required wrench vector τ_{CS} and instantly distributed among the actuators using Equation (3.15).

$$J_{CA}^T \, D_{AS} \, J_{CA} \, \dot{\theta}_{CS} \;=\; \tau_{CS} \;\stackrel{!}{=}\; J_{CA}^T \, F_{AS} \tag{3.37}$$

Due to the usage of twist rather than wrench, the stiffness K_Z of the virtual, WS impedance Z_{WS} is split up into a leading twist part K_{WS}, followed by the regular WS damping D_{WS}.

$$K_Z \;:=\; D_{WS} \, K_{WS} \tag{3.38}$$

The control twist v_{WS} is calculated by the Cartesian WS position delta Δ_{WS} and the so-called twist stiffness K_{WS}.

$$v_{WS} \;=\; K_{WS} \, \Delta_{WS} \tag{3.39}$$

The system is stabilized by computing K_{WS} in such a way that at least critical damping is achieved within the virtual impedance. For ease of computation, the whole two-link leg is assumed to behave like a single load with mass M. The twist stiffness K_{WS} is derived as follows:

$$2 \sqrt{K_Z \, M} \;=\; D_{WS} \tag{3.40}$$

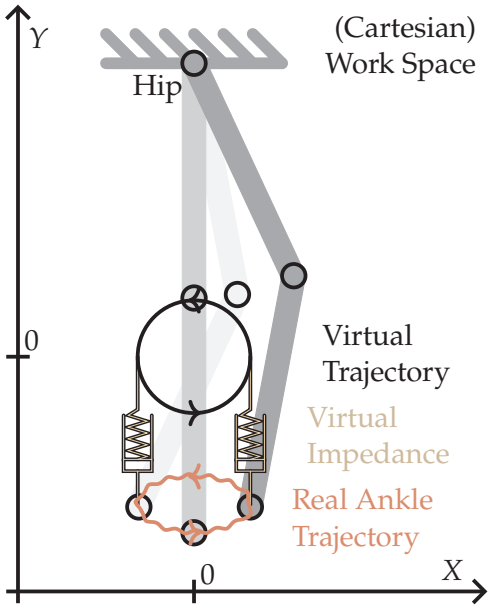
$$\Leftrightarrow \quad \sqrt{D_{WS} \, K_{WS} \, M} \;=\; \frac{1}{2} D_{WS}$$

$$\Rightarrow \quad D_{WS} \, K_{WS} \, M \;=\; \frac{1}{4} D_{WS}^2$$

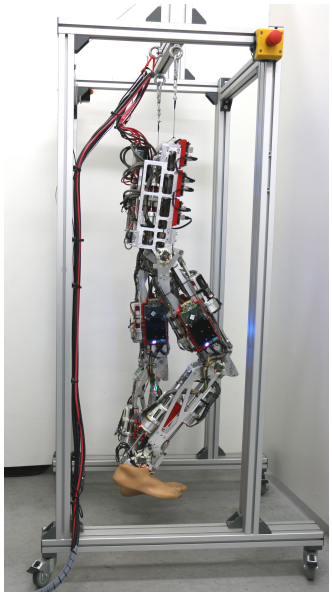
$$\Leftrightarrow \quad K_{WS} \;=\; \frac{1}{4M} D_{WS} \tag{3.41}$$

As given by Equation (3.41), the Cartesian WS stiffness K_{WS} depends on the Cartesian WS damping D_{WS}, which, as argued above, is unknown at this stage. Hence, the estimated mass M must be overestimated to keep the stiffness small enough to compensate for the imprecise damping approximation discussed in Section 3.4.3.

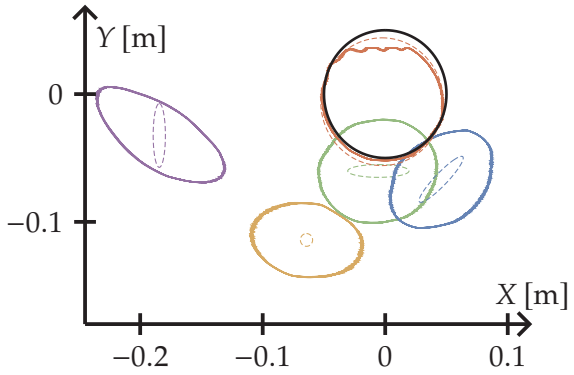
Two sets of five experiments are executed on a single leg of CARL. Throughout all experiments, the identical, circular virtual impedance trajectory is commanded in a feed-forward manner, as represented in all sub-parts of Figure 3.23. The circular trajectory has a radius of 5 cm and is centered 63 cm straight below CARL's hip joint. In the first set of experiments, this circular trajectory is followed at a constant speed



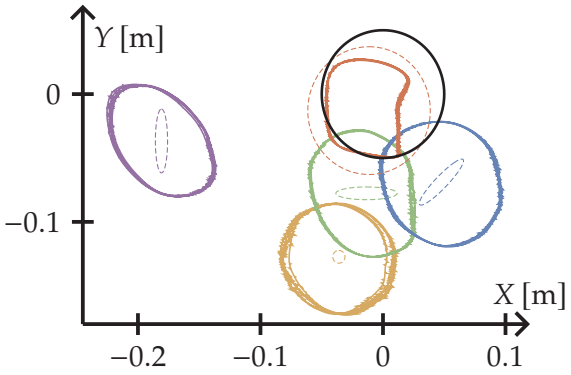
(a) Hybrid Impedance Experiment Setup



(b) CARL in a Compliant Pose



(c) Motion Recordings at 30 RPM



(d) Motion Recordings at 120 RPM

Figure 3.23: Experimental sketch and results of the hybrid, muscular impedance control applied to the ankle of the planar, single robotic leg CARL, as initially presented in [*Vonwirth 23b*]. On the top left, the kinematic sketch (a) illustrates the leg setup, the circular feed-forward trajectory, the virtual impedance, and an allusion to the expected emerging ankle trajectory. The robot CARL rendering a compliant leg pose is pictured in (b). Due to the lack of gravitational compensation, the virtual impedance trajectory is pictured as a perfect, black circle above the expected, uncertain ankle trajectory, indicated in red. Between both, the virtual spring-damper element is represented in orange. Two series of experiments are executed with each of five different WS impedance configurations. Plots (c) and (d) show the two experiments' recorded ankle trajectories at feed-forward trajectory velocities of 30 RPM and 120 RPM, respectively. Equal to the kinematic sketch, the feed-forward trajectory is given as a perfect black circle centered at the origin. The captured ankle motions are colored w.r.t. to their given commanded WS damping matrices, equal for both execution speeds. A sketched, dashed ellipse of the same color as the captured motion indicated the respective damping matrices. The five individual WS damping configurations, given in the standard SI-unit [kg/s], are:
Orange $\left[\begin{smallmatrix}50 & 0\\ 0 & 50\end{smallmatrix}\right]$, Purple $\left[\begin{smallmatrix}50 & 0\\ 0 & 250\end{smallmatrix}\right]$, Green $\left[\begin{smallmatrix}250 & 0\\ 0 & 50\end{smallmatrix}\right]$, Blue $\left[\begin{smallmatrix}150 & 100\\ 100 & 150\end{smallmatrix}\right]$, and Red $\left[\begin{smallmatrix}500 & 0\\ 0 & 500\end{smallmatrix}\right]$.

of 30 RPM, i.e., tracking 30 complete trajectory circles per minute. The second set of experiments is executed at 120 RPM, which equals an upper estimate of the average human step frequency ($\leq 120\,{}^{\text{steps}}/_{\text{minute}}$) during normal walking [Perry 92, p. 454].

Both experiments sets contain five execution runs at different desired WS impedance values, each defined by a desired damping matrix. An appropriate stiffness is calculated according to Equation (3.41). The shared estimated mass used to define those matrices is chosen to be 15 kg, i.e., about half the weight of CARL. For all experiments, the resulting ankle positions are recorded at a frequency of 1 kHz in WS coordinates without any post-processing. The results of the two experiments are plotted in Figures 3.23c and 3.23d each for the following five individually color-coded WS damping matrices:

$$\text{Orange} \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}, \quad \text{Purple} \begin{bmatrix} 50 & 0 \\ 0 & 250 \end{bmatrix}, \quad \text{Green} \begin{bmatrix} 250 & 0 \\ 0 & 50 \end{bmatrix}, \quad \text{Blue} \begin{bmatrix} 150 & 100 \\ 100 & 150 \end{bmatrix}, \quad \text{Red} \begin{bmatrix} 500 & 0 \\ 0 & 500 \end{bmatrix}$$

All damping matrices are given in the standard damping SI-unit ${}^{\text{kg}}/_{\text{s}}$. Small, dashed ellipses in the result plots illustrate the active, desired WS damping configuration for every recording. The damping matrices that are eventually generated depend on the momentary leg configuration. Hence, they vary throughout each motion cycle and cannot be visualized.

Both result sets demonstrate stable executed ankle trajectories. By comparing their recordings, the effects of the individual impedances and velocities can be reasoned.

As expected, all recorded trajectories are disturbed downward below the black reference circle due to uncompensated gravity. However, it can also be recognized that higher impedance values reduce this environmentally given error. The softest configuration is given by the orange trajectories. As expected, they are the lowest trajectories with the higher vertical deviation from the reference circle in both sets of experiments. Hence, these two trajectories are taken as reference w.r.t. the deviation against the commanded, black reference circle throughout the following. All other recordings clearly highlight the individual error reduction related to increased impedances in vertical, horizontal, diagonal, or all directions, independent of the motion velocity. Conclusively, on the robotic hardware of CARL, the herein-introduced hybrid muscular impedance control strategy, in general, behaves as expected and desired.

Among all the recordings, the left-most purple trajectories are of special interest. In both plots, the vertical deviation is clearly reduced. However, at the same time, the horizontal deviation is significantly increased although no impedance reduction has been made, w.r.t. the orange reference ones.

A detailed look at the physical leg configuration in those situations delivers an explanation of this unintended effect. With the ankle being pulled upward due to the increased vertical stiffness, the leg's knee is bending forward. This motion, however, also shifts the leg's Center of Mass (CoM) forward, which, in turn, leads to a passive backward swing torque of the whole leg around its hip joint. Due to the absence of both gravity compensation and horizontal stiffness, this torque is only minor counteracted and the whole leg's execution trajectory is deviated to the back.

A behavior nearly opposite to the purple recordings can be observed in the green ones. The green trajectories emerge from an increased horizontal stiffness. As expected, a reduced horizontal error can be observed in both result plots in Figure 3.23. However, unlike the purple trajectories, the green ones also show a significant error reduction in

the vertical direction, which has not been commanded. This time, an explanation of this effect can be found by a closer look into the internal damping distribution strategy, discussed in the previous Section 3.4.3.

By the morphological construction of Carl, the distributed horizontal damping always comes at the cost of unavoidable, additional vertical damping, too. As shown in the two virtual experiment results, given in Figures 3.20 and 3.21, horizontal damping is generated by the monoarticular hip actuator A_H and the biarticular actuator A_{HK}, independent of the utilized distribution weighting strategy. Unfortunately, both these actuators in Carl feature a more vertical rather than horizontal direction of action. Hence, the absence of an actuator suitable to act in the horizontal direction forces to damping distribution strategy to recruit undesired vertical damping in order to achieve the desired horizontal one. Note that this effect is only visible because of the absence of gravity-compensation mechanisms.

The highest impedance command, uniform in all directions, is illustrated in red. As expected, in both plots, those trajectories are closest to the commanded black reference circle. The top of the reference trajectory, however, slightly leaves the robot's reachable workspace due to Carl's mechanical knee joint limit at $-90°$. As can be seen by the flat upper part of the red trajectory at 30 RPM in Figure 3.23c, the red impedance has been chosen high enough to drive the robot partially into its mechanical limits. Notably, the recorded ankle motion demonstrates stability even at physical boundaries.

Aside from gravity, dynamic effects significantly influence the actual trajectories. Such effects can be observed by comparing pairs of equally colored (equal impedance) ankle plots across the two different execution velocities. First, it can be seen that, although existing, the dynamic's effects on the locations of the executed trajectories are limited compared to the gravitational disturbance. This effect is best visible in the orange, most compliant trajectory. The plot at 120 RPM shows significantly higher deviations between the repetitive circular motions compared to its orange counterpart at 30 RPM. Also noticeable is the fact that most deviation occurs in the horizontal direction. As already argued above, this observation can be explained by the absence of an explicit, horizontally operating actuator in Carl's morphology. More significant effects can be observed in the general shape and the variety of the executed motions.

Finally, the considerable deformation of the desired circular motion on both sides of the red trajectory at 120 RPM must be mentioned. Although being the most-precise trajectory at both speeds, it also features the most significant discrepancy between the two executions. An interpretation of this outstanding behavior is difficult but may be found in the robot's leg segment configuration in combination with the demanded high motion precision, i.e., impedance.

As stated above, the red trajectory operates close and even within the robot's mechanical boundary at its knee joint. Hence, a significant amount of mass is extended far to the front and outward of the leg's ankle axis, as defined in Figure 3.19a. This extended lever, together with the enforced high motion precision, hence velocity, may accelerate the robot-intrinsic disturbances due to dynamic effects. Although much less in effect, similar behavior can also already be observed in the green and very little in the blue trajectories at 120 RPM in Figure 3.23d.

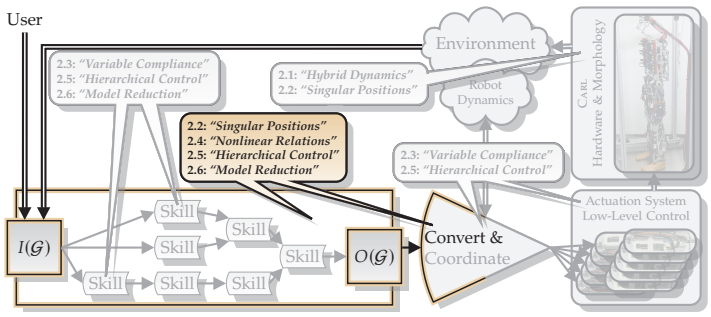
3.6 Conclusion

This chapter introduced a novel, bio-inspired hybrid impedance control philosophy for robots that are entirely actuated by compliant SEAs. The proposed antagonistic muscular control strategy satisfies all relevant natural control requirements derived from the identified natural gait characteristics throughout Section 2.1. Thus, the novel control concept makes a significant contribution to achieving the overall aim of this thesis, as described in Theorem 1.1: *"Natural Control"*. At the same time, the general control complexity of distributed impedance control has been reduced to the neuromuscular equivalent of only two required control commands per actuator, i.e., the equivalent of antagonistic muscle stimulation.

The proposed control approach was theoretically analyzed and experimentally evaluated to be functional on real-world hardware using the planar bipedal robot Carl. All sections highlighted the tight coupling between the chosen control strategy, its limits and capabilities, the emerging real-world robot behavior, and the robot's underlying morphology.

To achieve complete and consistent natural robot control, the following chapters develop and demonstrate a high-level control framework suitable for serving the demands of the novel low-level antagonistic muscular control strategy.

4. Probabilistic Behavior Networks – A Robot's Central Nervous System



In the previous chapter, the bio-inspired antagonistic muscular control for Series Elastic Actuator (SEA) hardware was developed. This novel low-level control implements a hybrid impedance control strategy designed to suit natural, entirely compliant, over-actuated, and multiarticular limb morphologies. Consecutively, this chapter covers the next higher hierarchical control layer required for an entire, well-fitting natural robot control framework. As highlighted in the introductory image, this consecutive step is to define an appropriate high-level control architecture. In the context of bio-inspired engineering, such a high-level robot control architecture defines the artificial replication and implementation of the natural Central Nervous System (CNS).

Robot technology in all related disciplines (see Table 2.1) has significantly evolved in recent years. Among others, materials, actuators, sensors, compute units, and algorithms have faced rapid, ongoing development. Nowadays, growing Artificial Intelligence (AI) capabilities, strongly pushing into robotics, further accelerate this development.

Unfortunately, with increasing system complexity, the number of sensors and actuators, as well as performance expectations, the required control complexity explodes. Such an explosion of complexity especially happens in the field of humanoid robots, where multiple cooperative and competitive tasks must be tracked in parallel. Thus, a suitable bio-inspired control architecture must, on the one hand, satisfy the natural control requirements derived from the six initially identified gait characteristics in Section 2.1. However, at the same time, it must also offer all necessary controls for its underlying, low-level control, e.g., the antagonistic muscular control derived in Section 3.2.3.

Simple monolithic control approaches have been proven perfectly fine for simple robots executing a single task. However, modern demands on versatility and task complexity require more sophisticated control architectures both from the capability and maintainability perspectives. Most likely, the biggest challenge in designing such an architecture is to balance its flexibility, efficiency, and standardization. Hence, a key design goal of a versatile robot control architecture is to find a simple but powerful set of axiomatic rules defining the whole architectural structure. These rules should satisfy the following guidelines:

Theorem 4.1: Guidelines on Robot Control Architecture Design Rules

Principle design rules for versatile robot control architectures should:

1. Not limit the architecture's overall capabilities.
2. Offer strict algebraic structure, allowing for theoretical analysis and formal verification.
3. Support modern software and control engineering techniques.
   E.g., Maintainability, Continuous Integration & Continuous Delivery *(CI/CD), Model Predictive Control (MPC), or Artificial Intelligence (AI).*
4. Be as simple as possible and just as complex as necessary.

Since so far none of the existing Behavior-Based (BB) robot control architectures has proven itself to outperform all others, a great variety of ideas and distinct approaches has been developed over the past decades.

Among others, one example of a robot control approach following no explicit architectural structure is the work in [Aceituno-Cabezas 18]. Aceituno-Cabezas et al. propose to use only a single global optimization strategy capable of handling simultaneous contact, gait control, and motion planning. Although this approach has successfully been demonstrated on real-world hardware, i.e., the quadruped robot HyQ[1] [Semini 11, Semini 15, Semini 16, Aceituno-Cabezas 18], its limits are clearly highlighted by the authors. In order to achieve new behaviors of the robot, the whole control algorithm has to be reworked. Hence, it neither easily extends to other types of quadrupedal gait nor other tasks than a gait. Due to the monolithic, centralized structure and built-in mathematical issues, this approach undoubtedly lacks desirable versatility and flexibility.

In 2006, Asfour et al. published a task coordination methodology entirely based on the well-known theory of *Petri nets*[2] [Asfour 06]. Obviously, such an approach unlocks the whole universe of existing algebraic theory on *Petri nets*, including all types of analyses and computer-aided, automatic proving. Its drawback, however, is the required high abstraction level of modeling. Each task has to be encoded together with its relation

[1] HyQ is a hydraulic-electric actuated quadruped robot by the *Italian Institute of Technology*.

[2] *Petri nets*, also called *place-transition nets*, embody a special graph theory based on finite automata. They are named after Carl Adam Petri, who introduced and extensively studied this theory in his 1962 Ph.D. thesis *"Kommunikation mit Automaten"* [Petri 62]. *Petri nets* are intended to formally model and analyze distributed, concurrent processes, e.g., communication protocols.

to others in the graph representation of a *Petri net*. Unfortunately, *Petri nets* have been designed for a different purpose. Hence, they do not out-of-the-box support robotic task representations. Furthermore, with the introduction of additional tasks into the model, the corresponding net graph keeps rapidly growing. Hence, corresponding models quickly become significantly large, complex, unhandy, and eventually challenging to understand or interpret.

With the ongoing advancements in AI, those techniques become increasingly important not only for perception systems but also for robot controllers. Successful examples of Machine Learning (ML)-involving robot control approaches in both simulation and real-world hardware can be found in [Geijtenbeek 13, Li 21, Shi 21]. Interestingly, none of those examples implement a so-called *end-to-end* solution. Instead, the Neural Networks (NNs) are embedded into surrounding control structures.

Geijtenbeek et al. learn control policies on top of muscle-imitating actuators and gait-encoding Finite State Machines (FSMs) per leg [Geijtenbeek 13]. The actuators feature impedance characteristics as presented in Figure 3.10. Li et al. train a NN as a control policy to follow a reference gait precomputed using Hybrid Zero Dynamics (HZD), introduced in Section 2.3.1 [Li 21]. The control commands are fed into low-pass filtered, dedicated joint controllers. Thus, the NN implements an abstract inverse dynamics model. A similar strategy is enrolled by Shi et al., using predefined decision models assisting a learned control policy driving dedicated joint controllers [Shi 21].

The above examples demonstrate that ML-based robot control is possible and, once trained, also computationally efficient and capable of deterministic, real-time execution. However, all those AI approaches are first trained to achieve only a single control goal and, second, are embedded in a surrounding digital infrastructure. Hence, such approaches are better treated as individual skills rather than an entire framework to interconnect those skills and interface with hardware.

Any mobile robot can be expected to operate in a dynamic environment. Thus, suitable controllers must feature fast response times and high situational adaptability. In the "*Springer Handbook of Robotics*" (2nd Edition) [Siciliano 16, chap. 13], Michaud and Nicolescu advise using BB control, especially for robots in such conditions. BB concepts offer high flexibility, maintainability, and extendability by exploiting the algebraic structure of interconnected control graphs. After its foundations in the 1980th, the BB philosophy emerged as one among other practical solutions for large-scale, multi-purpose, component-wise robot control architectures. I.e., especially in the case of multi-purpose, mobile walking robots in unstructured environments: Humanoids.

Chapter Outline

Throughout the following sections, this chapter will first introduce both biologically and technically motivated architectural requirements in Section 4.1. Section 4.2 then gives a brief introduction and overview of the continuous development of BB robot control. All subsequent sections consecutively develop and define the herein-proposed BB architecture, referenced as the so-called Probabilistic Behavior Networks (PBNs). Some of the most elementary parts of this architecture have already been defined and published in [*Vonwirth 24a*].

Section 4.3 defines the unified, axiomatic semantics underlying the Probabilistic Behavior Network (PBN) architecture: its data representation and calculation rules. Afterward, Section 4.4 specifies concrete, shared data representations for individual data types and use cases required to represent a bipedal robot's state and its hybrid impedance control commands, as derived in Chapter 3. The mandatory interfacing of probabilistic sense and control representations inside a PBN with the sharp data related to a robot's hardware is covered in Section 4.5. Then, Section 4.6 motivates the development of BB control composition and derives the respective unique PBN behavior equivalent. Sense and control abstraction and their realization within the PBN architecture are presented and discussed in Section 4.7. Finally, a simple virtual example is presented in Section 4.8, followed by a conclusion of this chapter in Section 4.9.

4.1 Biologically and Technically Motivated Requirements

It is well-known that embedded control software, and especially robot control systems, differ a lot from the usual, co-called *desktop* software products [Siciliano 16, p. 284]. Robot control systems, and hence system architectures, demand special, domain-specific requirements, mostly *non-functional* requirements. Among others, these may include:

- Real-time processing
- Calculation and system behavior stability
- Reliable resource management of both energy and computational capacity
- Uncertainty handling of both inputs and outputs
- Adaptation to dynamic environmental changes
- Varying temporal task scopes
- Coordination of potentially competitive, parallel tasks: multitasking
- self-observation and self-adaptation (e.g., calibration or future ML) possibilities
- *Bio-inspiration* in the form of the natural control requirements derived from the human gait characteristics, identified throughout Section 2.1

In the context of this work, a special focus is laid on the latter ones and their implications.

As illustrated in Figure 2.14, the most relevant architectural natural control Requirements are 2.2: *"Singular Positions"*, 2.4: *"Nonlinear Relations"*, 2.5: *"Hierarchical Control"*, and 2.6: *"Model Reduction"*. These bio-inspired requirements have been derived and discussed in detail within Sections 2.1.2, 2.1.4, 2.1.5, and 2.1.6. An overview of these, together with additional necessary technical requirements for a general robot control software architecture, is sketched in Figure 4.1. All of these requirements comply with the initially defined architectural robot control guidelines in Theorem 4.1.

The identified individual architecture requirements are:

1. Unified, Uncertain Data:
   A mobile, herein-legged robot is always meant to be deployed in a dynamic and changing environment. Thus, uncertainty is naturally embedded. In the context of legged locomotion, uncertainty in control was introduced in Requirement 2.3: *"Variable Compliance"*. Hence, by following the principle design guidelines in Theorem 4.1, it is evident that one unified, uncertain data representation is required for both sense and control processing.

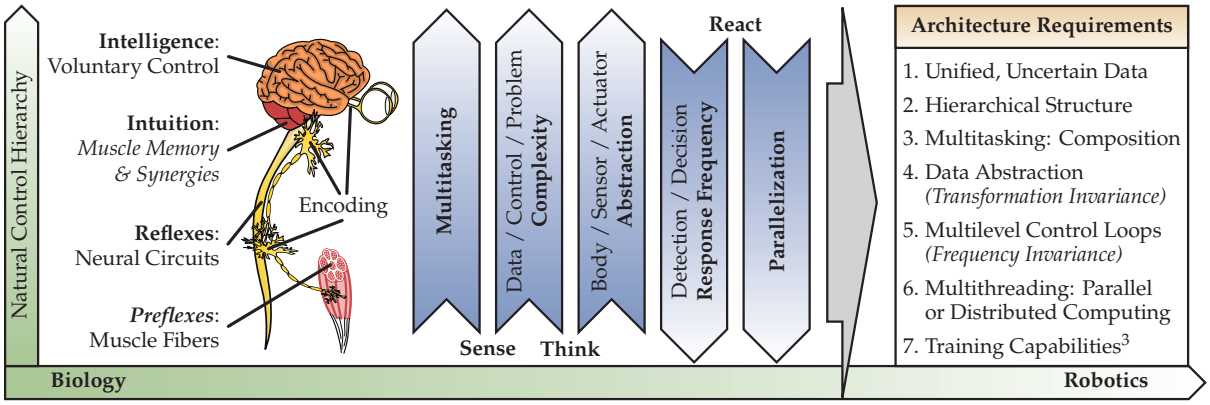


Figure 4.1: A sketch of architecture requirements of a natural robot control framework, derived from the natural control hierarchy in the human neuromuscular system and in compliance with Theorem 1.1. On the left, the principal control hierarchy of a human's CNS and muscular actuation system is sketched, as previously motivated in Figure 2.9 and Requirement 2.5: *"Hierarchical Control"* within Section 2.1.5. In the middle, the most significant, robot-relevant features and their changes in manifestation throughout that hierarchy are pictured. A list of all derived architecture requirements is given on the right. Note that the two natural control Requirements, 2.2: *"Singular Positions"* and 2.4: *"Nonlinear Relations"*, as listed in Figure 2.14, are both part of the architecture requirements to support (arbitrary) data abstraction.

2. Hierarchical Structure:
   In Section 2.1.5, analyses of the natural nervous system and its attached muscles have been described. The discussion reveals significant hierarchical control structures within the human CNS and muscles. Consequently, to achieve the overall aim of Theorem 1.1: *"Natural Control"*, support for hierarchical control structures becomes an essential architectural requirement.

3. Multitasking – Data Composition:
   The demand for proper data composition is equally argued as the unified, uncertain data representation. Sensor data composition is embedded in modern, multi-sensor hardware and model-based filter techniques such as the *Kalman* filter. Control data composition is embedded in the just-required architectural control hierarchy and the demanded multitasking capability, as derived from natural gait characteristics in Requirement 2.5: *"Hierarchical Control"* and illustrated in Figures 2.9 and 4.1.

4. Data Abstraction – Transformation Invariance:
   Embedded in the hierarchical structure of a human's CNS, a bottom-up gradient of task or control abstraction can be found. Mathematically, such an abstraction is realized by a coordinate transformation between spaces of different dimensions. Following the natural control Requirements 2.2: *"Singular Positions"*, 2.4: *"Nonlinear Relations"*, and 2.6: *"Model Reduction"*, arbitrary transformations, including nonlinear, non-injective, and non-surjective projections, must be supported. In Section 3.3.1, such coordinate spaces and transformations have already been introduced and graphically sketched in Figure 3.17.

5. Multilevel Control Loops – Frequency Invariance:
   Initially analyzed in Section 2.1.5 and illustrated in Figure 2.9, the natural neural delay increases with the level of abstraction, i.e., with the neural distance to the executing muscles. In addition to this delay, time-consuming mental preparation, reasoning, and planning processes significantly add further delay to the human neural response time when climbing up the CNS's neural hierarchy. Thus, as demanded by Requirement 2.5, tasks at different hierarchical levels should be executable at individual frequencies.

6. Multithreading – Parallel or Distributed Computing:
   The demand of processing sense and control data in parallel, distributed among multiple CPUs, is not necessarily bio-inspired. Instead, it is required by today's available hardware, computer infrastructures, and the sheer volume of data that needs to be processed. Two reasons are causing such significantly large amounts of data: First, large-scale data, such as high-resolution images, and second, high-speed data, such as centralized wrench control algorithms.

   At the lowest control layers, the natural neuromuscular system features concurrent neural reflex circuits and tunable mechanical *preflex* properties, as introduced in Figure 2.9 within Section 2.1.5. However, it is essential to note that the hybrid impedance control strategy, as developed in Chapter 3, already implements a strategy comparable to this neural processing distribution strategy.

7. Training Capabilities[3]:
   Although biology, obviously, is capable of learning, enhancing, and training its capabilities, this requirement is mainly intended for continued future work and arbitrary usability of the emerging control architecture beyond the scope of this work. Hence, this final requirement covers the architecture's compliance with modern agile software engineering processes and state-of-the-art robot control technology, such as continuous integration or AI, respectively.

4.2 Behavior-Based Robot Control Decomposition

In 1986, Brooks introduced his so-called *subsumption* architecture [Brooks 86]. Over time, this paper has become the de-facto standard work and the foundation of BB robotics. In this work, Brooks presents a fundamental new principle of decomposing complex robot control software. Instead of a sequential decomposition into *"functional modules"*, Brooks proposes a parallel decomposition into small, independent, so-called *"task-achieving behaviors"* [Brooks 86].

Following this BB decomposition strategy, a stack of individual control modules is constructed where all behavior modules run in parallel to each other. Thus, all individual control data must eventually be recomposed before being handed over to the underlying hardware or low-level control as one single command. A set of control composition

[3] Training Capabilities in this context are meant to include both manual and automatic enhancements. Hence, it embeds modern software engineering guidelines such as software composition, decomposition, and reorganization invariance, as well as AI integration via intrinsic support of basic ML techniques.

rules defines the correct control recomposition. The identical argumentation can be applied to the composition of redundant sensor data.

BB control, according to the just mentioned decomposition principle by Brooks, already satisfies many of the architectural requirements introduced in Figure 4.1 by design. Due to the independent parallel behaviors, multitasking, and parallelization capabilities exist by definition if the underlying data representation is suitable.

Similarly, as described above, data composition is one of the most crucial foundational definitions of any BB architecture. Together with the fundamental principle of behavior decomposition, the desired control reorganization capability follows directly. However, all those technical features are only valid when the respective technical invariances, as listed in Figure 4.1, are held throughout the entire BB control computation.

With the continuous growth of expectations, complexity, and computational power of embedded systems and robotics, the original *subsumption* architecture by Brooks underwent and still undergoes continuous evolutionary development. Numerous domain-specialized derivatives have been developed continuously adapting the behavior interconnection rules to the increasing demands on robot control software.

A representative but non-exhaustive collection of progressing BB control development is given by the following works in temporal order from 2002 till 2022: [Nicolescu 02, Vadakkepat 04, Scheutz 04, Proetzsch 07, Proetzsch 10b, Ropertz 17, Ropertz 19, Wolf 22].

Although all these architectures, throughout their development processes, rapidly diverted from the original *subsumption* implementation, the two most significant design principles of Brooks have been kept. In [Proetzsch 10a], Proetzsch summarizes these two fundamental architectural design principles in the following two rules:

Definition 4.1: Design Rules of Behavior-Based Robot Control Architectures

A Behavior-Based (BB) control architecture should satisfy the following two fundamental design rules:

Behavior Flexibility

A behavior module's internal computation must not be constrained or restricted.

Interface Duality

A behavior module's output data must not be constrained except for a strictly standardized metadata interface. These metadata are responsible for standardized inter-module connections defined by the architecture's data composition rules.

based on [Proetzsch 10a, p. 26 & 27]

Following these two architectural principles, the key design definition of a BB robot control network all comes down to its data composition rules.

Back in 1986, Brooks specified in his original *subsumption* architecture that behavior control data have to be either kept untouched or entirely overridden by others [Brooks 86]. About one and a half decades later, in 2002, Nicolescu and Mataric proposed a novel but still similar approach in [Nicolescu 02]. In their hierarchical organized control

architecture, behaviors are allowed to operate in parallel as long as they do not share required hardware resources, i.e., actuators.

On activation, a behavior must reserve all its required resources. If not all hardware components are available, the behavior must not become active. Consequently, this approach does not allow the parallel execution of competitive but not mutually exclusive tasks.

To overcome this limitation, concurrent execution and a mechanism to merge rather than override control data is required. The *fuzzy*-based strategy by Vadakkepat et al. and the Integrated Behavior-Based Control (iB2C) architecture by Proetzsch et al. both address this issue by introducing continuous weights, respectively, *"rule strengths"* [Vadakkepat 04] and *"behavior activities"* [Proetzsch 07, Proetzsch 10b].

While the binary (active or inactive) behavior activity by Brooks [Brooks 86] and Nicolescu and Mataric [Nicolescu 02] only allows data selection in a binary decision tree, a relaxed activity ranging between zero (full inactive) and one (full active) offers continuous weighting and balancing of competitive control data. Data composition is handled according to the respective *fuzzy* rule strengths or behavior activities. Both architectures define the same two data composition strategies: *maximum fusion* and *average fusion*.

The *maximum fusion* is a multiplexer forwarding the most relevant input data untouched. Thus, this functionality precisely realizes the well-known binary decision-making of the early BB concepts. In addition to this original *subsumption* concept, the *average fusion* merges all involved inputs by forwarding a weighted average according to the relevance of the input data.

By using averaged data, control goals or sensory inputs can be compromised, and multiple potentially competitive tasks can be followed in parallel. Thus, with the introduction of control composition, the BB control philosophy develops its architectural design from a simple stack of tasks to a full-featured data flow computer. Unfortunately, the standard weighted average is only well-defined on Cartesian vectors. A direct consequence is that neither architecture can handle cyclic data, such as angles or, more generally, orientations[4].

To correctly balance competitive control data in a meaningful manner, their metadata weights must be in a clear and known relation to each other. Thus, a semantic specification of those behavioral metadata by the architecture is a crucial design decision. Unfortunately, both the *fuzzy* BB architecture and the iB2C lack such a definition, leaving this interpretation open to the developers' individual personal opinions. Both architectures also do not explicitly specify their behavior in the case where only entirely inactive control data are available. I.e., a second crucial lack of specification, potentially caused as a follow-up of the already missing metadata semantics.

[4] Being well aware of the fact that Vadakkepat et al. include the control of their robot's orientation using a PID controller, the given statement about the orientation incapability remains valid [Vadakkepat 04]. First, applying such a linear controller onto an angle representation either leads to sudden jumps in the control whenever the angle representation flips (0° = 360°) or may allow for multiple complete rotations (torsional spring-like) when the angle representation is not flipped but fully enrolled. In any case, both variants lack a proper geometric meaning of a weighted average. See as a reference, e.g., [Caccavale 98a].

Analyses on the iB2C architecture by Ropertz uncovered serious mis- or non-usage problems due to semantically not-specified metadata in operational robot control networks [Ropertz 19, p. 50,51]. In response, Ropertz et al. introduce the concept of so-called *quality-based sensor fusion* [Ropertz 17], which has been successfully enrolled and tested on several operational off-road vehicles by Wolf [Wolf 22, p. 121-125]. This exclusive concept, along with the analyses by Ropertz [Ropertz 19], serves as the foundation of the novel PBN architecture defined throughout the rest of this chapter.

By adding the just-argued need for clearly defined semantics together with the desire for embedded algebraic structure, as demanded in Theorem 4.1, the former-cited BB architectural design rules in Definition 4.1 should be redefined:

Definition 4.2: Design Rules of Semantic BB Robot Control Architectures

A Behavior-Based (BB) control architecture should satisfy the following two fundamental design rules:

Behavior Flexibility

A behavior network's overall computational capability must not be limited by computational constraints on individual behaviors.

Interface Duality

A behavior module's (output and input) data must not be limited in its overall representation capability. However, a well-defined semantic, specified by the architecture's data composition rules, is required to ensure standardized inter-module connections.

reformulated from Definition 4.1

As described, the new formulations do not alter the overall intention of the original design rules. Instead, the rules' intentions are moved from a technical level to a functional, semantic one.

Behavior flexibility, as redefined in Definition 4.2, now allows restrictions on the internal computation of individual behaviors. However, such a restriction is only allowed if the overall computational capability of a wider behavior network is not limited. Such a computational network structure is very well-known nowadays. It is the same algebraic structure as in artificial NNs.

Similarly, the reformulated definition of *interface duality*, as given in Definition 4.2, now allows for data format restrictions within a BB control network. However, such a restriction must not limit the capability of representing arbitrary data semantics. For example, it is valid to prohibit Euler angle representations if, instead, rotation matrices or quaternions can be used to describe the same geometric property.

Additionally, the *interface duality* definition also raises the data composition rules to a semantic level. While in the original Definition 4.1, only a standardized (technical) interface for data composition is demanded, the new Definition 4.2 requests standardized semantic, thus meaningful, data composition rules.

4.3 Definition of a Probabilistic Behavior Network as a Data Flow Computer

As stated in Theorem 4.1, the collection of architectural requirements, presented in Section 4.1, should be achieved by a set of architectural rules as simple as possible. At the same time, those rules must be versatile and flexible enough so that the architecture's overall computational capability, as specified in Definition 4.2, is not restricted.

Following the two principles of *behavior flexibility* and *interface duality* given in Definition 4.1, the core definitions of a BB robot control decomposition architecture are its *data representation*, *calculation*, and *composition* rules. However, data composition can be treated as one specific instantiation of general data calculation. Thus, two axiomatic definitions can define an entire BB robot control architecture. These definitions describe BB control as a *directed, acyclic graph*, i.e., a strictly standardized data flow architecture.

Definition 4.3: A Probabilistic Behavior Network as an Acyclic Graph

Let $\mathcal{X}$ be a set of arbitrary PBN data representations with elements $X_n \in \mathcal{X}$. A Probabilistic Behavior Network (PBN) is a directed, acyclic graph $\mathcal{G}$. Its nodes, also called vertices, represent functional behaviors, and its edges describe the data connections between those behaviors' function inputs and result outputs.

$$\mathcal{G} := (B,\ E) \tag{4.1}$$

Where:

B is a set of behavior nodes $\mathcal{B}(\cdot) \in B$ representing a function $\mathcal{B} : \mathcal{X}^n \to \mathcal{X}$.
I.e.: $\mathcal{B}(X_1,\ X_2,\ \ldots,\ X_n) = X_{out}$

E is a set of triples $e \in E$ connecting behavior results to behavior inputs. $E := \{(\mathcal{B}_{src},\ \mathcal{B}_{dest},\ X_i) |\ \mathcal{B}_{src},\ \mathcal{B}_{dest} \in B,\ \mathcal{B}_{src} \neq \mathcal{B}_{dest},\ X_i \in \mathcal{X}\}$ such that $\mathcal{B}_{src}(\ldots) = X_i$, and $\mathcal{B}_{dest}(\ldots,\ X_i,\ \ldots)$ accepts X_i as its i-th input.
Furthermore, no two edges are allowed to connect to the identical input. Thus, if $e_1 = (\mathcal{B}_1,\ \mathcal{B}_{dest},\ X_i)$ and $e_2 = (\mathcal{B}_2,\ \mathcal{B}_{dest},\ X_i)$, then $e_1 = e_2$.

Following the PBN graph Definition 4.3, its corresponding probabilistic input and output interfaces are defined as subsets of the overall set of data representations $\mathcal{X}$:

1. Output Possibilities: $O_P(\mathcal{G}) \subset \mathcal{X}$
   The set of possible output data $O_P(\mathcal{G}) := \{X \in \mathcal{X} |\ \exists \mathcal{B} \in B,\ \mathcal{B}(\ldots) = X\}$ is the set of all behaviors result data.

2. Output Data: $O(\mathcal{G}) \subset \mathcal{X}$
   The set of actual output data $O(\mathcal{G}) \subseteq O_P(\mathcal{G})$ is a user-defined subset of all possible output data.

3. Input Data: $I(\mathcal{G}) \subset \mathcal{X}$
   The set of input data $I(\mathcal{G}) := \mathcal{X} \setminus O_P(\mathcal{G})$ is the set of all PBN data that are not behavior result data, thus behavior inputs.

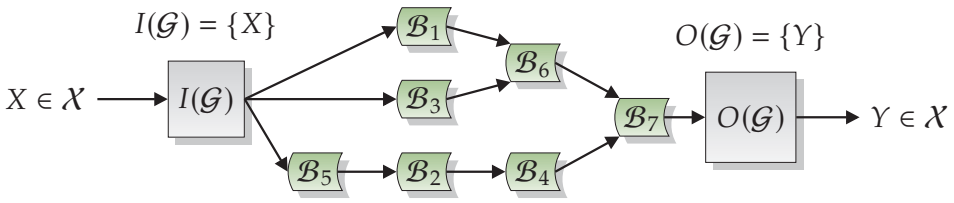


Figure 4.2: Exemplary PBN graph $\mathcal{G}$ featuring seven interconnected behaviors. Its input and output interfaces each contain only a single data element. The graph's functional equivalent is given underneath its graphical presentation. It is defined to map the PBN's set of inputs to its set of outputs, i.e., $\mathcal{G} : I(\mathcal{G}) \rightarrow O(\mathcal{G})$.

A PBN graph $\mathcal{G}$ describes a concatenation of functions, called behaviors $\mathcal{B}$, operating on the set of PBN data representations $\mathcal{X}$. Both the data representations and behavior functions will be specified by Definitions 4.4: *"PBN Data Representation"* and 4.6: *"PBN Data Calculation"*, respectively. Figure 4.2 provides a graphical and equivalent functional representation of a small example PBN graph defined as follows:

$$\begin{aligned}
\mathcal{X} &:= \{X,\ Y,\ Z_1,\ Z_2,\ Z_3,\ Z_4,\ Z_5,\ Z_6\} \\
B &:= \{\mathcal{B}_1,\ \mathcal{B}_2,\ \mathcal{B}_3,\ \mathcal{B}_4,\ \mathcal{B}_5,\ \mathcal{B}_6,\ \mathcal{B}_7\} \\
E &:= \{(\mathcal{B}_1,\ \mathcal{B}_6,\ Z_1)\,,\ (\mathcal{B}_2,\ \mathcal{B}_4,\ Z_2)\,,\ (\mathcal{B}_3,\ \mathcal{B}_6,\ Z_3)\,,\ (\mathcal{B}_4,\ \mathcal{B}_7,\ Z_4)\,, \\
&\qquad (\mathcal{B}_5,\ \mathcal{B}_2,\ Z_5)\,,\ (\mathcal{B}_6,\ \mathcal{B}_7,\ Z_6)\} \\
\mathcal{G} &:= (B,\ E) \\
O_P(\mathcal{G}) &= \{Y,\ Z_1,\ Z_2,\ Z_3,\ Z_4,\ Z_5,\ Z_6\} \\
O(\mathcal{G}) &:= \{Y\} \\
I(\mathcal{G}) &= \{X\}
\end{aligned} \tag{4.2}$$

The graph's functional equivalent $\mathcal{G} : I(\mathcal{G}) \rightarrow O(\mathcal{G})$ maps its set of inputs to its set of outputs, i.e., $\mathcal{G}(\{X\}) = \{Y\}$. It is given in the figure just underneath the graphical PBN presentation. Note that the data connections from and to the PBN's interfaces shown in Figure 4.2 are not part of the graph's edges. Instead, they automatically emerge from them, as defined.

In general, behaviors may also be defined to produce multiple results $\mathcal{B}^m : \mathcal{X}^n \rightarrow \mathcal{X}^m$. This way, several results of a single *source* behavior may be connected to the inputs of a single *destination* behavior, thus extending the PBN graph into a multigraph. However, such a multi-result behavior $\mathcal{B}^m$ can always be decomposed into m individual behaviors, one for each result. Thus, although recommended for practical implementation, the following definitions, without loss of generality, rely on single-result behaviors only.

Due to the well-researched graph structure, formal verification techniques such as reachability or dependency analyses can be applied to the entire robot control structure. Throughout the development process of the iB2C architecture, such analyses have already been developed and demonstrated in several doctoral theses [Proetzsch 10a, Armbrust 15, Kiekbusch 17, Ropertz 19]. However, due to the limited scope of this thesis, formal analyses w.r.t. the proposed PBN architecture are left open for future work.

The unified set of data representation $\mathcal{X}$ in a PBN, specifying the content of the underlying data flow graph edges, as described in Definition 4.3, is defined as follows:

Definition 4.4: PBN Data Representation

All data vectors $X \in \mathcal{X}$ sent through the edges of a PBN graph $\mathcal{G}$ represent a **multivariate normal distribution** or an adequate analog[a] for non-Euclidean spaces, such as Cartesian orientations.

$$X \sim \mathcal{N}(\mu, \Sigma) \tag{4.3}$$

With Mean $\mathbb{E}[X] = \mu$,
and Variance $\mathrm{Var}(X) = \Sigma \geq 0$, positive semi-definite.

[a] The normal distribution, also called Gaussian distribution, is defined only on Euclidean domains $\mathbb{R}^n$. For rotation or orientation representations $q \in \mathrm{SO}(n)$, the quaternion-relevant Bingham distribution on unit spheres is used instead [Bingham 74]. The Bingham distribution is considered the *"analogue of the normale distribution in Euclidean space"* [Schaeben 92, p. 199].

This definition follows the earlier demand on the iB2C architecture to annotate all input and output data with related certainty or tolerance information [Ropertz 19, p. 45]. However, in contrast to all the previously mentioned BB architectures, a PBN does not feature any data-independent, additional meta information, such as a dedicated behavior activity. Thus, the data-related satisfaction function is defined as follows:

Definition 4.5: PBN Data Satisfaction

The **satisfaction** of arbitrary probabilistic data $X \sim \mathcal{N}(\mu, \Sigma)$ against an equally dimensioned external mean value u is equal to the data's unscaled Probability Density Function (PDF).

$$\mathrm{sat}_X(u) := \alpha \, PDF_X(u) \in (0, 1] \tag{4.4}$$
$$\text{with } \alpha \text{ such that: } \quad \mathrm{sat}_X(\mu) = 1 \quad \text{and} \quad \mathrm{sat}_X(u \neq \mu) < 1$$

The PBN data satisfaction function inherits its underlying PDF's invariance against linear transformation.

$$\mathrm{sat}_{\mathcal{N}(\mu,\, \Sigma)}(u) \;=\; \mathrm{sat}_{\mathcal{N}(A\mu+B,\, A\Sigma A^T)}(Au + B) \tag{4.5}$$

The satisfaction calculation can be diagonalized by decomposition of the data's variance matrix Σ. For example, using the *Singular Value Decomposition* (SVD) or, computationally cheaper, the *Square Root Free Cholesky Decomposition* (LDL).

$$\text{using SVD}: \; \Sigma = Q\Lambda Q^T \quad \rightarrow \quad \mathrm{sat}_{\mathcal{N}(\mu,\, \Sigma)}(u) = \mathrm{sat}_{\mathcal{N}(Q^T\mu,\, \Lambda)}\left(Q^T u\right) \tag{4.6}$$
$$\text{using LDL}: \; \Sigma = LDL^T \quad \rightarrow \quad \mathrm{sat}_{\mathcal{N}(\mu,\, \Sigma)}(u) = \mathrm{sat}_{\mathcal{N}(L^{-1}\mu,\, D)}\left(L^{-1} u\right) \tag{4.7}$$

When using floating-point arithmetic, as is the standard case in modern CPUs, the satisfaction function may suffer from numerical instabilities around its maximum result

value of one. This numerical risk is due to the fact that the density of representable numbers in the floating-point format increases toward zero and thus decreases on an approach from zero toward one. Consequently, the satisfaction function, as defined above, is more suitable to represent low satisfaction values around zero than desired high values close to one. The following dissatisfaction function is defined to overcome this numerical threat:

Implementation Hint 4.1: PBN Data Dissatisfaction

The **dissatisfaction** of probabilistic data $X \in \mathcal{X}$ is its satisfaction minus 1.

$$\begin{aligned} \text{sat}^{-}_X(u) &:= \text{sat}_X(u) - 1 \in (-1, 0] \\ \text{with:}\quad \text{sat}^{-}_X(\mu) &= 0 \quad \text{and} \quad \text{sat}^{-}_X(u \neq \mu) < 0 \end{aligned} \tag{4.8}$$

Note that the satisfaction function will be defined by an outer exponential in the following scheme: $\text{sat}(\cdot) \cong \exp\left(-\frac{1}{2}\delta^2\right)$. In order to benefit from the higher floating-point representation density around zero, the dissatisfaction function must be implemented using the native expm1$(\cdot)$ function, explicitly implementing $\exp(\cdot) - 1$ for high-resolution floating-point results close to zero.

Following the above data and satisfaction Definitions 4.4 and 4.5, the unified result calculation rule for all behavior nodes in a PBN graph is defined as follows:

Definition 4.6: PBN Data Calculation

Let $X_{i\in I} \in \mathcal{X}$ be an arbitrary number of probabilistic input data. All behavior nodes $\mathcal{B}(X_{i\in I}) = X_{out}$ of a PBN graph $\mathcal{G}$ compute their result data $X_{out} \in \mathcal{X}$ with $X_{out} \sim \mathcal{N}(\mu_{out}, \Sigma_{out})$ by the following two rules:

1. Finding a **locally optimal** result mean value μ_{out} that **most satisfies** all its input data $X_{i\in I}$, respective to the behavior's task function $t(\cdot)$, defined such that $t(\mu_{i\in I}) \overset{!}{=} \mu_{out}$.

$$\arg\max_{\mu_{out}} \prod_{i\in I} \text{sat}_{X_i}\left(t^{-1}(\mu_{out})[i]\right) \tag{4.9}$$

   - To build a **proper**, i.e., information-preserving behavior, the function concatenation $(t^{-1} \circ t)(\cdot) = \text{ID}(\cdot)$ must result in the identity function.
   - If this identity is not given, but the optimization can still be calculated, an **improper**, i.e., information-lossy behavior emerges.

2. Finding a technically and semantically adequate result variance Σ_{out}, respective to the behavior's task $t(\cdot)$, and all its input variances $\Sigma_{i\in I}$. Note that the result variance Σ_{out} must preserve the PBN's most significant algebraic properties, such as commutativity or associativity, whenever needed.

Note that a proper behavior's general task function $t(\cdot)$ is not expected to be uniquely determinable, i.e., well-defined. Hence, its left (outer) inverse $t^{-1}(\cdot)$ defines a behavior's

result mean calculation. That means that $\left(t^{-1} \circ t\right)(\cdot) = \text{ID}(\cdot)$ must result in the identity function. However, this identity may not hold for the reverse concatenation, i.e., $\left(t \circ t^{-1}\right)$ may not even be well-defined. Thus, a proper behavior's task function $t(\cdot)$ must not be projective, thus information-preserving. In contrast, improper behaviors may implement an information-lossy data projection. Consequently, improper behaviors must only be used for processing data that must not be composed, i.e., merged, after projection. Details on this restriction will later be covered in more detail in Section 4.7.1. Since a behavior may have multiple inputs, the index i in the square brackets added to the task inverse function denotes the inverse w.r.t. the i-th input mean value μ_i.

Each behavior's mean optimization reward function, as defined in Equation (4.9), is a product of individual satisfaction functions. Due to a satisfaction's limited scalar result interval, as given in Definition 4.5, a product of multiple satisfactions is equally bound within the interval $(0, 1]$. However, the maximum satisfaction value of 1 is only achieved if all individual satisfactions are also evaluated to their unique maxima. Thus, the PBN calculation rule in Definition 4.6 offers a well-defined optimization with a unique global maximum to be desired.

According to Definition 4.5, the satisfaction function $\text{sat}(\cdot)$ is nothing else than a regular PDF, scaled by a constant factor. Inserting this definition into the behavior mean calculation Equation (4.9), it is easy to recognize that all those individual, constant scaling factors can be collected and merged into one single constant scaling factor of the entire product. However, argument optimizations are invariant against constant scaling. Thus, Equation (4.9) inside the behavior calculation Definition 4.6 can be rewritten using PDFs instead of the satisfaction functions.

$$\arg\max_{\mu_{out}} \prod_{i \in I} \text{sat}_{X_i}\left(t^{-1}\left(\mu_{out}\right)[i]\right) \;=\; \arg\max_{\mu_{out}} \prod_{i \in I} PDF_{X_i}\left(t^{-1}\left(\mu_{out}\right)[i]\right) \tag{4.10}$$

A PBN behavior, therefore, entirely complies with well-known probability calculus and can exactly be treated as such a computation instance. Thus, maximizing the satisfaction function, as defined in Equation (4.9) within Definition 4.6, is equivalent to finding one of the modes of the respective underlying PDF. The definition of the behavioral mean calculation rule via the prior-defined satisfaction instead of the underlying PDF's modes is due to technical reasons. The most significant are mathematical simplicity, better understandability without stochastic background, and algorithmic, numerical, and exponential stability.

Non-maximum satisfaction values become non-unique. Therefore, a PBN behavior implements a local-only optimization. As formulated in Requirement 2.3: *"Variable Compliance"*, control optimality is task-dependent and not mandatorily needed for natural locomotion. Moreover, Requirement 2.5: *"Hierarchical Control"* defines a local control decision hierarchy as one of the most significant natural control characteristics. Thus, in the context of the herein-desired smooth and natural control, as formulated in Theorem 1.1, continuous and reliable local optimization is preferable over global optimality. In nature, those fast response times are achieved by local control decisions via muscular *preflexes* and *reflexes*. Slower, more global optimization strategies are found in the higher-level CNS in the form of *muscle memory*, *muscle synergies*, and, eventually, the cerebrum's intelligence.

Undesired jumps of the optimization result can be avoided algorithmically by applying well-known, local-only gradient descent optimization strategies. Result smoothness can be guaranteed for equally smooth input sequences when so-called *hot starting* the gradient descent iteration at the optimization's prior result.

Furthermore, behavior-local optimization can be implemented by a whole variety of state-of-the-art control approaches. For example, this variety includes well-known classical robot control approaches, such as PID control, Optimal Control (OC), or Model Predictive Control (MPC). However, at the same time, state-of-the-art developments in AI are also supported.

Since artificial NNs themselves encode the result of a beforehand-executed (trained) optimization, they entirely comply with the herein-defined novel data calculation definition of a behavior node, as given in Definition 4.6. Error backpropagation and online behavior adaptation may also be implemented throughout a PBN if the involved task functions $t(\cdot)$ are suitable for such actions. Due to the task function's inverse usage $t^{-1}(\mu_{out})$ within the behavioral data optimization equation (4.9), reverse data flow, such as backpropagation, is always well-defined for proper behaviors. Updating the behavior's task function translates its internal potential satisfaction field toward the externally updated control goal.

4.3.1 Commutativity and Associativity

The requirement to support hierarchical control structuring has already been demanded for the iB2C architecture [Proetzsch 10a, p. 27]. However, due to the iB2C behavior's unique scalar activity, such a behavioral grouping must not preserve the iB2C's metadata inside the group. This structural problem has been addressed by the introduction of so-called *layer* modules, offering additional metadata interfaces [Wolf 22, p. 62]. However, the introduction of additional meta information to an iB2C behavior module breaks its original behavior definition. Furthermore, the fixed activity bounds of zero and one, by definition, prevent associativity of this significant iB2C meta information. With the introduction of the non-negative, upper unbound variance as the behaviors' novel meta information, the PBN architecture addresses both these known problems.

The following statement about computational PBN graphs can be derived from above Definitions 4.6 and 4.5:

Theorem 4.2: PBN Behavioral (Sub-) Network Calculation

Let $\mathcal{G} = (B, E)$ be a PBN graph defined on the set of data $\mathcal{X}$.

All subgraphs $\mathcal{G}_s \subseteq \mathcal{G}$ compute locally optimized mean output values according to their individually included behaviors.

implications

All PBN subgraphs $\mathcal{G}_s \subseteq \mathcal{G}$ satisfy the behavior calculation rule in Definition 4.6 relaxed to multi-result behaviors $\mathcal{B}^m$. Behavioral subgraphs can, therefore, be grouped and replaced by an equivalent single PBN sub-behavior $\mathcal{B}_s$.

$$\mathcal{G}_s(I(\mathcal{G}_s)) \rightarrow \mathcal{B}_s^{|\mathcal{O}(\mathcal{G}_s)|}(I(\mathcal{G}_s)) \tag{4.11}$$

Given a PBN graph $\mathcal{G} = (B, E)$ defined on a set of data $\mathcal{X}$. A PBN subgraph $\mathcal{G}_s \subseteq \mathcal{G}$ together with its data subset $\mathcal{X}_s \subseteq \mathcal{X}$ is entirely defined by the subgraph's subset of included behaviors. For every subset of behaviors $B_s \subseteq B$, it follows that

$$\begin{aligned} E_s &= \{(\mathcal{B}_{src},\ \mathcal{B}_{dest},\ X) \in E |\ \mathcal{B}_{src},\ \mathcal{B}_{dest} \in B_s\} && \subseteq E \\ \mathcal{X}_s &= \{X_i \in \mathcal{X} |\ \exists \mathcal{B} \in B_s, \text{ such that } \mathcal{B}(\ldots) = X_i \ \vee\ \mathcal{B}(\ldots,\ X_i,\ \ldots)\} && \subseteq \mathcal{X} \\ \text{and } \mathcal{G}_s &= (B_s,\ E_s) \text{ defined on } \mathcal{X}_s && (4.12) \end{aligned}$$

From Theorem 4.2, it follows that the entire PBN architecture inherits both commutativity and associativity from its behavior definition, i.e., Definition 4.6. With guaranteed commutativity and associativity, the PBN architecture allows function-invariant and position-invariant hierarchical control structuring, as required in Section 4.1.

For a behavior's meta data calculation, i.e., its variance, both properties are explicitly demanded to be preserved in Definition 4.6. In contrast, in the case of the primary data calculation, i.e., the mean, both those properties automatically emerge from the optimization-based return definition.

Commutativity and associativity inside a behavior are trivially given by the commutativity and associativity of the internal satisfaction product. However, a more detailed reasoning is required for the validity of both those properties among a network of multiple behaviors. Although the following argumentation does not provide an entirely valid proof of Theorem 4.2, it outlines a general sketch of such.

Argumentation on Theorem 4.2: *"PBN Behavioral (Sub-) Network Calculation"*

Assume a PBN with only two concatenated behaviors, each with only a single input. Let $t(x) = y$ and $u(y) = r$ be the two respective behavior task functions, and y be the data edge connecting both behavior nodes. Hence, the combined network task function $n(\cdot)$ is the composition of both behavior tasks $n(x) = (u \circ t)(x) = u(t(x)) = r$. It follows that the network itself optimizes the concatenated task inverse according to Definition 4.6.

$$\begin{aligned} & \arg\max_r \ \mathrm{sat}_x\left(n^{-1}(r)\right) && (4.13) \\ \Rightarrow \quad & \frac{\partial}{\partial r}\,\mathrm{sat}_x\left(n^{-1}(r)\right) && \stackrel{!}{=} 0 \\ \Leftrightarrow \quad & \frac{\partial \mathrm{sat}_x\left(n^{-1}(r)\right)}{\partial n^{-1}(r)}\ \frac{\partial t^{-1}\left(u^{-1}(r)\right)}{\partial u^{-1}(r)}\ \frac{\partial u^{-1}(r)}{\partial r} && \stackrel{!}{=} 0 \\ \Leftrightarrow \quad & \frac{\partial}{\partial x}\mathrm{sat}_x(x)\ \ \frac{\partial}{\partial y}t^{-1}(y)\ \ \frac{\partial}{\partial r}u^{-1}(r) && \stackrel{!}{=} 0 \qquad (4.14) \end{aligned}$$

From the definition of data satisfaction (see Definition 4.5), it can be assumed[5] that its derivative is non-zero unless the unique, perfect mean argument is given that results in a satisfaction value of one. Thus, Equation (4.14) is satisfied if either $\frac{\partial}{\partial y}t^{-1}(y) = 0$

[5] This assumption is valid for Cartesian data represented by regular multivariate normal distributions. In the case of orientation data represented as a Bingham distribution (see Section 4.4.2), further local extrema do exist [Caccavale 98b]. However, all but one of those local extrema are unstable and will, thus, not last in a noisy environment.

or $\frac{\partial}{\partial r} u^{-1}(r) = 0$. However, by analyzing the two behaviors individually, as they are computed in the PBN, it turns out that actually both conditions are met.

$$y = \arg\max_{y} \ \mathrm{sat}_x\left(t^{-1}(y)\right) \quad \Rightarrow \quad \frac{\partial}{\partial x}\mathrm{sat}_x(x) \ \frac{\partial}{\partial y} t^{-1}(y) = 0$$

$$\Rightarrow \quad \frac{\partial}{\partial y} t^{-1}(y) = 0 \tag{4.15}$$

$$r = \arg\max_{r} \ \mathrm{sat}_y\left(u^{-1}(r)\right) \quad \Rightarrow \quad \frac{\partial}{\partial y}\mathrm{sat}_y\left(y\right) \ \frac{\partial}{\partial r} u^{-1}(r) = 0$$

$$\Rightarrow \quad \frac{\partial}{\partial r} u^{-1}(r) = 0 \tag{4.16}$$

The fact that both criteria are met does not only satisfy the local stability criteria. It additionally engages a PBN's overall robustness against local behavior disturbances, e.g., caused by sensor noise, floating-point arithmetic, or damped optimization strategies.

Commutativity and associativity of a behavior network now directly follow from the trivially given commutativity and associativity of the network's equivalent single behavior, realizing the composed task function $n(x)$, as defined above.

4.3.2 Feedback, Loops, and Internal States

In the early design of the preliminary iB2C architecture, control oscillations have already been addressed as a potential serious risk to the overall system stability and performance. Especially in BB control setups, control decomposition is strongly motivated. Unfortunately, this motivation leads to systems that rapidly grow in the number of behaviors and their respective behavior connections, quickly exceeding manual overview capabilities. Thus, in [Proetzsch 10a, p. 51], Proetzsch lists oscillation analysis and detection techniques as one of the key design elements for BB robot control. However, after nearly a decade of experience and enhancements to the iB2C architecture, Ropertz still lists undesired control oscillation as one of the most significant safety issues that *"should be prevented"* [Ropertz 19, p. 114].

In addition to parasitic oscillations, inter-network data flow loops embed another severe stability issue. Since digital control networks are typically not treated as *pseudo-analog* circuits[6], every data flow loop must be broken up at some point to be sequentially executable by a standard CPU. At those break-up points, it is unavoidable to face unsynchronized, inconsistent, and delayed data. Unfortunately, data inconsistency and delay again cause a serious risk of oscillation and instability. Especially in highly dynamic setups, such as bipedal locomotion, these risks are significant and crucial.

[6] Analog circuit loops, as, e.g., well-known from standard amplifier circuits, automatically stabilize themselves at some electrical circuit fixpoint. Due to the analog nature of such an electrical circuit, this fixpoint iteration happens at infinite speed, thus offering an immediate stable signal. Respectively, a *pseudo-analog* circuit, herein, is meant as a digital loop performing an identical, entire digital fixpoint iteration (assuming a guaranteed eventual stop) before returning the stabilized result. One exemplary tool to describe such *pseudo-analog* behaviors is the programming and modeling language *Quartz* [Schneider 10, Schneider 17]. *Quartz* is specifically designed to compile or synthesize data flow programs.

The PBN architecture addresses both of the above issues by its definition. As initially defined, a PBN's data flow multigraph is explicitly required to feature an acyclic topology. Thus, data flow loops are, by definition, not allowed in a PBN control graph. Without such loops, network-internal control oscillations and inevitable data inconsistencies within circular dependencies must not occur in a PBN graph.

However, a versatile robot control architecture must not forbid internal states or feedback loops. To address these seemingly contradicting requirements, the PBN architecture, without loss of generality, treats all internal states and models as being part of the robot's PBN-external environment. Virtual environment and robot models are seen as being read by ideal virtual sensors and altered by ideal virtual actuators. Thus, all feedback loops, both intrinsic and extrinsic, leave and reenter the acyclic PBN control graph. The generalized robot control loop, as sketched in Figure 2.14 within Section 2.4, visualizes these feedback loops through arrows that return to the environment and the robot.

Defining feedback to occur solely outside the control architecture guarantees that the pure PBN control graph is stable. However, this does not prevent the overall system from becoming unstable or oscillating when closing the loop. This fact is well-known from classical systems theory when converting Single-Input Single-Output (SISO) systems from open-loop to closed-loop configuration. Section 3.2.6 has already addressed this same problem at the actuator level. The benefit of banning feedback from the PBN control graph lies in collecting and grouping stability-critical feedback loops in one place. Loop break-up is concretely defined to occur at precisely one known position in the data flow loop. Thus, it can be considered and addressed during the early design and development phase.

Stability verification may be achieved by ensuring that a model-affecting subnetwork implements a stable and finite fixpoint iteration w.r.t. the model itself. However, such analyses exceed the scope of this thesis and are, hence, left open to future work.

4.4 Unified Uncertain Data Representation

In the context of bipedal natural control, as declared in Theorem 1.1, only a limited set of individual data representations is relevant. This set includes all types required to represent a usual, legged, floating-base robot's state, merged with all types needed to encode the musculoskeletal hybrid impedance control commands developed in Chapter 3. Following the general Definition 4.4: *"PBN Data Representation"*, all those necessary concrete PBN data representations, together with their reasoning, are introduced in detail throughout the rest of this Section.

4.4.1 Cartesian Vectors

The probabilistic representation of arbitrary Cartesian vectors is the most simple and intuitive data representation. Nevertheless, dependent on how those data are intended to be used, different internal representations provide task-dependent computational and analytical benefits. Hence, the following notations are used:

Notation 4.7: Probabilistic Cartesian Data Vector

Cartesian data vectors $X \in \mathcal{X}$, representing a multivariate normal distribution $\mathcal{N}$, are represented in the following formats:

$$1. \quad {}^{\Sigma}X = {}^{\Sigma}\langle \mu, \Sigma \rangle \sim \mathcal{N}(\mu, \Sigma) \tag{4.17}$$

$$2. \quad X = \langle \mu, \Omega \rangle \sim \mathcal{N}(\mu, \Omega^{-1}) \tag{4.18}$$

$$3. \quad {}^{P}X = {}^{P}\langle p, \Omega \rangle \sim \mathcal{N}(\Omega^{-1}p, \Omega^{-1}) \tag{4.19}$$

$\Omega := \Sigma^{-1} \succeq 0$ represents a multivariate normal distribution's symmetric, positive semi-definite precision matrix. Note that the precision matrix may also be referred to as the so-called *concentration matrix* throughout the literature.

$p := \Omega\mu$ is herein named *precise mean*. Thus, the two notations ${}^{\Sigma}X$ and ${}^{P}X$ will be referenced as *sigma notation* and *precise notation*, respectively.

Following Definition 4.5, the satisfaction function of a Cartesian vector $X = \langle \mu, \Omega \rangle$ is the unscaled, multivariate normal distribution:

$$\text{sat}_{\langle \mu, \Omega \rangle}(u) := \exp\left(-\frac{1}{2}(u-\mu)^T \Omega (u-\mu)\right) \in (0, 1] \tag{4.20}$$

The satisfaction's derivative w.r.t. its argument u evaluates to:

$$\frac{\partial}{\partial u}\text{sat}_{\langle \mu, \Omega \rangle}(u) = -\text{sat}_{\langle \mu, \Omega \rangle}(u) \ (u-\mu)^T \Omega \tag{4.21}$$

Taking into account that $\text{sat}_{\langle \mu, \Omega \rangle}(u) \neq 0$, it follows that the derivative only becomes zero if $u = \mu$ (or $\Omega = 0$).

4.4.2 Rotor Orientations

In order to avoid non-continuous angle representations and the well-known *Gimbal lock* problem, all orientations should only be represented as so-called *unit rotors* $r \in \text{SO}(n)$. Thus, complex numbers $r \in \mathbb{C}$ should be used in the planar case, and quaternions $r \in \mathbb{H}$ should represent three-dimensional orientations. Generalized, rotors can be represented as special unit multivectors in any Projective Geometric Algebra (PGA) of arbitrary dimension $r \in \text{SO}(n) \subset \mathbb{R}_{n,0,1}$. Unfortunately, the normal distribution, defined on n-dimensional Euclidean spaces $\mathbb{R}^n$, does not offer a valid semantic interpretation of the set of $(n+1)$-dimensional unit rotors.

Hence, in this special case of periodic orientation representations, the Bingham distribution on unit spheres is used instead of the Cartesian normal distribution [Bingham 74]. Against alternative distributions, such as the Mises-Fisher distribution [Khatri 77], the Bingham distribution has been considered most profitable for the herein intended purpose [Schaeben 92]. This evaluation is furthermore backed by the studies of Caccavale et al. back in 1998, where the authors rely on the Bingham distribution to generate continuous, lock-free rotational impedance for robot arms [Caccavale 98a, Caccavale 98b, Caccavale 99]. The close similarity between the Bingham

distribution and the normal distribution will later be derived for the planar case and is shown at the hand of an Example in Figure 4.3.

In order to achieve notation uniformity, the herein-used notations for probabilistic orientation representations are similar to the Cartesian ones. They are:

Notation 4.8: Probabilistic Rotor Representations

Unit rotors $R \in \mathcal{X}$, representing a Bingham distribution $\mathcal{BN}$, are represented in the following formats:

$$1. \quad {}^{\Sigma}R = {}^{\Sigma}\langle r,\, 4\Sigma\rangle \sim \mathcal{BN}\left(r,\, -2\Sigma^{-1}\right) \tag{4.22}$$

$$2. \quad R = \left\langle r,\, \tfrac{1}{4}\Omega\right\rangle \sim \mathcal{BN}\left(r,\, -\tfrac{1}{2}\Omega\right) \tag{4.23}$$

Where $\frac{1}{2}\Omega := 2\Sigma^{-1} \geq 0$ represents the $(n+1)$-dimensional, symmetric, positive semi-definite precision matrix of the Bingham distribution[a]. The distribution's mean unit rotor $r \in \mathrm{SO}(n)$ is an $(n+1)$-dimensional unit vector $\|r\| = 1$.

[a] In contrast to the normal distribution, the standard Bingham distribution description $\mathcal{B}$ usually contains its negative characteristic precision matrix, also called the *concentration matrix*, instead of its variance. The Bingham distribution $\mathcal{BN}(0,\, A)$ respectively describes a *"normal distribution with zero mean and covariance matrix* $-\frac{1}{2}A^{-1}$*"* [Bingham 74, p. 1202].

Following Definition 4.5, the respective orientation satisfaction function of a unit rotor $R = \left\langle r,\, \frac{1}{4}\Omega\right\rangle$ is the unscaled Bingham distribution with Geometric Algebra (GA)-compliant, multiplicative mean deviation. Due to notation uniformity, the orientation satisfaction is written in a format similar to the prior Cartesian satisfaction definition:

$$\mathrm{sat}_{\left\langle r,\, \frac{1}{4}\Omega\right\rangle}(u) := \exp\left(-\frac{1}{2}\, u^T\, \mathrm{L}[\tilde{r}]^T\, \Omega \mathrm{L}[\tilde{r}]\; u\right) \in (0, 1] \tag{4.24}$$

With $\tilde{r}$ being the rotor's complex conjugate, i.e., its reverse rotation r^{-1}. The operator $\mathrm{L}[\tilde{r}]$ describes the Cartesian left multiplication matrix equivalent of a rotor's left multiplication with $\tilde{r}$ within their respective GA. Thus, the expression $(\mathrm{L}[\tilde{r}]\, u)$ implements the rotational equivalent of the Cartesian mean deviation expression $(u - \mu)$. Note that $\mathrm{L}[\cdot]$ is not a Cartesian rotation matrix as it rotates a rotor, not a Cartesian vector.

Consequently, the satisfaction's derivative w.r.t. its argument u evaluates to:

$$\frac{\partial}{\partial u}\mathrm{sat}_{\left\langle r,\, \frac{1}{4}\Omega\right\rangle}(u) = -\mathrm{sat}_{\left\langle r,\, \frac{1}{4}\Omega\right\rangle}(u)\; u^T \left(\mathrm{L}[\tilde{r}]^T\, \Omega \mathrm{L}[\tilde{r}]\right) \tag{4.25}$$

Note that both the mean rotor and its respective precision matrix feature one spare dimension w.r.t. their represented orientation entity. This extra dimension, however, must not carry any further, non-existing information. In fact, the additional element in the unit rotor compensates for the information loss due to a rotor's invariance against negation, i.e., $r = -r$. The distribution's precision matrix describes the probability of the rotor's deviation from its mean, i.e., its first derivative. However, the derivative lies within the rotor unit sphere's tangent plane located at the rotor's point on the sphere. The

rotor unit vector itself embodies the plane's negation-invariant normal vector. Hence, the distribution's tangent precision plane features one dimension less than the rotor's unit sphere and Cartesian precision matrix representation. Thus, the spare precision matrix dimension must be non-linearly dependent on the others, according to the non-linear unit length constraint of the rotor description vectors.

A unit rotor r is constructed as a composition of a scalar s and a multivector ϵ that's basis vectors all square to -1. Herein, s is assumed to be the first element in r.

$$r = \left[s = \cos\left(\tfrac{\theta}{2}\right) \ , \ \epsilon = \sin\left(\tfrac{\theta}{2}\right) e \right]^T \tag{4.26}$$

Within this definition, θ is the angle of rotation, and e is the unit multivector in an arbitrary dimensioned PGA describing the generalized normal to the plane in which the rotation is happening.

- In 2-D : e becomes the space itself and vanishes to the unit pseudoscalar.
- In 3-D : e becomes the intuitive *axis of rotation*.
- In $\geq$4-D : e becomes a plane or a hyperplane, still embodying the respective higher dimensional normal of the always two-dimensional plane of rotation.

Independent of the dimension, the scalar s in a rotor is only necessary to define the rotation direction. Since the desired probability distribution is symmetric in its mean, this element can be dropped for evaluating the PDF without loss of relevant information. Consequently, the valid orientation distribution can be constructed from a precision matrix $\Omega_\star$ of one lower dimensionality than its respective rotor. In order to rematch the rotor's dimensionality, the reduced precision matrix is filled with zeros so that the rotor's scalar element is canceled out in the satisfaction equation.

Definition 4.9: Construction of a Probabilistic Rotor Representation

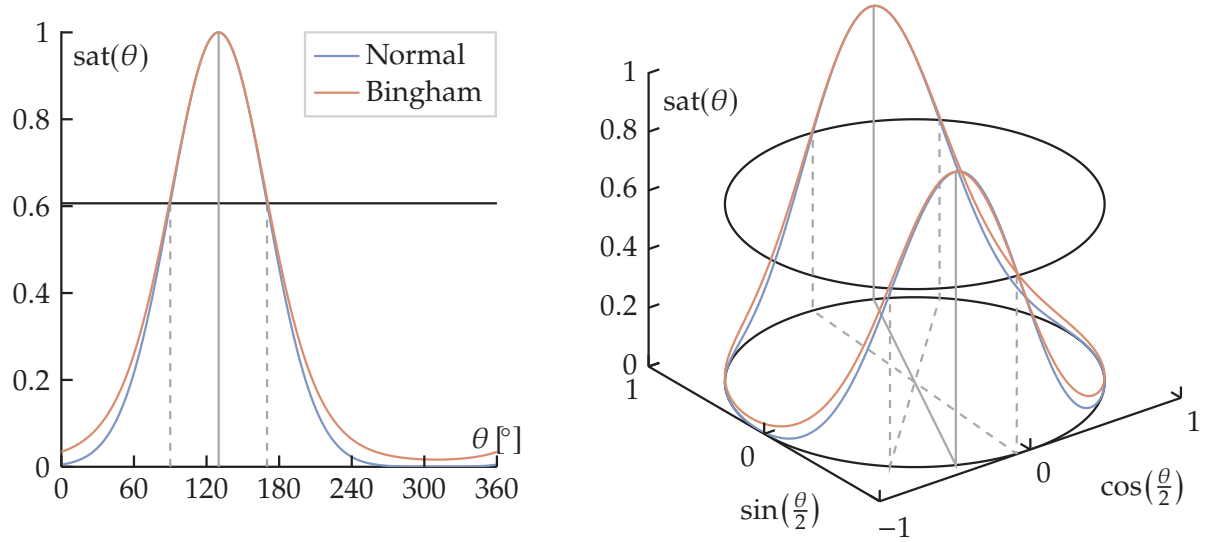
Following Notation 4.8, a probabilistic unit rotor R may be constructed in the following formats:

$$1. \quad {}^{\Sigma}R = {}^{\Sigma}\langle r, 4\Sigma_\star \rangle \sim \mathcal{BN}\left(r, -2\begin{bmatrix} 0 & \cdots \\ \vdots & \Sigma_\star^{-1} \end{bmatrix}\right) \approx \mathcal{N}\left(r, \begin{bmatrix} 0 & \cdots \\ \vdots & \Sigma_\star \end{bmatrix}\right) \tag{4.27}$$

$$2. \quad R = \left\langle r, \tfrac{1}{4}\Omega_\star \right\rangle \sim \mathcal{BN}\left(r, -\frac{1}{2}\begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star \end{bmatrix}\right) \approx \mathcal{N}\left(r, \begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star^{-1} \end{bmatrix}\right) \tag{4.28}$$

Where $\Omega_\star := \Sigma_\star^{-1} \geq 0$ represents an n-dimensional, symmetric, positive semi-definite precision matrix according to the mean rotor $r \in \mathrm{SO}(n)$ represented as $(n+1)$-dimensional unit vector in Cartesian space $\|r\| = 1$. Without loss of generality, the scalar element of the rotor r is assumed to be the rotor's top-most element in its vector representation.

Since the directional information of the rotor must not be dropped in general, the above definition does not stay valid throughout calculations. Thus, it may only be



(a) Satisfaction in the Cartesian Subspace

(b) Satisfaction in the 2D-Rotor Unit Circle

Figure 4.3: Comparison of the Cartesian satisfaction based on the normal distribution in blue against the analog orientation satisfaction based on the Bingham distribution in red. Both graphs are based on a single angle distribution with its mean at $\mu = 130°$ and a standard deviation of $\sigma = 40°$, both highlighted by grey lines in the two plots. To the left, Plot (a) shows the two respective satisfaction functions w.r.t. the angular value of one full rotation. Plot (b) on the right visualizes the identical satisfaction functions w.r.t. a two-dimensional rotor, indicated by the bottom unit circle. It can be seen that the Bingham distribution mostly coincides with the normal distribution, especially in the most significant area around the mean value.

used for initial representation construction. Furthermore, due to the loss of directional information in the rotor, a precise notation, as described in the Cartesian Notation 4.7, must not be used for orientation data.

Following the above construction Definition 4.9, the Bingham distribution's mean-compensated precision matrix is given by the following equation:

$$A = \left(\mathrm{L}[\tilde{r}]^T \begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star \end{bmatrix} \mathrm{L}[\tilde{r}] \right) \tag{4.29}$$

It can be seen that A is not of full rank even if $\Omega_\star$ is so. Let now $A =: Q\Lambda Q^T$ be its SVD, and the unit vector $u =: Qv$ be an arbitrary rotor argument of the satisfaction derivative, as given in Equation 4.25. It follows that $u^T A = v^T \Lambda Q^T$. Since A is not of full rank, at least one eigenvalue of Λ must be zero. When multiplied with Q^T, at least one entire row of Q^T is canceled out, and the resulting column vectors of ΛQ^T are linearly dependent. Thus, if Ω is strictly positive-definite, there exists precisely the number of rotor dimensions sign-invariant solution rotor pairs, such that the orientation satisfaction derivative becomes zero. In fact, these solutions correspond to the desired stable rotor $u = r$ and one unstable rotor at exactly 180° rotation from r in all spatial dimensions [Caccavale 98b].

Figure 4.3 compares a Cartesian and a respective orientation satisfaction function at the hand of a scalar angle example. The angle distributions plotted in the figure have a mean value of $\mu = 130°$ and a standard deviation of $\sigma = 40°$. To make the Cartesian normal distribution periodic in one full rotation, it is wrapped around itself. This wrapping means that for every input angle θ, not only the satisfaction at the angle θ is calculated, but all satisfactions of all (infinitely many) θ-equivalent angles are summed. Afterward, the sum is rescaled to return the maximum value one at all the PDF modes.

$$\mathrm{sat}^{\circ}_{\Sigma\langle \mu,\, \sigma^2 \rangle}(\theta) := \frac{\sum_{i\in\mathbb{Z}} \mathrm{sat}_{\Sigma\langle \mu,\, \sigma^2 \rangle}(\theta + i\ 360°)}{\sum_{i\in\mathbb{Z}} \mathrm{sat}_{\Sigma\langle \mu,\, \sigma^2 \rangle}\big(\mu + i\ 360°\big)} \tag{4.30}$$

Since the satisfaction function is a scaled PDF, its entire area under its curve is finite. Consequently, the infinite sums over all integer values within Equation (4.30) must be finite, too. Thus, the periodic satisfaction function $\mathrm{sat}^{\circ}_{\Sigma\langle \mu,\, \sigma^2 \rangle}(\theta)$ is well-defined.

In the planar case, which is the only relevant for the herein-discussed control of the robot Carl[7] [Schütz 17][Schütz 20a], the Bingham distribution can be derived from a Cartesian normal distribution, thus highlighting their close similarity.

1. Let $X = \big\langle \theta,\ \frac{1}{4}\Omega_\star \big\rangle$ be a scalar normal distribution around the mean angle θ with the scalar precision value of $\frac{1}{4}\Omega_\star$.

2. Let furthermore $f_\pm(\theta) := \pm \Big[\ \cos\!\Big(\frac{\theta}{2}\Big),\ \sin\!\Big(\frac{\theta}{2}\Big) 1\ \Big]^T =: \pm r$ be the two functions that construct the two equivalent planar rotor representations $\pm r$ of the angle θ, as defined in Equation 4.26.

   The two function derivatives are: $f'_\pm(\theta) = \frac{d}{d\theta} f_\pm(\theta) = \pm\frac{1}{2}\Big[\ -\sin\!\Big(\frac{\theta}{2}\Big),\ \cos\!\Big(\frac{\theta}{2}\Big)\ \Big]^T$ with $f'_\pm(0) = \pm\frac{1}{2}\big[0 \quad 1\big]^T$ being the functions' linearizations at their respective distributions' modes, i.e., their most-significant values, their means.

3. Now, X can be transformed into a new normal distribution by defining its variance to be transformed via the above-defined derivative linearization $f'_\pm(0)$

$$\begin{aligned} f_\pm\big(X = \big\langle \theta,\ \tfrac{1}{4}\Omega_\star \big\rangle\big) &= {}^{\Sigma}\big\langle f_\pm(\theta)\,,\ f'_\pm(0)\, 4\Omega_\star^{-1}\, f'_\pm(0)^T \big\rangle \\ &= {}^{\Sigma}\Big\langle \pm r,\ \pm\big[0 \quad 1\big]^T \Omega_\star^{-1} \pm \big[0 \quad 1\big] \Big\rangle \\ &= \Big\langle \pm r,\ \begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star \end{bmatrix} \Big\rangle \end{aligned} \tag{4.31}$$

   Note that the rotor r is sign-invariant and that the individual $\pm$ signs in the distribution's variance cancel out each other, as required to preserve positive semi-definiteness. Thus, both possible transformation functions, f_+ and f_-, construct equivalent rotor distributions.

4. The equivalence of the above still-Cartesian normal distribution with its respective Bingham distribution follows from their equivalent satisfaction functions. This equivalence is achieved by replacing the Cartesian mean deviation calculation with its multiplicative rotor equivalent, i.e., $\big(u - \mu\big) \mapsto \big(\mathrm{L}[\tilde{r}]\, u\big)$.

[7] Carl (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU. URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025)

As shown in Figure 4.3, the Bingham distribution-based satisfaction tracks the periodic normal distribution-based satisfaction very well. Moreover, both functions can be considered identical in the most significant area around their equal mean value. Notable deviations occur at a 180° phase shift from the mean. This observation is explainable since Equation (4.31) within the above derivation of the rotor satisfaction function uses the transformation function's first derivative, linearized at the mean.

4.4.3 Cartesian Symmetric Matrices

When applying the musculoskeletal, hybrid impedance control approach introduced in Chapter 3, damping becomes a control property. However, unlike Cartesian properties such as force, torque, or velocity, damping represents a mapping between those properties. As pictured in Figure 3.17 within Section 3.3.1, this (linear or linearized) mapping can be represented by a symmetric, positive semi-definite matrix. Thus, PBNs are required to represent such matrices. In the following, the notation and encoding of general symmetric matrices is introduced. Nevertheless, the optional property of positive semi-definiteness, or even strict positive definiteness, is preserved within all operations without additional notice.

Since matrices are defined within Cartesian spaces, a matrix normal distribution exists and is defined as $\mathcal{MN}(M,\ \Sigma,\ V)$, with M being the distribution's mean matrix and Σ, V being the variance matrices of the rows and columns of M, respectively. Its respective satisfaction function derives to:

$$\begin{aligned}\mathrm{sat}_{\mathcal{MN}(M,\ \Sigma,\ V)}(U) \ &:= \exp\left(-\frac{1}{2}\mathrm{tr}\left(V^{-1}\,(U-M)^T\,\Sigma^{-1}\,(U-M)\right)\right) \\ &= \exp\left(-\frac{1}{2}\mathrm{vec}(U-M)^T\,(V\otimes\Sigma)^{-1}\,\mathrm{vec}(U-M)\right) \in (0,1] \end{aligned} \tag{4.32}$$

Equation (4.32) shows that the multivariate normal distribution on vectors is a special case of the more general matrix normal distribution. With M being a 1-column matrix and V becoming a scalar value that can, hence, be integrated into Σ, the general matrix normal distribution degenerates to its vector equivalent. However, both matrices and vectors are multidimensional notations that follow identical calculation rules. Therefore, matrix normal distributions can also be rewritten into equivalent, multivariate vector normal distributions [Soch 22].

$$\mathcal{MN}(M,\ \Sigma,\ V) \quad \Leftrightarrow \quad \mathcal{N}(\mathrm{vec}(M)\,,\ V\otimes\Sigma) \tag{4.33}$$

The equivalence of the respective satisfaction functions is given by the latter expression in Equation (4.32). To preserve representation simplicity, probabilistic matrix data should be represented as their equivalent vectorization.

In the special case of representing symmetric matrices, as herein desired, the row variance must equal the column variance $V = \Sigma$. Furthermore, in order to achieve equivalence with the vector version in the special one-dimensional, scalar case, both row and column variances have to be the square root of the overall matrix variance.

$$\mathcal{MN}\left(M,\ \sqrt{\Sigma},\ \sqrt{\Sigma}\right) \quad \Leftrightarrow \quad \mathcal{N}\left(\mathrm{vec}(M)\,,\ \sqrt{\Sigma}\otimes\sqrt{\Sigma}\right) \tag{4.34}$$

Since Σ is symmetric, its square root is well-defined and unique. Let $\Sigma =: Q\Lambda Q^T$ be the SVD of the variance matrix. Its unique square root evaluates to $\sqrt{\Sigma} = Q\sqrt{\Lambda}Q^T$.

Definition 4.6 defines a PBN behavior to determine its result mean vector by solving an optimization problem. In the special case of this mean vector being a symmetric matrix vectorization, the optimization semantic must be reinterpreted to its equivalent, non-vectorized matrix semantic:

Let $M \in \mathcal{R}^{n \times n}$ be an arbitrary, symmetric matrix of dimension $\dim(M) = n$ and vectorization $\text{vec}(M)$. Let furthermore $\sqrt{\Omega}$ be a symmetric positive semi-definite weighting matrix of identical size.

Minimizing the weighted square error of the matrix vectorization $\text{vec}(M)$ using the weight $\sqrt{\Omega} \otimes \sqrt{\Omega}$ is equal to the minimization of the integral of the weighted square error over all existing vectors right-multiplied to the matrix M sandwiched by the weight matrix $\sqrt{\Omega}$.

$$\begin{aligned}
& \arg\min_X \ \frac{1}{2} \int_{q \in \mathcal{R}^n} \left(\sqrt[4]{\Omega}\,(X - M)\,\sqrt[4]{\Omega}\,q \right)^2 && (4.35) \\
\Leftrightarrow \quad & \arg\min_X \ \frac{1}{2} \operatorname{tr}\left(\left(\sqrt[4]{\Omega}\,(X - M)\,\sqrt[4]{\Omega} \right)^T \left(\sqrt[4]{\Omega}\,(X - M)\,\sqrt[4]{\Omega} \right) \right) \\
\Leftrightarrow \quad & \arg\min_X \ \frac{1}{2} \operatorname{tr}\left(\left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right)^T \left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right) \right) - \operatorname{tr}\left(\left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right)^T \left(\sqrt[4]{\Omega} M \sqrt[4]{\Omega} \right) \right) \\
\Leftrightarrow \quad & \arg\min_X \ \frac{1}{2} \operatorname{vec}\left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right)^T \operatorname{vec}\left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right) - \operatorname{vec}\left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right)^T \operatorname{vec}\left(\sqrt[4]{\Omega} M \sqrt[4]{\Omega} \right) \\
\Leftrightarrow \quad & \arg\min_X \ \frac{1}{2} \left(\operatorname{vec}\left(\sqrt[4]{\Omega} X \sqrt[4]{\Omega} \right) - \operatorname{vec}\left(\sqrt[4]{\Omega} M \sqrt[4]{\Omega} \right) \right)^2 \\
\Leftrightarrow \quad & \arg\min_X \ \frac{1}{2} \left(\sqrt{\left(\sqrt{\Omega} \otimes \sqrt{\Omega} \right)} \operatorname{vec}(X - M) \right)^2 \\
\Leftrightarrow \quad & \arg\min_X \ \frac{1}{2} \operatorname{vec}(X - M)^T \left(\sqrt{\Omega} \otimes \sqrt{\Omega} \right) \operatorname{vec}(X - M) && (4.36)
\end{aligned}$$

Note that the weight inside the squared error term is applied in its square root, e.g., $\sqrt{\sqrt{\Omega}} = \sqrt[4]{\Omega}$. Reducing the integral over all vectors $q \in \mathcal{R}^n$ to the trace of the squared inner matrix in the first reformulation is argued as follows:

1. The inner matrix is diagonalized to $Q\Lambda Q^T$ using the SVD.
2. Since the outer integral includes all existing vectors q, the bijective Q-transformation vanishes inside the integration vector q.
3. What is left is an integration over the squared diagonal matrix of eigenvalues $\int_{q \in \mathcal{R}^n} q\Lambda^2 q$.
4. Furthermore, all those vectors q can be written as a linear combination of the eigenvectors of Λ^2, thus simplifying the whole integral to the sum of squared eigenvalues $\|\Lambda\|^2$.

5. The equality with the trace of the squared matrix directly arises from the property of the trace being equal to the sum of eigenvalues.

Due to the optimization's invariance against constant scaling, all scalar factors are neglected throughout the above argumentation.

Due to the matrix symmetry, non-diagonal elements occur twice in the matrix and its respective vectorization. Hence, nearly half of the elements can be removed without loss of information. In detail, the number of elements in the vectorization can be reduced from n^2 to $\frac{1}{2}n(n+1)$. Accordingly, the number of elements in the associate variance matrix can be reduced by the square of this vector reduction. Roughly estimated, the actual variance information only requires about a quarter of the entire Kronecker product. Such a reduction can be achieved by applying a special, linear, projective transformation to the distribution. Since normal distributions are invariant against linear transformations, doing so will not affect its semantics.

A *kernel* matrix K is constructed to linearly map a reduced vector representation of a symmetric matrix to its entire, full-dimensional matrix vectorization. Elements representing diagonal matrix entries are copied unchanged. Off-diagonal elements are scaled by a factor of $\sqrt{0.5}$. Thus, the kernel matrix K_2 for a vectorized two-dimensional symmetric matrix is given as follows:

$$K_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{0.5} & 0 \\ 0 & \sqrt{0.5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.37}$$

By using this concrete kernel construction, the kernel's left inverse becomes its transpose, i.e., $K^T\ K = I$. This property is crucial to preserve variance symmetry, as required by the normal distribution definition. Multiplying K^T to the right of K, however, only results in an identity mapping of vectorizations of symmetric matrices. In fact, multiplication with KK^T returns the symmetric part of a matrix. Thus, when being applied to symmetric matrix vectorizations, K^T can be treated as the general inverse, i.e., $K^T = K^{-1}$.

$$\begin{aligned} KK^T \mathrm{vec}(M) &= \mathrm{vec}\left(\tfrac{1}{2}(M + M^T)\right) \\ &= \mathrm{vec}(M) \qquad : \text{if } M = M^T \end{aligned} \tag{4.38}$$

In the exemplary case of the K_2 kernel matrix, the left and right transpose products become the following square matrices:

$$K_2^T K_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{and} \qquad K_2 K_2^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.39}$$

Following the linear transformation rules for multivariate normal distributions, the following two encoding functions can be defined:

$$\text{Mean Encoding:} \qquad \mu(S) = K^T \mathrm{vec}(S) \tag{4.40}$$

$$\text{Precision Encoding:} \qquad \omega(\Omega) = K^T \left(\sqrt{\Omega} \otimes \sqrt{\Omega}\right) K \tag{4.41}$$

Due to the definition of K and $\omega(\cdot)$, the following important properties hold:

$$\omega(\Omega)^{-1} = \left(K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)K\right)^{-1} = K^T\left(\sqrt{\Omega^{-1}}\otimes\sqrt{\Omega^{-1}}\right)K = \omega\left(\Omega^{-1}\right) \tag{4.42}$$

and:

$$\begin{aligned}
\exp(\omega(\Omega)) &= \exp\left(K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)K\right) \\
&= \frac{K^TK}{0!} + \frac{K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)K}{1!} + \frac{K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)KK^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)K}{2!} + \dots \\
&= K^T\left(\frac{I}{0!} + \frac{\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)}{1!} + \frac{\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)}{2!} + \dots\right)K \\
&= K^T\exp\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)K \qquad (4.43) \\
\text{but:} \quad &\neq \omega\left(\exp(\Omega)\right) \qquad (4.44)
\end{aligned}$$

With the above encoding functions (4.40) and (4.41), the following notations for probabilistic symmetric matrices can be defined:

Notation 4.10: Probabilistic Symmetric Matrix Data

Symmetric matrices $X \in \mathcal{X}$, representing a matrix normal distribution $\mathcal{MN}$, are represented by their equivalent, reduced Cartesian vectorizations in the following formats:

$$\begin{aligned}
1.\quad X &= \langle S,\ \Omega\rangle \sim \mathcal{MN}\left(S,\ \sqrt{\Omega^{-1}},\ \sqrt{\Omega^{-1}}\right) \\
&= \left\langle \mu(S),\ \omega(\Omega)\right\rangle \sim \mathcal{N}\left(\mu(S),\ \omega\left(\Omega^{-1}\right)\right) \qquad (4.45) \\
2.\quad {}^{P}X &= {}^{P}\langle P,\ \Omega\rangle \sim \mathcal{MN}\left(\sqrt{\Omega^{-1}}P\sqrt{\Omega^{-1}},\ \sqrt{\Omega^{-1}},\ \sqrt{\Omega^{-1}}\right) \\
&= {}^{P}\left\langle \mu(P),\ \omega(\Omega)\right\rangle \sim \mathcal{N}\left(\omega\left(\Omega^{-1}\right)\mu(P),\ \omega\left(\Omega^{-1}\right)\right) \qquad (4.46)
\end{aligned}$$

$\Omega := \Sigma^{-1} \succeq 0$ represents a symmetric matrix normal distribution's symmetric, positive semi-definite *precision* matrix.

$P := \sqrt{\Omega}S\sqrt{\Omega}$ is herein named *precise mean matrix*, analog to the Cartesian vector's precise notation introduced in Notation 4.7.

Note that the last notation equality holds due to the following equality:

$$\begin{aligned}
\mu(P) &= \mu\left(\sqrt{\Omega}S\sqrt{\Omega}\right) &&= K^T\mathrm{vec}\left(\sqrt{\Omega}S\sqrt{\Omega}\right) \\
&= K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)\mathrm{vec}(S) &&= K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)KK^T\mathrm{vec}(S) \\
&= \omega(\Omega)\,\mu(S) && \qquad (4.47)
\end{aligned}$$

The kernel matrix K is highly sparse, containing only a single non-zero element per row. Thus, efficient implementations of the matrix mean encoding function $\mu(\cdot)$ and its

inverse $\mu^{-1}(\cdot)$ can be formulated by exploiting the kernel's sparsity. Size-independent implementations for both are proposed in Algorithms 4.1 and 4.2.

Algorithm 4.1: Matrix Encoding: $\mu(S) = K^T \text{vec}(S) =: v$ – Equation (4.40)

Data: A quadratic matrix S.
Let $n = \dim(S)$.
Initialize the result vector v with $\frac{1}{2}n\,(n+1)$ elements.
for $col = 0$ ***to*** $(n-1)$ **do**
  for $row = 0$ ***to*** $(n-1)$ **do**
    /* Calculate the lower-left element equivalent */
    $col_{ll} = \min\{col, row\}$
    $row_{ll} = \max\{col, row\}$
    /* Calculate the index in the result vector */
    $idx = \frac{1}{2}col_{ll}\,(2n - 1 - col_{ll}) + row_{ll}$
    if $row \geq col$ **then**
      $v[idx] = S[row, col]$ /* Copy lower-left elements */
    else
      $v[idx] = \sqrt{0.5}\,(v[idx] + S[row, col])$ /* Apply kernel */
    end
  end
end
return v

Algorithm 4.2: Matrix Decoding: $\mu^{-1}(v) = \text{vec}^{-1}(Kv) =: S$ – Encoding Inverse

Data: An encoded matrix vectorization v, according to Algorithm 4.1.
Let $n = \dim(S)$.
Initialize the result matrix $S \in \mathcal{R}^{n \times n}$.
for $col = 0$ ***to*** $(n-1)$ **do**
  /* Calculate the diagonal index in the input vector */
  $idx = \frac{1}{2}col\,(2n + 1 - col)$
  $S[col, col] = v[idx]$ /* copy diagonal elements */
  for $row = (col+1)$ ***to*** $(n-1)$ **do**
    /* Calculate the off-diagonal index in the input vector */
    $idx = \frac{1}{2}col\,(2n - 1 - col) + row$
    $S[row, col] = \sqrt{0.5}\,v[idx]$ /* Apply lower triangular kernel */
    $S[col, row] = \sqrt{0.5}\,v[idx]$ /* Apply upper triangular kernel */
  end
end
return S

A similar argumentation can be given for the matrix precision encoding function, as defined in Equation (4.41). Furthermore, the encoding itself should be implemented in such a manner that its result symmetry is guaranteed. Hence, the following mathematical

reformulation is proposed for implementation. Let $Q\Lambda Q^T := \Omega$ be the SVD of Ω and, hence, $\sqrt{\Omega} = Q\sqrt{\Lambda}Q^T$. It follows that:

$$\begin{aligned}\omega(\Omega) &= K^T\left(Q\sqrt{\Lambda}Q^T \otimes Q\sqrt{\Lambda}Q^T\right)K \\ &= K^T(Q\otimes Q)\left(\sqrt{\Lambda}\otimes\sqrt{\Lambda}\right)(Q\otimes Q)^T K \\ &= K^\star(Q)\left(\sqrt{\Lambda}\otimes\sqrt{\Lambda}\right)K^\star(Q)^T \end{aligned} \tag{4.48}$$

using:

$$K^\star(Q) := \left[\mu\left(\mathrm{col}_1(Q)\,\mathrm{col}_1(Q)^T\right),\ \mu\left(\mathrm{col}_1(Q)\,\mathrm{col}_2(Q)^T\right),\ \cdots,\ \mu\left(\mathrm{col}_n(Q)\,\mathrm{col}_n(Q)^T\right)\right]$$

Note that the inner matrix $\left(\sqrt{\Lambda}\otimes\sqrt{\Lambda}\right)$ is pure diagonal and that the mean encoding $\mu\left(\mathrm{col}_i(Q)\,\mathrm{col}_j(Q)^T\right)$ inside $K^\star(Q)$ takes non-symmetric input matrices for $i \neq j$.

The inverse of the precision encoding function $\omega^{-1}(V) = \Omega$ can be achieved by multiplication with the constant, vector-encoded identity matrix $\mu(I)$, followed by the mean encoding inverse.

$$\omega^{-1}(V) = \Omega = \mu^{-1}\left(V\mu(I)\right) \tag{4.49}$$

The proof of the above identity is given by a the following derivation.

$$\begin{aligned}\mu^{-1}\left(V\mu(I)\right) &= \mu^{-1}\left(\omega(\Omega)\,\mu(I)\right) \\ &= \mu^{-1}\left(K^T\left(\sqrt{\Omega}\otimes\sqrt{\Omega}\right)KK^T\mathrm{vec}(I)\right) \\ &= \mu^{-1}\left(K^T\mathrm{vec}\left(\sqrt{\Omega}I\sqrt{\Omega}\right)\right) \\ &= \mu^{-1}\left(\mu(\Omega)\right) \\ &= \Omega \end{aligned} \tag{4.50}$$

4.4.4 Combined Data Representation

While processing probabilistic data in a PBN, the previously introduced pure data representations are likely to be required to be merged. As long as those merged types stay stochastically independent, i.e., feature block-diagonal variance or precision matrices, they can be handled independently. However, when off-diagonal (non-block-diagonal) elements occur, the respective satisfaction function and derivative become more complex. Since Cartesian matrices are represented as Cartesian vectors, the only interesting case is a combination of an orientation rotor with a Cartesian vector.

For the special planar case, Gilitschenski et al. address this exact same problem in their work [Gilitschenski 14], using reduced dual quaternions, i.e., dual complex numbers, as combined representation for planar rotation and translation. Although dual complex numbers, analog to rotations, store only half of their Cartesian translation representation internally, the approach by Gilitschenski et al. is directly generalizable to the dimensionally unrestricted case presented in the following. However, it might be

beneficial for future work to investigate possible data unification by representing all the necessary data types in some suitably dimensioned, uniform GA multivector.

In the beforehand-introduced notation scheme, combined probabilistic data are represented in the following notation:

Notation 4.11: Combined Probabilistic Data Representation

Let $C := \begin{bmatrix} R^T & X^T \end{bmatrix}^T \in \mathcal{X}$ be a stacked, probabilistic data vector containing a rotor R on top of a Cartesian vector X.

It is represented in the following format:

$$C = \langle c,\ \Omega_c \rangle = \left\langle \begin{bmatrix} r \\ \mu \end{bmatrix},\ \begin{bmatrix} \frac{1}{4}\Omega_r & \frac{1}{2}\Omega_s^T \\ \frac{1}{2}\Omega_s & \Omega_\mu \end{bmatrix} \right\rangle \tag{4.51}$$

Note that $\frac{1}{4}\Omega_r$ and Ω_μ are the symmetric, positive semi-definite, pure orientational, and Cartesian parts of Ω_c, respectively. However, the off-diagonal matrix Ω_s is an arbitrary, possibly non-square matrix, such that the overall precision matrix is again symmetric, positive semi-definite $\Omega_c \succeq 0$, as required. Using the two satisfaction Equations (4.24) and (4.20) for the orientation and Cartesian vector parts, respectively, the following combined satisfaction function can be formulated:

$$\text{sat}_{\left\langle \begin{bmatrix} r \\ \mu \end{bmatrix},\ \begin{bmatrix} \frac{1}{4}\Omega_r & \frac{1}{2}\Omega_s^T \\ \frac{1}{2}\Omega_s & \Omega_\mu \end{bmatrix} \right\rangle} \left(\begin{bmatrix} u \\ v \end{bmatrix} \right) := \exp\left(-\frac{1}{2} \begin{bmatrix} \mathrm{L}[\tilde{r}]\, u \\ v - \mu \end{bmatrix}^T \begin{bmatrix} \Omega_r & \Omega_s^T \\ \Omega_s & \Omega_\mu \end{bmatrix} \begin{bmatrix} \mathrm{L}[\tilde{r}]\, u \\ v - \mu \end{bmatrix} \right) \tag{4.52}$$

In order to simplify Equation (4.52), the general distance function Δ is defined.

$$\Delta_c \ :=\ \Delta_c(x) \ =\ \begin{cases} \mathrm{L}[\tilde{c}]\, x\ , & \text{for } c \in \mathrm{SO}(n) \ \text{ being a rotor} \\ x - c\ , & \text{for } c \in \mathbb{R}^n \ \text{ being a Cartesian vector} \end{cases} \tag{4.53}$$

It follows that the exponent in Equation (4.52) can be rewritten to:

$$\begin{aligned} & -\frac{1}{2} \begin{bmatrix} \Delta_r \\ \Delta_\mu \end{bmatrix}^T \begin{bmatrix} \Omega_r & \Omega_s^T \\ \Omega_s & \Omega_\mu \end{bmatrix} \begin{bmatrix} \Delta_r \\ \Delta_\mu \end{bmatrix} \\ =\ & -\frac{1}{2}\, \Delta_r^T \Omega_r \Delta_r \ - \Delta_\mu^T \Omega_s \Delta_r \ - \frac{1}{2}\, \Delta_\mu^T \Omega_\mu \Delta_\mu \\ =\ & -\frac{1}{2}\, \Delta_r^T \left(\Omega_r - \Omega_s^T \Omega_\mu^{-1} \Omega_s \right) \Delta_r \ - \frac{1}{2} \left(\Delta_\mu + \Omega_\mu^{-1} \Omega_s \Delta_r \right)^T \Omega_\mu \left(\Delta_\mu + \Omega_\mu^{-1} \Omega_s \Delta_r \right) \\ =\ & -\frac{1}{2}\, \Delta_r^T \left(\Omega_r - \Omega_s^T \Omega_\mu^{-1} \Omega_s \right) \Delta_r \ - \frac{1}{2}\, \Delta_{\left(\mu - \Omega_\mu^{-1} \Omega_s \Delta_r\right)}^T \Omega_\mu \Delta_{\left(\mu - \Omega_\mu^{-1} \Omega_s \Delta_r\right)} \end{aligned} \tag{4.54}$$

Hence, the combined satisfaction can be split into a product of two satisfactions, one orientational and one Cartesian.

$$\text{sat}_{\left\langle \begin{bmatrix} r \\ \mu \end{bmatrix},\ \begin{bmatrix} \frac{1}{4}\Omega_r & \frac{1}{2}\Omega_s^T \\ \frac{1}{2}\Omega_s & \Omega_\mu \end{bmatrix} \right\rangle} \left(\begin{bmatrix} u \\ v \end{bmatrix} \right) = \text{sat}_{\left\langle r,\ \frac{1}{4}\left(\Omega_r - \Omega_s^T \Omega_\mu^{-1} \Omega_s\right) \right\rangle}(u) \cdot \text{sat}_{\left\langle \mu - \Omega_\mu^{-1} \Omega_s \mathrm{L}[\tilde{r}] u,\ \Omega_\mu \right\rangle}(v) \tag{4.55}$$

Note that the above reformulation can similarly be made for combined data vectors containing multiple rotors as long as at least one Cartesian element is present inside the combined data vector. Furthermore, the concrete formula of the split satisfaction may change when taking a behavior task function $t(\cdot)$ into account, as specified in the behavior calculation Definition 4.6.

The emerging multiplication in the split satisfaction Equation (4.55) perfectly integrates into the satisfaction product-based optimization of Definition 4.6. In order to maximize the combined satisfaction, the product of both separated ones has to be maximized. However, due to the reformulation, they can be handled independently.

It is important to see that the orientation satisfaction is independent of the latter Cartesian satisfaction. However, its optimization result u is restricted to embody a valid unit rotor, i.e., $\|u\| = 1$. In contrast, the vector satisfaction is unrestricted w.r.t. its result vector v, but depends on the orientation result rotor u. Hence, the Cartesian result vector v can be computed as an expression depending on u, i.e., $v = \mu - \Omega_\mu^{-1}\Omega_s \mathrm{L}[\tilde{r}]\, u$.

Without loss of optimality, the combined satisfaction can be optimized in two steps:

1. Determine the orientation result rotor u that maximizes the split orientation satisfaction.
2. Use the determined result rotor u to compute the Cartesian satisfaction equation and eventually solve this equation for its maximizing result vector v.

Note that additional orientation result evaluations may be added before the eventual, unrestricted Cartesian result part is calculated.

4.4.5 Differentiation, Integration, and Abstraction

Due to its Definition 4.5, satisfaction inherits important algebraic properties from its underlying PDF. One of the most important properties is its invariance against linear, non-projecting transformations. Since both Differentiation and Integration are linear transformations, it follows that the satisfaction-based PBN data semantics and behavior calculations are invariant against those, too.

Note that the derivative of a rotor becomes a Cartesian vector. Thus, in the special case of deriving a rotor, the satisfaction-related underlying PDF changes from a Bingham distribution to a normal distribution. The claimed invariance, hence, holds only up to the discrepancy between these two probability distributions. As shown in Figure 4.3, the two distributions mostly coincide in the region of interest.

Given this invariance, data can be derived or integrated throughout a PBN without altering its satisfaction-encoded semantics, as long as these data undergo only linear transformations. Preserving data semantics is a crucial property that ensures a behavior network's development flexibility and maintainability, as well as its enhancement and training capabilities, as stated in the architectural requirements in Section 4.1. However, motivated by the natural control Requirements 2.2: *"Singular Positions"*, 2.4: *"Nonlinear Relations"*, and 2.5: *"Hierarchical Control"*, arbitrary, nonlinear data abstraction must also

be supported throughout the entire behavior graph. Both these contradictory demands can be met by limiting control data processing to a linearized state, represented by its first derivative.

Definition 4.12: PBN Control Data

Control data vectors X, meant to be composed or converted into representation spaces within a PBN graph, should be represented as the control's respective uncertain first derivative $\dot{X} \in \mathcal{X}$.

I.e., control data are represented by the desired change of the momentary control value μ_0 and its accepted tolerance in the desired change.

$$\dot{X} = {}^{\Sigma}\langle \dot{\mu},\ \sigma^2 \rangle \qquad \text{with} \qquad \int \dot{X} = {}^{\Sigma}\langle \mu_0 + \int \dot{\mu},\ (0 + \int \sigma)^2 \rangle$$

Note that the integral of the control data $\int \dot{X}$ starts at time t_{NOW} with the probabilistic constant representation ${}^{\Sigma}\langle \mu_0,\ 0 \rangle$. The constant represents the momentary control value μ_0 that is about to be changed by $\dot{X}$. Since the control value μ_0 is exactly known, its respective variance equals zero, i.e., $\sigma_0^2 = 0$.

First introduced in Figure 3.17 within Section 3.3.1, data abstraction is achieved by arbitrarily projecting a data vector into an equal or lower-dimensional subspace. Since Definition 4.6 specifies a behavior's mean calculation rule using the left inverse task function, such a projection, although well-defined, results in an improper, possibly information-lossy behavior. Thus, such an arbitrary projection may only be used for sensor data processing that must not be composed afterward.

Control data, however, is meant to flow in the opposite direction. For control processing behaviors, the just-mentioned data projection becomes the actual left inverse task function, thereby leading to a proper definition of behavior calculation. Furthermore, the first derivative of such an arbitrary function, i.e., its Jacobian matrix, represents a momentary linearization. Thus, derivative control processing satisfies both of the above-mentioned requirements. It enables arbitrary coordinate transformations while simultaneously maintaining transformation linearity throughout the control flow. A single integrator is eventually placed at the end of the data flow when the probabilistic PBN representations are converted into concrete hardware commands.

The following additional benefits automatically arise derivative control commands:

1. By definition, C^0-continuity of the eventual signal integral is guaranteed.
2. All control data are unified Cartesian vectors.
3. The BB-related problem of undefined semantics of entirely inactive control data, as described in [Ropertz 19, p. 141], is avoided due to the intrinsic C^0-continuity.

   If no active control change is demanded by any of the behaviors in the control network, the eventual integrator should be intended to hold its current value. Consequently, inactive or momentarily unnecessary behaviors may switch off their internal control calculation routines or be entirely removed from the network, thus saving computational capacity and energy.

4. The problem of competitive runaway integration[8] is structurally avoided.

   By returning derivatives, integral controllers become proportional controllers, and proportional controllers become derivative controllers within a PBN.

In order to avoid derivative control components becoming dependent on arithmetically-induced noisy second derivatives, these may be handled as separate, secondary control commands to be applied outside the PBN. Note that separating this derivative component implements exactly the low-level control scheme of the musculoskeletal, hybrid impedance control approach developed in Chapter 3.

4.4.6 Data Series

Motivated by Requirement 2.5: *"Hierarchical Control"* and derived as an architecture requirement in Section 4.1, frequency invariance can be achieved by extending the probabilistic data to functions of such in time. In practice, this means that control behaviors not only return the momentary data but, instead, all predicted control data within a future time interval. This way, subsequent behaviors can under- or oversample this function. In the case of sensor data processing, the same function may be reversed in time, representing the sense data representations over some past time interval. However, since controllers should always operate with the latest, most accurate sense data available, the practical relevance of long-lasting sense data series is questionable. Hence, without loss of general applicability to sense data, this work focuses only on the more relevant control data series.

As discussed in Section 2.1.5, the control frequency is intended to reduce with the hierarchical level of abstraction. At the same time, control complexity may rise with the reduced frequency due to the increase of available computation time per control cycle. Figure 4.1 lists those coherent properties in the middle between the biological CNS and the robot control architecture requirements. Hence, following the control data flow from high levels of abstraction down to the hardware actuators, these control data will never be undersampled. Thus, control information in control data functions is preserved in the entire control data flow throughout the architectural hierarchies, as required by the behavior Definition 4.6. Analog to the data abstraction described in the previous Section 4.4.5, this information conservativeness is not given in the case of sense data processing due to its opposite data flow direction, thus resulting in improper behaviors. Hence, both frequency reduction and data abstraction form a consistent architectural hierarchy within the definitions of PBNs.

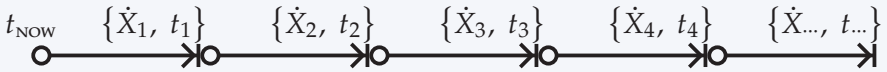
As declared in Definition 4.12, control data series represent a command's first derivative. The actual control value is, hence, determined by an integral of this derivative. Due to the non-negative standard deviation, its integral is also non-negative and additionally monotonically increasing in time. Thus, it is guaranteed that controls become less precise the further they are placed in the future, as expected. Since integration can be expressed as a summation of infinitely small, piecewise constant rectangles, the above-introduced data series can be assumed to be piecewise constant without loss of generality. A control sequence can, hence, be defined as follows:

[8] If two or more integrating controllers sharing the same output data operate against each other, they may individually run into contrary infinities.

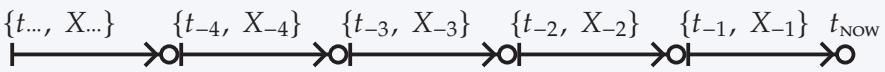
Definition 4.13: PBN Data Series

A data series $X(t)$ in a PBN is a sequence of pairs $\{t_n,\ X_n\}$ containing a PBN data vector $X_n \in \mathcal{X}$ according to Definition 4.4 and a timestamp t_n. The timestamps t_n within the data series define the time intervals in which their respective data vectors X_n are in charge.

1. In the case of future **control series**, each data vector $\dot{X}_n \in \mathcal{X}$ lasts within the time interval $(t_{n-1},\ t_n]$ with $t_{n-1} < t_n$, and $t_{n-1} = t_{\text{NOW}}$, if not given.



2. In the case of past **sense series**, each data vector $X_n \in \mathcal{X}$ lasts within the time interval $[t_n,\ t_{n+1})$ with $t_n < t_{n+1}$, and $t_{n+1} = t_{\text{NOW}}$, if not given.



Note that the control data vectors $\dot{X}_n$ are given in their respective first derivative, as demanded in Definition 4.12 within the previous Section 4.4.5. In order to handle multiple data series of different lengths at once, every series is defined to return zero precision at all times outside its specified interval.

Due to the discrete-time and discrete-value domains in digit control, this assumption even holds for finitely small time steps. In practice, the shortest duration of relevance T_{base} is determined by the highest control frequency $T_{base} = \frac{1}{f_{max}}$. In the case of a PBN control network, this is the lowest hierarchical layer, i.e., the frequency of the eventual control integrator that forwards the emerging commands to the hardware. Thus, all higher-level, lower control cycles should be natural multiples of that shortest base duration $T_n = n \cdot T_{base}$, or $f_n = \frac{1}{n} f_{max}$. Control data in between the concrete time steps of the control integrator would lead to some required interpolation and are, hence, not of practical use. An example control data series in discrete time, together with its respective control integral, is given in Figure 4.4.

Implementation Hint 4.2: Natural, Discrete Time Quantization

In order to simplify digital time arithmetic, it is recommended to discretize time to natural numbers representing multiples of the shortest controlled cycle time.

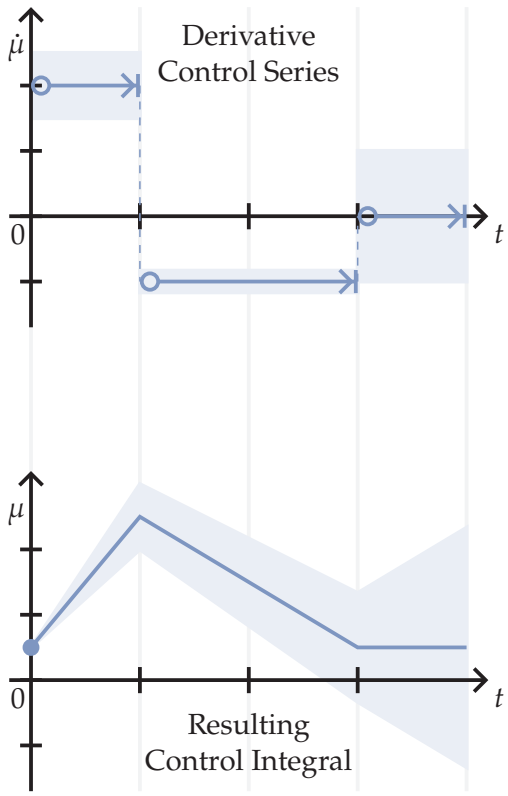
With the above introduction of data series, data in a PBN may no longer be simple data but instead represent functions of simple data in time. Thus, respective behaviors no longer implement a regular function but a functional instead. Accordingly, functional behaviors receive data functions as inputs and return a result function, i.e., a data series, as output. It follows that the behavior calculation, as given in Equation (4.9) within Definition 4.6, must be updated to handle data series instead of regular data.

Time t	Control $\dot{X} = \langle \dot{\mu}, \Omega \rangle$	Std.Dev. $\sigma = \sqrt{1/\Omega}$
NOW	–	–
1	$\langle 2, 4 \rangle$	$1/2$
3	$\langle -1, 36 \rangle$	$1/6$
4	$\langle 0, 1 \rangle$	1

(a) Control Series

Time t	Integral $\int \dot{X} = \langle \mu, \Omega \rangle$	Std.Dev. $\int \sigma$
NOW	$\langle 0.5, \infty \rangle$	0
1	$\langle 2.5, 4 \rangle$	$1/2$
2	$\langle 1.5, 2.25 \rangle$	$2/3$
3	$\langle 0.5, 1.44 \rangle$	$5/6$
4	$\langle 0.5, \sim 0.30 \rangle$	$1\,5/6$

(b) Control Integral



(c) Control Visualizations

Figure 4.4: Example of a derivative control series (a) and its resulting control integral (b) on the left. Visualizations (c) of both of those probabilistic data series are given on the right. Both graphs plot the mean values within their respective 1σ-environments. The commanded control series contains three control derivatives lasting until the discrete timestamps 1, 3, and 4, respectively. The resulting control integral starts at the exact value of 0.5 and is evaluated consecutively for all discrete timestamps. Following the standard stochastic calculus, the piecewise constant integration of $\dot{X}_n$ is a continuous cumulation of both the mean and standard deviation integral. As shown in the bottom plot, the uncertainty of the control integral strictly increases in time, as expected.

Definition 4.14: PBN Data Series Calculation

Let $X_{i\in I}(t) = \langle \mu_i(t)\,,\ \Omega_i(t) \rangle$ be an arbitrary number of input data series. Let also $T(\cdot)$ be the behavior's task functional, such that $T\big(\mu_{i\in I}(t)\big) \stackrel{!}{=} \mu_{out}(t)$.
A functional behavior's result mean series is then defined as:

$$\underset{\mu_{out}(t)}{\arg\max} \int_{-\infty}^{+\infty} \prod_{i\in I} \mathrm{sat}_{X_i(t)}\big(T^{-1}\big(\mu_{out}(t)\big)[i]\big)\, dt \tag{4.56}$$

According to the original Definition 4.6, a functional behavior's output precision function $\Omega_{out}(t)$ has to be technically and semantically adequate while preserving commutativity and associativity at all times.

Note that the above optimization Equation (4.56) is generally not well-defined due to its unbound time integral. However, this issue is solved by defaulting $\Omega_{out}(t)$ to zero, i.e., leaving $X_{out}(t)$ unspecified, whenever all input data series are unspecified, too, i.e., also defaulting to zero precision. This way, the infinite integral can be reduced to the time interval where at least one of the input series is specified. Hence, Equation (4.56) becomes well-defined for all finite input series. In the case of control data, only finite future control horizons are accepted by PBN behaviors.

Generally, the mean calculation Equation (4.56) optimizes for an arbitrary function of mean in time $\mu_{out}(t)$ w.r.t. arbitrary input data functions. However, following Hint 4.2, discretizing the time into natural numbers, those arbitrary functions also discretize and become vectors.

Furthermore, the prior integral in time degenerates to a sum over the vector elements. A quadratic optimization formulation is now achieved by replacing the maximization of satisfaction functions with a minimization of the satisfaction-internal quadratic equations. This quadratic form is well-known from the standard quadratic cost function in MPC.

The required predictive dynamic model is embedded in the behavior's task functional. Thus, a functional PBN behavior can implement a well-known standard MPC policy, as demanded in Requirement 2.1: *"Hybrid Dynamics"*. The mandatory dynamic model may be either hard-coded in the behavior task itself or fed as a parameter by a PBN-external robot environment model, as illustrated in Figure 2.14.

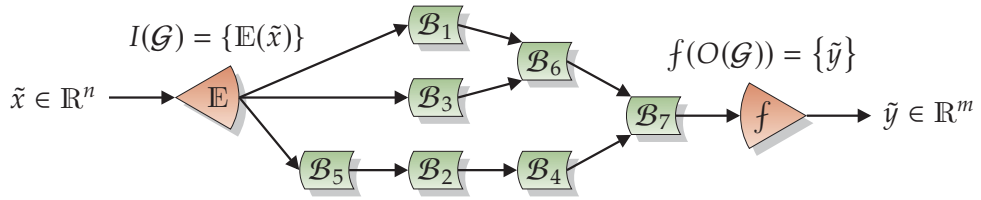
Using a parameterized setup offers several benefits over its hard-coded counterpart. First, it is better engineering to separate the data from the data processing algorithm for technical reasons such as software flexibility or maintainability. More significant, however, is the possibility of offline or closed-loop online model enhancements without underlying computational code updates. Thus, using PBN-external models to parameterize behaviors enables state-of-the-art ML techniques to be easily integrated into the robot's overall control architecture, as required in Section 4.1.

4.5 Interfacing the Environment

As introduced in Section 4.3 and defined in Definition 4.3, the PBN architecture is a consistent, acyclic, directed data flow graph. Therefore, to achieve a closed-loop control system, it has to be embedded within an outer, overall control loop, as pictured in Figure 2.14. Two different loop closures are sketched within this overall setup.

First, a pure virtual control loop feeds information from the robot's control domain, i.e., the PBN, back into its internal state memory. The second loop closure is physical, involving the real-world robot hardware and environment.

In Figure 2.14, this robot-intrinsic state memory is named *Model* and is part of the overall *Environment*. The model includes all intrinsically memorized information independent of their complexity and usage. Examples of such intrinsic models are, e.g., the internal state of a simple SISO digital signal filter, the robot's whole body floating-base dynamic model, or a complex geometric representation of the robot's past and current environment.



$$\mathcal{G}(\tilde{x}) = f(\mathcal{B}_7(\mathcal{B}_6(\mathcal{B}_1(\mathbb{E}(\tilde{x}))\,,\ \mathcal{B}_3(\mathbb{E}(\tilde{x})))\,,\ \mathcal{B}_4(\mathcal{B}_2(\mathcal{B}_5(\mathbb{E}(\tilde{x})))))) = \tilde{y}$$

Figure 4.5: Exemplary PBN graph $\mathcal{G}$, as introduced in Figure 4.2. The graph's input and output interfaces have been replaced by their respective data annotation and integration functions. Underneath the graphical graph presentation, its non-probabilistic functional equivalent is given. The function continuously maps a concrete input signal $\tilde{x} \in \mathbb{R}^n$ into a continuous concrete output signal $\tilde{y} \in \mathbb{R}^m$, i.e., $\mathcal{G}(\tilde{x}) = f(\mathcal{G}(\mathbb{E}(\tilde{x}))) = \tilde{y}$.

Since those states are purely virtual and user-defined, there is no technical limitation to their PBN integration. However, depending on the exact use case, preserving the stochastic information from the PBN control graph within the memorized models is advisable, as already stated in Section 4.4.1. Two interesting examples of stochastic models in robotics are the well-known *Kalman* filter and the so-called *"Gaussian Process Dynamical Models"* for predictive sense and control, respectively [Kalman 60, Beckers 22].

In contrast, the second physical loop closure is subject to real-world-induced constraints. The feedback loop involves the sequence of digital control commands being physically executed by the robot's actuators, mechanically processed through the hardware morphology and the real-world environment, and eventually observed again by the robot's sensors.

When controlling bipedal robots, stochastic electrical or mechanical actions do not exist. Thus, to interface a PBN control graph with the non-stochastic, real-world environment, sense and control signals must be pre- and postprocessed when entering or leaving a PBN graph via its interfaces.

PBN input data preprocessing is described via the function $\mathbb{E}(\tilde{x}) = X$ with $\tilde{x} \in \mathbb{R}^n$ being a concrete signal and $X \in \mathcal{X}$ being its respective probabilistic data representation. The function $\mathbb{E}(\cdot)$ converts its argument signal into a signal of expected values annotated with respective precisions. It is information-generative and, thus, may be implemented in various manners.

Due to the necessary removal of the probabilistic control signal's precision information, PBN output data postprocessing is information-lossy. As control data in a PBN are restricted to only occur as time series of derivatives (Definitions 4.12 and 4.13), the respective postprocessing function is herein written as an integral with a minus sign $f(\cdot)$. The function $f\left(\dot{X}(t)\right) = \tilde{x}$ continuously integrates its probabilistic data derivative series $\dot{X}(t)$ into a concrete, continuous command signal $\tilde{x} \in \mathbb{R}^n$.

Replacing the interfaces of a PBN graph with the above pre- and postprocessing functions, an entirely non-probabilistic control function is achieved. Figure 4.5 shows the example PBN graph introduced in Figure 4.2 with its interfaces replaced by the respective interface functions.

4.5.1 Exponential Sensor Signal Precision Annotation

Sensor signal preprocessing is based on individual, concrete signal samples and is, therefore, always a statistical application rather than a stochastic one. Based on a set of n individually measured samples x_i of a signal $\tilde{x}$, the signal's mean $\mathbb{E}[\tilde{x}] = \tilde{\mu}$ and its statistical variance $\tilde{S}^2 = \frac{n}{n-1}\mathrm{Var}(\tilde{x})$ are generally defined as follows:

$$\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.57}$$

$$\tilde{S}^2 = \frac{n}{n-1}\mathbb{E}\Big[\big(\tilde{x}-\tilde{\mu}\big)^2\Big] = \frac{1}{n-1}\sum_{i=1}^{n}\big(x_i-\tilde{\mu}\big)^2 \tag{4.58}$$

Thus, the corresponding signal annotation function is defined on individual signal samples.

$$\mathbb{E}(\tilde{x}): \;\; \mathbb{E}(x_i) \;\rightarrow\; {}^{\Sigma}\big\langle \mu_i,\, \Sigma_i \big\rangle = \big\langle \mu_i,\, \Omega_i \big\rangle \tag{4.59}$$

By adding additional weight factors to the individual signal samples, the above formulations perfectly integrate into fixed-window Finite Impulse Response (FIR) signal filters. Unfortunately, in most robotics applications, infinite-window Infinite Impulse Response (IIR) signal filters, especially exponential filters, are preferred due to their digital computational simplicity and efficiency. The most simple and most popular one is the single-stage exponential filter defined by the following identical explicit and recursive mean calculations:

$$\begin{aligned} & \mu_i = \sum_{n=0}^{\infty} \alpha\,(1-\alpha)^n\, x_{i-n} && \text{explicit} \\ \text{or:} \quad & \mu_i = \alpha x_i + (1-\alpha)\,\mu_{i-1} && \text{recursive} \end{aligned} \tag{4.60}$$

Where $\alpha \in (0,1)$ characterizes the exponential weight decrease when traversing backward through the series of signal samples. Because of $\sum_{n=0}^{\infty} \alpha\,(1-\alpha)^n = 1$ for all $0 < \alpha < 1$, Equation (4.60) is a well-defined, exponentially weighted, valid mean calculation for an infinite set of signal samples.

According to the infinite mean summation, the statistical variance is weighted identically. However, due to the infinite sample size, the statistical variance correction term vanishes, and thus $\tilde{S}^2 = \mathrm{Var}(\tilde{x}) = \tilde{\Sigma}$. As a last adaptation, each sample's squared deviation from the distribution's mean is replaced by the samples' squared deviation from their momentary mean. Since measured sensor signals are expected to be non-constant in time, this last step is necessary to forget previous signal deviations continuously. Using a running mean within the variance calculation will definitely lead to an incorrect evaluation, not returning the actual variance value. However, the resulting variance will be close enough to its correct value so that practical usability, in this case, is to be favored over the last bit of statistical correctness. Note that Equation (4.60) does also not

return the real signal value. The following two identical explicit and recursive equations formulate the resulting so-called *exponential variance*[9]:

$$\begin{aligned} \Sigma_i &= \sum_{n=0}^{\infty} \alpha\,(1-\alpha)^n \left(x_{i-n} - \mu_{i-n}\right)^2 && \text{explicit} \\ \text{or:} \quad \Sigma_i &= \alpha \left(x_i - \mu_i\right)^2 + (1-\alpha)\,(\Sigma_{i-1}) && \text{recursive} \end{aligned} \tag{4.61}$$

The equivalent recursive precision equation is achieved by an inversion of the variance.

$$\Omega_i = \frac{1}{\Sigma_i} = \frac{\Omega^\star_{i-1}}{1 - \alpha + \alpha\Omega^\star_{i-1}\left(x_i - \mu_i\right)^2} \tag{4.62}$$
$$\text{for:} \quad \Omega^\star_{i-1} := \max\{\Omega_{i-1}, \epsilon\} \text{ with } \epsilon > 0$$

Note that the latter rise of Ω_{i-1} to at least some small value $\epsilon > 0$ is necessary to escape zero precision, which should not be of practical relevance except for a zero precision startup procedure. The probabilistic data $\langle \mu_i,\ \Omega_i \rangle$ will be stored outside the PBN as part of the robot's stochastic model storage, as sketched in Figure 2.14. These model data will be fed back into the PBN as $\langle \mu_{i-1},\ \Omega_{i-1} \rangle$ for the next execution cycle.

The above-described methodology is suitable for scalar, non-circular raw signals. Variations for Cartesian vector data and orientation data represented as unit rotors can be implemented similarly with only a few adaptations. However, depending on the signals' context, complexity, and potential inbuilt preprocessing, other solutions are possible and recommended to best fit each particular sensor unit's individual needs.

4.5.2 Probabilistic Integration of Control Data Series

In the case of sensor information being fed into a PBN, additional signal precision information is generated and added to the data. In contrast, control commands are required to lose their attached tolerance annotations. Thus, prior important information must be removed. Unfortunately, removing such information while preserving all required control invariants and characteristics collected in Section 4.1 is much more challenging than introducing additional information.

Remember that Definitions 4.12 and 4.13 restrict robot control commands to be represented by time series of the controls' first derivatives. Thus, control data must be integrated when leaving a PBN. According to the control and PBN invariants, requirements of such a probabilistic integration can be formulated, as partially published in [*Vonwirth 24a*]:

1. A control derivative annotated with zero precision must not change its underlying actual control value.

2. The higher a control derivative's precision, the more significantly it must alter its underlying actual control value toward its integrated pure mean.

[9] Without an existing reference, credits are going out to the unknown former member of the RRLab who initially implemented this *exponential variance* within our code base.

3. If a control behavior aims toward a steady control goal, the satisfaction of its control integral against its underlying actual control value must not decrease.

4. A steady control goal with non-zero precision must eventually be achieved.

5. Probabilistic integration must be invariant against linear, information-preserving, i.e., non-projective transformations, as derived in Section 4.4.5.

6. Probabilistic integration must be invariant against its execution frequency, i.e., against under- and oversampling.

7. Probabilistic integration must preserve the integral's functional shape integrity, i.e., a piecewise constant control derivative must integrate into a piecewise linear, C^0-continuous control command.

A possible but not unique implementation can be defined as follows:

Definition 4.15: Probabilistic Integration

Let $\dot{X}(t) = \left\langle \dot{\mu}(t)\,,\ \Omega(t) \right\rangle$ be a series of probabilistic Cartesian control derivatives acting on the concrete command signal $\tilde{x} \in \mathbb{R}^n$. Define x_t as the command sample at timestamp t, directly following its previous control cycle's command sample x_{t-1} at timestamp $t-1$.
The probabilistic integration function $\int\!\!\!\!\!\!\int \left(\dot{X}(t)\right) \to \tilde{x}$ is defined recursively in the result signal samples. At time t:

$$\int\!\!\!\!\!\!\int \left(\dot{X}(t)\right) = x_t \ :=\ E\, x_{t-1} + (I - E)\left(x_{t-1} + \int_{t-1}^{t} \dot{\mu}(t)\, dt\right) \tag{4.63}$$

with

$$E \ :=\ \exp\left(-\alpha \int_{t-1}^{t} \Omega(t)\, dt\right) \tag{4.64}$$

The scalar factor $\alpha \geq 0$ is determined such that

$$\left(1 - \mathrm{sat}_{\left(x_{t-1} + \int_{t-1}^{t} \dot{X}\right)}(x_t)\right) \ =\ (1-\gamma)\left(1 - \mathrm{sat}_{\left(x_{t-1} + \int_{t-1}^{t} \dot{X}\right)}(x_{t-1})\right) \tag{4.65}$$

with $\gamma \ :=\ \exp\left(-\frac{1}{2}p^2\right) \in (0, 1]$ is a user-defined control tolerance parameter p.
The control commands x_t are guaranteed to be inside the p-σ-environment around the integrated control mean w.r.t. the integrated control precision.

The control command sample x_t is calculated as a multivariate weighted average of its previous sample x_{t-1} and its ideal control integral $x_{t-1} + \int \dot{\mu}$. Both sides' weight matrices are related to the negative, scaled matrix exponential of the control's integrated precision. Due to this definition, the matrix exponential E is positive-definite $E > 0$ and inside the *matrix-interval* $E \in (0, I]$ independent of the scalar factor $\alpha \geq 0$. Thus, both weight matrices E and $(I - E)$ are non-negative and sum to the identity matrix I.

Equation (4.63) implements a single-stage, exponential IIR filter with adaptive parameterization. The parameterization is chosen so that the result is not updated if the control precision Ω equals zero and continuously shifts toward the commanded integrated mean with increased precision. Thus, Equation (4.63) satisfies the Requirements 1 and 2.

It is important to highlight that the proposed implementation of the probabilistic integration can directly be applied to encoded symmetric matrix data, as introduced in Section 4.4.3, without any need for adaptations. This versatility arises from using the matrix exponential in Equation (4.64), the property of the exponential encoded matrix precision derived in Equation (4.43), and the respective satisfaction definition.

Equation (4.65) implements an exponential approximation of the desired control's satisfaction against the actual commanded control $\mathrm{sat}_{\left(x_{t-1}+\int_{t-1}^{t}\dot{X}\right)}(x_t)$ toward the maximum satisfaction value of 1. Whenever the desired control is most unsatisfied with the previous control, i.e., $\mathrm{sat}_{\left(x_{t-1}+\int_{t-1}^{t}\dot{X}\right)}(x_{t-1}) \approx 0$, the satisfaction against the upcoming control command x_t is defined to equal the value of γ. Due to the definition of γ w.r.t. parameter p and the Cartesian satisfaction, as given in Equation (4.20), x_t must exactly feature p-σ-deviation from its ideal, integrated mean. Since the satisfaction value exponentially approximates its maximum value of 1, the ideal mean control value must eventually be achieved. Thus, Equation (4.65) satisfies the Requirements 3 and 4.

Furthermore, Requirement 5 is met by using the satisfaction function in Equation (4.65). The probabilistic integrator's invariance against linear, non-projective transformations is directly inherited from the satisfaction, i.e., from the normal distribution's PDF.

Invariance against undersampling, as demanded in Requirement 6, is of no practical relevance. Following the time and execution frequency discretization given in Hint 4.2, undersampling is, by definition, impossible to occur. However, by executing the integration multiple times, higher sampling rates than the actual one can easily be simulated. Thus, the demanded invariance remains valid.

Without undersampling, the probabilistic control series $\dot{X}(t)$ becomes constant within all individual time intervals $([t-1],\ t]$ according to Definition 4.13. Thus, all integrals in Definition 4.15 simplify to linear functions w.r.t. the integration's cycle time T.

$$\int_{t-1}^{t} \dot{X}(t)\,dt = \left\langle T\ \dot{\mu},\ \tfrac{1}{T^2}\,\Omega \right\rangle \qquad \text{with} \quad T = (t - [t-1]) \tag{4.66}$$

Thus, Equations (4.63) to (4.65) can be simplified into

$$\begin{aligned} & x_t \ = x_{t-1} + T\,(I-E)\,\dot{\mu} && \text{with} && E \ = \exp(-\alpha\,\Omega) \\ \Leftrightarrow\quad & x_t \ = x_{t-1} + T\Delta_x && \text{with} && \Delta_x \ = \tfrac{1}{T}\,(x_t - x_{t-1}) = (I-E)\,\dot{\mu} \end{aligned} \tag{4.67}$$

and the scalar factor $\alpha \geq 0$ be determined such that

$$\begin{aligned} \left(1 - \mathrm{sat}_{\dot{X}}(\Delta_x)\right) \ &= \ (1-\gamma)\left(1 - \mathrm{sat}_{\dot{X}}(0)\right) \\ \Leftrightarrow \qquad \mathrm{sat}^{\text{-}}_{\dot{X}}(\Delta_x) \ &= \ -\,\mathrm{expm1}\left(-\tfrac{1}{2}p^2\right)\mathrm{sat}^{\text{-}}_{\dot{X}}(0) \end{aligned} \tag{4.68}$$

Note that the linear time factor $\frac{1}{T^2}$ of the evaluated integration of Ω is now included within the new scalar factor $\alpha \geq 0$. The integration result x_t will be stored outside the

PBN as part of the robot's stochastic model storage, sketched in Figure 2.14. However, since the integral result is perfectly known, its respective variance equals zero, i.e., $x_t = {}^{\Sigma}\langle x_t,\ 0\rangle$, as specified in Definition 4.12. The stored result x_t will be fed back into the PBN as ${}^{\Sigma}\langle x_{t-1},\ 0\rangle$ for the next execution cycle.

Inserting the Equation of E into Δ_x and both into the dissatisfaction Equation (4.68), it can be seen that the scalar factor α is independent of any time or integral result information, thus being only defined by the momentary derivative data $\dot{X}$ and the constant parameter p. Since data series are defined piecewise constant, $\dot{X}$ will be constant throughout oversampling. It follows that the scalar factor α, thus the matrix E, and eventually the vector Δ_x will all stay constant during oversampling.

Let the cycle time $T = \int_{t-n}^{t} = T_n + \cdots + T_2 + T_1$ be oversampled with the consecutive smaller cycle times $T_n, \ldots, T_2, T_1$. The eventual integration result after all consecutive probabilistic integrations is then

$$\begin{aligned} x_t &= x_{t-1} + T_1\Delta_x \\ &= x_{t-2} + T_2\Delta_x + T_1\Delta_x \\ &= x_{t-n} + T_n\Delta_x + \cdots + T_2\Delta_x + T_1\Delta_x \\ &= x_{t-n} + (T_n + \cdots + T_2 + T_1)\,\Delta_x \qquad = x_{t-n} + T\Delta_x \end{aligned} \tag{4.69}$$

The resulting identity of the consecutive integrations with a single one proves the probabilistic integration to be invariant against oversampling, as demanded in the above Requirement 6.

As visible in Equation (4.67), the probabilistic integral result sample x_t is described as a linear function in the integration interval T with the constant derivative Δ_x. Due to the piecewise constant series of derivatives $\dot{X}(t)$, and thus the piecewise constant $\Delta_x(t)$, the result integral in time $x(t)$ becomes piecewise linear. C^0-continuity also directly follows from the linear integration Equation (4.67). Thus, the only remaining initial Requirement 7 is fulfilled by the probabilistic integration Definition 4.15.

In order to evaluate the probabilistic integral, Equation (4.68) has to be solved for the scalar factor α. Unfortunately, for higher dimensional $\dot{X}$, this equation contains a summation of different exponential functions which, to the best of the author's knowledge, does not feature a closed-form solution, yet. Hence, a standard root-finding optimization is applied to the $\operatorname{lnp1}(\cdot)$ transformed values of both sides of Equation (4.68) to solve for α. Applying the $\operatorname{lnp1}(\cdot)$-function to both sides of the equation does not change the result due to the strict monotony of the natural logarithm and $\operatorname{lnp1}(\cdot)$ being well-defined for the dissatisfaction's entire image $(-1, 0]$. The root-finding function $f(\alpha)$ is defined as follows:

$$\begin{aligned} f(\alpha) &:= \ln\left(\operatorname{sat}_{\dot{X}}(\Delta_x)\right) - \operatorname{lnp1}\left(-\operatorname{expm1}\left(-\tfrac{1}{2}p^2\right)\operatorname{sat}^{-}_{\dot{X}}(0)\right) \\ &= -\tfrac{1}{2}\dot{\mu}^T E\Omega E\dot{\mu} - s_d \end{aligned} \tag{4.70}$$

With $E = \exp(-\alpha\Omega)$ being dependent on α, and $s_d = \operatorname{lnp1}\left(-\operatorname{expm1}\left(-\tfrac{1}{2}p^2\right)\operatorname{sat}^{-}_{\dot{X}}(0)\right)$ being constant in α. The first derivative of $f(\alpha)$ evaluates to:

$$\begin{aligned} f'(\alpha) &:= \dot{\mu}^T E\Omega^2 E\dot{\mu} && \geq 0 \\ &> 0 && \text{if } \Omega\dot{\mu} \neq 0 \end{aligned} \tag{4.71}$$

Thus, the respective Newton iteration is formulated as follows.

$$\begin{aligned} \alpha_0 &= 0 \\ \alpha_{n+1} &= \alpha_n + \delta_n \qquad \text{with} \quad \delta_n = -\frac{f(\alpha_n)}{f'(\alpha_n)} \end{aligned} \tag{4.72}$$

Note that $f(\alpha)$ is a strictly monotonic scalar function with an exponential-like shape for all non-zero probabilistic data, i.e., $\Omega\dot{\mu} \neq 0$. Hence, it features a single, unique root guaranteed to be eventually found by the above Newton iteration.

However, the scalar factor α does not need to be calculated. In order to see this, the iterative approximation process of the exponential weight matrix E, as defined in Equation (4.67), has to be studied.

$$\begin{aligned} E_0 &= \exp(-\alpha_0\Omega) &&= \exp(-0\Omega) &&= I \\ E_{n+1} &= \exp(-\alpha_{n+1}\Omega) &&= \exp(-\alpha_n\Omega - \delta_n\Omega) &&= \exp(-\delta_n\Omega)\, E_n \end{aligned} \tag{4.73}$$

Note that the last matrix exponential transformation holds due to the commutativity of the symmetric precision matrix with itself. Since the consecutive calculation of E is anyhow required within the iterative evaluation of δ, the actual value of α becomes obsolete. The iteration, therefore, can be executed on the value of δ until its convergence to zero, i.e., until $\delta < \epsilon$.

Implementation Hint 4.3: Numerical Stability on Floating-Point Arithmetic

It is important to highlight that the following three implementation-related details are all crucial for achieving numerical stability with floating-point arithmetic:

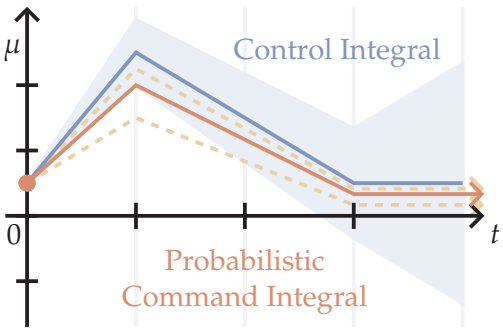
1. Using the dissatisfaction function, as defined in Hint 4.1.
2. Using the lnp1(·)-function, as the numerically stable counterpart of the expm1(·)-function.
3. Reformulating the approximation iteration in favor of δ converging toward zero instead of approximating the unbound value of α.

Figure 4.6 presents an example execution of the just-derived probabilistic integration at the hand of the control series example introduced in Figure 4.4. An efficient implementation following Hint 4.3 is given in Algorithm 4.3.

It is important to note that the above-formulated probabilistic integration has been proven to meet its initially stated requirements but not of being the only possible solution. Similar but different realizations of Equations (4.63) to (4.65) do exist. For example, Equation (4.64) could be replaced by $E := \left(\alpha \int_{t-1}^{t} \Omega(t)\, dt + I\right)^{-1}$ generating a similar characteristic. However, due to the given argumentation, the comfortable embedding into the rest of the PBN architecture, and the natural exponential characteristics, other solutions can be assumed not to behave more suitably or more naturally than the given one. Future investigations may be done to uncover the optimal solution.

Time	Control	Prob. Integral	
t	$\dot{X} = \langle \dot{\mu}, \Omega \rangle$	$\Delta_x\|_{p=1}$	x_t
NOW	–	–	$1/2$
1	$\langle 2, 4 \rangle$	$1\,1/2$	2
2		$-5/6$	$1\,1/6$
3	$\langle -1, 36 \rangle$	$-5/6$	$1/3$
4	$\langle 0, 1 \rangle$	0	$1/3$

(a) Control and Probabilistic Integral



(b) Control Visualizations

Figure 4.6: Illustration of a probabilistic integration at the hand of the control's first derivative series $\dot{X}$, introduced in Figure 4.4. On the left, the derivative control series (a) is listed aside from the respective probabilistic integral x_t. The probabilistic integration is calculated according to Equations (4.67) and (4.68) using parameter $p = 1$, following Definition 4.15. A visualization (b) is given on the right. The blue background and graph plot the integrated probabilistic control series $\int \dot{X}$, as first presented in Figure 4.4c. In the foreground, the solid red graph plots the control command integral x_t, emerging from the probabilistic integration. As parameterized, the control command stays within the $p = 1$-σ-environment of the control mean. The two dashed, orange graphs above and below the red one are probabilistic integrals of the identical input series $\dot{X}$ using the integration parameterizations $p = 0.5$ and $p = 2$, respectively. As intended, the parameterization p inverse correlates with the probabilistic integration precision.

Algorithm 4.3: Probabilistic Integration According to Definition 4.15

Data: The momentary Cartesian control derivative ${}^{P}\dot{X} = {}^{P}\langle \dot{p}, \Omega \rangle$ in precise notation.
The algorithm parameters ϵ and p.
The integration cycle time T.
The previous command sample x_{t-1}.

`/* Execute a single, linear probabilistic integration step: */`

Decompose the control input precision via SVD: $\Omega = Q \Lambda Q^T$

Extend ${}^{P}\dot{X}$ with *no-change* data in all irrelevant directions to ensure Λ being invertible.

Calculate s_d according to Equation (4.70)

Initialize $E_p = Q^T \dot{p}$ `/* A decomposed precise equivalent to (4.73) */`

`/* Execute the Newton iteration as defined in Equations (4.70) to (4.72) using the decomposed precise control` ${}^{P}\langle \dot{p}, Q \Lambda Q^T \rangle$`: */`

repeat

$\delta = \dfrac{\frac{1}{2} E_p^T \Lambda^{-1} E_p + s_d}{E_p^T E_p}$ `/* Evaluate the decomposed update (4.72) */`

$E_p \mathrel{+}= \operatorname{expm1}(-\delta \Lambda)\, E_p$ `/* Update the precise exponential (4.73) */`

until $\|\delta \Lambda\|_\infty < \epsilon$;

`/* Reconstruct the command update derivative (4.67) */`

$\Delta_x = (I - E)\, \Omega^{-1} \dot{p} = Q \Lambda^{-1} \left(Q^T \dot{p} - E_p \right)$

return $x_t = x_{t-1} + T \Delta_x$

4.6 Data Composition

As introduced in Section 4.2, BB control decomposition eventually boils down to control command composition. The development of such control composition started with binary decision-making rules, i.e., no actual composition at all [Brooks 86, Nicolescu 02]. Actual multitasking capability has been introduced with continuous but still bounded data weighting mechanisms and thus possible respective weighted average composition [Vadakkepat 04, Proetzsch 10b]. Lastly, Ropertz et al. introduced unbound weighted average sensor signal composition [Ropertz 17]. Their composition is still a weighted average, but it uses the inverses of the individual sensor measurement standard deviations as respective weights. With the introduction of stochastic data properties, it is important to highlight that this last development first introduces interpretable semantics to BB data composition, thus BB robotics. However, this principle has never been intended to be used also for control purposes.

Due to the lack of generalized semantics and embedded algebraic structure, modern BB robot control architectures, such as the iB2C architecture, have been blown up with numerous functionally distinct composition possibilities, leaving the control networks uninterpretable, unpredictable, and thus unreliable. Additional developments of online decision strategies to change composition strategies during runtime, e.g., [Scheutz 04, Ropertz 19], have also not contributed to better predictability and usability of BB robot control.

In order to overcome this thread, the herein-proposed PBN design reduces all of the just-mentioned functionalities into a single composition strategy. This composition will implement an unbound weighted average for both Cartesian and orientation data types. Due to the unbound weighting based on the identical signal properties of sensor data variance and control tolerance, the weighted average intrinsically degenerates into a binary decision in the case of an infinite input weight.

Following Definitions 4.4 and 4.6, the proposed PBN architecture only allows a single, unique composition behavior implementation per distinct data type. As declared in Section 4.3, data composition is to be treated as just one concrete instantiation of a regular PBN behavior. Due to the uniqueness of the composition behavior, it is denoted with the dedicated behavior function $\mathcal{B}_{\text{composition}}(\cdot) = \Sigma(\cdot)$. Independent of its composing data type, composition behaviors must not alter their inputs. Thus, all composition behaviors' task functions are the identity $t_{\Sigma}(\cdot) = \text{ID}(\cdot)$.

The small example PBN graph in Figure 4.5 can now be updated with the introduced dedicated composition notation. This updated example is shown in Figure 4.7. Compared to its prior representation, the two control-composing behaviors have been replaced by respective composition notations in both the updated graph and its functional equivalent.

4.6.1 Cartesian Vectors

Let $X_{i\in I} = \langle \mu_i, \Omega_i \rangle$ be an arbitrary number of probabilistic Cartesian data vectors according to Notation 4.7. By setting the behavior task function to the identity function $t(\cdot) = \text{ID}(\cdot)$ as defined beforehand, the Cartesian composition behavior is derived

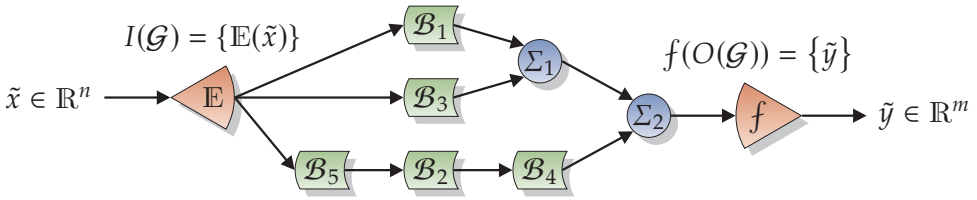


Figure 4.7: Exemplary PBN graph $\mathcal{G}$, updated from its equivalent shown in Figure 4.5. The graph's two composition behaviors have been replaced with the respective dedicated composition notation $\Sigma(\cdot)$. Underneath the graphical presentation, its functional equivalent is given with equally updated composition functions.

following the PBN behavior calculation Definition 4.6 and the Cartesian satisfaction Equation (4.20).

$$\begin{aligned}
\mu_{out} &= \arg\max_{\mu_{out}} \prod_{i \in I} \mathrm{sat}_{X_i}(\mu_{out}) \\
&= \arg\max_{\mu_{out}} \prod_{i \in I} \exp\left(-\frac{1}{2}\left(\mu_{out} - \mu_i\right)^T \Omega_i \left(\mu_{out} - \mu_i\right)\right) \\
&= \arg\max_{\mu_{out}} \exp\left(-\frac{1}{2}\sum_{i \in I}\left(\mu_{out} - \mu_i\right)^T \Omega_i \left(\mu_{out} - \mu_i\right)\right)
\end{aligned} \tag{4.74}$$

Since the exponential function is strictly monotonic, it follows that:

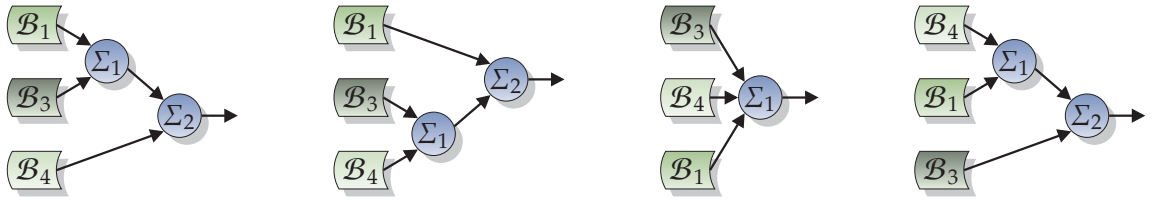
$$\begin{aligned}
\mu_{out} &= \arg\max_{\mu_{out}} -\frac{1}{2}\sum_{i \in I}\left(\mu_{out} - \mu_i\right)^T \Omega_i \left(\mu_{out} - \mu_i\right) \\
&= \arg\min_{\mu_{out}} \frac{1}{2}\sum_{i \in I}\left(\mu_{out}^T \Omega_i \mu_{out} - 2\mu_i^T \Omega_i \mu_{out} + \mu_i^T \Omega_i \mu_i\right) \\
&= \arg\min_{\mu_{out}} \frac{1}{2}\,\mu_{out}^T \left(\sum_{i \in I} \Omega_i\right)\mu_{out} - \sum_{i \in I}\left(\mu_i^T \Omega_i\right)\mu_{out}
\end{aligned} \tag{4.75}$$

Note that the terms $\mu_i^T \Omega_i \mu_i$ are constants and, therefore, neglected in the last term. Since the above equation represents a convex, unconstrained quadratic optimization, it can be solved in closed form by setting its first derivative to the transposed zero vector 0^T.

$$0^T \stackrel{!}{=} \frac{\partial}{\partial \mu_{out}} \frac{1}{2}\,\mu_{out}^T \left(\sum_{i \in I} \Omega_i\right)\mu_{out} - \sum_{i \in I}\left(\mu_i^T \Omega_i\right)\mu_{out} \tag{4.76}$$

$$\Leftrightarrow \quad 0^T \stackrel{!}{=} \mu_{out}^T \left(\sum_{i \in I} \Omega_i\right) - \sum_{i \in I}\left(\mu_i^T \Omega_i\right)$$

$$\Rightarrow \quad \mu_{out} = \left(\sum_{i \in I} \Omega_i\right)^{-1} \sum_{i \in I}\left(\Omega_i \mu_i\right) \tag{4.77}$$



$$\Sigma_2(\Sigma_1(\mathcal{B}_1,\ \mathcal{B}_3)\,,\ \mathcal{B}_4) \quad = \quad \Sigma_2(\mathcal{B}_1,\ \Sigma_1(\mathcal{B}_3,\ \mathcal{B}_4)) \quad = \quad \Sigma_1(\mathcal{B}_3,\ \mathcal{B}_4,\ \mathcal{B}_1) \quad = \quad \Sigma_2(\Sigma_1(\mathcal{B}_4,\ \mathcal{B}_1)\,,\ \mathcal{B}_3)$$

Figure 4.8: Four topologically different but functionally identical PBN example graphs, thus visualizing the composition behavior's commutativity and associativity. The leftmost graph is a subgraph of the example PBN in Figure 4.7.

As desired, Equation (4.77) implements a weighted average of all input mean vectors μ_i weighted by their respective symmetric positive semi-definite precision matrices. In order to preserve the behavior network's commutativity and associativity as demanded by Definition 4.6, the resulting precision of Cartesian vector composition must be defined as the sum of all input precisions.

$$\Omega_{out} := \sum_{i \in I} \Omega_i \tag{4.78}$$

Hence, the Cartesian composition behavior is defined by the following function:

Definition 4.16: Cartesian Data Composition

Let $\Sigma(X_1,\ X_2,\ \ldots)$ be the behavior function describing the composition of arbitrary many Cartesian input data $X_{i \in I} = \langle \mu_i,\ \Omega_i \rangle$.

The unique Cartesian composition behavior is defined as:

$$\Sigma(X_1,\ X_2,\ \ldots) := \left\langle \left(\sum_{i \in I} \Omega_i \right)^{-1} \sum_{i \in I} (\Omega_i \mu_i)\,,\ \sum_{i \in I} \Omega_i \right\rangle \tag{4.79}$$

By using the **precise notation** ${}^P X_{i \in I} = {}^P \langle p_i,\ \Omega_i \rangle$, as given in Equation (4.19) within Notation 4.7, the above Cartesian composition calculation simplifies to the following precise Cartesian composition formula:

$$\begin{aligned} \Sigma\left({}^P X_1,\ {}^P X_2,\ \ldots\right) &= \left\langle \sum_{i \in I} \Omega_i \left(\sum_{i \in I} \Omega_i \right)^{-1} \sum_{i \in I} p_i,\ \sum_{i \in I} \Omega_i \right\rangle \\ &= {}^P \left\langle \sum_{i \in I} p_i,\ \sum_{i \in I} \Omega_i \right\rangle \\ &= \sum_{i \in I} {}^P X_i \end{aligned} \tag{4.80}$$

Commutativity and associativity trivially follow from the composition summation in precise notation. Accordingly, Figure 4.8 shows four topologically different but functionally equivalent subgraphs of the example PBN in Figure 4.7.

Implementation Hint 4.4: Control Data Representation

Due to the simplicity of the precise Cartesian composition, as given in Equation (4.80), robot control data should be represented in precise notation.

Since the PBN architecture is defined stochastically, it inherently embeds stochastic properties. Thus, well-known stochastic algorithms and methodologies are intrinsically supported. Among others, three examples relevant to legged robotics are:

1. The Cartesian data composition behavior implements the proven optimal stochastic solution of merging normal distributed sensor data into one, again normal distributed, substitute representation [Elmenreich 07].
2. A *Kalman* filter [Kalman 60] is easily realized by a stochastic model that is added as a virtual sensor. Stochastic model updates with the merged hardware and model data achieve the necessary loop closure, as sketched in Figure 2.14.
3. State-of-the-art stochastic modeling, including online ML strategies such as Gaussian Process Regression (GPR) or dynamics modeling [Umlauft 20, Beckers 22], seamlessly integrates into the stochastic sense and control processing of a PBN.

4.6.2 Rotor Orientations

In analogy to the Cartesian case, let $R_{i\in I} = \left\langle r_i, \frac{1}{4}\Omega_i \right\rangle$ be an arbitrary number of probabilistic unit rotor orientations according to Notation 4.8. By setting the behavior task function to the identity function $t(\cdot) = \text{ID}(\cdot)$, the unit rotor composition behavior is derived according to the PBN behavior calculation definition, given in Definition 4.6.

$$r_{out} = \arg\max_{r_{out}} \prod_{i\in I} \text{sat}_{R_i}(r_{out}) \tag{4.81}$$

Inserting the unit rotor satisfaction Equation (4.24), then leads to:

$$\begin{aligned}
r_{out} &= \arg\max_{r_{out}} \prod_{i\in I} \exp\left(-\frac{1}{2}\, r_{out}^T \mathrm{L}[\tilde{r}_i]^T\ \Omega_i\ \mathrm{L}[\tilde{r}_i]\, r_{out}\right) \\
&= \arg\max_{r_{out}} \exp\left(-\frac{1}{2} \sum_{i\in I} r_{out}^T \mathrm{L}[\tilde{r}_i]^T\ \Omega_i\ \mathrm{L}[\tilde{r}_i]\, r_{out}\right) \\
&= \arg\max_{r_{out}} -\frac{1}{2} \sum_{i\in I} r_{out}^T \mathrm{L}[\tilde{r}_i]^T\ \Omega_i\ \mathrm{L}[\tilde{r}_i]\, r_{out} \\
&= \arg\min_{r_{out}} \sum_{i\in I} r_{out}^T \mathrm{L}[\tilde{r}_i]^T\ \Omega_i\ \mathrm{L}[\tilde{r}_i]\, r_{out} \\
&= \arg\min_{r_{out}} r_{out}^T \left[\sum_{i\in I} \mathrm{L}[\tilde{r}_i]^T\ \Omega_i\ \mathrm{L}[\tilde{r}_i]\right] r_{out} \\
\text{s.t.: } \|r_{out}\| &= 1
\end{aligned} \tag{4.82}$$

Exploiting the definition of r_{out} being a unit rotor, i.e., $r_{out} = -r_{out}$ and $\|r_{out}\| = 1$, the minimization result is given by the eigenvector q_1 corresponding to the smallest eigenvalue λ_1 of the symmetric, positive semi-definite inner sum of matrices.

Define the matrix H as:

$$H := \sum_{i \in I} \left(\mathrm{L}[\tilde{r}_i]^T \ \Omega_i \ \mathrm{L}[\tilde{r}_i] \right) \tag{4.83}$$

Let $Q \Lambda Q^T := H$ be the ordered SVD of the matrix H, i.e.:

$$\begin{aligned} H &= Q \Lambda Q^T \\ \Lambda &= \mathrm{diag}(\lambda_1, \ \ldots, \ \lambda_{n+1}) \\ &\text{where } 0 \leq \lambda_1 \leq \cdots \leq \lambda_{n+1} \\ \text{and appropriately: } Q &= \left[q_1, \ldots q_{n+1} \right] \end{aligned} \tag{4.84}$$

It follows that:

$$r_{out} := \mathrm{col}_1(Q) = q_1 \tag{4.85}$$

In order to preserve the behavior network's commutativity and associativity, as demanded by Definition 4.6, the precision of a rotor orientation composition is defined as the above-derived inner matrix sum H, sandwiched by the result rotor's left-multiplication matrix $\mathrm{L}[r_{out}]$.

$$\Omega_{out} := \mathrm{L}[r_{out}]^T \ H \ \mathrm{L}[r_{out}] \tag{4.86}$$

This definition replicates the general additive semantics of the Cartesian precision composition. Precision always increases when composed. However, a proof of probabilistic correctness w.r.t. the Bingham distribution is left for the interested reader.

The following definition of probabilistic orientation composition can be given:

Definition 4.17: Rotor Orientation Composition

Let $\Sigma(R_1, \ R_2, \ \ldots)$ be the function describing the orientation composition behavior for arbitrary many input rotors $R_{i \in I} = \left\langle r_i, \ \frac{1}{4}\Omega_i \right\rangle$. The orientation composition function is defined as:

$$\Sigma(R_1, \ R_2, \ \ldots) := \left\langle q_1, \ \mathrm{L}\left[q_1\right]^T H \ \mathrm{L}\left[q_1\right] \right\rangle \tag{4.87}$$

With $H := \sum_{i \in I} \left(\mathrm{L}[\tilde{r}_i]^T \ \Omega_i \ \mathrm{L}[\tilde{r}_i] \right)$ and q_1 being the eigenvector corresponding to the smallest eigenvalue of H.

By defining the *normalized notation* ${}^N R_{i \in I} = {}^N \langle r_i, \ N_i \rangle$ using the normalized precision matrices $N_i := \mathrm{L}[\tilde{r}_i]^T \ \Omega_i \ \mathrm{L}[\tilde{r}_i]$, the above composition function simplifies to

$$\Sigma\left({}^N R_1, \ {}^N R_2, \ \ldots \right) := {}^N \left\langle q_1, \ \sum_{i \in I} N_i \right\rangle \tag{4.88}$$

where q_1 is the eigenvector corresponding to the smallest eigenvalue of $\sum_{i \in I} N_i$. Note that the $\mathrm{L}[\tilde{r}]$ matrices for all rotor r are still rotation operations with determinant

$\det(\mathrm{L}[\tilde{r}]) = 1$. Thus, equal to the Cartesian case, the more data are collected, the higher the resulting precision, as expected.

As expected, both commutativity and associativity of the orientation composition trivially follow from its normalized notation. Thus, the PBN graph identities presented in Figure 4.8 also hold for orientation representations. Moreover, due to the behaviors representation-independent calculation definition, these properties generally hold for all valid PBN data representations complying with Definitions 4.4 and 4.6.

Interestingly, the composition's result rotor is independent of any input rotor. Thus, the actual rotor must be entirely embedded within the one extra spare dimension of the rotational precision matrix, as constructed in Definition 4.9. Consequently, the unit rotor representation of a probabilistic orientation data vector may be dropped entirely without loss of information. An arbitrary rotor representation $R = \left\langle r,\ \frac{1}{4}\Omega_\star \right\rangle$, according to the rotor representation construction, as given in Definition 4.9, features the following normalized precision matrix N_R:

$$N_R = \mathrm{L}[\tilde{r}]^T \begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star \end{bmatrix}_{=\Omega} \mathrm{L}[\tilde{r}] \tag{4.89}$$

It can be seen that the vector $q_1 = \begin{bmatrix} 1,\ 0,\ \ldots \end{bmatrix}^T$ is an eigenvector of Ω with the respective eigenvalue of $\lambda_1 = 0$. Since $\Omega \geq 0$ is positive semi-definite, all its eigenvalue are non-negative and λ_1 must be the smallest eigenvalue. λ_1 becomes unique, if is strictly positive-definite $\Omega_\star > 0$.

Left-multiplying the rotor r with $\mathrm{L}[\tilde{r}]$ reverses the orientation represented by r, thus resulting in the identity rotor r_{ID}. However, according to the rotor construction definition in Equation (4.26), this identity rotor equals the following:

$$r_{\text{ID}} = \begin{bmatrix} \cos\left(\frac{0}{2}\right), \sin\left(\frac{0}{2}\right) e \end{bmatrix}^T = \begin{bmatrix} 1,\ 0,\ \ldots \end{bmatrix}^T = q_1 \tag{4.90}$$

It follows that:

$$N_R\, r \ = \ \mathrm{L}[\tilde{r}]^T\, \Omega\, \mathrm{L}[\tilde{r}]\ r \ = \ \mathrm{L}[\tilde{r}]^T\, \Omega\, q_1 \ = \ \mathrm{L}[\tilde{r}]^T\, \lambda_1\, q_1 \ = \ 0 \tag{4.91}$$

The rotor r is always an eigenvector of the normalized precision matrix N_R featuring the smallest possible eigenvalue of $\lambda_r = 0$.

For example, an arbitrary, planar rotor representation $R = \left\langle \left[\begin{smallmatrix} s \\ \epsilon \end{smallmatrix}\right],\ \frac{1}{4}\Omega_\star \right\rangle$, with s, ϵ, and $\Omega_\star$ being scalar values, features the following 2×2 normalized precision matrix N_R:

$$N_R = \mathrm{L}[\left[\begin{smallmatrix} s \\ -\epsilon \end{smallmatrix}\right]]^T \begin{bmatrix} 0 & 0 \\ 0 & \Omega_\star \end{bmatrix} \mathrm{L}[\left[\begin{smallmatrix} s \\ -\epsilon \end{smallmatrix}\right]] = \Omega_\star \begin{bmatrix} \epsilon^2 & -s\epsilon \\ -s\epsilon & s^2 \end{bmatrix} \tag{4.92}$$

The two eigenvectors of N_R are $q_1 = \left[\begin{smallmatrix} s \\ \epsilon \end{smallmatrix}\right]$ and $q_2 = \left[\begin{smallmatrix} -\epsilon \\ s \end{smallmatrix}\right]$. Their respective eigenvalues are $\lambda_1 = 0$ and $\lambda_2 = \Omega_\star$.

However, since such orientation representations are only used for sensor data processing (see Section 4.4.5), orientation composition is expected to occur only once throughout an entire PBN. Thus, implementing a normalized notation representation cannot be expected to offer practical benefits during operation.

4.6.3 Combined Orientation and Cartesian Data

To compose combined vector and orientation data, no special procedure is required. As derived in Section 4.4.4, the combined satisfaction function can be split into separate orientation and Cartesian satisfactions using Equation (4.55). These individual subfunctions can then be solved consecutively and independently, starting with the constraint orientations and ending with the unbound Cartesian elements. Furthermore, dividing the overall optimization problem into smaller, independent subproblems generally reduces computational costs relative to the problem size.

As recommended in Implementation Hint 4.4, Cartesian control vectors should be represented in the precise format defined in Notation 4.7. Using this representation, the satisfaction split Equation (4.55) simplifies to the following:

$$\mathrm{sat}_{P\left\langle \begin{bmatrix} r \\ p \end{bmatrix}, \begin{bmatrix} \frac{1}{4}\Omega_r & \frac{1}{2}\Omega_s^T \\ \frac{1}{2}\Omega_s & \Omega_p \end{bmatrix} \right\rangle} \left(\begin{bmatrix} u \\ v \end{bmatrix} \right) = \mathrm{sat}_{\left\langle r,\ \frac{1}{4}\left(\Omega_r - \Omega_s^T \Omega_p^{-1} \Omega_s\right) \right\rangle}(u) \cdot \mathrm{sat}_{P\left\langle p - \Omega_s \mathrm{I}[\tilde{r}]u,\ \Omega_p \right\rangle}(v) \tag{4.93}$$

Note that only the Cartesian vector is present in the precise format in the above-used combined data vector. The rotor part is represented in its standard notation, as is clearly distinguished in the two split sub-satisfactions.

4.6.4 Derivatives and Data Series

As outlined in Sections 4.4.5 and 4.4.6, the two respective Definitions, 4.12 and 4.13, declare robot control commands to be handled as data series of the controls' first derivatives. An example of such a control series has been given in Figure 4.4.

Since differentiation and integration are linear functions, the satisfaction function is invariant against those transformations. Subsequently, the behavior result optimization is invariant, too. Instead of composing the integrated actual control commands and deriving them again before being returned, it is possible to compose the signal derivatives directly. Moreover, as a composition behavior's task functional is the identity, no temporal coupling exists between data at different points in time. Therefore, the functional satisfaction optimization, given in Equation (4.56), can be entirely decomposed into individual data compositions per time interval specified by the input series. Hence, the composition of data derivative series, such as robot control commands, equals the series of time-individually composed data derivatives.

Let $\Sigma\big(\dot{X}_1(t)\,,\ \dot{X}_2(t)\,,\ \ldots\big)$ be the functional describing the composition behavior for arbitrary many input series of data derivatives. The composition functional can be defined as:

$$\Sigma\big(\dot{X}_1(t)\,,\ \dot{X}_2(t)\,,\ \ldots\big) := \Sigma\big(\dot{X}_1(t)\,,\ \dot{X}_2(t)\,,\ \ldots\big)(t) \tag{4.94}$$

Note that the left-hand-side $\Sigma()$ describes the composition functional, while the right-hand-side $\Sigma()(t)$ describes the regular composition function with its inputs evaluated at time t.

4.7 Data Abstraction

Motivated by the natural control Requirements 2.5: *"Hierarchical Control"* and 2.6: *"Model Reduction"*, as derived in Sections 2.1.5 and 2.1.6, the two architectural requirements of transformation invariance, i.e., data abstraction and control hierarchy, have been postulated in Section 4.1. A general sketch of such a hierarchical control layout has been introduced in Section 3.3.1 and visualized in Figure 3.17.

To support both above requirements, Definition 4.12 within Section 4.4.5 restricts control data to be processed in their first derivative, i.e., linearized. Furthermore, the general behavior definition (Definition 4.6 within Section 4.3) requires all control processing behaviors to be proper, information-preserving behaviors. Thus, their task function $t(\cdot)$ must be non-projective and information-preserving, i.e., left-invertible $\left(t^{-1} \circ t\right)(\cdot) = \text{ID}(\cdot)$.

Improper behaviors, which feature information-lossy projective task functions, can be used for sensor data processing if they are not composed with other identically projected data afterward. By definition, the representation space interconnection and possible abstraction, as sketched in Figure 3.17, meet all the just-mentioned PBN conditions for hierarchical, abstract robot control.

As defined in Equation (3.6), robot representation spaces in higher hierarchical layers must have lower or equal dimensionality than those below. Consequently, sensory information representations may be projected into more abstract, thus lower dimensional, higher hierarchical control layers via information-lossy improper behaviors. However, at the same time, control data flow in the opposite direction, thus always from lower dimensional spaces to higher dimensional ones. This way, control data are guaranteed to be reverse-projected solely via information-preserving proper behaviors.

Due to their clear semantic and technical relevance within a PBN, behaviors performing a data abstraction in either direction are denoted with a dedicated behavior function.

1. An Improper Projection Behavior $\mathcal{B}_{\text{projection}}(\cdot) = \mathcal{P}(\cdot)$
2. A Proper Back Projection Behavior $\mathcal{B}_{\text{back projection}}(\cdot) = \mathcal{P}^{-1}(\cdot) = \mathcal{T}(\cdot)$

According to the required architectural transformation invariance, back projection behaviors may also be called transformation behaviors $\mathcal{T} = \mathcal{P}^{-1}$.

It is important to highlight that these two contrary data abstraction behaviors only occur pairwise in reasonable robot control graphs. Such a pair always sandwiches some arbitrary PBN subgraph $\mathcal{P}^{-1} \circ \mathcal{G} \circ \mathcal{P}$. Define the so-called Projection Space (PS) as the inner, more abstract representation of such a sandwich and its outer representation as the so-called Transformation Space (TS) with $\dim(PS) \leq \dim(TS)$. Sensory data are always projected from the outer TS into the inner PS, while control data are back-projected from the inner PS into the outer TS.

Following the behavior calculation Definition 4.6, a back projection behavior $\mathcal{P}^{-1}$ is only required to implement the right inverse of its respective projection behavior $\mathcal{P}$. Thus, the entire behavior sandwich may become information-lossy, as desired from a control abstraction. An example of such an abstraction sandwich is given in Figure 4.9. The example shows the identical PBN graph previously presented in Figure 4.7 with two original behaviors replaced by a pair of a projection and a respective back projection behavior in their dedicated notations.

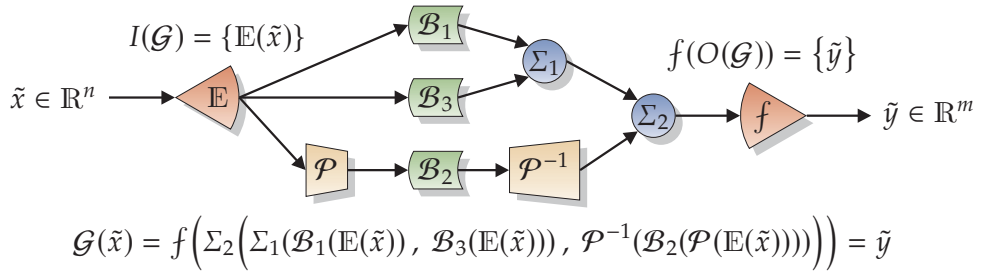


Figure 4.9: Exemplary PBN graph $\mathcal{G}$, updated from its equivalent graph as presented previously in Figure 4.7. Herein, the graph's bottommost behavior $\mathcal{B}_2$ is sandwiched by a projection behavior to its left and the respective back projection behavior to its right, i.e., $\mathcal{P}^{-1}(\mathcal{B}_2(\mathcal{P}(X)))$. Since $\mathcal{P}^{-1}$ may only implement the right inverse of $\mathcal{P}$, this behavior sandwich may be information-lossy, as expected from a control abstraction. The graph's functional equivalent underneath the graphical presentation is updated accordingly.

4.7.1 Improper Sensory Data Projection

Hierarchical data abstraction, as demanded by the architectural requirements listed in Section 4.1, is modeled as a specific, dedicated task function $t(\cdot)$. This task function implements the projection of an arbitrary state description vector, here given in TS coordinates, into a lower- or equal-dimensional representation vector in PS.

An example of such a data projection is the Work Space (WS) abstraction function $f_{WS}(\theta_{CS})$, shown in Figure 3.17 within Section 3.3.1. This function represents the possibly lossy transformation from a robot's Configuration Space (CS) into an arbitrarily lower- or equal-dimensional WS.

The respective PBN behavior can be assumed to expect only a single input. Thus, let $X_{TS} = \langle \mu_{TS},\ \Omega_{TS} \rangle = {}^{\Sigma}\langle \mu_{TS},\ \Sigma_{TS} \rangle$ be an arbitrary probabilistic data representation in TS coordinates. Assume $t^{-1}(\cdot)$ to be the possibly ambiguous right inverse of the projection behavior's task function $t(\mu_{TS}) = \mu_{PS}$. The respective PBN mean projection calculation is derived according to Definition 4.6.

$$\mu_{PS} = \underset{\mu_{PS}}{\arg\max}\ \mathrm{sat}_{X_{TS}}\left(t^{-1}(\mu_{PS})\right) \tag{4.95}$$

It directly follows that

$$\mu_{TS} = t^{-1}(\mu_{PS}) \tag{4.96}$$

and hence

$$t(\mu_{TS}) = t\left(t^{-1}(\mu_{PS})\right) = \mu_{PS} \tag{4.97}$$

Let $t'(\mu) = \frac{\partial}{\partial \mu} t(\mu)$ be the well-defined, momentary derivative of the projection task function $t(\cdot)$. The projected variance and precision matrices are determined according to the calculation properties of the PDF underlying the PBN data representation.

$$\Sigma_{PS} = t'(\mu_{TS})\, \Sigma_{TS}\, t'(\mu_{TS})^T \tag{4.98}$$

$$\Omega_{PS} = \left(t'(\mu_{TS})\, \Omega_{TS}^{-1}\, t'(\mu_{TS})^T\right)^{-1} = t'(\mu_{TS})^{-T}\, \Omega_{TS}\, t'(\mu_{TS})^{-1} \tag{4.99}$$

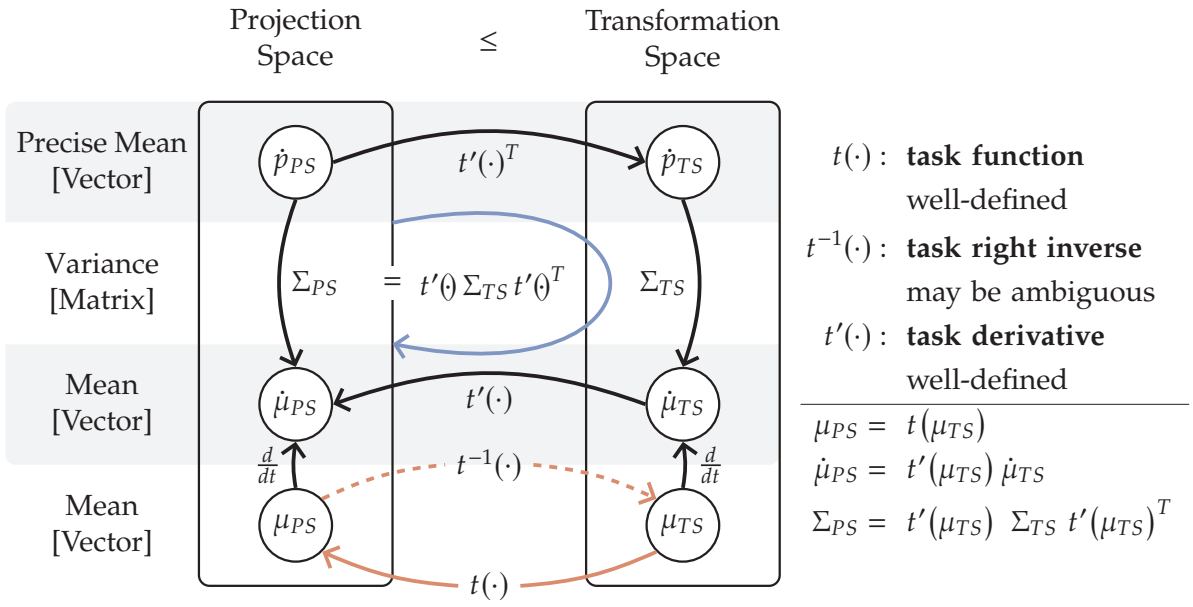


Figure 4.10: Illustration and derivation of the correlations between PBN data across two representation spaces of different dimensionality. To the left, a graph of projections visualizes the kinematic coupling between each representation's mean μ, mean derivative $\dot{\mu}$, variance Σ, and precise mean p. The projection behavior's task function is represented as a red arrow between the spaces' mean vectors. Its possibly ambiguous right inverse is drawn as a dashed red arrow in the opposite direction. The well-defined projection of the variance matrix is constructed and highlighted in blue. On the right, the inter-space projections and their respective projection equations are listed as sketched.

Figure 4.10 sketches the individual data correlations between the original TS and the projected PS. The relation diagram covers the mean data, their respective variance matrices, and the precise mean representations.

A general improper projection behavior is defined as follows:

Definition 4.18: Improper Data Projection

Let $X \in \mathcal{X}$ be an arbitrary PBN data in either regular or sigma notation, i.e., $X = \langle \mu,\ \Omega \rangle = {}^{\Sigma}\langle \mu,\ \Sigma \rangle = {}^{\Sigma}X$. Also define $t(\cdot)$ as an arbitrary, potentially projective, behavior task function, according to Definition 4.6.

An improper data projection behavior $\mathcal{P}$ is defined as:

$$\mathcal{P}(X) := {}^{\Sigma}\left\langle t(\mu)\,,\ \left(t'(\mu)\,\Omega^{-1}\,t'(\mu)^T\right)^{-1} \right\rangle \tag{4.100}$$

$$\text{or } {}^{\Sigma}\left\langle t(\mu)\,,\ t'(\mu)^{-T}\,\Omega\,t'(\mu)^{-1} \right\rangle \tag{4.101}$$

$$\mathcal{P}({}^{\Sigma}X) := \left\langle t(\mu)\,,\ t'(\mu)\,\Sigma\,t'(\mu)^T \right\rangle \tag{4.102}$$

Equation (4.102) shows that a projection behavior, although improper, is well-defined when its data are processed in sigma notation. The projection behavior is furthermore

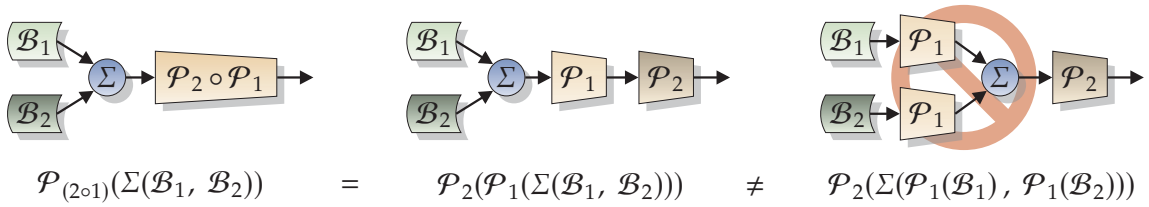


Figure 4.11: Three PBN example graphs with similar topologies. The two graphs on the left and middle are functionally identical, thus demonstrating the projection behavior's invariance against consecutive projection concatenation. However, the right graph may be unequal to the others due to the projection's volatility against PBN composition. Since the projection behaviors $\mathcal{P}_1$ may be information-lossy, composing data after identical projection is forbidden in a PBN without loss of functionality or generality.

well-defined for regular data representation if both input and output precision matrices are invertible. In the case of $t(\cdot)$, and thus also $t'(\cdot)$, being invertible, the projection behavior becomes even a proper, information-preserving behavior.

The following two important properties hold for improper data projections:

1. Invariance Against Concatenation:
   Improper data projection is invariant against concatenation and, thus, suitable for consecutive hierarchical abstraction, as required.

2. Variance (Volatility) Against Composition:
   Improper data projection is *not* invariant against composition.

Let $\mathcal{P}_2(\cdot)$ and $\mathcal{P}_1(\cdot)$ be improper PBN data projections with respective task functions $t_2(\cdot)$ and $t_1(\cdot)$. With $X = {}^{\Sigma}\langle \mu,\ \Sigma \rangle$ being a probabilistic data vector in the definition space of $t_1(\cdot)$, the following composition equality holds:

$$\begin{aligned}
(\mathcal{P}_2 \circ \mathcal{P}_1)(X) &= {}^{\Sigma}\left\langle (t_2 \circ t_1)(\mu)\,,\ \frac{\partial\,(t_2 \circ t_1)}{\partial \mu}(\mu)\,\Sigma\,\frac{\partial\,(t_2 \circ t_1)}{\partial \mu}(\mu)^T \right\rangle \\
&= {}^{\Sigma}\left\langle t_2(t_1(\mu))\,,\ t_2'(t_1(\mu))\,t_1'(\mu)\,\Sigma\,t_1'(\mu)^T\,t_2'(t_1(\mu))^T \right\rangle \\
&= \mathcal{P}_2\left({}^{\Sigma}\left\langle t_1(y)\,,\ t_1'(\mu)\,\Sigma\,t_1'(\mu)^T \right\rangle\right) \\
&= \mathcal{P}_2(\mathcal{P}_1(X))
\end{aligned} \tag{4.103}$$

The volatility against composition directly follows from the information loss within the data projective task function $t(\cdot)$, as described in Definition 4.6. A graphical and equivalent functional representation of the above invariance and variance properties is presented in Figure 4.11.

Due to the above-stated properties of improper data projection, the simplicity of Equation (4.102), and the data composition Definitions 4.16 and 4.17, the following guidelines on sensory data processing within a PBN architecture can be stated:

Implementation Hint 4.5: Efficient Sensory Data Processing

1. Sensory data should be composed as early as possible in their most raw representation but must not be improperly projected before composition, as highlighted in Figure 4.11.
2. Before composition, sensory data should be represented in standard or precise notation to simplify their respective compositions.
3. If not composed anymore, sensory data should be represented in sigma notation to linearize respective projections into more abstract, higher hierarchical control layers.

Note that a precision matrix Ω_R of an orientation representation before composition is not invertible. Thus, $\Sigma_R = \Omega_R^{-1}$ is not well-defined. However, in this case, Ω_R is structured as constructed with $\Omega_R = \begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star \end{bmatrix}$, and its respective construction inverse $\Sigma_R = \Omega_R^{-1} := \begin{bmatrix} 0 & \cdots \\ \vdots & \Omega_\star^{-1} \end{bmatrix}$ can be used, as described in Definition 4.9.

4.7.2 Proper Control Data Back Projection

Following Definitions 4.12 and 4.13, control data within a PBN are represented as time series of control derivatives. Thus, a transformation behavior is a functional behavior. In order to correctly determine a series of back-projected control derivatives, the optimization in Definition 4.14 has to be solved.

However, for simplicity reasons, it is assumed that no bidirectional temporal coupling is present within the behavior's task functional. This means that when writing the result series' data as a single, stacked data vector, the functional's Jacobian matrix should be respectively block triangular. With this assumption, each control time step can be optimized individually, in either forward or backward temporal order. Thus, the functional transformation behavior can be treated as a standard behavior, following the calculation Definition 4.6. The behavior's task function will thus be called consecutively for each time step in its input series.

As initially introduced and visualized in Figure 4.9, data abstraction always occurs as a pair of sense input projection and control output transformation, i.e., back projection. Thus, in contrast to the previous data projection, the respective transformation behavior task function $t(\cdot)$ may not be well-defined. It may be possible to find multiple solutions to identical inputs. However, the task function's left inverse $t^{-1}(\cdot)$ becomes mandatorily well-defined. An example of such a back projection would be the possibly not well-defined, right inverse twist mapping J_{CW}^{-1} in Figure 3.17. Two conclusions can be drawn from the definitions of the task function and its left inverse.

As desired, the well-definition of the task function's left inverse satisfies the requirement of a proper, information-preserving behavior $\left(t^{-1} \circ t\right)(\cdot) = \text{ID}(\cdot)$, as stated in Definition 4.6. Unfortunately, the reverse concatenation $\left(t \circ t^{-1}\right)(\cdot)$ may not be well-defined. Consequently, the projective result mean calculation in Equation (4.97) may also not be well-defined in a transformation behavior. Instead, the behavioral mean optimization

Equation (4.9) within Definition 4.6 becomes well-defined and must be solved as a real optimization problem to achieve proper back projection.

Due to Definition 4.12 and Hint 4.4, PBN control transformation can be limited to only process precise Cartesian derivative data. Let ${}^{P}\dot{X}_{PS} = {}^{P}\langle \dot{p}_{PS},\ \Omega_{PS} \rangle$ with $\dot{\mu}_{PS} = \Omega_{PS}^{-1}\ \dot{p}_{PS}$ be such an arbitrary Cartesian control input representation in some abstract PS. Assume $t^{-1}(\mu_{TS}) = \mu_{PS}$ to be a transformation behavior's principle left inverse task function that projects a non-derivative control result μ_{TS} in TS coordinates back to a non-derivative input μ_{PS} in PS coordinates. The control transformation behavior's actual left inverse task function is the principle function's time derivative. A visualization of this function and its transformation from PS into TS is given in Figure 4.12. Since the actual transformation function may be ambiguous, its well-defined inverse is:

$$\tfrac{d}{dt} t^{-1}(\mu_{TS}) = \tfrac{\partial}{\partial \mu_{TS}} t^{-1}(\mu_{TS})\ \dot{\mu}_{TS} = J_{PT}^{-1}\ \dot{\mu}_{TS} = \dot{\mu}_{PS} \tag{4.104}$$

With $J_{PT}^{-1} = \frac{\partial}{\partial \mu_{TS}} t^{-1}(\mu_{TS})$ being the well-defined left inverse Jacobian matrix.

The behavior calculation optimization (Equation (4.9)) can now be rewritten according to the Cartesian satisfaction Equation (4.20).

$$\begin{aligned}
& \arg\max_{\dot{\mu}_{TS}}\ \mathrm{sat}_{\langle \Omega_{PS}^{-1}\dot{p}_{PS},\ \Omega_{PS} \rangle}\left(\tfrac{d}{dt} t^{-1}(\mu_{TS})\right) \\
\Leftrightarrow \quad & \arg\min_{\dot{\mu}_{TS}}\ \frac{1}{2}\left(J_{PT}^{-1}\ \dot{\mu}_{TS} - \Omega_{PS}^{-1}\ p_{PS}\right)^{T} \Omega_{PS}\left(J_{PT}^{-1}\ \dot{\mu}_{TS} - \Omega_{PS}^{-1}\ p_{PS}\right) \\
\Leftrightarrow \quad & \arg\min_{\dot{\mu}_{TS}}\ \frac{1}{2}\ \dot{\mu}_{TS}^{T}\ J_{PT}^{-T} \Omega_{PS}\ J_{PT}^{-1}\ \dot{\mu}_{TS}\ -\ p_{PS}^{T}\ J_{PT}^{-1}\ \dot{\mu}_{TS}
\end{aligned} \tag{4.105}$$

Note that the optimization (4.105) is well-defined.

Analog to Equation (4.99), the precision transformation follows from its underlying PDF calculus.

$$\Omega_{TS} = J_{PT}^{-T} \Omega_{PS}\ J_{PT}^{-1} \tag{4.106}$$

Assuming J_{PT}^{-1} is constant, Equation (4.105) is quadratic and can be solved by setting its derivative to zero.

$$\begin{aligned}
& \frac{\partial}{\partial \dot{\mu}_{TS}}\ \frac{1}{2}\ \dot{\mu}_{TS}^{T}\ J_{PT}^{-T} \Omega_{PS} J_{PT}^{-1}\ \dot{\mu}_{TS}\ -\ \dot{p}_{PS}^{T} J_{PT}^{-1}\ \dot{\mu}_{TS}\ =\ 0 \\
\Leftrightarrow \quad & \dot{\mu}_{TS}^{T}\ J_{PT}^{-T} \Omega_{PS} J_{PT}^{-1}\ =\ \dot{p}_{PS}^{T} J_{PT}^{-1}
\end{aligned} \tag{4.107}$$

Using the precise mean definition $\dot{p}_{TS} = \Omega_{TS}\ \dot{\mu}_{TS} = J_{PT}^{-T} \Omega_{PS}\ J_{PT}^{-1}\ \dot{\mu}_{TS}$, it follows that:

$$\dot{p}_{TS} = J_{PT}^{-T} \dot{p}_{PS} \tag{4.108}$$

Note that J_{PT}^{-T} is well-defined. An illustration of the individual correlations between the origin and back-projected mean data, their precision matrices, and precise mean representations is given in Figure 4.12.

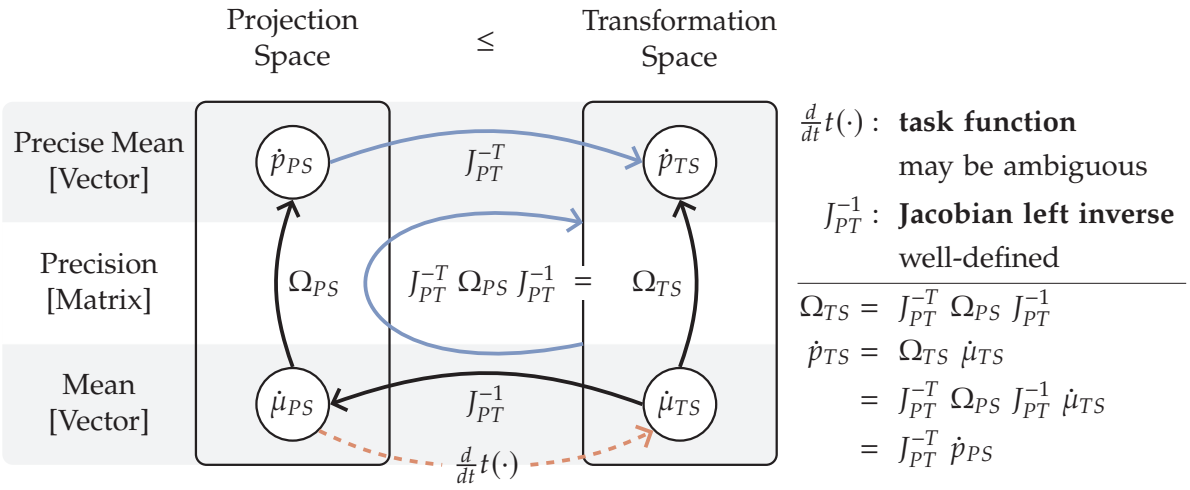


Figure 4.12: Illustration and derivation of the correlations between PBN data across two representation spaces of different dimensionality. To the left, a graph of transformations visualizes the kinematic coupling between each representation's mean derivative $\dot{\mu}$, precision Ω, and precise mean derivative $\dot{p}$. The transformation behavior's possibly ambiguous task function is the dashed red arrow. In contrast, the two well-defined transformations of the precision and precise mean are highlighted in blue. On the right, the inter-space transformations are listed, and the transformed precise mean calculation is derived as given in Equation (4.108).

Definition 4.19: Proper Control Transformation

Let ${}^{P}\dot{X} \in \mathcal{X}$ be an arbitrary Cartesian PBN control data in precise notation, i.e., ${}^{P}\dot{X} = {}^{P}\langle \dot{p},\ \Omega \rangle$. Assume $t(\cdot)$ to be an arbitrary, potentially ambiguous principle transformation task function, according to Definition 4.6.

A proper control data transformation behavior, i.e., a back projection behavior, $\mathcal{T} = \mathcal{P}^{-1}$ is defined as:

$$\mathcal{T}\left({}^{P}\dot{X}\right) := {}^{P}\left\langle J^{-T}\dot{p},\ J^{-T}\Omega J^{-1} \right\rangle \tag{4.109}$$

With $J^{-1} := \frac{\partial}{\partial \mu} t^{-1}(\mu)$ being the well-defined left inverse Jacobian matrix.

The following two architectural significant properties hold for proper, precise control data transformation:

1. Invariance Against Concatenation:
   Proper Cartesian control transformation is invariant against concatenation and, thus, suitable for consecutive hierarchical control abstraction. The invariance directly follows from the linearity of Equation (4.109).

2. Invariance Against Composition (Control Commutativity):
   Proper Cartesian control transformation is invariant against composition and, thus, suitable for arbitrary hierarchical control grouping. The control commutativity directly follows from the composition's distributivity with linear transformations. See Equation (4.80) for reference.

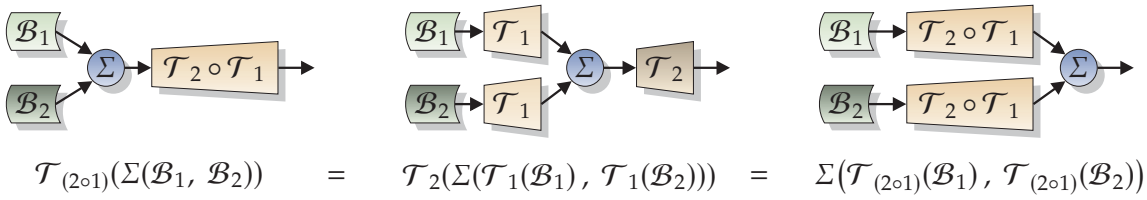


Figure 4.13: Four topologically different but functionally identical PBN example graphs, thus visualizing the control transformation's invariance against concatenation and composition.

Figure 4.13 illustrates these two important invariances at the hand of three topologically different but functionally identical PBN graphs.

Although well-defined, there are two important issues left within Definition 4.19:

1. First, the task function's left inverse Jacobian matrix J_{PT}^{-1} is implicitly assumed to stay constant throughout the transformation time horizon.
2. Second, unlike the architectural requirements, J_{PT}^{-1} is unfortunately expected to be non-singular.

As introduced in Equation (4.104), the definition of the task Jacobian depends on the non-derivative control mean $J_{PT}^{-1} = \frac{\partial}{\partial \mu_{TS}} t^{-1}\big(\mu_{TS}\big) = J_{PT}^{-1}\big(\dot{\mu}_{TS}\big)$. Unfortunately, the control mean itself μ_{TS} is intended to be altered by the desired resulting control derivative. Thus, throughout the control's time horizon $(t-1,\ t]$, μ_{TS} should correctly be assumed to be probabilistically integrated according to Definition 4.15.

$$\mu_{TS} \;=\; \oint_{t-1}^{t} \Big({}^{P}\big\langle \dot{p}_{TS},\ \Omega_{TS} \big\rangle(t) \Big) \tag{4.110}$$

$$\text{thus:}\quad J_{PT}^{-1} \;\mapsto\; J_{PT}^{-1}\Big({}^{P}\big\langle \dot{p}_{TS},\ \Omega_{TS} \big\rangle(t) \Big) \tag{4.111}$$

Consequently, Equation (4.108) becomes non-linear in the function of $J_{PT}^{-1}(\cdot)$.

However, for small control cycles, i.e., at hierarchically lower control layers, the change of J_{PT}^{-1} within this small time horizon may be neglected. Assume μ_{t-1} to be the prior control mean. Then, J_{PT}^{-1} can be approximated constant as $J_{PT}^{-1} \approx J_{PT}^{-1}\big(\mu_{t-1}\big)$.

Whenever this simplification becomes unfeasible, e.g., in the case of longer cycle times, the order of simplification has to be reduced to preserve feasibility. As the Jacobian function $J_{PT}^{-1}(\cdot)$ depends on the integral of the desired probabilistic result ${}^{P}\big\langle \dot{p}_{TS},\ \Omega_{TS} \big\rangle$, Equation (4.109) becomes an Ordinary Differential Equation (ODE) in time with the above simplification implementing an explicit *Euler* integration. Thus, nearby alternative solutions are implicit *Euler* integration or Runge-Kutta (RK) integration, such as the most popular RK4 variant.

Unfortunately, none of the above methodologies can solve the singularity problem. Explicit methods fail for J_{PT}^{-1} being singular, while implicit methods fail when J_{PT}^{-1} is intended to become singular. However, derived from the natural control Requirement 2.2, singularity support is an architectural requirement, as initially listed in Section 4.1.

To overcome algorithmic singularity locks, the transformation minimization given in Equation (4.105) must not be simplified to a quadratic optimization. In general, the time derivative of the transformation task function does not need to be a constant Jacobian matrix, as assumed in Equation (4.104). For a position-controlled robot with an arbitrary kinematic transformation task function $t^{-1}(\cdot) = k(\mu)$, the following general derivatives can be formulated:

$$t^{-1}(\cdot) : \quad \mu_{TS} = k(\mu) \tag{4.112}$$

$$\tfrac{d}{dt}t^{-1}(\cdot) : \quad \dot{\mu}_{TS} = \tfrac{d}{dt}k(\mu) = J_k(\mu)\,\dot{\mu} \tag{4.113}$$

$$\tfrac{d^2}{dt^2}t^{-1}(\cdot) : \quad \ddot{\mu}_{TS} = \tfrac{d^2}{dt^2}k(\mu) = J_k(\mu)\,\ddot{\mu} + \left(\tfrac{\partial}{\partial\mu}J_k(\mu)\,\dot{\mu}\right)\dot{\mu} \tag{4.114}$$

If not kept constant, the kinematic Jacobian matrix $J_k(\mu)$ is a function in μ. Thus, the optimization Equation (4.105) becomes a quartic (4th-degree) polynomial. When solved via gradient descent approximation, the optimization's gradient not only contains the (simplified constant) Jacobian matrix but the entire acceleration equation $\ddot{\mu}_{TS}$. Note that the Jacobian derivative within the acceleration Equation (4.114) is a third-rank tensor and describes a quadratic function in $\dot{\mu}$.

Derivative or gradient vectors are orthogonal to the surfaces from which they originate. Therefore, if not entirely vanishing, the Jacobian derivative tensor fills the Jacobian singularities, if any. In the usual robotic case with rotational joints where singularities do occur, they can always be resolved. Thus, in all relevant cases, such a 4th-degree optimization always features approximable local minima, even if J_{PT}^{-1} is singular. A detailed derivation and analysis of a 4th-degree optimization within a singular position is published in [*Vonwirth 24b*], which is based on the preliminary work in [*Vonwirth 22*].

As desired by the natural control Requirement 2.2 in Section 2.1.2, singularity support can be achieved within a PBN at the cost of increased computational effort while maintaining the architectural local-only optimality criteria formulated in Definition 4.6. Due to the argumentation on Theorem 4.2 within Section 4.3.1, both earlier stated invariances still hold in this more complex setup.

However, different PBN topologies, as shown in Figure 4.13, are only guaranteed to feature identical local minima but may not result in identical ones. The identity of the local minima, of course, assumes identical simplifications being applied within the individual transformations.

4.8 Virtual 1D Position Control Example

In the upcoming chapter, the yet well-known topology of the example PBN graph $\mathcal{G}$, as lastly illustrated in Figure 4.9, is used to control the planar bipedal robot Carl. Due to the high dimensionality required to control Carl, a detailed illustration and discussion at the PBN architectural level is not possible with the actual control implementation. Hence, to provide a better understanding of how the individual behavior nodes within this later control graph interact with each other, a virtual control experiment is presented within this section using the most simplified possible version of that identical control topology. References to the respective control components deployed on Carl throughout Chapter 5 are provided for all relevant behavior nodes.

The most simple replacement for Carl is a virtual, one-dimensional robot R without any noise, intrinsic, or extrinsic dynamics. Thus, the robot R is ideally and immediately position-controlled $R = \tilde{y} = \int\left(\dot{Y}(t)\right)$, and its state is perfectly observed $X = {}^{\Sigma}\langle R,\ 0\rangle$.

Figure 4.14 sketches the entire virtual control system setup, covering (a) the entire robot control loop and (b) its stepwise execution visualization. The formal topology definition of the PBN control graph $\mathcal{G}$ is given by Equation (4.2) within Section 4.3. The control's behavior nodes and their respective later real-world control references are:

$\mathcal{B}_1$: Trajectory Tracking: **Squat Motion in Section 5.3.1**
This behavior wants to track the position trajectory $\left[\begin{smallmatrix} t \\ R \end{smallmatrix}\right] = \left[\begin{smallmatrix} 1 & 2 & 3 & 4 & 6 & 8 & 9 \\ 1.25 & 2.25 & 2.75 & 3 & 3.5 & 3.5 & 3.5 \end{smallmatrix}\right]$ with an end position tolerance of $\sigma = 0.25$. It becomes inactive for $t > 9$.

$\mathcal{B}_3$: Aiming: **Compliant Pose Control in Section 5.1.3**
Activated at discrete simulation time $t \geq 6$, the behavior $\mathcal{B}_3$ aims to achieve the constant robot position $R = 1$ with a tolerance of $\sigma = 0.75$ within one control cycle of constant duration $T = 2$. Its probabilistic result series $\dot{Y}(t)$ is calculated as:

$$\dot{Y}(t) = \left\{t + T,\ {}^{P}\left\langle (1 - R)\,\tfrac{T}{\sigma^2},\ \tfrac{T^2}{\sigma^2}\right\rangle\right\} \tag{4.115}$$

$\mathcal{B}_2$: Avoidance: **Ankle Strategy Balancing in Section 5.2.2**
This behavior continuously tries to avoid reaching the projected position $p = 0$ with a tolerance of $\sigma = \frac{1}{3}$ within its projection space, defined by the projection behavior $\mathcal{P}$. $\mathcal{B}_2$ stays inactive if $p > \sigma$. Its result series $\dot{P}(t)$ is calculated as:

$$\dot{P}(t) = \begin{cases} \left\{t + T,\ {}^{P}\left\langle (\sigma - p)\,\tfrac{T}{p^2},\ \tfrac{T^2}{p^2}\right\rangle\right\} & if\ \ |p| < \sigma \\ \left\{t + T,\ {}^{P}\langle 0,\ 0\rangle\right\} & else \end{cases} \tag{4.116}$$

Where the behavior's cycle time is $T = 1$, and t is the momentary discrete time.

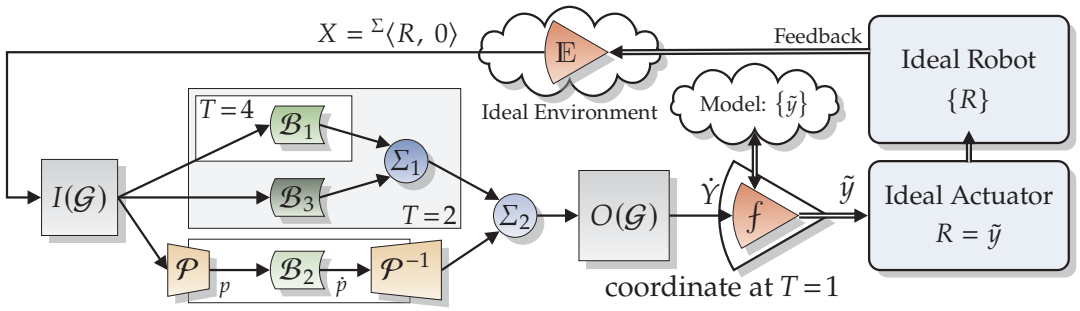
$\mathcal{P}$: Projection: **Support Work Space in Section 5.2.1**
This sense projection behavior $\mathcal{P} = \mathcal{B}_5$ maps R into its projection space $p = 1 - \frac{R}{3}$.

$\mathcal{P}^{-1}$: Back Projection: **Configuration Space in Section 5.2.1**
This control transformation behavior $\mathcal{P}^{-1} = \mathcal{B}_4$ realizes the respective control back projection according to $\mathcal{P}$, i.e., $\dot{y} = \frac{d}{dt}\left(3 - 3p\right) = 3\dot{p}$.

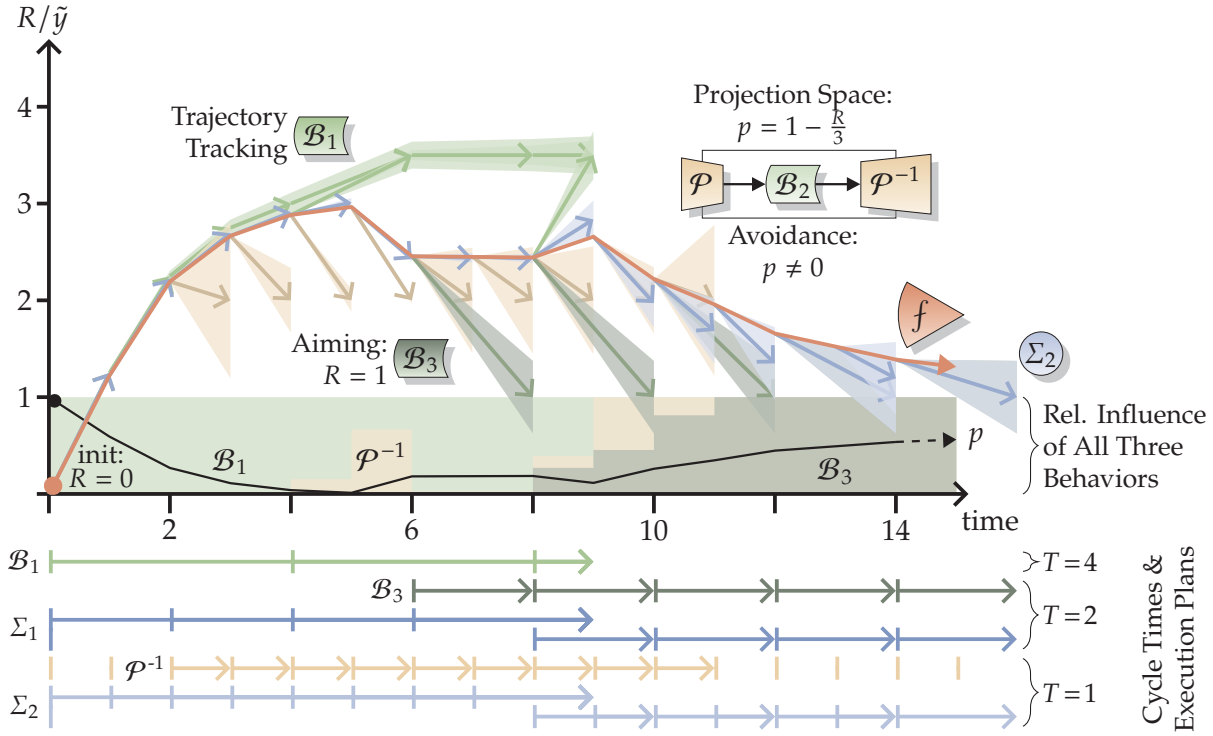
Σ : Composition:
$\Sigma_1 = \mathcal{B}_6$ and $\Sigma_2 = \mathcal{B}_7$ are regular PBN control compositions.

In this example, seven behavior nodes are executed at three different frequencies, including one representation projection and two control groups. Thus, the PBN architecture's overall functionality and compliance with all architectural requirements listed in Section 4.1 is demonstrated.

In the upper part of Figure 4.14b, several graphs plot all planned control series of all relevant behaviors, i.e., $\mathcal{B}_1$, $\mathcal{B}_3$, $\mathcal{P}^{-1}$, Σ_2, and the eventual probabilistic integration $\int$. For improved readability, only the first planned control derivative of the composition Σ_2



(a) Closed Virtual Robot Control Loop Example with Integrated PBN Control Graph $\mathcal{G}$



(b) Stepwise Visualization of the PBN Control Execution and the Resulting Robot Value R

Figure 4.14: Visualization of an exemplary, one-dimensional, PBN-controlled robot R. In order to highlight the undisturbed internals of the control graph, the virtual, so-called robot is a scalar number without intrinsic or extrinsic dynamics. On top, graphic (a) sketches the entire virtual robot control loop setup as motivated in Figure 2.14 within Section 2.4. The PBN control graph is embedded on the lower left of the loop. Below, Figure (b) visualizes the control simulation within the discrete time interval $t \in [0..14]$. Plotted are the means and standard deviations of the control-relevant nodes, their influences, and the planned control series in time. The overall control result, i.e., the robot R, equals the result of the probabilistic integration node $\int$. It is, therefore, plotted as the red arrow without a surrounding 1-σ-environment. The respective projection value $p = \mathcal{P}(X) = 1 - \frac{R}{3}$ is similarly plotted in black.

result series is drawn. Since the control is computed in its first derivative, all planned control series are represented as chains of changes in the robot R and time t. These chains are visualized by consecutive arrows of the planned mean derivatives embedded in their cumulating preplanned 1-σ-environments. The eventual composed control result $\tilde{y} = R$ is given as the pure red arrow, starting at the initial state of $R = 0$. Similarly, the respective projection value $p = \mathcal{P}(X) = 1 - \frac{R}{3}$ is plotted in black.

The behaviors' relative influences on the eventual control are visualized as background coloring between the R values zero and one. The influence of $\mathcal{P}^{-1}$ therein equals the influence of behavior $\mathcal{B}_2$. Note that the relative influence depends on the controls' precision values, which are the squared inverses of the represented standard deviations.

As described in Figure 4.14a, the example control graph $\mathcal{G}$ is organized in three hierarchical levels and one projection space. The projection space subgraph $\mathcal{P}^{-1} \circ \mathcal{B}_2 \circ \mathcal{P}$ is executed at the shortest base duration of $T = 1$. Both other control behaviors, $\mathcal{B}_3$ and $\mathcal{B}_1$, are executed at the longer cycle times of $T = 2$ and $T = 4$, respectively. The individual update rates and planned control horizons of all control-relevant modules are sketched below the robot axis of Figure 4.14b along the shared time axis.

The robot-time graph of Figure 4.14b shows that the red trace of the simulated robot scalar R follows the behavior controls according to their respective influences as expected. Initially, behavior $\mathcal{B}_1$ is solely active at a relatively high precision. Thus, the robot R perfectly follows the desired trajectory of $\mathcal{B}_1$. With R approaching the value 3 and thus p approaching zero within the projection space, behavior $\mathcal{B}_2$ becomes active with increasing precision. In the projection space, $\mathcal{B}_2$ aims to push p away from its undesired origin. Thus, outside the projection space, $\mathcal{P}^{-1}$ desires to shrink R. After $\mathcal{B}_2$ succeeds with its desire at time $t = 6$, the robot R continues to follow the trajectory derivatives of $\mathcal{B}_1$ but at a degraded value. This behavior occurs due to the long cycle time of $\mathcal{B}_1$, which will only be reevaluated at time $t = 8$. Eventually, behavior $\mathcal{B}_3$ exponentially drives the robot R toward its aim of $R = 1$. As desired, the value of $\tilde{y} = R$ always stays within the 1-σ-environment of the last composition's control Σ_2 before integration.

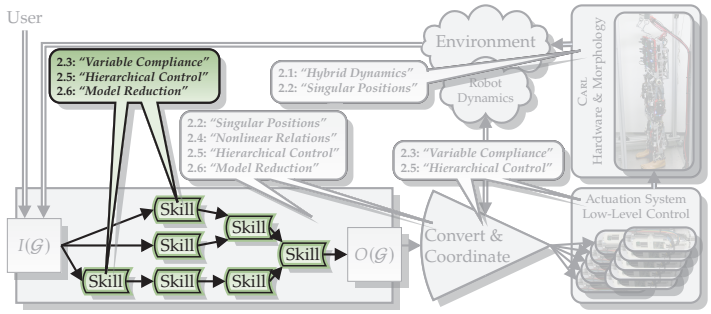
4.9 Conclusion

This chapter developed a probability-based control decomposition strategy in the style of BB robot control. The novel architecture is organized in so-called *Probabilistic Behavior Networks* (PBNs), derived from only two axiomatic definitions of *data representation* and *calculation*. PBNs firstly unify and strictly structure BB control algebraically, thus ensuring important control properties, such as *commutativity* and *associativity*.

The PBN architecture satisfies all the relevant natural control requirements derived throughout Section 2.1. At the same time, it entirely complies with the demands of the novel low-level control philosophy introduced in Chapter 3. Thus, with the definition of a proper and suitable high-level control framework, this chapter significantly contributes to this thesis' overall aim, as stated in Theorem 1.1: *"Natural Control"*. The architecture's suitability has been demonstrated through a small virtual control example, whose PBN control topology has iteratively been developed throughout this chapter.

A real-world scenario following the identical control topology is presented in the upcoming chapter using the bipedal robot Carl.

5. Natural Squats – PBN-Based Hybrid Impedance Control on CARL



The foundations of natural control, stated in Theorem 1.1, have been laid in the previous chapters. Chapter 2 identified essential characteristics of an undisturbed human gait and derived six fundamental natural control requirements. These requirements were then assigned to the individual subsystems and techniques involved in robot control. Antagonistic muscular control, a novel, bio-inspired, low-level, hybrid impedance control strategy, was developed in Chapter 3. Chapter 4 introduces the novel concept of Probabilistic Behavior Networks (PBNs), a stochastic-based high-level control decomposition. Thus, this chapter provides the last missing step toward the aim of natural control. It combines all the above foundations into one overall natural robot control system. The combined control is deployed and evaluated on real-world hardware using the bipedal robot CARL[1] [Schütz 17][Schütz 20a].

CARL's human-like morphology and actuation hardware have been introduced in Sections 3.1.3 and 3.1.2. A first proof of concept of the novel antagonistic muscular control principle has been demonstrated in Section 3.5. However, this first evaluation was executed as stand-alone experiments without further higher-level control. Thus, to finalize the proof of concept of the herein-proposed novel natural control philosophy, the antagonistic muscular control is integrated into the high-level PBN architecture. Experiments on CARL demonstrate their combined functionality.

[1] CARL (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU. URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025)

Unfortunately, a proof of concept by executing a planar walking gait is not possible at the current hardware development stage of Carl. Due to the absence of an upper body and the heavy actuators' placement, most of the robot's total mass is in the legs. This mass distribution has two immediate consequences for the robot's dynamics. First, the estimated Center of Mass (CoM) is below the hip joints and thus significantly depends on the legs' configurations. Second, fast alternating accelerations of the two legs cause enormous parasitic torques perpendicular to the legs' direction of motion, i.e., outside the robot's planar environment. Hence, mechanical reconstruction of Carl, including a supporting frame to compensate for non-planar forces, is needed to compensate for those issues. However, upgrading the robot's hardware to allow experimental validation of real-world walking is beyond the scope of this thesis and left for future work. Instead, the proof of concept is given at the hand of a squat motion, which is less vulnerable to the described non-planar disturbances.

Chapter Outline

First, Section 5.1 defines the necessary integration of antagonistic muscular control, defined in Section 3.2.3, into the PBN architecture developed throughout Chapter 4. Afterward, the most essential skill of all bipedal systems, upright balancing, is presented in Section 5.2. Section 5.3 combines horizontal balancing with mostly vertical-oriented spring leg behavior. The result is a compliant, whole-body squat motion autonomously executed by the planar, bipedal robot Carl. Critical analyses of the recorded robot motion and control data, the applied control system, and the robot hardware are provided throughout all sections. Finally, this chapter is concluded in Section 5.4.

5.1 Antagonistic Muscular Control via Probabilistic Behavior Networks

The PBN architecture is intentionally motivated to serve the needs of the antagonistic muscular control. However, the PBN behavioral calculation rules in Definitions 4.6 and 4.19 introduce one crucial requirement w.r.t. proper control data abstraction. A proper control transformation behavior must feature a well-defined left inverse of its actually desired transformation function. With this restriction, the kinematic coupling graph between Work Space (WS), Configuration Space (CS), and Actuation Space (AS), as introduced in Figure 3.17, has to be slightly reorganized. Figure 5.1 highlights the required PBN control data and their data flow from an abstract PBN WS through the interfacing CS into the hardware-representing AS.

Since a proper control transformation requires a well-defined inverse, the antagonistic muscular control data flow is best described in its reverse direction, i.e., from the hardware-representing AS backward into an abstract WS PBN control graph. Figure 5.1 represents this reverse data flow from right to left.

As discussed in Section 3.2.3, the antagonistic muscular control expects force and damping control commands. In the AS representation, these are the force vector F_{AS} and the non-negative, diagonal damping matrix D_{AS}, respectively. Following the

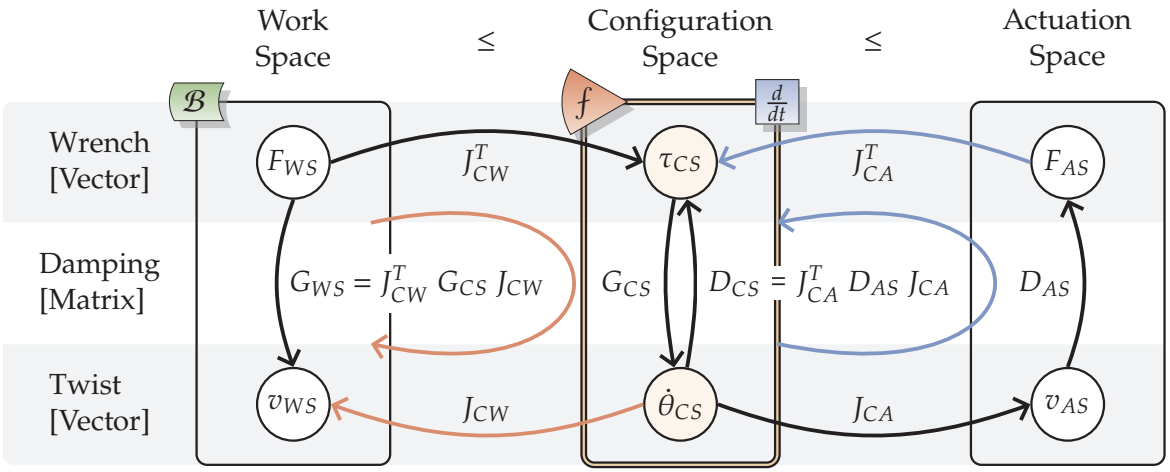


Figure 5.1: Sketch of the control-relevant data flows from WS through CS into AS. The control-relevant properties are the wrench and twist vectors and the damping or responsiveness matrices. As indicated, the WS is part of the PBN control graph. Its control data flow into the CS is colored in red. As already shown in Figure 3.17, the Jacobian transformation's well-defined direction reverses within the CS. Thus, the probabilistic integration interfacing the PBN graph is located within the CS. Since the antagonistic muscular control principle is based on velocity rather than position, the probabilistically integrated controls are derived again. Thus, a twist control vector and a responsiveness matrix are converted into a wrench vector and a corresponding damping matrix. The CS control distribution to the AS-represented actuation system is colored in blue. It is realized by optimizations, as introduced in Section 3.4.

equations (3.13) and (3.14) within Section 3.4, both controls are determined by individual optimizations according to their respective reverse transformations.

$$\tau_{CS} \stackrel{!}{=} J_{CA}^T\, F_{AS} \tag{5.1}$$

$$D_{CS} \stackrel{!}{=} J_{CA}^T\, D_{AS}\, J_{CA} \tag{5.2}$$

Figure 5.1 highlights these two reverse transformations in blue.

As already shown in Figure 3.17, the Jacobian's well-defined transformation directions reverse at the CS. In order to carry control data from the CS backward into an arbitrary WS, the control data have to follow this transformation reversion. Thus, damping has to be replaced with its inverse, the so-called responsiveness $G_{CS} = D_{CS}^{-1}$, and torque has to be replaced with rotational velocity $\dot{\theta}_{CS} = G_{CS}\tau_{CS}$. The velocity and responsiveness transformations from CS backward into WS are well-defined. Figure 5.1 highlights these reverse transformations in red.

Due to the required non-linear inversion operation of the damping matrix, any higher-level PBN control graph has to be exited before this operation. Thus, the minimal coordinates of the CS define the PBN interface layer.

Note that this mandatory consequence is only due to the Cartesian data representation. However, inversion is linearized when using a hyperbolic projection space, e.g., Conformal Geometric Algebra (CGA), for data representation. Investigations in robot and control representations using higher-dimensional projective algebras, such as Projective Geometric Algebra (PGA) or CGA, seem promising for future research.

To the left of the CS in Figure 5.1, a PBN graph will coordinate arbitrary WS control behaviors. Its eventual, uncertain PBN control data are converted into hardware controls via probabilistic integration, time derivation, and respective inversion within the CS. Note that the probabilistically integrated position control data are immediately derived back into velocities and then converted into torques using the CS damping. This step is necessary since the underlying low-level antagonistic muscular control, derived in Section 3.2.3, is based on force commands rather than position commands. To the right of the CS, those commands are eventually distributed among the available actuation hardware. Feedback from these control distributions might be used in future work to prevent potential runaway integration by considering the mechanical hardware construction limits and sensory feedback.

5.1.1 Responsiveness to Damping Approximation

As introduced in Equation (3.14) within Section 3.4, distributing arbitrary CS damping matrices to a limited set of available hardware actuators requires an approximation strategy. Such a strategy would be best integrated into the probabilistic integration, as given in Definition 4.15. Thus, the weighted average Equation (4.63) becomes a quadratic minimization. Let a_t be the desired valid approximation. Using Equations (4.66) and (4.67), it follows that:

$$\begin{aligned} a_t &= \arg\min_{a_t} \frac{1}{2} E \left(x_{t-1} - a_t\right)^2 + \frac{1}{2} \left(I - E\right) \left(x_{t-1} + T\dot{\mu} - a_t\right)^2 \\ &= \arg\min_{a_t} \frac{1}{2} a_t^T E a_t - a_t^T E x_{t-1} + \frac{1}{2} a_t^T \left(I - E\right) a_t - a_t^T \left(I - E\right) \left(x_{t-1} + T\dot{\mu}\right) \\ &= \arg\min_{a_t} \frac{1}{2} a_t^2 - a_t^T \left(x_{t-1} + \left(I - E\right) T\dot{\mu}\right) \\ &= \arg\min_{a_t} \frac{1}{2} a_t^2 - a_t^T x_t \end{aligned} \tag{5.3}$$

The initial weighted squared error optimization inside the probabilistic integration can be reformulated into an equivalent but unweighted squared error minimization w.r.t. the regular probabilistic integration result x_t. Thus, the PBN control extraction and the consecutive hardware control distribution can be decoupled.

Furthermore, as initially stated and pictured in Figure 5.1, the probabilistic control integral must be immediately derived again. This step is required since the underlying antagonistic muscular control principle, derived in Section 3.2.3, requires velocity-based control commands rather than position-related controls. Given a probabilistic encoded responsiveness series representation ${}^{P}\dot{X}_{CS}(t) = {}^{P}\left\langle \mu\left(\dot{P}\right), \omega(\Omega)\right\rangle(t)$ in precise notation, the derivation of the probabilistic control integral is derived from the probabilistic integration calculation in Equation (4.67).

$$\dot{x}_t = \frac{d}{dT} \int \left({}^{P}\dot{X}_{CS}(t)\right) = \frac{d}{dT}\, x_t = \frac{d}{dT} \left(x_{t-1} + T\Delta_x\right) = \Delta_x \tag{5.4}$$

Δ_x is calculated by the probabilistic integration Algorithm 4.3. Furthermore, by reducing the probabilistic integration to return only Δ_x, the algorithm becomes independent

of the integration cycle time T and past control samples x_{t-1}. Thus, integrating the low-level antagonistic muscular control with high-level PBNs perfectly satisfies the architectural requirement of frequency invariance, as listed in Section 4.1 and Figure 4.1.

A derived probabilistic integration determines the encoded, desired CS responsiveness matrix $\Delta_x \equiv \mu(G_{CS})$. However, the separated calculation of Δ_x, as defined in Equation (4.67), contains one significant problem regarding responsiveness extraction. Due to its derivative character, it naturally defaults toward zero for decreasing precision annotations. Unfortunately, this behavior leads to an infinite default damping and, thus, to undesired infinite default low-level actuator control commands. In order to achieve zero default damping, the implicit weighted average calculation of Δ_x is altered to tend toward infinite responsiveness G_∞ rather than zero.

$$\begin{aligned} & \Delta_x \;= (I-E)\,\dot{\mu} \;\; = \; E\,0 \;\;\;\;\;\; + (I-E)\,\dot{\mu} \\ \Rightarrow \quad & \Delta_G \;:= \;\;\;\;\;\;\;\;\;\;\;\; E\,G_\infty \;\;\;\;\;\; + (I-E)\,\dot{\mu} \;\; = \; G_\infty + (I-E)\left(\dot{\mu} - G_\infty\right) \end{aligned} \tag{5.5}$$

Note that all responsiveness data Δ_G, $\dot{\mu}$, and G_∞ are encoded matrix vectorizations as defined by Equation (4.40). Since the encoding function and all utilized algorithms have been defined and proven valid for general symmetric matrices rather than only symmetric positive semi-definite damping or responsiveness matrices, the subtraction of G_∞ prior to the derived probabilistic integration does not harm the validity of the probabilistic integration Algorithm 4.3.

A suitable and finite G_∞ is estimated using the minimum actuator damping control vector $d_{AS_{min}}$ introduced in Equation (3.30) and some safety factor $s \geq 1$.

$$G_\infty = s\left(J_{CA}^T \operatorname{diag}\left(d_{AS_{min}}\right) J_{CA}\right)^{-1} \tag{5.6}$$

The responsiveness vector Δ_G is decoded and inverted into the desired CS damping control matrix G_{CS}^{-1}. Since a responsiveness matrix is just positive semi-definite, it may feature zero eigenvalues, and its inverse is, thus, determined via decomposition.

Let $G_{CS} = \mu^{-1}(\Delta_G) = LDL^T$ be the LDL decomposition of the decoded responsiveness matrix G_{CS}. Its inverse is defined as:

$$G_{CS}^{-1} := \left(L^{-T} D^\dagger L^{-1}\right) + \left(LD^\star L^T\right) \tag{5.7}$$

with

$$D^\dagger := [d_i^\dagger] \;=\; \begin{cases} \frac{1}{d_i} & \text{, if } \; d_i \geq \frac{1}{\tilde{d}_{max}} \\ 0 & \text{, else} \end{cases} \qquad \text{diagonal, positive semi-definite} \tag{5.8}$$

$$D^\star := [d_i^\star] \;=\; \begin{cases} \tilde{d}_{max} & \text{, if } \; d_i < \frac{1}{\tilde{d}_{max}} \\ 0 & \text{, else} \end{cases} \qquad \text{diagonal, positive semi-definite} \tag{5.9}$$

and

$$\tilde{d}_{max} \;:=\; s\,\frac{\dim(AS)}{\dim(CS)}\,\max\left\{\operatorname{diag}\left(J_{CA}^T \operatorname{diag}(d_{AS_{max}})\,J_{CA}\right)\right\} \;>\; 0 \tag{5.10}$$

The vector $d_{AS_{max}}$ is the actuators' maximum stable damping vector, as introduced in Equation (3.30). $s \geq 1$ is a scalar parameter adding a safety margin to the estimation of $\tilde{d}_{max}$, e.g., $s \approx 1.1$. The estimate $\tilde{d}_{max}$ ensures maximum actuator damping in the directions of close-to-zero responsiveness at a suitably low, thus numerically stable, optimization input damping matrix. Note that the two summands of G_{CS}^{-1} both project into each other's null spaces. Thus, the artificially constructed second summand does not interfere with the non-maximum-generating first summand. Therefore, the sum of both is a symmetric, positive-definite, full-rank damping matrix.

As the last step, the full-rank CS damping matrix is approximated by a diagonal AS damping matrix, representing the available hardware actuators. This damping distribution uses the optimization Equation (3.30), introduced in Section 3.4.2. Due to the decoupling of probabilistic integration and approximation given in Equation (5.3), the damping distribution is executed with no, i.e., unit weight $W = I$.

$$\begin{aligned} \underset{d_{AS}}{\arg\min} \; & \frac{1}{2}\, d_{AS}^T\, L^T\, L\, d_{AS} \; - d_{AS}^T\, L^T \mathrm{vec}\left(G_{CS}^{-1}\right) \\ \text{with} \; & L := \left[\left(\mathrm{col}_1\left(J_{CA}^T\right) \otimes \mathrm{col}_1\left(J_{CA}^T\right)\right),\, \left(\mathrm{col}_2\left(J_{CA}^T\right) \otimes \mathrm{col}_2\left(J_{CA}^T\right)\right),\, \cdots\right] \\ \text{s.t.} \; & 0 \leq d_{AS_{min}} \leq d_{AS} \leq d_{AS_{max}} \end{aligned} \tag{5.11}$$

Without weighting, the positive definiteness of the optimization's *Hessian* matrix $L^T L$ is entirely defined by the Jacobian matrix J_{CA} encoding the robot's actuation morphology. This sole dependency again underscores the importance of a robot's mechanical design in supporting the underlying low-level control. Herein, this is the antagonistic muscular control developed in Section 3.2.3. For non-singular J_{CA}, the *Hessian* matrix $L^T L$ is guaranteed to be positive-definite. The effects of an unweighted damping distribution were studied in Section 3.4.3 and are shown in Figures 3.19b and 3.20.

Implementation Hint 5.1: Floating-Point-Normalized Optimization

During the iterative numerical optimization, the *Hessian* matrix is decomposed and inverted. Furthermore, the usual stop criteria for such optimization processes are defined via an absolute reference error $\epsilon > 0$.

In order to ensure numerical stability, the result vector elements can be scaled individually using a diagonal matrix S and a minimization vector substitution.

$$\underset{s_{AS}}{\arg\min} \; \frac{1}{2}\, s_{AS}^T\, S\, L^T\, L\, S\, s_{AS} \; - s_{AS}^T\, S\, L^T \mathrm{vec}\left(G_{CS}^{-1}\right) \tag{5.12}$$

$$\text{s.t.} \quad S^{-1}\, d_{AS_{min}} \leq s_{AS} \leq S^{-1}\, d_{AS_{max}}$$

$$\text{and:} \quad d_{AS} \; = \; S\, s_{AS} \tag{5.13}$$

An efficient implementation of S is a direct manipulation of the exponents of the affected floating-point representations in the *Hessian*, gradient, and result vector. Note that an addition to the floating-point exponent equals a multiplication with the summand's power of two. A good estimate for S that keeps the *Hessian* matrix close to the identity is to use half of its original diagonal element exponents.

According to Definition 2.1, distributed skeletal damping is a non-visible but measurable characteristic. Thus, hardware impedance rendering is not influenced by external disturbances and can be assumed to be perfectly generated. Since the internal limitations of minimum and maximum stable damping are considered within the optimization (5.11), the achieved responsiveness matrix can be calculated and used as an optional measure for hardware capabilities and limitations. However, the invertibility of the following formulation again depends on $d_{AS_{min}} > 0$ and a non-singular Jacobian J_{CA}.

$$G_{CS} = \left(J_{CA}^T \operatorname{diag}(d_{AS}) J_{CA} \right)^{-1} \tag{5.14}$$

Algorithm 5.1 summarizes the consecutive steps of PBN responsiveness approximation:

Algorithm 5.1: PBN Responsiveness Control Distribution

Data: The desired probabilistic CS responsiveness ${}^{P}\dot{X}_{CS} = {}^{P}\left\langle \mu(\dot{P}) \, , \, \omega(\Omega) \right\rangle$ containing an encoded Cartesian precise mean matrix according to Notation 4.10.
The momentary mechanical morphology configuration as Jacobian matrix J_{CA}.
The actuator boundary conditions $d_{AS_{min}}$ and $d_{AS_{max}}$.

`/* 1st: Derived Probabilistic Integration */`

1 $\mu(G_{CS}) = G_\infty + \frac{d}{dT} \int_{t-1}^{t} \left({}^{P}\dot{X}_{CS} - G_\infty \right) = \Delta_G$ `/* Equation (5.5) */`

`/* 2nd: Responsiveness Inversion */`

2 $D_{CS} = G_{CS}^{-1} = \left(L^{-T} D^{\dagger} L^{-1} \right) + \left(L D^{\star} L^T \right)$ `/* Equation (5.7) */`

`/* 3rd: Damping Distribution */`

3 $\arg\min_{s_{AS}} \; \frac{1}{2} \, s_{AS}^T \, S \, L^T \, L \, S \, s_{AS} \; - s_{AS}^T \, S \, L^T \operatorname{vec}(D_{CS})$ `/* Following Hint 5.1 */`

4 $d_{AS} = S \, s_{AS}$

5 **return** d_{AS}

6 ---

`/* Optional: Actually Achieved Responsiveness */`

7 $G_{CS} = \left(J_{CA}^T \operatorname{diag}(d_{AS}) J_{CA} \right)^{-1}$ `/* Equation (5.14) */`

Again, it is important to highlight that according to the intended definitions, the responsiveness control should default to zero. Thus, its respective damping should default to infinity, i.e., maximum possible damping. At first glance, defaulting to maximum damping seems a suitable and stable system state. However, Section 5.1.2 will use this default damping to determine the CS control torques by multiplication with the desired control velocities. This multiplication generates the numerically unstable situation of multiplying zero with infinity. Luckily, such an unstable system initialization is practically stabilized by the real-world hardware limits, leaving infinity quite small. E.g., in the case of Carl, infinite damping is about 10^5 kg/s.

Thus, interesting future research may investigate completely removing the independent responsiveness control and determining the actuator damping from the velocity control's precision annotation. This way, the overall control is simplified and better mimics the natural muscle's characteristic of coupling high accelerations, i.e., forces, with high precision, i.e., damping. See Equation (3.4) within Section 3.2.3. However, a proper semantic precision reduction, i.e., a neural inhibition mechanism, must be integrated into the PBN architecture to support intentional muscle relaxation with such a design.

5.1.2 Velocity to Force Optimization

Converting a probabilistic PBN velocity control into a respective actuator force distribution requires nearly identical conversion steps as the responsiveness approximation in Section 5.1.1. However, in contrast to the actuators' damping distribution, force distribution optimizes the generation of a control command rather than approximating it. This difference was already introduced in Equation (3.13) within Section 3.4.

Unfortunately, actuator forces are mechanically, electrically, and thermally limited by the involved hardware. However, as described in Section 3.4.1 and pictured in Figure 3.11, these limits are applied distributed within the individual actuator-local impedance and current control loops. Thus, the velocity control can be assumed to be perfectly achieved by the over-actuated hardware if the distribution cares about the individual actuator capabilities.

Analog to the responsiveness distribution, a probabilistic PBN velocity is converted into a concrete velocity control command with a probabilistic integration followed by an immediate differentiation. Given a momentary probabilistic velocity representation in CS coordinates ${}^{P}\dot{X}_{CS} = {}^{P}\langle \dot{p},\ \Omega \rangle$, the low-level velocity control v_{CS} is evaluated identical to Equation (5.4).

$$v_{CS} = \dot{x}_t = \frac{d}{dT} \int \left({}^{P}\dot{X}_{CS}(t) \right) = \frac{d}{dT}\, x_t = \frac{d}{dT} \left(x_{t-1} + T\Delta_x \right) = \Delta_x \tag{5.15}$$

This velocity is converted into a CS torque vector τ_{CS} by multiplication with the actually applied actuator damping controls d_{AS} determined by the responsiveness approximation in Hint 5.1.

$$\tau_{CS} = J_{CA}^{T}\, \mathrm{diag}(d_{AS})\, J_{CA} v_{CS} \tag{5.16}$$

The final step is to find a suitable AS force vector F_{AS} that correctly generates the desired CS torque control vector τ_{CS}, thus the desired velocity v_{CS}. As first introduced in Section 3.4.1, such a force distribution can be found by an arbitrary quadratic minimization subject to the desired control goal, formulated as a linear constraint. Since no actuator individual force limits are enrolled within this minimization, its optimization criteria must favor a relatively equal force distribution.

Following such criteria, the actuators can be assumed to bottom out roughly simultaneously. Thus, the herein-used optimization strategy slightly adapts the so-called normalized thermal Actuator Force Distribution (AFD) strategy given in Equation (3.16) within Section 3.4.1. Actuator forces are distributed according to their normalized expected force generation. The expected generated force of an actuator is its control force, corrected by the expected actuator-intrinsically rendered damping force.

$$\begin{aligned} \underset{F_{AS}}{\arg\min} \quad & \frac{1}{2} \left(\frac{F_{AS} - \alpha F_d - \beta F_c}{F_{AS_{max}}} \right)^2 \\ s.t. \quad & J_{CA}^{T}\, F_{AS} - \tau_d = \tau_{CS} \end{aligned} \tag{5.17}$$

with

$$F_d := \mathrm{diag}(d_{AS})\ J_{CA}\ v_{CS} \qquad \text{the expected desired damping force} \tag{5.18}$$

$$F_c := \mathrm{diag}(d_{AS})\ J_{CA}\ v_c \qquad \text{the expected current damping force} \tag{5.19}$$

τ_d is an additional CS torque vector, and α, β are parameters satisfying $\alpha,\ \beta \geq 0$ and $\alpha + \beta \leq 1$. The torque vector τ_d may be used for model-based dynamics and gravity compensation as, e.g., studied in [*Hackenberg 23*, *Giebler 24*]. $F_{AS_{max}} \geq 0$ is the non-directional, positive, maximum force vector generable by the individual actuators.

Due to the limited scope of this thesis, no sophisticated dynamics compensation model has been implemented. Instead, a simple integrator compensates for the inevitable steady-state errors induced by friction hystereses, passive mechanical compliance, and similar disturbances. Unfortunately, this setup is not yet capable of highly dynamic motions beyond the movements presented herein.

As analyzed in Section 2.1.1, Requirement 2.1 highlights the hybrid dynamics of legged systems and demands model-based control, such as Model Predictive Control (MPC). Thus, future work on this topic must investigate the deployment of state-of-the-art modeling and a model-based control approach. To match the introduced approach of control tolerance, research in the direction of probabilistic MPC can be recommended, e.g., [Farina 13, Mesbah 16, Landgraf 23].

Each actuator is intrinsically limited in its force generation capability. However, this limitation is applied after the impedance calculation. It is, therefore, possible to correctly generate a force higher than the actuator's electrical maximum if the actuator-intrinsic damping compensates for the overexerted force demand.

The parameter α determines the amount of compensation of expected, desired damping using the momentary velocity control command v_{CS}. The parameter β specifies the compensation w.r.t. the momentary, measured CS velocity v_c. To prevent overcompensation, the sum of both parameters must not exceed one, i.e., $\alpha + \beta \leq 1$. If both parameters are zero, the damping compensation is switched off, leaving only the original normalized thermal AFD optimization. It is important to note that the parameters α and β influence the force distribution but not the cumulated CS torque.

In order to solve the optimization in Equation (5.17), the substitution variable s is defined as the entire inner optimization term.

$$s := \frac{F_{AS} - \alpha F_d - \beta F_c}{F_{AS_{max}}} \tag{5.20}$$

$$\text{and thus:} \qquad F_{AS} = F_{AS_{max}}\, s + \alpha F_d + \beta F_c \tag{5.21}$$

Substituting s, the minimization (5.17) simplifies to

$$\begin{aligned} \arg\min_{s} \quad & \frac{1}{2}\, s^2 \\ s.t. \quad & J_{CA}^T\, F_{AS_{max}}\, s = \tau_{CS} + \tau_d - \alpha J_{CA}^T F_d - \beta J_{CA}^T F_c \end{aligned} \tag{5.22}$$

By inserting the definitions in Equations (5.16), (5.18), and (5.19), the minimization's constraint becomes:

$$J_{CA}^T\, F_{AS_{max}}\, s = J_{CA}^T\, \mathrm{diag}(d_{AS})\, J_{CA}\, \left(v_{CS} - \left(\alpha v_{CS} + \beta v_c\right)\right) + \tau_d \tag{5.23}$$

The simplified minimization (5.20) is well-known to be explicitly solved by the *Moore-Penrose* pseudo inverse of the left-hand-side matrix of the vector s [Boyd 07].

$$\begin{aligned} D_{CA} &:= \operatorname{diag}(d_{AS})\ J_{CA} \\ v_d &:= \alpha v_{CS} + \beta v_c \\ JF^+ &:= F_{AS_{max}}\ J_{CA} \left(J_{CA}^T\ F_{AS_{max}}^2\ J_{CA} \right)^{-1} \\ s &= JF^+ \left(J_{CA}^T\ D_{CA}\ (v_{CS} - v_d) + \tau_d \right) \end{aligned} \tag{5.24}$$

Reverse substitution with Equation (5.21) leads to the following final actuator force distribution calculation:

$$F_{AS} = F_{AS_{max}}\ JF^+ \left(J_{CA}^T\ D_{CA}\ (v_{CS} - v_d) + \tau_d \right) + D_{CA}\ v_d \tag{5.25}$$

Note that dependent on the parameterization α and β, the actuator forces F_{AS} may become independent of the sensed velocity v_c but always depends on the velocity control command v_{CS}, as expected.

As given in Equation (5.25), the actuator forces are calculated as the sum of the additional torque τ_d and the torques generating the desired velocity control command v_{CS}. The first summand of Equation (5.25) is a force distribution optimization. The second summand is a straightforward term. While τ_d is always part of the optimization, the velocity control is distributed between the optimization and the straightforward term dependent on the parameterization. In the three extreme parameterization cases, Equation (5.25) simplifies to:

1. $\alpha = 1:\quad F_{AS} = F_{AS_{max}}\ JF^+\ \tau_d + D_{CA}\ v_{CS}$ (5.26)
2. $\beta = 1:\quad F_{AS} = F_{AS_{max}}\ JF^+ \left(J_{CA}^T\ D_{CA}\ (v_{CS} - v_c) + \tau_d \right) + D_{CA}\ v_c$ (5.27)
3. $\alpha, \beta = 0:\quad F_{AS} = F_{AS_{max}}\ JF^+ \left(J_{CA}^T\ D_{CA}\ v_{CS} + \tau_d \right)$ (5.28)

In the first case, the velocity control v_{CS} is calculated entirely straightforward as the ideal damping compensation. The second case compensates the current damping forces $D_{CA}v_c$ while optimizing the difference between current and control velocity $(v_{CS} - v_c)$. Thus, the second case with $\beta = 1$ optimizes the control acceleration. In the third case, the damping compensation is completely switched off. All torques are distributed without considering damping compensation forces at all.

As stated before, the actuator forces F_{AS} are limited actuator-intrinsic. Hence, the actually achieved control velocity may be estimated as an optional measure of momentary hardware limitations, analog to the achieved responsiveness. The achieved control velocity is estimated in reverse to the force distribution formulation.

$$\begin{aligned} F_{AS}^\star &:= F_{AS}\Big|_{D_{CA}v_c - F_{AS_{max}}}^{D_{CA}v_c + F_{AS_{max}}} \qquad \text{the estimated limited actuator forces} \\ v_{CS} &= G_{CS} \left(J_{CA}^T F_{AS}^\star - \tau_d \right) \end{aligned} \tag{5.29}$$

Note that the above estimation does not require any actuator feedback but relies only on the momentary measured CS velocities v_c. Thus, the actuators stay entirely decoupled from the control and may still be operated via pure feed-forward commands. Such a procedure has several advantages:

1. A regular closed-loop control system of the CS velocity is achieved.
2. Noise from the passive elasticities in the Series Elastic Actuator (SEA) hardware is not injected back into the higher-level control.
3. The communication between the CPU and the actuators can be minimized, thus reducing communication overhead and potential delay.

Algorithm 5.2 summarizes the above-described consecutive steps of the PBN velocity distribution using the actuator force optimization.

Algorithm 5.2: PBN Velocity Control Distribution

Data: The desired probabilistic CS velocity ${}^{P}\dot{X}_{CS} = {}^{P}\langle \dot{p},\ \Omega \rangle$ in precise notation.
The additional torque vector τ_d and the damping control vector d_{AS}.
The current measured CS velocity v_c and the mechanical morphology J_{CA}.
The non-directional, positive, maximum actuator force vector $F_{AS_{max}}$.
Optional: The achieved CS responsiveness matrix G_t.

/* 1st: Derived Probabilistic Integration */

1 $v_{CS} = \frac{d}{dT} \int_{t-1}^{t} \left({}^{P}\dot{X}_{CS}\right) = \Delta_x$ /* According to Algorithm 4.3 */

/* 2nd: Force Distribution */

2 $F_{AS} = F_{AS_{max}} J F^{+} \left(J_{CA}^{T} D_{CA} \left(v_{CS} - v_d \right) + \tau_d \right) + D_{CA}\, v_d$ /* Equation (5.25) */

3 **return** F_{AS}

4 ---

/* Optional: Actually Achieved Control Velocity */

5 $v_{CS} = G_t \left(J_{CA}^{T} F_{AS}^{\star} - \tau_d \right)$ /* Equation (5.29) */

5.1.3 Compliant Pose Control

Compliant pose control using the low-level antagonistic muscular control developed in Section 3.2.3 was initially demonstrated in Section 3.5. Thus, the identical control principle can be integrated into the PBN architecture. The entire compliant pose control is implemented in one PBN behavior $\mathcal{B}_{Pose}$. Only two slight adaptations to the original control equations are necessary to satisfy the PBN architecture's requirements.

1. Responsiveness:
   The control, as presented in Section 3.5, uses a velocity control vector in combination with a damping matrix. In order to fit into the PBN control hierarchy, this damping control has to be replaced by its respective responsiveness matrix, as visualized in Figure 5.1. Thus, Equations (3.40) and (3.38) are reformulated:

$$G_{WS} = \frac{1}{2} \left(\sqrt{K_Z\, M} \right)^{-1} \tag{5.30}$$

and therefore also:

$$\begin{aligned} v_{WS} &= K_{WS}\, \Delta_{WS} \\ &= G_{WS}\, K_Z\, \Delta_{WS} \end{aligned} \tag{5.31}$$

K_Z and M are the desired impedance stiffness and virtual mass, respectively.

2. Precision:
   Every valid PBN control data must be annotated with an appropriate precision matrix. In the case of the velocity controls, those precisions can semantically be determined from the desired control task's tolerance according to Definition 4.15. Respective control tolerance is visualized in Figures 4.4 and 4.6 within Sections 4.4.6 and 4.5.2.
   Regarding responsiveness precision, Equation (5.3) within Section 5.1.1 shows that the control precision can be correlated with the weight of the damping distribution approximation given in Equation (3.30) within Section 3.4.2. Thus, suitable responsiveness precisions may be chosen from the three weighting examples studied in Section 3.4.3. However, applying equal precisions on both velocity and responsiveness controls may also be suitable.

The resulting complete robot control system yet features the compliant pose behavior $\mathcal{B}_{Pose}$ as its only available high-level control skill. Figure 5.2 illustrates this entire control system. Thereby, the illustration used is identical to the generalized robot control system initially introduced in Figure 2.14 within Section 2.4.

On the left of Figure 5.2, the mostly empty PBN control graph is sketched. At this point, the control features only the compliant pose control behavior $\mathcal{B}_{Pose}$. As described above, the compliant pose control is realized equal to the initial non-PBN compliant trajectory control of Section 3.5. Thus, the behavior's function visualization shows Figures 3.23a and 3.23b for reference.

The robot hardware, i.e., its morphology and actuation system, is represented by the light blue boxes all to the right of Figure 5.2. Pictured are the planar bipedal robot CARL and its compliant bio-inspired actuation system featuring ten RRLabSEAs, introduced in Section 3.1.2. All actuators locally run the low-level antagonistic muscular control, as developed in Section 3.2.3. CARL's human-like musculoskeletal morphology is described in detail in Section 3.1.3.

Required interfacing components connecting the high-level PBN control with the robot hardware and low-level actuation control can be found in Figure 5.2 between both. Due to the lack of extrinsic sensors, the environment model features only the robot-intrinsic state space, i.e., the angular positions and velocities of the robot's floating-base and its entire CS. Exponential precision annotation modules, as introduced in Section 4.5.1, implement these sensory interfaces.

Control coordination is realized according to Algorithms 5.1 and 5.2 within Sections 5.1.1 and 5.1.2, respectively. As previously mentioned in Section 5.1.2, due to the limited scope of this work, the robot dynamics compensation torque τ_d is delivered by a simple steady-state error-compensating integrator instead of a dynamic model controller, which would actually be necessary.

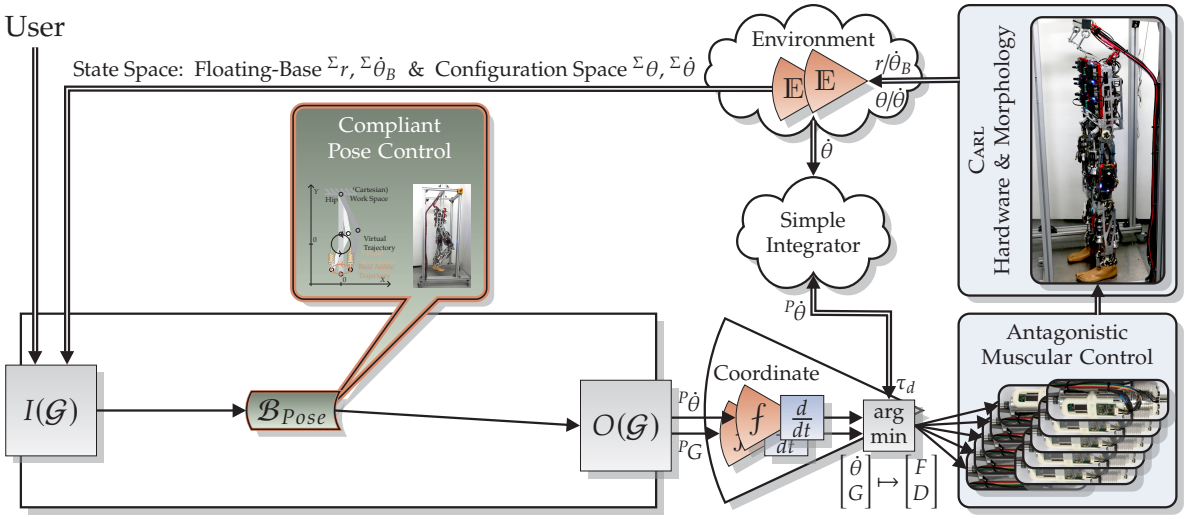


Figure 5.2: The control system of the bipedal robot Carl. From left to right, the diagram shows the high-level PBN control, the required hardware interconnection modules, and the robot hardware, including the RRLabSEAs building the compliant actuation system. Currently, the PBN control is mostly empty, only featuring the compliant pose control behavior. The data flow starts on the upper right, probabilistically annotating the available robot-intrinsic sensor information. This state information then enters the PBN control on the left. To the right, the respective control data leave the high-level control, are converted into low-level antagonistic muscular control commands, and are finally distributed among the available actuators.

5.1.4 Technical and Theoretical Limits of the Deployed System

At this point, it is essential to highlight some critical perspectives on the above-presented system. Thus, the following thoughts aim to motivate future research, development, and continuous improvement of the natural control philosophy herein developed.

With only the compliant pose control behavior applied on top of the low-level antagonistic muscular control, this basic control architecture satisfies three of the desired natural control requirements derived throughout Section 2.1. Requirement 2.3: *"Variable Compliance"* from Section 2.1.3 is satisfied on all levels, i.e., by the hardware mechanics, the hybrid low-level impedance control, and the high-level PBN. Furthermore, by design, the layered architecture of the hybrid impedance control satisfies the basics of Requirement 2.5: *"Hierarchical Control"* from Section 2.1.5. Finally, Requirement 2.2: *"Singular Positions"* is satisfied by two different methodologies. First, the possibility of implementing controllers directly in the singular-free CS, as in the case of the above-described compliant pose control behavior. Second, by back-projecting PBN velocity control data from an arbitrary, possibly singular WS into the singular-free CS via acceleration-based optimization, as sketched via Equation (4.114) in Section 4.7.2 and described and analyzed in detail within [*Vonwirth 24b*].

In contrast, Requirement 2.1: *"Hybrid Dynamics"* from Section 2.1.1 cannot be achieved by the simplified setup pictured in Figure 5.2. Unfortunately, this fact is caused by two main reasons. First, the simple integrator, used instead of a necessary model-based dynamics compensation, must be tuned to one concrete system and, thus, cannot handle hybrid dynamics. Second, the PBN architecture's hierarchy, as presented in Section 4.7,

so far ignores possible passive Degrees of Freedom (DoFs). While this is not a problem on the sensor side, it is a problem for the controls.

Analog to that, the probabilistic integration defined in Section 4.5.2 is also not meant to handle passive DoFs. Thus, the PBN architecture must be adapted to embed and intrinsically support model-based control, such as MPC, to enable suitable hybrid dynamic control. However, determining a correct and applicable model representation goes far beyond the scope of this work. Additionally, the proprioceptive, commercial prosthetic feet currently built into Carl cannot reliably distinguish between the required hybrid model states: ground contact and non-contact [Schütz 20b]. A promising direction for future studies is to abandon classical robot representation via Cartesian CS minimal coordinates and instead turn toward constrained, projective maximal coordinates represented in PGA or CGA if further linearization is necessary.

In Section 3.2.2, the unsuitability of fully distributed impedance control is reasoned due to the enormous amount of required control commands. The same argument can be raised for the above-declared high-level control using probabilistic velocity and probabilistic damping commands independent from each other. Theoretically, the high-level control behavior must generate an overwhelming amount of 69 individual control values to control only six mechanical DoFs of Carl. Following Definition 2.1, the controls can be split into visible and measurable motion controls. The visible ones, i.e., the actual motion controls, are the six-dimensional velocity vector and its attached 21-DoF symmetric, positive semi-definite 6×6 precision matrix. All others, i.e., two further 21-DoF symmetric matrices for responsiveness and responsiveness precision, are used to describe the motion precision, as aimed for in Theorem 1.1.

Obviously, this approach will not scale to full-featured, three-dimensional humanoids, just as distributed low-level impedance control does not. Interestingly, the eventual purpose of the responsiveness control, i.e., motion precision, is already encoded within the precision matrix annotating the velocity, i.e., the actual motion control vector. Thus, this finding, combined with the above-argued need to embed a robot model into the overall PBN control architecture, seems promising for eliminating the need for an independent responsiveness control.

However, to do so, the entire control interface must be redefined, i.e., the probabilistic integration and the subsequent low-level impedance control distribution. As discussed above, the control interface presented herein is unsuitable for hybrid, partially passive systems and thus warrants further investigation and improvement. Furthermore, an inhibition mechanism must be introduced to support intended relaxation. Note that thoughts about future adaptations to these issues have already been given in Sections 4.5.2 and 5.1.1, respectively.

5.2 Upright Balancing

In the context of bipedal locomotion, the very first control task that has to be mastered is an upright, balanced stance. A physical description of that task is the well-known Inverted Pendulum (IP) problem. Since this problem features unstable intrinsic dynamics, active control is required to stabilize an IP against gravity. Using robotic vocabulary,

balancing a legged robot modeled as IP means eventually keeping the robot's CoM calm upon its support polygon.

Among others, two simple approach principles are well-known in robotics to tackle the IP, i.e., the upright balancing problem.

1. A more static approach:
   Keeping a robot's kinematic Center of Pressure (CoP) inside its support polygon.
   E.g., a bio-inspired ankle strategy *or single IP*

2. A more dynamic approach:
   Keeping a robot's dynamic Zero Moment Point (ZMP) inside its support polygon.
   E.g., a bio-inspired hip strategy *or double IP, a technical whole-body CoM impedance*

For reference, see Definitions 2.2, 2.3, and 2.4 within Sections 2.1.1 and 2.1.6.

The bio-inspired balancing approaches, the *ankle* and *hip strategy*, result from extensive human reaction and Electromyography (EMG) studies, e.g., [Winter 98, Loram 02, Peterka 02, Sakanaka 16, Groote 17]. Both strategies have first been introduced in Figure 2.10 within Section 2.1.6. Their control relies on distributed controllers, each with only a limited set of affected actuators in the body. The two approaches are combined in a hierarchical order so that the more complex dynamic *hip strategy* takes over control from the static and, thus, less capable *ankle strategy* when necessary [Creath 05].

A theoretical analysis of the general stability regions of both those strategies and a *step strategy* on top of those is given by Benjamin Stephens in [Stephens 07]. Experimental results on virtual and real-world robots have proven these bio-inspired balancing strategies functional in robotic applications [Hyon 09, Mergner 10, Zhao 14]. Figure 5.3 illustrates the IP abstraction model w.r.t. real-world *ankle strategy* balancing of the planar, bipedal robot Carl.

On the other hand, purely technically motivated balancing strategies are usually built upon one centralized, coordinated, whole-body torque controller. On top of this fundamental low-level control, different balancing strategies are applied, often based on a virtual impedance around the CoM or the ZMP. Various possible strategies can be found in [Henze 14, Henze 16, Caron 18, Hosokawa 18, White 20].

By deploying one bio-inspired and one technical balance strategy on the same hardware, Ott et al. compared both types of balancing against each other [Ott 16]. Exposed to CoM displacements, the two balancing strategies react with completely different compensation motions. However, in their conclusion, Ott et al. state that, although highly different, none of those approaches can be claimed to be better or worse than the other because both strategies fulfill their control goal. Each approach has its individual strengths and drawbacks. Due to its centralization, the technical whole-body balance control is more versatile and predictable than the distributed bio-inspired strategy. Vise-versa, the distributed topology of the bio-inspired balancing makes it more robust against signal delay.

Since this work explicitly aims toward natural control, the bio-inspired *ankle strategy* is derived and implemented within the PBN framework to demonstrate real-world upright balance control. At the same time, the robot's body abstraction to a one-dimensional IP demonstrates the PBN model reduction capability, derived in Section 2.1.6 and demanded in Requirement 2.6.

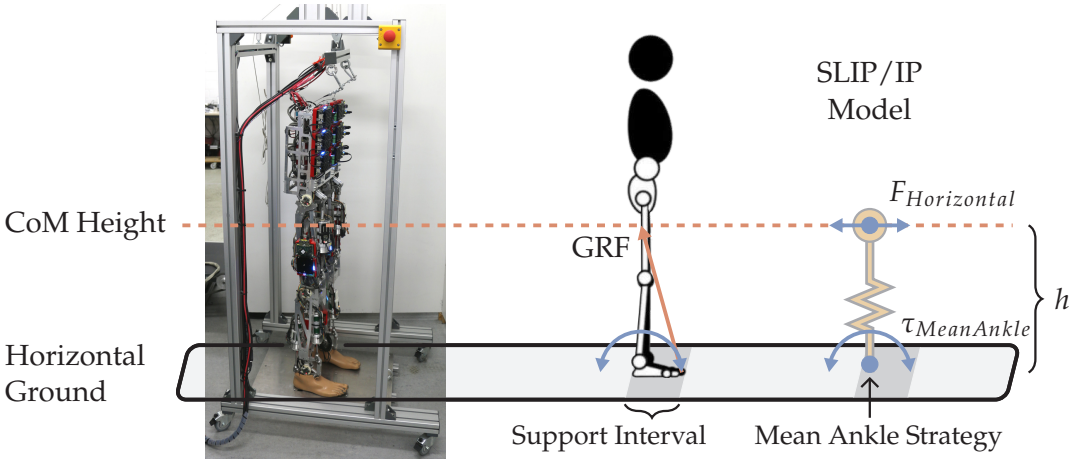


Figure 5.3: Model of the bipedal robot CARL as an abstract one-dimensional IP model. Following Requirement 2.6: *"Model Reduction"*, the IP abstraction implements the Spring Loaded Inverted Pendulum (SLIP) model initially introduced in Figure 2.10 within Section 2.1.6. Due to its heavy legs and the lack of an upper body, CARL's CoM, unfortunately, is located below its hip joints. It is assumed that both feet are always in complete contact with the ground. Thus, the planar positions of the heels and toes define the robot's support polygon, as emphasized in gray. In the picture on the left, CARL stands free, applying the *ankle strategy* for upright self-balancing.

5.2.1 Support Work Space

Following the analysis of Benjamin Stephens [Stephens 07], balancing via the natural *ankle strategy* can generally be reduced to a projection into the horizontal plane, i.e., the support polygon as defined in Definition 2.2. In the planar case, as relevant for the bipedal robot CARL, the horizontal plane is reduced to a horizontal line, and the balancing strategy becomes a one-dimensional problem. Since no dedicated robot dynamics and gravity compensation model has been integrated into the PBN's velocity-force interface described in Section 5.1.2, the theoretical stability analysis in [Stephens 07] can be used as a blueprint for the actual *ankle strategy* implementation. Thus, to tackle this problem, a dedicated one-dimensional support polygon WS is defined.

The actual *ankle strategy* is realized as a PBN behavior $\mathcal{B}$ inside this WS. A PBN projection behavior $\mathcal{P}$ and a corresponding transformation behavior $\mathcal{T}$ sandwich the *ankle strategy* according to Definitions 4.18 and 4.19. The correspondent PBN control subgraph topology is pictured in Figure 5.4. Note that the topology has already been introduced surrounding behavior $\mathcal{B}_2$ in Figure 4.9 within Section 4.7.

The one-dimensional support polygon WS is defined by its projection task function t_{CoM}, following Definition 4.18. The following preliminaries are used:

1. Define the robot's floating-base position as the origin and its orientation being measured against gravity.

2. Let $P_{CoM} : \mathbb{R}^6, \mathbb{C} \mapsto \mathbb{R}$ be the kinematic function that calculates the robot's horizontal CoM projection w.r.t. its floating-base origin. The inputs of P_{CoM} are all six range-limited joint angles of CARL $\mathbb{R}^6$ in radians, and the floating-base orientation $\mathbb{C}$ as a two-dimensional rotor $r \in \mathrm{SO}(2) \subset \mathbb{C}$.

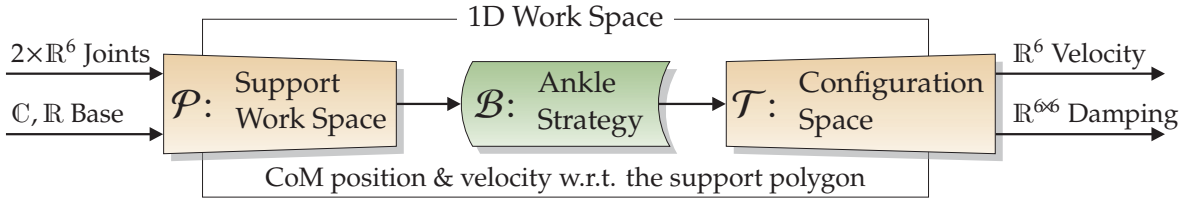


Figure 5.4: Topology of the PBN subgraph implementing the natural *ankle strategy* for static upright balance control. On the left, the robot's configuration state, i.e., position and velocity of all six joints, and the floating-base orientation and rotational velocity enter the WS projection behavior $\mathcal{P}$. Within the one-dimensional support polygon WS, the actual *ankle strategy* is realized by its contained behavior $\mathcal{B}$. Finally, on the right, the transformation behavior $\mathcal{T}$ converts the one-dimensional WS controls back into the robot's CS controls: A probabilistic joints velocity vector and a respective symmetric, positive-definite damping matrix.

3. Let $P_{LH}, P_{LA}, P_{LT}, P_{RH}, P_{RA}, P_{RT} : \mathbb{R}^3, \mathbb{C} \mapsto \mathbb{R}$ be analog kinematic functions computing the horizontal projections of the left and right legs' heels, ankles, and toes. Each function takes its leg angles $\mathbb{R}^3$ and the floating-base orientation $\mathbb{C}$ as input.

The projection's task function $t_{CoM} : \mathbb{R}^6, \mathbb{C} \mapsto \mathbb{R}$ is defined as the CoM position w.r.t. the ankles' mean $P_{\tilde{A}} := \frac{1}{2}(P_{LA} + P_{RA})$.

$$t_{CoM} := P_{CoM} - P_{\tilde{A}} \tag{5.32}$$

Respectively, the robot's state velocity projection $\dot{t}_{CoM} : \mathbb{R}^6, \mathbb{R} \mapsto \mathbb{R}$ is the task function's time derivative:

$$\dot{t}_{CoM} := \tfrac{d}{dt} t_{CoM} = \begin{bmatrix} J_{CW} & J_{Base} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{CS} \\ \dot{\theta}_{Base} \end{bmatrix} \tag{5.33}$$

$\dot{\theta}_{CS} \in \mathbb{R}^6$ is the vector of joint velocities, and $\dot{\theta}_{Base} = \frac{d}{dt} r \in \mathbb{R}$ is the scalar time derivative of the orientation rotor r, i.e., the robot's floating-base rotational velocity.

Additionally, the horizontal projection of the support polygon reduces to a support interval from heels to toes w.r.t. the ankles' mean $P_{\tilde{A}}$.

$$I_{Support} = [I_{min},\ I_{max}] \tag{5.34}$$

where

$$\begin{aligned} I_{min} &:= \min\{P_{LH}, P_{RH}\} - P_{\tilde{A}} && \text{the back-most heel} \\ I_{max} &:= \max\{P_{LT}, P_{RT}\} - P_{\tilde{A}} && \text{the front-most toes} \end{aligned}$$

Following the intention of the PBN data abstraction described in Section 4.7, the low-dimensional WS behavior should be sandwiched between a sensor data projection $\mathcal{P}$ and its inverse $\mathcal{P}^{-1}$, which back-project the control values. By using Definition 4.19, the transposed Jacobian matrix $\begin{bmatrix} J_{CW} & J_{Base} \end{bmatrix}^T$ of Equation 5.33 should realize this reverse projection. See Figures 4.10 and 4.12 for reference. However, using the transpose of that equation would return not only CS joint controls but also a control for the robot's passive floating-base orientation.

Furthermore, the actual robot motion is not generated by the control of the projected CoM but by the mirrored Ground Reaction Force (GRF) that the ground surface applies to the robot in response to its CoM control. Thus, the transposed control Jacobian would have to be additionally concatenated by the inverse of the CoM-support transformation Jacobian. However, to do so, the CoM and the support interval should be represented in complete and thus invertibly transformable planar pose coordinates, i.e., a two-dimensional position and an orientation.

In this particular case, an adequate simplification of the just-described calculation process is to model the CS joints' effects on the robot's CoM by hand. The effects are given by the vertical levers between the controlled joints and the CoM. Thus, the control transformation behavior $\mathcal{T}$ is realized by the following transposed inverse transformation behavior task function:

$$J_{WC}^{-T} := \begin{bmatrix} 0 & h_{knee} & h & 0 & h_{knee} & h \end{bmatrix}^T \tag{5.35}$$

The levers of the two ankle joints are given by the CoM height h, as pictured in Figure 5.3. Additionally, the knees are also actuated with their respectively reduced lever arm $h > h_{knee} > 0$. In the case of the biped CARL, its CoM is located below its hip joints. Thus, the two hips are intentionally not actuated by the *ankle strategy* balancing and feature a zero lever arm within the transformation Jacobian J_{WC}^{-T}.

5.2.2 Ankle Strategy

Gravity stabilization is achieved by balancing potential and kinetic energy. The reduced system's potential energy is given by the spatial integral of the horizontal force required to counteract the mean ankle torque applied by gravity. Let x be the CoM's position w.r.t. to the ankles' mean, and $\dot{x}$ its respective velocity.

$$F_{Horizontal}\ h = \tau_{MeanAnkle} = M\ g\ x \tag{5.36}$$

$$\Rightarrow \quad E_{pot} = \int F\ dx = \frac{1}{2}\ M\ \frac{g}{h}\ x^2 \quad \Rightarrow K = M\ \frac{g}{h} \tag{5.37}$$

$$\Rightarrow \quad E_{kin} = \int p\ d\dot{x} = \frac{1}{2}\ M\ \dot{x}^2 \tag{5.38}$$

M is the system's overall point mass, and g is the gravitational constant. From the energy equality, a desired position-velocity relation is derived

$$E_{pot} \stackrel{!}{=} E_{kin} \quad \Rightarrow \quad \dot{x} \stackrel{!}{=} \sqrt{\frac{g}{h}}\ x = \omega\ x \tag{5.39}$$

where $\omega = \sqrt{g/h} > 0$ is the pendulum's angular frequency in the upright position.

The PBN control parameters, the responsiveness G and velocity v, are derived from the horizontal force Equation (5.36) and the desired energy equality Equation (5.39).

$$F_{Horizontal} = M\ \frac{g}{h} x = M\ \omega\ \dot{x}$$

$$\Rightarrow \quad G = \frac{1}{D} = \frac{1}{M\ \omega} > 0 \tag{5.40}$$

By using the damping value $D = M\ \omega > 0$, the control velocity v is

$$\begin{aligned} F_{Control} &= D\ v \stackrel{!}{=} -\frac{d}{dx}\left(E_{pot} + E_{kin}\right) \\ \Rightarrow \qquad v &= -\frac{1}{M\ \omega}\left(K\ x + M\ \omega\ \dot{x}\right) = -\omega\left(x + \tfrac{1}{\omega}\dot{x}\right) = -\omega\ \xi \end{aligned} \tag{5.41}$$

where $\xi = \left(x + \frac{1}{\omega}\dot{x}\right)$ is the so-called Extrapolated Center of Mass (XCoM) [Hof 08].

The static *ankle strategy* can physically balance a robot as long as its XCoM stays within its support interval $\xi \in I_{Support}$. As visualized in Figure 5.3, the support interval for a human-sized robot with respective feet dimensions is quite narrow. Thus, with only the *ankle strategy* in use, Carl's balancing capability within this work is limited to compensate for only light impacts, less dynamic motions, and sensor noise.

The above control rule mandatorily balances the robot above its ankles. In order to balance within the whole stability region, i.e., the support interval, an additional stiffness $K \geq 0$ with equilibrium $e \in I_{Support}$ is added. Since multiple parallel linear stiffnesses combine into one single linear replacement stiffness, the above control can be rewritten into

$$v = -(\omega + K)\left(\xi - \frac{K}{\omega + K}\ e\right) = -K_\epsilon\left(\xi - \epsilon\right) \tag{5.42}$$

where $K_\epsilon = (\omega + K)$ and $\epsilon = \frac{K}{\omega + K}\ e$ are the respective replacement stiffness and equilibrium position.

As demanded by the PBN data representation Definition 4.4, the velocity control variance is defined by the stability region, i.e., the support interval.

$$v_{min} := -K_\epsilon\left(\xi - \frac{K}{\omega + K}\ I_{min}\right) \tag{5.43}$$

$$v_{max} := -K_\epsilon\left(\xi - \frac{K}{\omega + K}\ I_{max}\right) \tag{5.44}$$

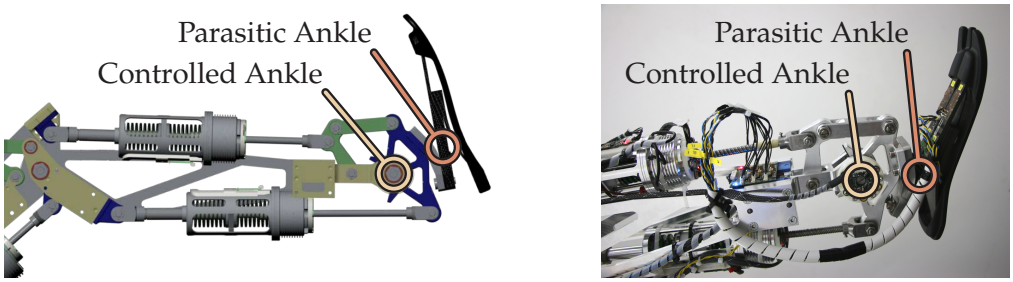
A suitable variance allows a velocity standard deviation within the support interval.

$$\Sigma_v := \min\left\{(v - v_{min})^2\ ,\ (v - v_{max})^2\right\} \tag{5.45}$$

Thus, the velocity control's precision grows to infinity when the XCoM approaches the support interval boundaries. The responsiveness control precision can be set similarly and scaled by the velocity $\dot{x}$. This way, the system automatically relaxes if it stays calm.

Unfortunately, Carl's mechanical construction causes one significant control problem. Introduced in Section 3.1.3 and visible in Figure 3.7, Carl features compliant commercial prosthetic feet. These medical feet are designed to replace an entire human ankle-foot complex and to *"deliver[] a high energy return"*[2]. In the case of the robot Carl, this means that its prosthetic feet add passive, compliant, unmeasured, and low-damped parasitic secondary ankle joints. Figure 5.5 illustrates the double ankle joints in the shank-foot complex of the bipedal robot Carl.

[2] By Ottobock Product URL: `https://www.ottobock.com/en-us/product/1C63` (2025)



(a) CAD Design **(b)** Assembled Hardware

Figure 5.5: Double ankle joints in the leg of the bipedal robot CARL. Both figures, the (a) CAD design and (b) assembled hardware, highlight the two distinct ankles. The active, i.e., the measured and controlled ankle joint, is an intended part of CARL's designed rigid leg skeleton. It is highlighted in orange. On the other hand, the undesired parasitic secondary ankle joint is marked in red. This joint is embedded in the compliant construction of the prosthetic foot and, thus, neither measured nor controlled.

It is generally possible to measure the intrinsic deformation of CARL's feet using strain gauges. Research by Schütz et al. shows the extraction of the GRF and the CoP within the foot sole gathered from those measured deformations [Schütz 20a, Schütz 20b]. Unfortunately, very high signal noise, continuous signal drift over time (probably due to thermal effects), and low reliability under light loads cause these measurements jet to be unusable for control purposes. However, future work on this topic, including mechanical, electrical, and post-processing improvements, seems promising to eventually achieve a sensory system entirely embedded in the robot's skeleton, i.e., without additional space and mass requirements.

As demonstrated by the antagonistic muscular control principle developed for the RRLab SEA control in Section 3.2.3, it is generally possible to stabilize uncontrolled mechanical stiffness using adjacent damping. Unfortunately, the dynamic analysis in Section 3.2.6 highlights one mechanical requirement for this method. This requirement demands that the actively damped mass attached to the passive stiffness be smaller than the external load on the other side. See Figure 3.15 for reference.

In the case of CARL's parasitic ankle joints within the prosthetic feet and under the assumption of a floating-base, the actively damped part is just the mechanical frame between both ankles. The external load is the whole rest of the robot. Thus, the above requirement is far from being fulfilled. Experimental trials with various ankle damping all eventually lead to slowly increasing oscillations, validating the above theory.

In order to counteract the rotational motion of the parasitic ankle joints, a second-derivative control term is added to Equation (5.42). Thus, the final velocity control calculation, applied as *ankle strategy* on the bipedal robot CARL, is

$$v = -K_\epsilon (\xi - \epsilon) - K_a \ddot{x} \tag{5.46}$$

where $K_a > 0$ is the additional control gain, counteracting WS accelerations. Unfortunately, using a second-derivative component comes at the cost of very high sensor noise being introduced into the control calculations.

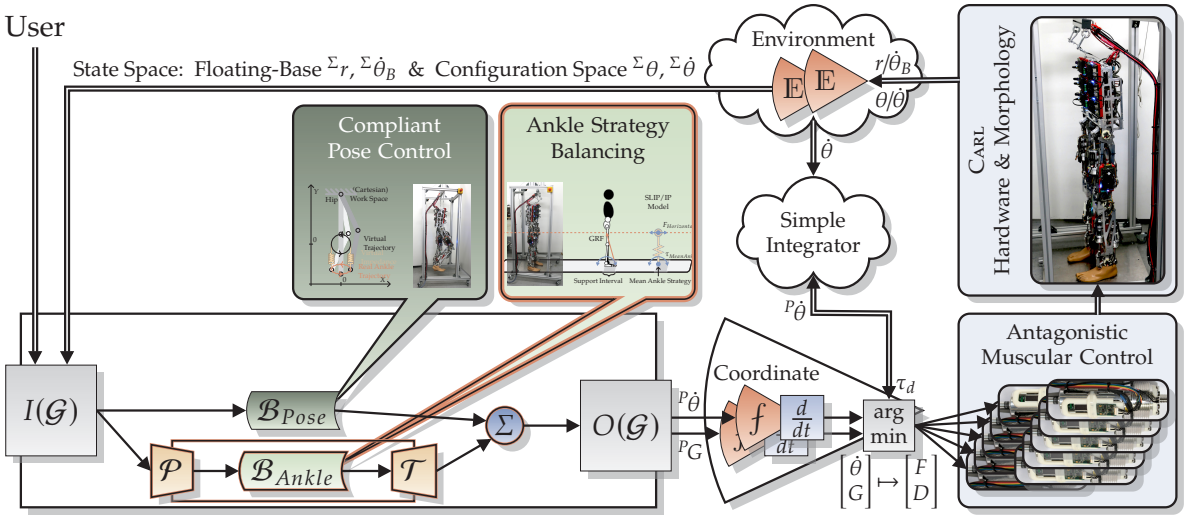


Figure 5.6: The control system of the bipedal robot Carl. From left to right, the diagram visualizes the high-level PBN control graph, the hardware connection modules, and the robot hardware, including the RRLabSEAs that build the compliant actuation system. Compared to Figure 5.2, the PBN control is extended by the *ankle strategy* subgraph, introduced in Figure 5.4. The balancing data flow first projects the robot's CS state space into the one-dimensional support WS. Within this, the behavior $\mathcal{B}_{Ankle}$ implements the balance control before its control data are transformed back into high-dimensional CS control vectors. A composition Σ merges the two independent control data of the compliant pose control and the ankle strategy.

5.2.3 Experimental Results of Ankle Strategy Balancing on Carl

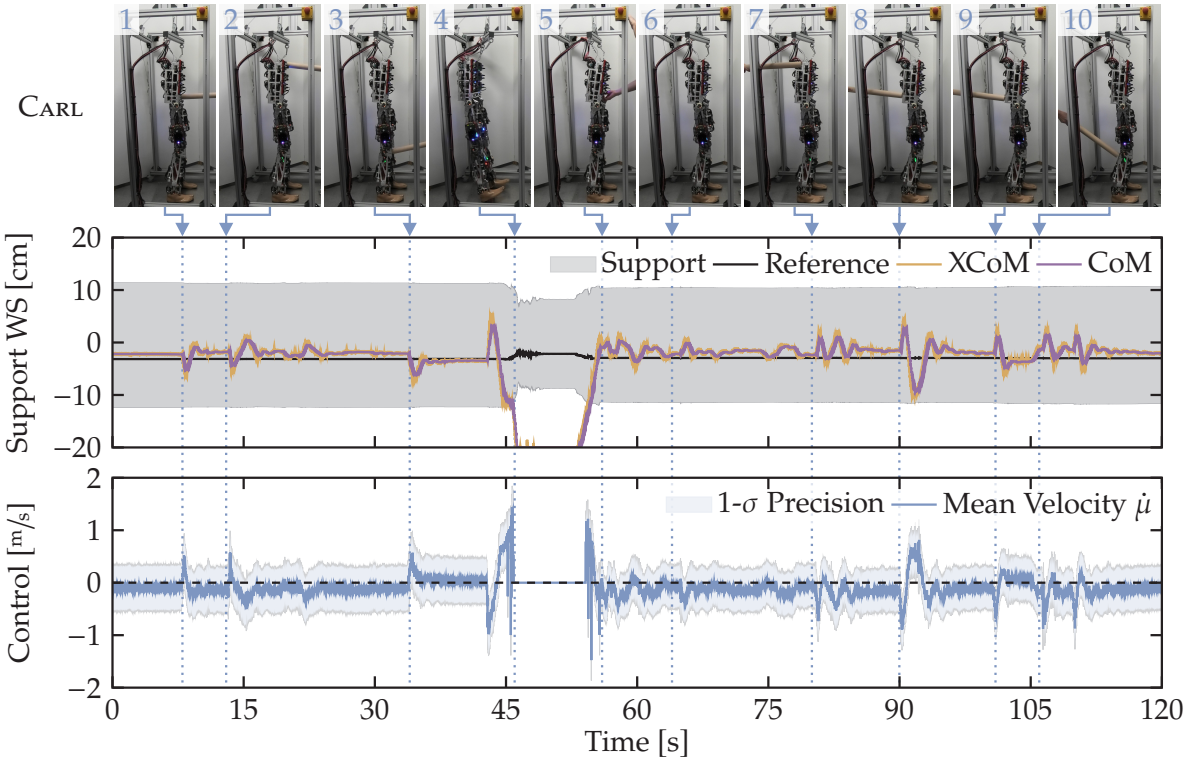
To evaluate the *ankle strategy* balancing approach under real-world conditions, it is deployed on the bipedal robot Carl. Hence, the respective control subgraph, shown in Figure 5.4, is integrated into the overall control architecture introduced in Figure 5.2.

Within the high-level PBN control graph, the *ankle strategy* subgraph is added in parallel to the existing compliant pose control behavior. Both controls are merged by a PBN composition behavior Σ before exiting the high-level control graph into the underlying low-level hardware coordination module. The entire control system, including both pose and balance behaviors, is sketched in Figure 5.6.

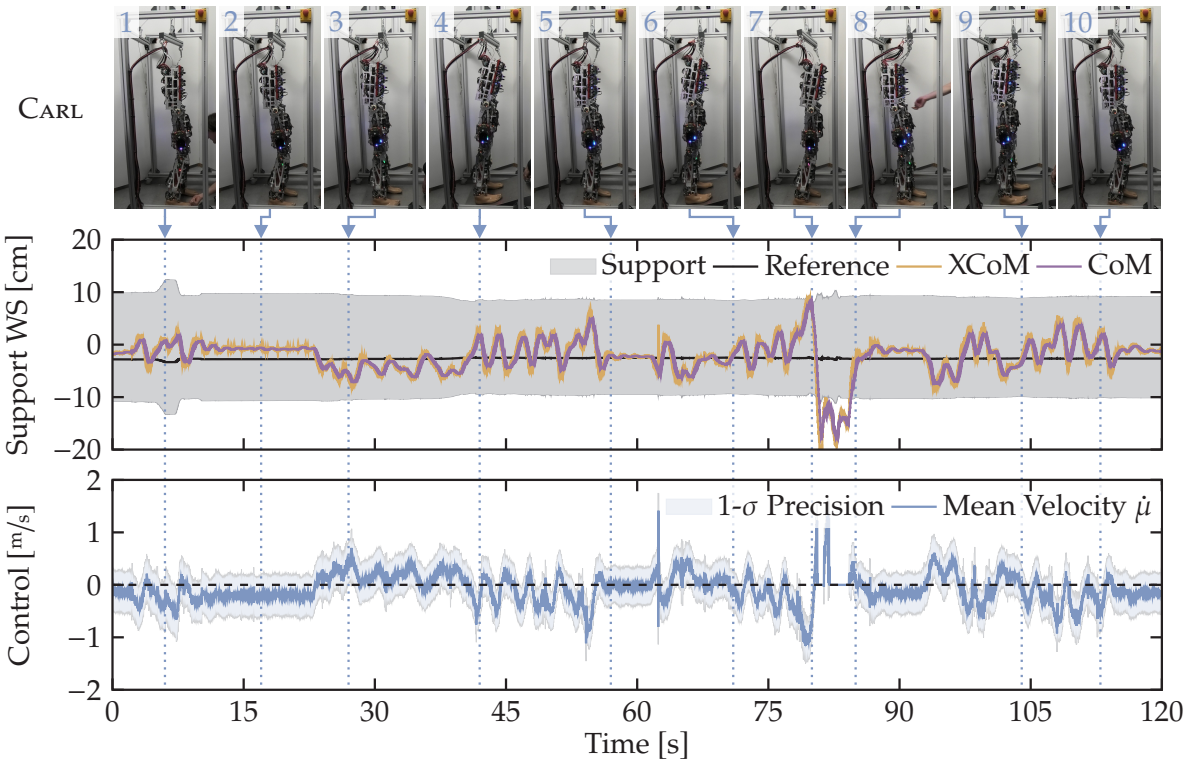
Two distinct experiments with different disturbances have been conducted to test the control's stability. However, the experimental setup is the same for both. At first, the compliant pose control behavior initializes Carl into a stretched position. Then, the robot is manually turned over into an upright standing position in which the *ankle strategy* behavior automatically activates and takes over the upright balancing control.

The balancing behavior is implemented to self-activate if the robot's XCoM is detected inside the support interval boundaries, i.e., inside the *ankle strategy*'s stability region. Vise versa, it also automatically deactivates if it detects that static balancing cannot handle the situation. Once upright, the disturbances are applied by hand to ensure randomization and non-predictability.

In the first experiment, the robot is manually pushed and pulled at various spots among the entire frame in both directions. The spots are the upper trunk, the hip, and the knee



(a) *Ankle strategy* balancing by CARL exposed to body disturbances



(b) *Ankle strategy* balancing by CARL exposed to ground disturbances

Figure 5.7: Static, upright balancing of CARL, applying the *ankle strategy*, sketched in Figure 5.6. The robot is disturbed by (a) pushes on its frame and (b) ground rotation. All data are recorded at 1 kHz frequency. However, all plots are down-sampled without visual smoothing using overlapping, alternating minimum and maximum windows of 50 ms to become representable.

joints. A representative excerpt of this disturbed balancing setup is visualized, and its control data are plotted in Figure 5.7a.

The second experiment does not disturb the robot's body but its feet and the ground on which it balances. Carl is set up to stand on a thin metal plate that significantly bends under the robot's weight and the applied pressure when actively balancing. First, the left and right foot are pushed backward one after another while maintaining its upright position. Afterward, the ground plate is hand-lifted on one side. This way, the Carl is disturbed in both ground orientation and elevation. Respective experiment visualizations and control data plots are given in Figure 5.7b.

Both plots in Figure 5.7 visualize two-minute excerpts of the bipedal robot Carl standing and balancing upright. The static balance is achieved by applying the bio-inspired *ankle strategy*, pictured in Figure 5.3, embedded in the natural control architecture, sketched in Figure 5.6. Several snapshots of the real robot hardware are given throughout the represented experiment interval.

For both experiments, two plots visualize the WS information most relevant for balancing, as derived in Section 5.2.2. Sensory information is presented in the WS plots, while the emerging WS control data is given in the velocity graphs. The upper WS plots each show the support interval, the desired reference position, the XCoM, and the CoM. In the graphs below, the precise control velocity is visualized by its mean value embedded in the controlled 1-σ standard deviation environment.

The chosen reference is a compromise between the most stable middle position within the support interval, i.e., the zero position in the plots, and the more backward-located, most efficient balancing position straight above the ankle joints. Due to very high noise in the signal, the presented control mean is smoothed by a 50 ms Gaussian filter. The unfiltered control velocity of Figure 5.7a is plotted in Figure 5.8b.

The first three snapshots in Figure 5.7a picture backward pushes applied to the robot's hip, trunk, and right knee. In the support WS, these disturbances are detected by a negative displacement of both CoM and XCoM. As expected, these negative displacements are counteracted by positive control velocities.

Afterward, the robot is strongly pulled forward, causing a significant over-swing backward at about 45 s in the plot. The toes start lifting off the ground when the XCoM reaches the *ankle strategy*'s stability boundaries, given by the support interval. This exact moment is captured by the given snapshot of Carl and the start of the support interval shrinking. For safety reasons, the *ankle strategy* behavior then automatically deactivates itself, thus handing the ankle and knee control back to the parallel pose control behavior, as sketched in the control architecture given in Figure 5.6. As expected, Carl then eventually falls over to the back.

The robot is then manually pulled upright, and the balancing behavior self-engages, stabilizing Carl in the desired upright standing position. For the rest of the presented time, Carl is pushed from the back at its trunk, hip, and right knee, as visualized by the last four snapshots. In all the cases, the XCoM stays within the support interval, and the robot self-stabilizes as desired.

In the second experiment, Carl is first pushed on its feet. The first experiment snapshot in Figure 5.7b pictures the right foot being pushed backward while Carl self-stabilizes.

Consequently, the support interval increases. With the left foot pushed backward equally, the support interval shrinks again.

Throughout the rest of the experiment, the thin metal plate on which Carl stands is lifted and subsequently lowered on its front three times. At about 80 seconds in the plot, the pictured snapshot captures the front-most possible stable position. However, analog to the first experiment, this extreme position causes a significant backward overshoot and an eventual fall to the back. Thus, at about 85 seconds, Carl is again manually repositioned upright, as pictured in the 8th snapshot.

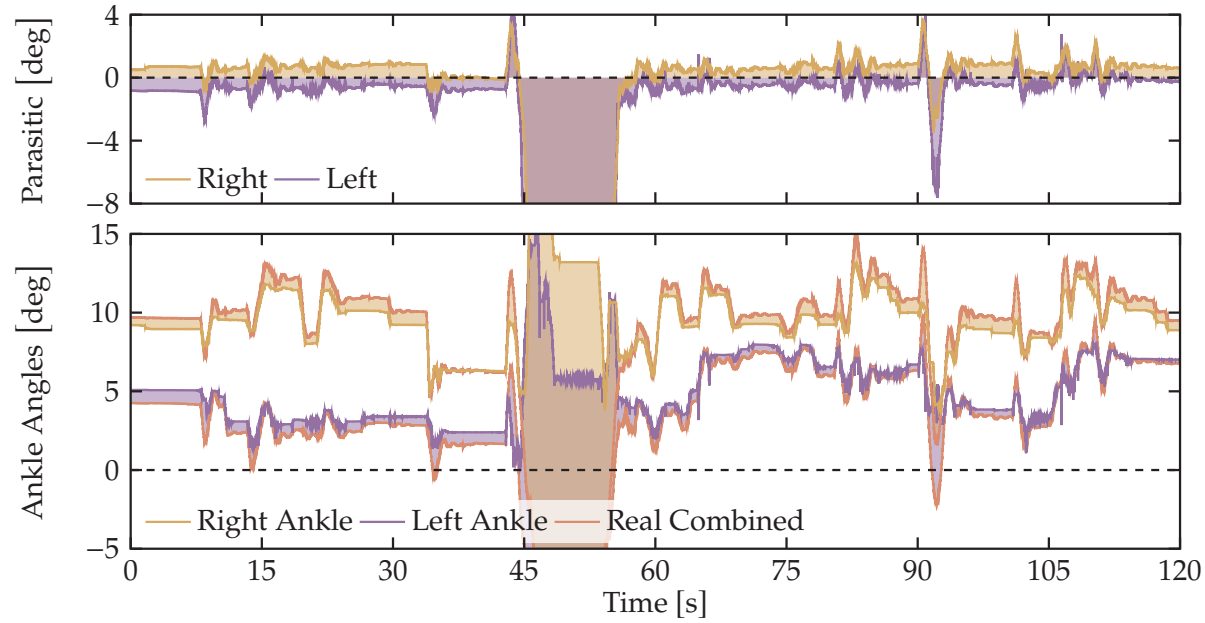
When comparing both experiments, the primary observable difference is the significantly increased oscillations within the second experiment. The WS graph in Figure 5.7b contains four regions in which the oscillations are damped. These are always when the ground plate rests on the floor, as pictured by the second, fifth, eighth, and last snapshots.

Between the stable phases, the ground is actively moved, thus permanently inputting changes to the system. Furthermore, the mechanical compliance of the thin ground plate adds additional soft stiffness in series with the passive parasitic ankle joints, shown in Figure 5.5. As explained in Section 5.2.2, the current system realization, which uses entirely distributed damping via the low-level antagonistic muscular control principle, cannot actively damp such an environment. Thus, it cannot fully stabilize the robot either. However, as is nicely visible around 45 seconds right after snapshot four, the robot achieves at least marginal stability within its support limits.

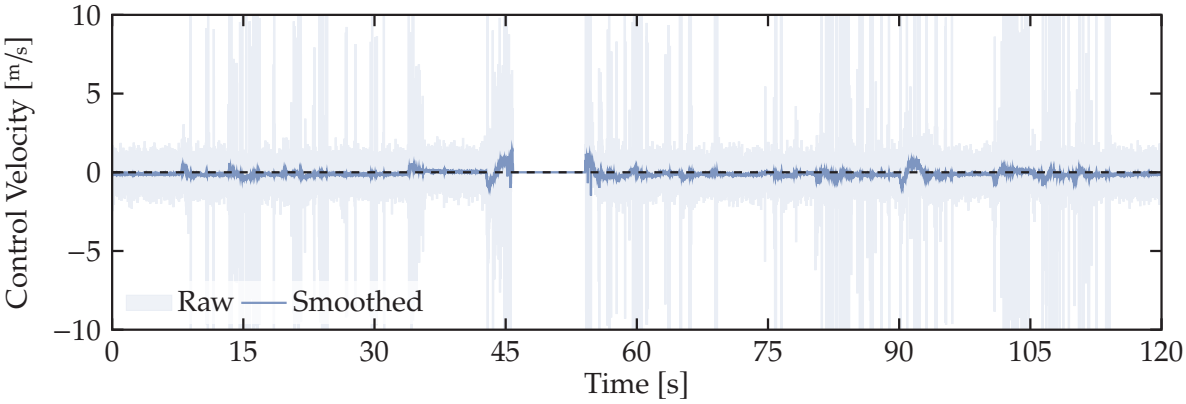
In general, the two balancing experiments demonstrate the real-world functionality of the applied *ankle strategy*, the PBN control architecture, and the low-level antagonistic muscular control. As just analyzed and visualized in both WS graphs in Figure 5.7, the bipedal robot Carl can stabilize itself under various disturbances and falls exactly when it is supposed to. However, the following four significant issues in the combined hardware and control software can be identified:

1. Huge stabilization overshoots in the backward direction.
2. The passive and thus uncontrolled compliance in the robot's prosthetic feet.
3. A significant amount of noise injected into the control calculation caused by the third-derivative term and imperfect hardware.
4. Runaway integration due to the closed-loop kinematics and non-existing dynamic model control.

The first issue is very likely to be caused by two factors. First, the robot's actual CoM is further back than the modeled CoM, caused by massive model simplifications. Second, the lack of control over the passive, parasitic ankles within the prosthetic feet leads to a lack of required low-level damping applied to those joints. The three other issues are expected by design and have been discussed in Sections 5.2.2 and 4.4.5. Figures 5.8 and 5.9 show those control problems at the hand of the first balancing experiment where Carl is exposed to body disturbances on steady ground, shown in Figure 5.7a.



(a) Double ankle joints of Carl



(b) Raw WS control velocity

Figure 5.8: Visualization of the control problems within Carl at the hand of the disturbed upright balancing experiment on static ground, as presented in Figure 5.7a. The two plots in (a) highlight Carl's mechanical double ankle joint problem, illustrated in Figure 5.5. Plot (b) demonstrates the actual noise within the *ankle strategy*'s WS velocity control signal. The unmeasured angles of Carl's parasitic ankle joints are calculated under the assumption of a steady and perfectly horizontal ground with the feet in full contact. All data are recorded at 1 kHz frequency. However, all plots are down-sampled without visual smoothing using overlapping, alternating minimum and maximum windows of 50 ms to become representable.

Throughout this first balancing experiment, the ground is assumed to be stable and horizontal, and the feet are in full contact. Thus, together with the relative joint angles and floating-base, i.e., the trunk's absolute orientation, all absolute and relative angles of all skeletal segments of Carl can be derived, including the angular deflections of the parasitic ankles within the prosthetic feet. The upper graph of Figure 5.8a plots those two parasitic ankle angles in degree throughout the entire represented experiment interval. In the lower graph, the measured ankles of both feet are represented and

combined with their parasitic ones. Thus, the two respective red graph edges represent the two legs' real combined ankles, i.e., the angles between shanks and foot soles.

It can be seen that the parasitic deflection is mostly about ±1° under calm conditions but strongly reacts to the applied disturbances. Note that in the interval between 45 s and about 55 s, CARL has fallen over, and its feet were no longer in contact with the ground. Thus, the data within this period are not legit. The observed deflection of about ±1° can be explained as the natural passive spring deflection of the prosthetic feet under the static load of CARL's weight.

However, the natural ankle deflection of the prosthetic feet under an upright load is negative, i.e., bending toward the heel. In Figure 5.8a, this natural behavior can be observed on the left leg. In contrast, the right foot reveals an unnatural positive deflection, i.e., pressuring the toes. The difference in deflection between both feet can be explained by the applied actuator forces plotted in Figure 5.9.

As expected, more or less no forces are applied on CARL's left leg. Thus, the robot's weight passively rests on the prosthetic left foot, leading to its natural deformation. As plotted in Figure 5.9a, the right leg is more active.

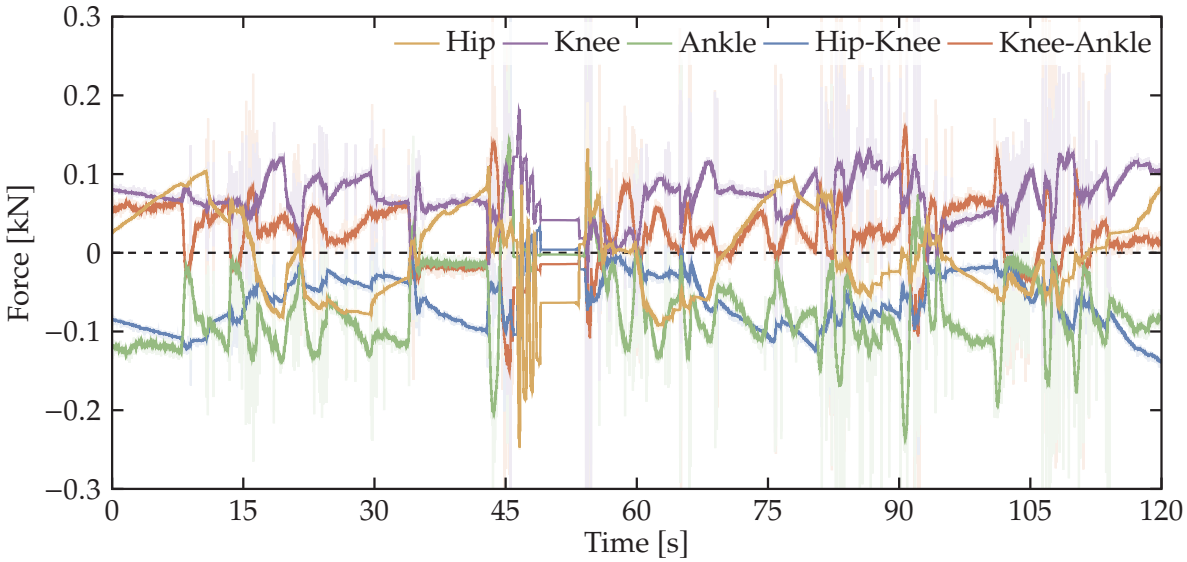
It can be seen that both actuators acting on the robot's controlled right ankle generate a negative torque. Since the feet are in full ground contact and cannot move, this controlled negative ankle torque is counteracted by a respective positive torque generated by the unnaturally deflected right prosthetic ankle. See Figure 3.6 within Section 3.1.3 to reference CARL's musculoskeletal morphology.

Figure 5.8b visualizes the WS velocity control signal generated by the *ankle strategy* behavior. The blue graph in the foreground is the same Gaussian-smoothed control velocity signal as previously plotted in Figure 5.7a. However, the raw signal is given in the background. Three main observations can be drawn from this representation:

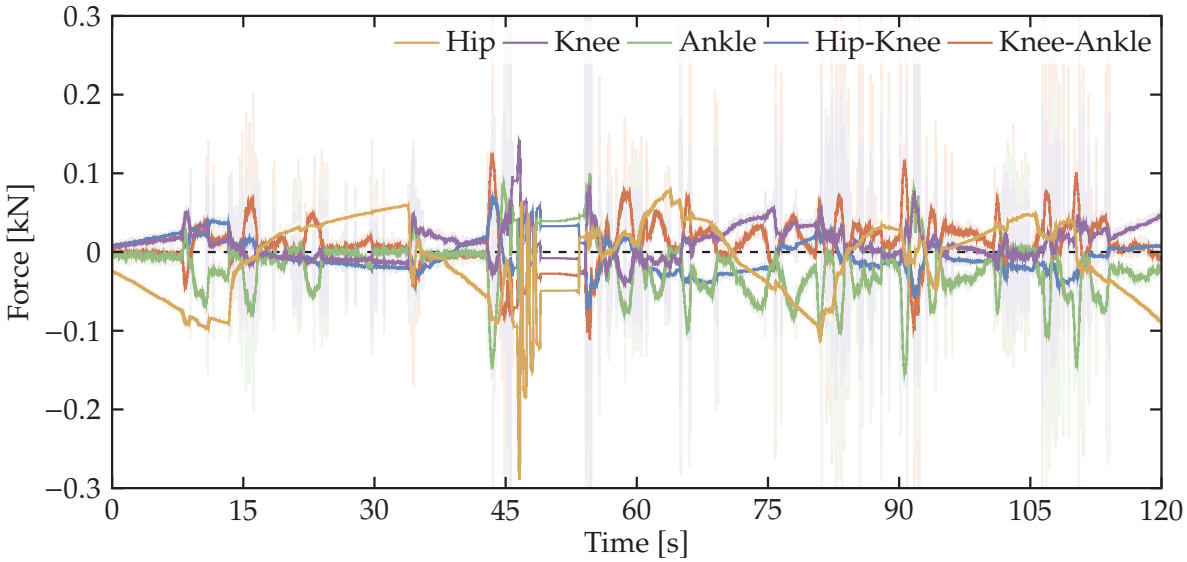
1. The signal's noise is a whole order of magnitude larger than the signal itself.
2. There are two different types of noise present in the signal.
3. The signal can be recovered despite the extraordinary amount of noise.

As mentioned in Section 5.2.2, the enormous ambient noise level in the control signal results from the second-derivative term in the control Equation (5.46). Other causes of the ambient noise are the non-real-time Operating System (OS) used to operate the control software and the non-real-time UDP/IP ethernet communication between the central control computer and the robot hardware, i.e., the RRLab SEAs. Empirical studies on communication delays revealed a mostly stable communication at about 1 kHz roundtrip frequency with regular single signal losses. Fatal communication failures with a signal interruption for more than 10 ms that, for safety reasons, trigger a hardware STO occur only very sporadic [*Sivak 22*].

In contrast, the second type of noise, the thin peaks up and beyond $\pm 10\,{}^{\mathrm{m}}/_{\mathrm{s}}$, are most likely valid reactions to undesired, sudden, but correct hardware movements. Due to hardware tolerances, the mounting points of the left leg's biarticular knee-ankle actuator



(a) Right leg's applied actuator forces



(b) Left leg's applied actuator forces

Figure 5.9: Visualization of all actuator forces of Carl at the hand of the disturbed upright balancing experiment on static ground, presented in Figure 5.7a. The two graphs each plot all five applied RRLab SEA forces of the robot's (a) right and (b) left leg. Plotted are the eventual propulsion forces of the antagonistic muscular control, as sketched in Figure 3.11. Thus, each plotted actuator force is the sum of the commanded feed-forward force and the actuator-intrinsically determined damping force. All data are recorded at 1 kHz frequency but down-sampled to become representable using overlapping, alternating minimum and maximum windows of 50 ms, avoiding visual smoothing. A 50 ms Gaussian filter is applied to smooth the foreground data of all RRLabSEAs. Their respective raw data are given in the background. It can be seen that the hip joint, as described, is not involved in the *ankle strategy* balancing. Thus, both legs' orange monoarticular hip actuators and blue biarticular hip-knee actuators do not suffer from the prior-described peak noise induced by the legs' double ankles. However, the two orange monoarticular hip actuators notably display runaway integration against each other due to their mechanically identical joint.

are not perfectly planar. Consequently, this actuator is under permanent tension, causing a lot of friction within its ball bearings and ball screw gear.

The resulting breakaway torque is transmitted into the left ankle joint. Small ripples can be seen in the purple left ankle plots in Figure 5.8a, which are correlated with the massive raw control velocity peaks in Figure 5.8b. It is important to additionally recognize that those ripples are not present in the right leg's orange ankle plots.

However, as the plotted smoothed signal demonstrates, the intended control is still recoverable. As discussed in Sections 3.2.3, 4.4.5, and 4.5.2, the low-level antagonistic muscular control and the high-level PBN architecture are designed for damping control, control tolerance, and control smoothing. Due to these design decisions, Carl's overall control system inherently acts as a low-pass filter. Thus, the robot intrinsically recovers the desired control, keeping the implemented balancing control approach functional.

The significant noise reduction from WS velocity control to hardware force commands can be seen in Figure 5.9. Analog to Figure 5.8b, both plots in Figure 5.9 visualize Gaussian-smoothed actuator forces in the foreground while the respective raw data are plotted in the background. It can be observed that the ambient noise of the high-level velocity control mostly disappears on the lowest possible actuator control level. However, the peak noise clearly remains, causing noticeable disturbances in the system. Since this peak noise originates in poor hardware, future improvements to the actuator mountings can be expected to immediately and significantly improve the control signal quality and, hence, the system's stability without software changes.

The last control problem observable in Figure 5.9 is so-called runaway integration. This phenomenon is best visible at the hand of the two right and left hip actuators, both plotted in orange. In the represented upright balancing experiment, both of Carl's feet are in permanent contact with the ground. This extrinsic environmental condition leads to a so-called closed-loop kinematic.

Consequently, although technically independent, the two hip joints and, thus, the two monoarticular hip actuators are mechanically coupled. In the first 10 to 15 seconds of the represented experiment, the monoarticular right hip actuator can be observed to linearly increase its force exactly opposite to the left leg's monoarticular hip actuator. Thus, both hip actuators generate mechanical tension in the robot's frame but, at the same time, cancel out each other.

Runaway integration is a well-known problem of concurrent, competitive control in general. Hence, it is one of the most essential challenges inherent to all Behavior-Based (BB) control systems. Within the herein-developed PBN architecture, this integration issue has been discussed and addressed in Sections 4.4.5 and 5.1. However, due to the limited scope of this thesis, the design-intended and necessary dynamics model is simplified to an integrator, introduced in Section 5.1.2 and sketched in Figure 5.2.

Thus, although the herein-proposed concept avoids runaway integration in principle, the momentary simplified implementation that is deployed on Carl still suffers from this issue. However, as demonstrated by the successful experiments pictured in Figure 5.7, this structural problem is acceptable within the limited use cases of this work. To avoid runaway integration in future control developments, integrating a modern, model-based, robot-dynamics control approach is inevitable.

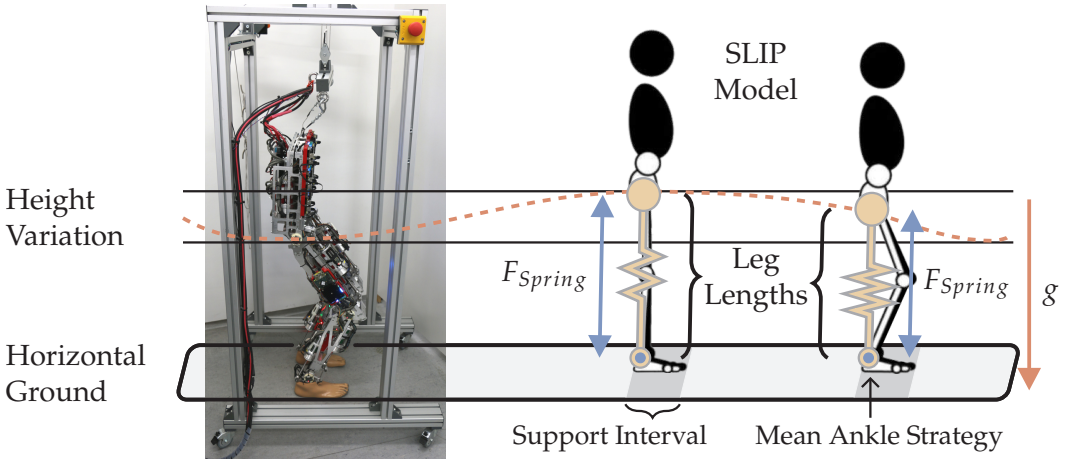


Figure 5.10: Planar SLIP model of the bipedal robot CARL. The SLIP abstraction follows the natural control Requirements 2.4: *"Nonlinear Relations"* and 2.6: *"Model Reduction"*. Both requirements were already illustrated by their respective human gait characteristics in Figures 2.8 and 2.10, respectively, within Sections 2.1.4 and 2.1.6. Due to the lack of model-based gravity compensation, the virtual spring legs compress under the robot's gravitational load. Height variation is actively achieved through the continuous modulation of the equilibrium length of the two spring legs. The bio-inspired *ankle strategy* balancing approach of Section 5.2 is applied in parallel to maintain the robot in an upright position. Thus, both feet are assumed always to have full ground contact.

5.3 Squat Motion via Linear Spring Leg Behavior

Upright balancing is the most fundamental skill for any bipedal robot. However, when realized with the bio-inspired *ankle strategy*, as demonstrated in the previous section, it is a relatively static and mostly horizontal task, as illustrated in Figure 5.3. Thus, a vertical-oriented task is a natural combination with basic balancing to achieve a complete whole-body motion. Herein, this vertical task is realized by a squat motion, deployed and executed on the real-world hardware of the bipedal robot CARL.

Following the Spring Loaded Inverted Pendulum (SLIP) model, introduced in Section 2.1.4, the squat motion is achieved by simulating virtual linear springs with moving equilibrium lengths per leg. Mathematically, this realization is a one-dimensional vertical projection of the compliant, rotational motion executed in Section 3.5 that has been used to initially evaluate the proper real-world functionality of the low-level antagonistic muscular control, defined in Section 3.2.3. Thus, this approach intentionally does not result in a perfect motion due to the still absence of proper dynamics and gravity compensation, as mentioned in Section 5.1.3.

Instead, it highlights the embedded whole-body compliance and the robot's reactions to intrinsic and extrinsic dynamic disturbances. This way, both the capabilities and limitations of the overall control architecture and the robot hardware are visible in the motion and control data. Figure 5.10 illustrates the SLIP abstraction and the respective compliant squat motion.

As for the upright balancing of Section 5.2, the bipedal robot CARL is modeled as an ideal SLIP. However, this time, each leg is modeled individually, and the virtual spring leg is

independent of the robot's CoM between the ankle and hip joints. In an upright position, gravity accelerates the robot's mass downward, compressing both virtual spring legs according to their stiffnesses.

A sinusoidal squat motion of the whole robot is generated by a sine wave motion of the spring legs' equilibria. Due to the lack of explicit gravity compensation and the introduced control tolerances, the actual virtual leg lengths are expected to be a little compressed, equal to the initial antagonistic muscular control experiment in Section 3.5.

As stated in Theorem 1.1, this work's overall aim is natural control. The respective necessary control requirements are derived in Chapter 2 at the hand of a regular human gait. Unfortunately, as already argued, Carl's momentary hardware setup does not allow such dynamic motions without additional planar guidance.

Hence, the above-described sinusoidal compliant squat motion is the best approximation for such a walking gait. The similarity can be seen by comparing the intended sinusoidal height variation of the robot's trunk within Figure 5.10 to the CoM's sinusoidal height variation in the human gait's SLIP model in Figure 2.8 within Section 2.1.4.

5.3.1 Linear Spring Legs

Generally, a spring leg behavior could be achieved using an identical PBN subgraph topology as used by the *ankle strategy* balancing shown in Figure 5.4 within Section 5.2.1. However, as stated in Theorem 4.2 within Section 4.3.1, every PBN control subgraph can be substituted with a single behavior. Thus, due to the simplicity of the spring leg behavior and to demonstrate this structural invariance of the PBN decomposition strategy, a complex abstraction architecture with leg-individual linear WSs is avoided. Instead, the whole individual control processing for both legs is implemented in a single PBN behavior, operating directly on CS coordinates, i.e., on the robot's joint level.

As introduced in Figure 3.17 within Section 3.3.1, arbitrary WS stiffness can be transformed into an equivalent CS representation by the following equation:

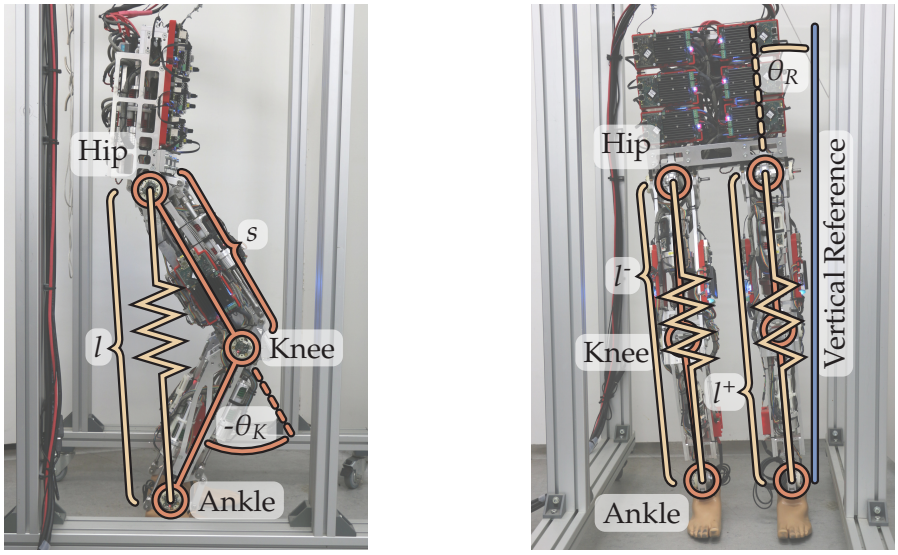
$$K_{CS} = J_{CS}^{T} \; K_{WS} \; f_{WS}(\theta_{CS}) \tag{5.47}$$

In the case of the desired spring leg behavior of the bipedal robot Carl, the general equation (5.47) instantiates to:

$$K_{CS} = s \, \sin\!\left(\frac{1}{2}\theta_K\right) \; K_{WS} \; \left(l_0 - 2s \, \cos\!\left(\frac{1}{2}\theta_K\right)\right) \tag{5.48}$$

where $s = 0.42\,\text{m}$ is the equal length of Carl's thigh and shank, $l_0 \leq 2s$ is the virtual spring's equilibrium length, and $\theta_K \in \left[-\frac{\pi}{2},\, 0\right]$ is the knee joint angle.

Figure 5.11a illustrates the entire spring leg geometry at the hand of the real robot hardware during a squat execution. The two Equations (5.47) and (5.48) clearly show that the spring leg behavior relies on the demanded natural control Requirements 2.3: *"Variable Compliance"* and 2.4: *"Nonlinear Relations"*, derived and defined within Sections 2.1.3 and 2.1.4, respectively.



(a) Spring Leg Geometry **(b)** Non-Planar Side Roll

Figure 5.11: Spring Leg behavior of the bipedal robot Carl. On top of real-world pose pictures in the background, the robot's joints, links, relevant angles, and virtual spring legs are sketched. The (a) general planar spring leg geometry in sketched on the left at the hand of a side-view of Carl. To the right, a frontal view of Carl pictures (b) an inevitable, non-planar sideward roll of the robot that undesirably but emerges from the imperfect compliant control. Problematically, a sideward leading does not exist in Carl's planar world and, thus, control design.

A simple sine-wave signal is used to continuously adapt the virtual spring's equilibrium length l_0, resulting in a continuous squat motion of the Carl over time t.

$$l_0(t) = 2s - \frac{a}{2}(1 - \cos(\omega t)) \qquad \leq 2s \tag{5.49}$$

Equation (5.49) generates a sinusoidal wave signal for the spring's equilibrium length $l_0(t)$ with amplitude $a \geq 0$ and frequency $f = {}^{\omega}/_{2\pi}$. The motion starts at its maximum length with an entirely stretched leg $l_0(0) = 2s$. Due to the system's compliance and behavior-external control components, the robot's actual leg lengths l are expected to be compressed under gravity and thus shorter than the equilibrium, thus $l \leq l_0$.

Next to the above stiffness, the virtual leg's responsiveness and motion precision are similarly evaluated using the scalar Jacobian value J_{CS}, as explained in Figures 5.1 and 4.12 with Sections 5.1 and 4.7.2, respectively. Equal to the compliant pose control behavior introduced in Section 5.1.3, the responsiveness is determined according to Equation (5.30).

The above one-dimensional virtual spring leg controls only the scalar angular knee velocity $\dot{\theta}_K$. However, due to the velocity control, this would not lead to a straight squat but a rotation around the knee joint. In order to achieve a straight squat, both hip and ankle joints must move appropriately opposite to the knee.

Within the morphology of Carl featuring identical thigh and shank lengths, these proper movements are both half the knee's negative velocity, i.e., $\dot{\theta}_H = \dot{\theta}_A = -\frac{1}{2}\dot{\theta}_K$. The scalar motion precision of the spring leg is respectively oriented within the CS, thus

only affecting joint control that generates a straight squat. Note that, in a perfectly upright position with perfectly parallel aligned legs, this orientation is orthogonal to the horizontal control orientation of the prior *ankle strategy* behavior.

Entirely in or close to the singular position where both legs are stretched, most of the robot's mass rests on its rigid skeleton rather than its compliant actuators. With the knees being bent during squats, this situation rapidly changes. Consequently, CARL's mass of about 33 kg compresses the virtual spring legs, the mechanical springs within its ten RRLabSEAs, and tensions the electric motors to withstand their respective external load forces.

Unfortunately, the compression of both legs depends on the load distribution and the individual actuators' mechanical control accuracies. Thus, unequal leg lengths are inevitable. The significantly problematic consequence of unequal leg lengths is sketched at the hand of a real-world pose picture of CARL in Figure 5.11b.

In the background of Figure 5.11b, CARL is pictured with the right leg l^- being slightly more bent than the left one l^+. It can be seen that this difference in leg lengths causes the robot to lead over to its right side, i.e., to the pictured left. The roll angle of the captured pose is 5°. This angle already causes the robot's CoP to be located at the inside edge of the bent leg's foot, so most of the robot's weight is carried by the compressed leg, pictured on the left.

Unfortunately, such a sideward lead is an unstable system, similar to a regular IP. The more the robot's weight compresses one leg more than the other, the more weight is shifted onto this particular leg, causing it to compress even more. Thus, although sideward motion, by definition, does not exist in CARL's planar world, it will inevitably cause the robot to fall over to one side during squats if not compensated.

The best solution to stop Carl from non-planar sideward roll is to mechanically restrict it to its planar world. In order not to interfere with the robot's planar dynamics, such mechanics require sophisticated constructions. However, a software solution alternative is realized herein to keep CARL autonomous and free of external support.

Thus, roll balancing is achieved by a non-planar PID controller redistributing the control load between the two legs. Depending on the roll angle, the controller subtracts a portion of the longer leg's spring equilibrium length l^+ and adds this exact portion to the opposite, shorter leg l^-. This way, a roll moment toward the straight upright position is achieved while, at the same time, the two legs' summed overall virtual spring force that carries the robot's total weight stays unchanged. As this method rolls CARL sideward over its feet-ground contacts, the stability region of this roll controller is equal to the support interval of the *ankle strategy*, as described in Section 5.2.2.

5.3.2 Experimental Results of Compliant Natural Squats on CARL

As for the previous experiments, the just-introduced linear spring leg behavior is evaluated under real-world conditions on the bipedal robot CARL. To achieve that, the single respective CS PBN behavior $\mathcal{B}_{Squat}$ is added to the existing control graph, sketched in Figure 5.6. Exploiting the control composition capability of the PBN architecture, the novel squat behavior is placed in parallel to the two prior behaviors, i.e., pose control and upright balancing.

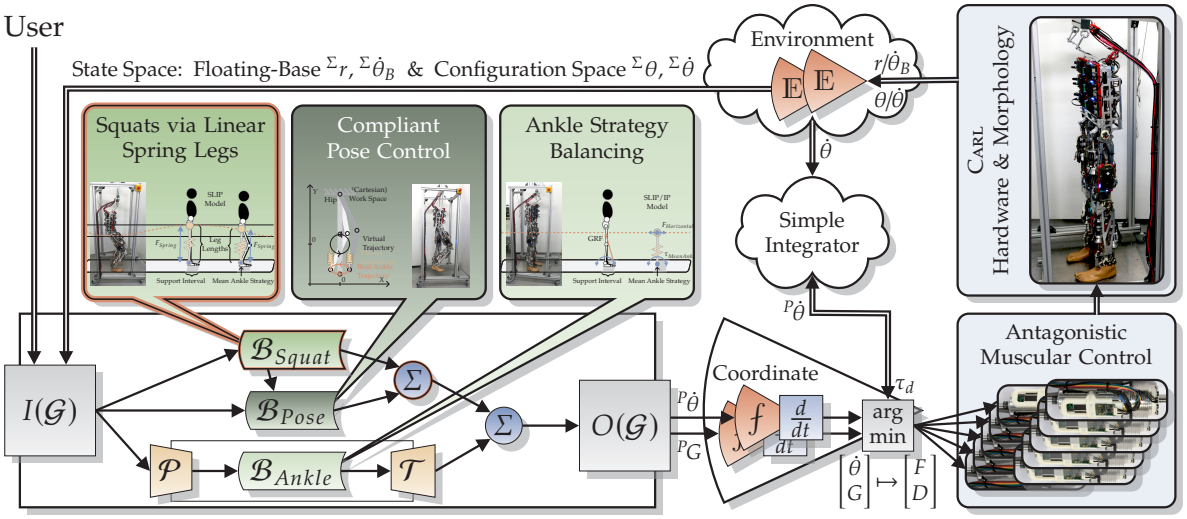


Figure 5.12: The complete, novel-developed control system of the bipedal robot CARL. As in Figures 5.2 and 5.6, the high-level PBN control graph, the hardware connection modules, and the robot hardware, including the RRLabSEAs, are visualized from left to right. The pictured PBN control graph encompasses three concurrent behaviors that implement a linear spring legs emulation, a compliant joint pose control, and static balancing behavior using the natural *ankle strategy*. The topology of the PBN graph is known from Chapter 4, and its insights have been discussed by the simplified one-dimensional control execution example in Figure 4.14 within Section 4.8. An additional hierarchical connection from behavior $\mathcal{B}_{Squat}$ to $\mathcal{B}_{Pose}$ keeps CARL's trunk upright throughout squats. Altogether, the pictured control architecture exploits all five natural control Requirements from 2.2: *"Singular Positions"* to 2.6: *"Model Reduction"*.

The control data of the squat behavior is first composed with the compliant pose control behavior $\mathcal{B}_{Pose}$. A second composition afterward then fuses these combined control data with the controls of the *ankle strategy* balancing behavior $\mathcal{B}_{Ankle}$. The entire, herein-developed, control system of the bipedal robot CARL is sketched in Figure 5.12.

The emerging overall PBN control topology, as pictured in Figure 5.12, is well-known from the development of the PBN architecture in Chapter 4. At the hand of a simplified, virtual example experiment, presented in Figure 4.14, all individual features of the real-world control graph of CARL have been discussed in detail within Section 4.8.

CARL's control behavior $\mathcal{B}_{Squat}$ replicates the virtual experiment's behavior $\mathcal{B}_1$. Both behaviors aim to execute a predefined trajectory. In the case of CARL, this trajectory is the compliant sinusoidal squat motion, as defined in Equation (5.49). By implementing a static robot pose control goal, Behavior $\mathcal{B}_{Pose}$ equals the virtual behavior $\mathcal{B}_3$. Finally, the real-world balancing behavior $\mathcal{B}_{Ankle}$ and the virtual behavior $\mathcal{B}_2$ share identical subgraph topologies and avoidance desires. Both behaviors operate within a projection space. In the case of the real-world control graph, the PBN behavior $\mathcal{B}_{Ankle}$ aims to prevent CARL from falling.

Nevertheless, one slight topological difference is added to the PBN control graph of the robot CARL. This difference is an additional connection from $\mathcal{B}_{Squat}$ to $\mathcal{B}_{Pose}$. During a squat, the robot's thighs rotate away from their upright, vertical standing position. Since the compliant pose control behavior aims for static hip angles between the trunk and the thighs, this would result in the trunk leaning backward in line with the leg thighs.

Although such movement is possible, it looks odd and limits the vertical squat stroke due to CARL's tight mechanical ankle joint limits of ±30° w.r.t. the shanks. A backward trunk orientation would cause the robot's CoM to shift backward. Consequently, the *ankle strategy* balancing behavior would compensate for this by counter-rotating forward around the ankles. Unfortunately, the rotational ankle degrees consumed for balancing are not available for squatting. Figure 5.15 later shows the two ankle joints already closely approaching their mechanical limits during the executed squat.

Thus, the $\mathcal{B}_{Squat}$ behavior forwards an upright pose as a static compliant control goal to the behavior $\mathcal{B}_{Pose}$, further exploiting the PBN's natural control Requirement 2.5. Altogether, the complete control architecture, including the high-level PBN behaviors and the low-level antagonistic muscular control of the RRLab SEAs, as illustrated in Figure 5.12, demonstrates all five natural control Requirements from 2.2: *"Singular Positions"* to 2.6: *"Model Reduction"*.

The compliant, natural squat experiment setup is similar to the prior balancing experiments. At first, the compliant pose control behavior $\mathcal{B}_{Pose}$ initializes CARL in a stretched position from where the robot is manually moved upright, automatically activating the *ankle strategy* balancing behavior $\mathcal{B}_{Ankle}$. Once standing autonomously, the novel squat behavior $\mathcal{B}_{Squat}$ is activated. Once active, the squat behavior generates a continuous sine wave of desired spring leg equilibria according to Equation (5.49).

Due to the precision-based PBN control composition, the two knee joints are, once active, primarily influenced by the squat behavior. Thus, CARL generally follows the predefined squat trajectory while, in parallel, balancing the robot's CoM upright within the support interval by the *ankle strategy* behavior. An excerpt of one complete squat executed this way is presented in Figure 5.13.

Figure 5.13 is organized in three rows, similar to the two balancing experiments pictured in Figures 5.7a and 5.7b within Section 5.2.3. On the top, ten representative pictures capture CARL throughout its autonomous squat motion. As in the previous experiments, the given pictures are video snapshots of the same experiment whose data are plotted below them. Thus, the picture sequence shows CARL in exactly the respective real-world pose as referenced to the sense and control data plots.

The two plots below the picture sequence in Figure 5.13 represent the most relevant internal data of the two behaviors, $\mathcal{B}_{Ankle}$ and $\mathcal{B}_{Squat}$. In the upper plot, the *ankle strategy* balancing-related data are represented identically to the plots in Figure 5.7. Hence, the robot's support interval, its desired CoM reference position within the support interval, and the estimated XCoM and CoM are visualized.

Compared to the balancing experiments, it can be seen that there is significantly less horizontal motion, as expected for an undisturbed squat. However, the plot also shows severe balancing issues and oscillations during the second half of the squat, i.e., the rising motion. Due to a more parallel alignment of the two feet, CARL's available support interval is slightly smaller than recorded during balancing.

Finally, the bottom plot of Figure 5.13 visualizes the spring leg-related data of the squat behavior $\mathcal{B}_{Squat}$. The overall range of possible leg lengths is given as the interval R, depending on CARL's legs segment length s and the knees' mechanical range of motion.

$$R := \left[2s\cos\left(\tfrac{-60^\circ}{2}\right),\ 2s\cos\left(\tfrac{-0^\circ}{2}\right)\right] = [72.75,\ 84.00]\,\text{cm} \tag{5.50}$$

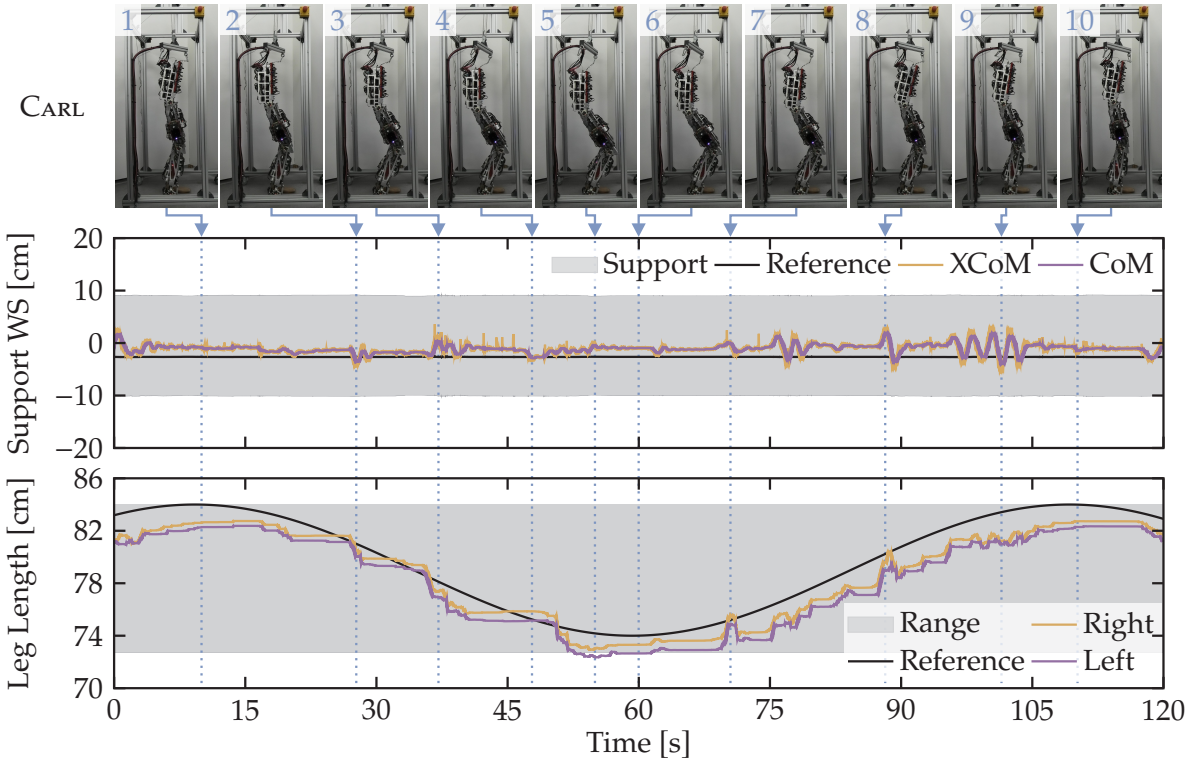


Figure 5.13: Compliant squat experiment of the bipedal robot Carl executing the control architecture of Figure 5.12. Visualized is one squat out of a continuous sequence. The squat starts in an upright pose at 10 seconds and lasts until 110 seconds. A sequence of snapshots pictures Carl at various moments throughout the experiment. Below, the support interval WS of the *ankle strategy* balancing behavior is plotted, as known from Figure 5.7. At the bottom, the novel squat behavior's equilibrium reference leg length and Carl's actual leg lengths are plotted throughout the squat. All data are recorded at 1 kHz frequency but down-sampled via overlapping, alternating minimum and maximum windows of 50 ms for representation.

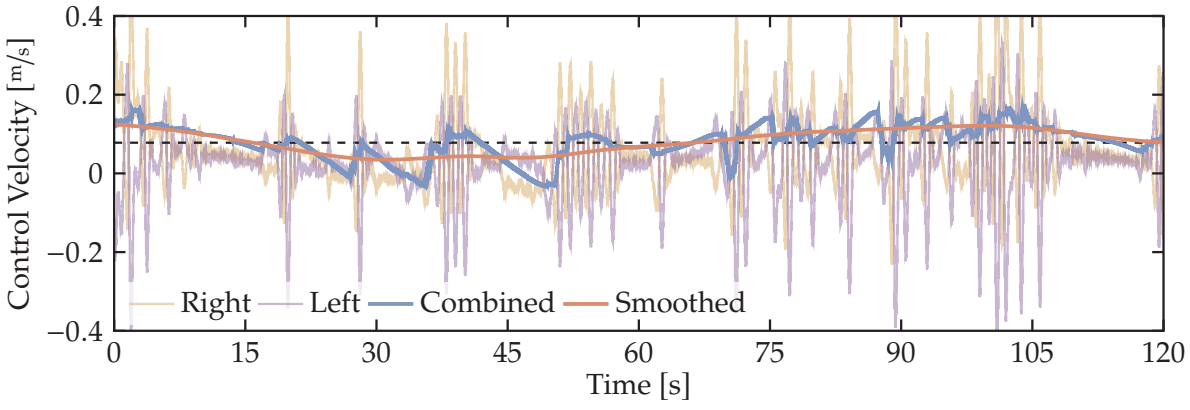


Figure 5.14: Visualization of the compliant leg length control velocities generated by the novel squat behavior on the bipedal robot Carl during the experiment shown in Figure 5.13. The two legs' raw control velocities are given in the background. In the foreground, the legs' combined control velocity is plotted in blue, and a 15 s Gaussian smoothed average of those controls is given in red. All data are recorded at 1 kHz frequency but down-sampled via overlapping, alternating minimum and maximum windows of 50 ms for representation.

Although the knees can bend up to −90°, Carl's ankle joints are mechanically limited to ±30°. Hence, when standing upright on flat ground, the ankle joints reach their mechanical limits at a knee angle of −60° far before the knees' limits[3]. The perfect black sinusoidal wave trajectory represents the spring leg control's equilibrium reference length l_0. The squat starts at the maximum possible range and features an amplitude of $a = 10\,\text{cm}$. As described in Section 5.3.1, the ideal virtual spring legs are expected to be compressed under the gravitational load of Carl's weight. Thus, as expected, the two orange and purple plots of the robot's respectively emerging actual right and left legs display shorter lengths than the predefined reference. However, both legs generally follow the controlled trajectory.

Unfortunately, the steps within both legs' signals highlight a non-neglectable amount of friction in the system. During the first half of the squat, i.e., during the downward motion, this friction helps to withstand gravity and to keep the robot in a more upright position than desired. Due to the spring legs' controlled soft compliance, this friction hysteresis overshoots if pulled down, and the robot gets stuck in a position below the desired equilibrium.

On the other hand, such a motion overshoot is much less likely during the upward motion in the second half of the squat since motion and gravity are continuously aligned against each other. Thus, the downward part of the squat motion is much more sensitive to friction than the upward part, as visible in Figure 5.13.

Interestingly, the upward motion causes significantly larger horizontal balancing disturbances than the downward motion, although it is principally smoother. Without extensive investigations into this phenomenon, two possible explanations can be given. First, it might be possible that the smaller steps simply better match the natural frequency of Carl's weight and its parasitic ankle compliances. A second explanation is about the opposite energy flow.

Going down generally removes energy from the overall system, thus causing an inherently stabilizing effect. In contrast, energy has to be brought into the robot when standing up. Consequently, the intrinsic potential of destabilizing occurrences is raised.

Finally, Figure 5.13 shows the left leg being constantly shorter than the right one. Thus, the left leg should produce higher control velocities according to the linear spring leg behavior. However, the control velocities plot in Figure 5.14 uncovers the opposite.

On average, the left leg's spring leg control velocity plotted in light purple is below the right leg's light orange control velocity graph. This discrepancy is visible in particular by the sharp control peaks. Those peaks are generated by the induced roll controller that redistributes control between the two legs to prevent any non-planar sideward roll of Carl. This additional control compensates for the higher spring leg controls of the shorter left leg to keep both legs in balance. Due to the redistribution characteristic of this non-planar controller, the blue combined control velocity does not feature those control peaks as they, by definition, cancel out each other.

The red graph of Figure 5.14 is a 15 s Gaussian smoothed average of the blue combined control velocity data. It can be seen that the averaged control signal has a noticeable sinusoidal shape. Its overall mean is visualized as a dashed line at about $0.8\,\text{m/s}$.

[3] See [Schütz 17] for further information on Carl's mechanical construction details.

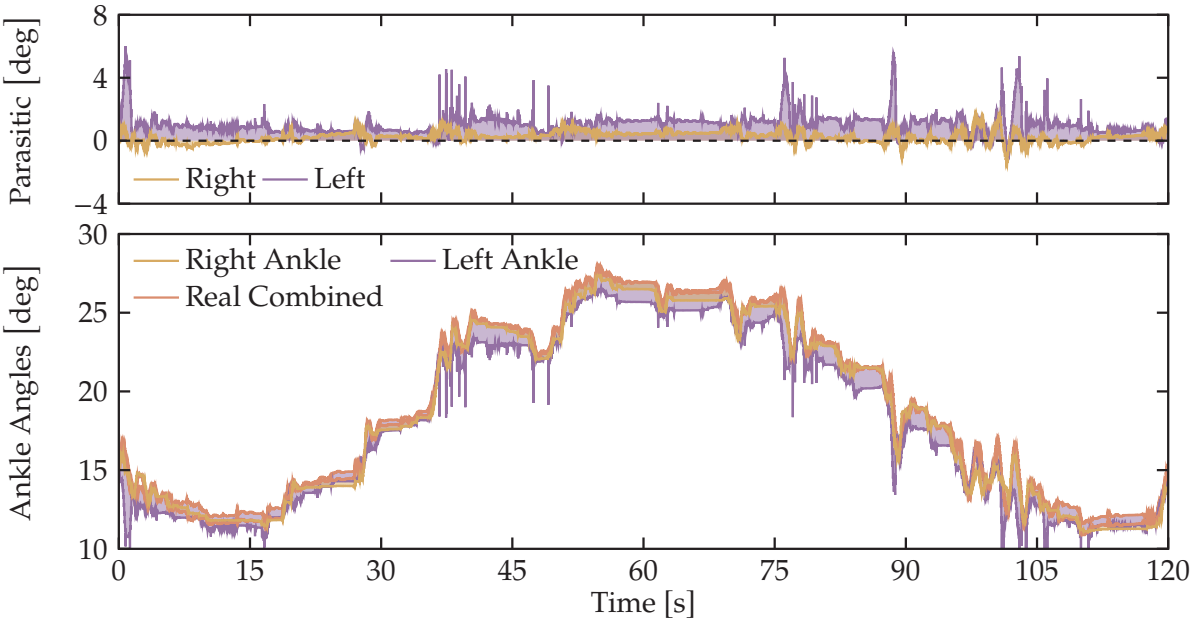


Figure 5.15: Visualization of the double ankle joints of the bipedal robot Carl during the compliant squat experiment, presented in Figure 5.13. The upper plot presents the two legs' estimated parasitic ankle joints, pictured in Figure 5.5. Below, the controlled and combined ankle angles are plotted. Both ankles follow the sinusoidal squat motion of the experiment.

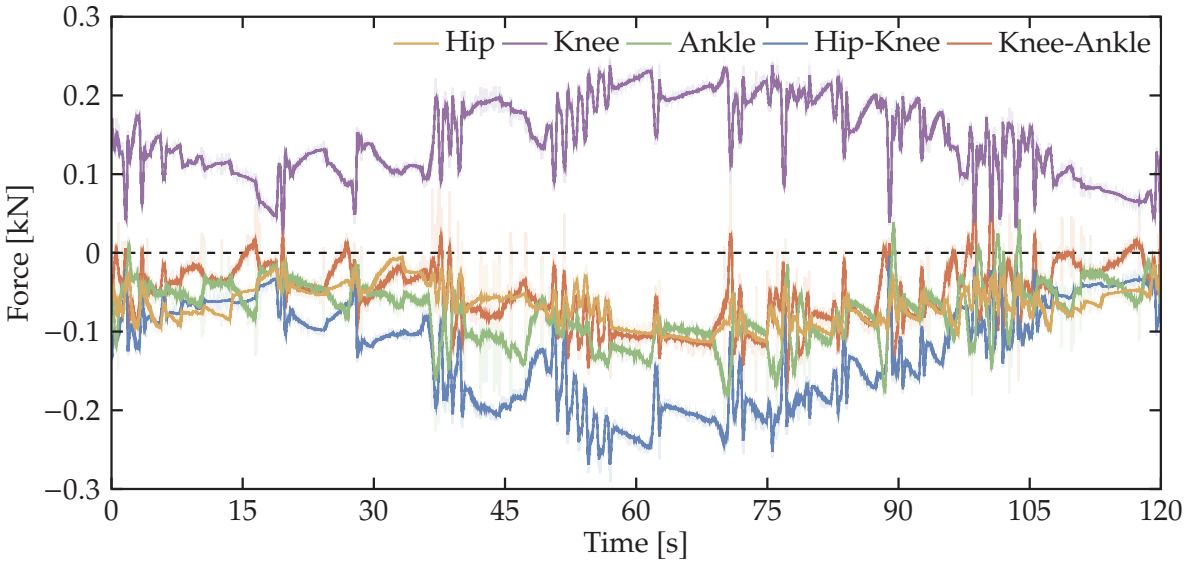
Compared to the sinusoidal reference and actual leg length trajectories in Figure 5.13, the red control signal roughly features a 90° phase shift. Since these control commands embody the time derivative of the leg lengths, the signal's shape and phase shift occur exactly as expected.

The legs' length discrepancy, as observable in Figure 5.13, has been demonstrated to be compensated and balanced by the controls presented in Figure 5.14. However, the cause of this discrepancy can be seen in the ankle angle plots in Figure 5.15. The upper plot of the secondary parasitic ankles introduced in Section 5.2.2 shows that the right foot is more or less continuously neutral. In contrast, the left parasitic ankle is continuously and significantly bent to a positive angle. With the foot sole flat on the ground, this positive angle means the robot is leaning forward.

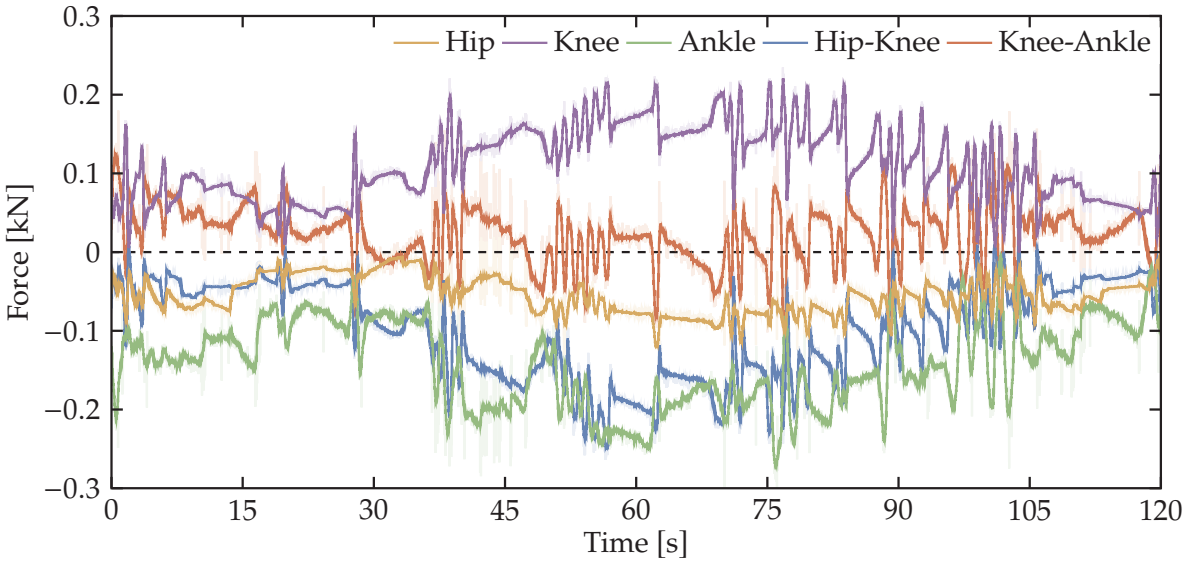
However, due to the prosthetic feet' internal leaf spring construction, pictured in Figure 5.5, a forward pitch bending also causes a respective upward bending. Hence, throughout the entire squat experiment recording, the left foot is taller than its right counterpart. Consequently, the left leg is constantly shortened to keep the robot balanced and prevent non-horizontal, non-planar roll orientation.

The lower plot in Figure 5.15, showing the combined ankle angles, does not uncover further motion details. However, it shows both ankles nicely following the legs' sinusoidal squat motion trajectory, as introduced in Figure 5.13. The two legs' nearly identical combined robotic and parasitic ankle joints indicate close-to-perfect parallel leg orientation throughout this squat experiment.

In the parasitic ankle plot in Figure 5.15, it can be seen that the right prosthetic foot is generally unloaded. This observation is confirmed by the applied actuator forces



(a) Right leg's applied actuator forces



(b) Left leg's applied actuator forces

Figure 5.16: Visualization of all actuator forces of CARL performing the compliant squat, as presented in Figure 5.13. Equal to Figure 5.9, the two graphs each plot all five applied RRLab SEA forces of the robot's (a) right and (b) left leg. Plotted are the eventual propulsion forces of the antagonistic muscular control, as sketched in Figure 3.11. All data are recorded at 1 kHz frequency but down-sampled to become representable using overlapping, alternating minimum and maximum windows of 50 ms, avoiding visual smoothing. A 50 ms Gaussian filter is applied to smooth the foreground data of all RRLab SEAs. Their respective raw data are given in the background. Compared to Figure 5.9, the runaway integration at the hip joints has been solved, and the signals generally show significantly less noise. The noise reduction is caused by the squat behavior, which substantially overrides the parallel *ankle strategy* behavior's noisy knee commands. Thus, the characteristic peak noise of the control data of the prior balancing experiments is restricted to both legs' green monoarticular ankle actuators and red biarticular knee-ankle actuators, all directly affected by the parasitic secondary ankle joints.

plotted in Figure 5.16. The plotted green monoarticular ankle force is significantly larger in the left leg than in the right. Note that the negative actuator force generates a negative torque in the morphological ankle and, thus, as expected, a positive pitch in the parasitic ankle within the prosthetic foot.

Pictured in Carl's leg morphology in Figure 3.6 within Section 3.1.3, biarticular actuators are placed on the leg's front while monoarticular actuators are located at the backside. Thus, biarticular actuators have a negative gear ratio w.r.t. their spanned joints, while monoarticular actuators feature positive gear ratios. It can be concluded that the *ankle strategy* balancing behavior active on Carl is mostly solely executed by the left leg's SEAs throughout the recorded compliant squat experiment.

In contrast, the robot's weight is carried slightly more by the right leg than the left. This observation is visible by the three knee-related actuators, the purple monoarticular knee SEA and the two blue and red biarticular SEAs. All three produce slightly higher forces in Carl's right leg. The spring leg control plot in Figure 5.14 explains this behavior.

As shown and mentioned before, the non-planar roll controller regularly unloads the left leg to balance the robot's overall body. However, to do so, Carl must have regularly leaned over its right side beforehand. Consequently, as initially discussed in Section 5.3.1, Carl's CoP regularly moves toward its right side, thus enforcing higher loads on the right leg.

Overall, the actuator plots in Figure 5.16 display the generated forces to follow the legs' squat trajectories, introduced in Figure 5.13. It is also visible that the forces in the first half of the squat are significantly lower than in the second half. This observation further strengthens the issue of significant friction within Carl's mechanical musculoskeletal construction, as discussed beforehand at the hand of the non-smooth, step-like leg lengths during the squat.

Two improvements can be observed compared to the prior balancing experiment represented in Figure 5.9 within Section 5.2.3. First, the orange monoarticular hip actuator plots prove the runaway integration of the hip joints being solved and no longer present in Figure 5.16. This improvement has been achieved by a pairwise, loose coupling of both leg's joint integrators.

Furthermore and more relevant, the control signals of the squat experiment display significantly less noise compared to the balancing experiment. The most noise can still be observed at the ankle-related SEAs, i.e., the green ankle and red knee-ankle actuator plots. However, these leftover noises come with significantly reduced amplitude.

At the same time, the noises in the monoarticular knee forces have vanished entirely in the squat experiment. The reduced noise in the knee actuators is caused by the squat behavior dominating the knee control.

As described in Equation (5.48) within Section 5.3.1, the spring leg control of the squat behavior solely depends on the knee joint sensors, thus mostly decoupling the knee control from the legs' problematic double ankle complexes. The noise reduction in the ankle and knee-ankle actuators, thus, highlights the intrinsic self-stabilizing effect of multiarticular actuators within an over-actuated musculoskeletal system, as inspired by human nature.

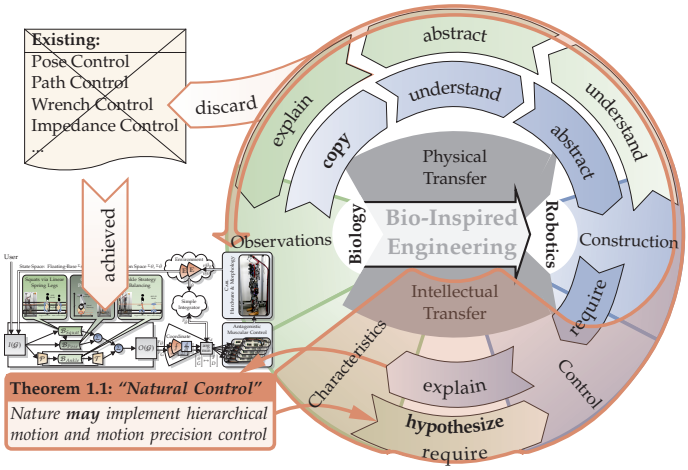
5.4 Conclusion

In the previous chapters, a novel low-level and high-level natural control philosophy has been developed. This chapter combined them into one consistent, top-to-bottom robot control system, sketched in Figure 5.12. At the hand of the planar bipedal robot CARL, this overall system architecture has been evaluated and proven functional under real-world conditions.

Three consecutive experiments were executed, eventually enabling CARL to perform compliant squats autonomously without external support. Throughout these experiments, upcoming issues with both the architecture's design and the robot CARL are highlighted and discussed in detail. Because of its unconstrained planarity, CARL currently cannot walk without additional external support. Due to the absence of such planar support structures, squat motions best possibly approximate all the natural control requirements, as derived in Chapter 2.

With the successful real-world deployment of the combined prior-developed theories, this chapter provides the essential proof of concept for the novel natural control methodology developed throughout this thesis.

6. Summary and Outlook



To achieve the overall aim of artificial natural robot control, as hypothesized in Theorem 1.1, several objectives have been accomplished throughout this thesis. Altogether, they build a consistent novel natural control philosophy based on low-level hybrid impedance control and high-level probability-based control decomposition. The natural control hypothesis is theoretically elaborated, implemented, and experimentally evaluated on real-world robotic hardware, i.e., the bipedal robot Carl[1] [Schütz 17][Schütz 20a]. Carl's prior-designed bio-inspired leg morphology replicates a planar projection of a natural human leg. Thus, Carl's legs are over-actuated, transparent, and entirely compliant, i.e., a mechanical setup mostly *uncontrollable* for existing control approaches.

Summary Outline

Section 6.1 first summarizes the most significant scientific contributions this work has achieved. Afterward, Section 6.2 briefly outlines the scope of the technical prerequisite work that has not been mentioned but underlies these objectives. Section 6.3 evaluates the novel natural control against the initial objective of six identified *"Characteristic Natural Control Requirements "*. A brief discussion of the three most interesting undisclosed issues and respective future research recommendations is provided in Section 6.4. Finally, Section 6.5 closes this thesis with a brief yet concise conclusion.

[1] Carl (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU. URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025)

6.1 Objective-Specific Contributions

This work's primary contribution and superior aim is the introduction of the novel so-called *natural control* philosophy, as hypothesized in Theorem 1.1 within Section 1.2 of the introduction. Unlike other well-known and capable robot control methodologies, the bio-inspired natural control strategy is capable of reliable and stable control of entirely compliant, i.e., nature-like, robotic morphologies. The preceding chapters covered the specification, derivation, implementation, high-level embedding, and eventual real-world evaluation of the first fully functional, consistent, top-to-bottom natural control system. A sketch of the control architecture finally deployed on Carl is pictured in Figure 5.12 within Section 5.3.2.

The overall aim of this thesis has been achieved through multiple smaller objectives, each serving the superior hypothesis of natural control. A complete summary of all these individually addressed objectives is outlined in detail in Section 1.3. In the following paragraphs, the objectives' four most significant contributions are entitled and summarized in the order of their respective chapters within this work:

Characteristic Natural Control Requirements **Chapter 2**

A small but powerful set of fundamental natural control requirements has been derived purely from six identified essential human gait characteristics independent from technical constraints. These requirements are *"Hybrid Dynamics"*, *"Singular Positions"*, *"Variable Compliance"*, *"Nonlinear Relations"*, *"Hierarchical Control"*, and *"Model Reduction"*. Altogether, they build a small but powerful foundation for bio-inspired robot and control engineering, aiming for more natural robotics and, specifically, natural robot control.

Due to their specifically intended technical independence, these six essential natural control requirements are not limited to planar natural gait but are generally valid. Thus, their contained principles and requirements are suitable for application in bio-inspired engineering outside the scope of this thesis.

Antagonistic Muscular Control – A Hybrid Impedance Approach **Chapter 3**

Antagonistic muscular control is a reduced hybrid impedance control approach for low-level Series Elastic Actuator (SEA) hardware. The bio-inspired control is specifically designed for use within mechanically over-actuated, entirely compliant hardware configurations, such as in a human's musculoskeletal system. By exploiting the benefits of antagonistic muscular control, this work firstly achieves reliable hybrid impedance control among the entirely compliant, human-like morphology of the robot Carl.

Compared to the preliminary, entirely distributed impedance control strategy, which was first deployed on Carl [*Vonwirth 17*], the herein-developed novel approach reduces the number of required control commands by 60% down to only two commands per actuator. This way, technically fast and reliable actuator-level impedance control becomes computable and scalable for more complex systems with large numbers of Degrees of Freedom (DoFs) actuators. At the same time, the minimal number of two controls per actuator replicates the antagonistic control principle of the neuromuscular control in the human Central Nervous System (CNS). Thus, the proposed hybrid impedance control layout simultaneously satisfies *bio-inspiration* from the natural neuromuscular structure and technical recommendations, such as given in [Zhao 19].

Probabilistic Behavior Networks – A Control Decomposition **Chapter 4**

To serve the uncertain, compliant low-level antagonistic muscular control principle, a consistent high-level control architecture, organized in so-called Probabilistic Behavior Networks (PBNs), is defined. Based on the established probability theory of multivariate normal distributions, this novel control calculus aims to enable Behavior-Based (BB) control layouts similar to the human CNS. Thus, PBN-based high-level robot control satisfies the prior-defined natural control requirements while simultaneously featuring a solid, semantically defined algebraic foundation.

Compared to preliminary BB control systems, specifically the so-called Integrated Behavior-Based Control (iB2C) architecture [Proetzsch 07, Ropertz 17], the herein-defined PBNs unify sense and control processing within a single probabilistic calculus. The introduction of the normal distribution's precision as novel behavior activity, for the first time, assigns a semantic meaning to this crucial, previously abstract value within BB control systems. Although strictly defined, the PBN's probabilistic representation is powerful enough to unify all state and control-relevant data representations, i.e., Cartesian vectors, symmetric matrices, and non-Cartesian orientations. At the same time, this same strict definition ensures that important algebraic properties of PBN graphs, such as associativity and commutativity, are maintained. Both are crucial for functional composition and decomposition but are absent in preliminary BB designs.

Real-World Demonstration at the Hand of Natural Squats **Chapter 5**

The experimental proof-of-concept of the combined two prior low- and high-level control concepts under real-world conditions. Several experiments have been conducted on the entirely compliant, human-shaped, lower-body robot Carl. These experiments included compliant in-air leg pose and path control, variously disturbed upright standing and balancing, and entirely autonomous compliant squats without external non-planar support structures. The experiments' most significant control and motion data are critically discussed, and undisclosed hardware and software limits are emphasized, analyzed, and explained in detail.

In comparison to preliminary control approaches deployed on Carl [*Vonwirth 17*, *Nejadfard 18a*], the herein-developed system, for the first time, can constructively coordinate all available over-actuated SEAs and stabilize the emerging motion without any external supporting structures. To the best of the author's knowledge, no other equally compliant and naturally over-actuated robot exists that is similarly and equally reliably controlled, as is Carl by the novel natural control philosophy developed throughout this thesis. The most similar natural musculoskeletal leg morphology can be found in the robot BioBiped3[2] [Sharbafi 16, Sharbafi 17b]. The most similar single-actuation principle can be found in the knee joint of Cassie[3], but, at least visually, not anymore in Cassie's commercial successor robot, Digit[4].

[2] BioBiped3 is a planar scientific lower body constructed at the *Technical University of Darmstadt* featuring series elastic actuation.

[3] Cassie is the bipedal predecessor robot of Digit, both produced by the company *Agility Robotics*. URL: `https://agilityrobotics.com/` (2025)

[4] Digit is a commercial humanoid developed and produced by the company *Agility Robotics*. URL: `https://agilityrobotics.com` (2025)

6.2 Technical Support Work

Aside from the previously described scientific contributions, additional necessary technical support work has been undertaken throughout the entire research, development, and evaluation process of the work at hand. Experience gathered during the work on this thesis has shown that, unexceptionally, all involved components, from the first screw to the latest implemented mathematical control theory, create a toolchain that, commonly known, *"is just as strong as its weakest link"*.

Thus, a perfectly designed, implemented, and theoretically proven top-level control policy can, will, and, throughout the work on this thesis, did fail due to a single missing bit in a sensor's filtered signal data resolution. The mechanical, electrical, and logical hardware systems underlying the herein-developed novel control approaches are not only supporting technologies but critically essential foundations the higher-level, more abstract scientific contributions entirely rely on.

Among others, the most significant unmentioned technical labor that has been done to enable this thesis' scientific contributions includes the following:

1. Assembly:
   The mechanical and electrical assembly of the bipedal robot Carl, including 10 stand-alone FPGA-based sense and control boards[5] embedding distributed, low-latency, and concurrent hardware interfaces and signal preprocessing, such as high-frequency filtering.

2. Calibration:
   The mechanical and digital calibration and parameterization of all 26 skeletal and SEA sensors[6] and 10 brushless DC motors[7] in Carl's musculoskeletal hardware.

3. Communication:
   The design and implementation of an ETH-based UDP/IP communication protocol connecting the embedded FPGAs with a central standard CPU computer at a bidirectional 1 kHz sense and control frequency.

4. Logic Synthesis:
   The design and synthesis of the embedded FPGA SoC logic. The logic includes the ETH communication interface, the sensor signal preprocessing, and the logic-level implementation of the antagonistic muscular control, as sketched in Figure 3.11 within Section 3.2.3.

5. Framework Integration:
   The integration of the centralized robot communication interface, the high-level PBN robot control architecture, and the individual *compliant pose control*, ankle strategy *balancing*, and *linear spring leg squat* behaviors, as sketched in Figure 5.12 within Section 5.3.2, into the robot control framework Finroc[8].

[5] The boards used throughout this work are based on an Altera Cyclone IV FPGA (today Intel).

[6] All encoders built into Carl are the iC-Haus iC-MU magnetic absolute position encoders.

[7] All actuators of Carl are TQ-Group TQ-RoboDrive ILM50x14 and ILM70x18 brushless DC motors, individually controlled by Elmo Motion Control Gold Twitter servo drives using current control mode.

[8] Finroc is a modular and concurrent robot control framework developed at the RPTU.
URL: `https://www.finroc.org/` (2025) [Reichardt 12, Reichardt 13]

6.3 Achieved and Demonstrated Natural Control Requirements on CARL

In Chapter 2, the small set of human gait characteristics has been identified and derived into six fundamental natural control requirements for natural robot control. A representative review of various existing bipedal robot and simulation control approaches, along with analyses of their compliance with these requirements, has been provided throughout Section 2.3. These reviews include and are grouped into rigid control approaches, compliant control approaches, and muscular control approaches. Figures 2.11, 2.12, and 2.13 each visualize the individual evaluations of two representative robots and their respective control approaches against the six demanded natural control requirements.

Analog to those analyses, the bipedal robot CARL, equipped with the novel natural control system, is analyzed in the same manner. A non-empirical visualization of the respective analysis conclusion for CARL compared to the prior-analyzed state-of-the-art approaches is given in Figure 6.1. It is important to note that the presented evaluation results solely reflect the approaches' compliance with the six demanded natural control requirements. Thus, there is no statement about the individual approaches' general motion and control capabilities within the given diagram.

As shown in Figure 2.14 within Section 2.4, Requirement 2.1: *"Hybrid Dynamics"* is a primarily mechanical requirement on a robot's morphology. The robot CARL's hardware fully satisfies this requirement by using only mechanically compliant SEAs for actuation throughout its entire musculoskeletal construction. The single-leg version of CARL, equipped with an entirely different control approach, demonstrated its morphology's general capability of hybrid dynamics through the successful execution of a cyclic gait on a treadmill [*Vonwirth 17*][Schütz 20a]. Furthermore, the experiments described herein in Sections 3.5, 5.2.3, and 5.3.2 cover successful stable leg motion control both in the air and on the ground.

However, a real-world experiment that combines and thus couples these two dynamic domains while running the novel-developed natural control system is yet to be conducted. Thus, although principally satisfied, additional experimental proof-of-concept is necessary to verify the hybrid dynamics requirement entirely.

The natural control Requirement 2.2: *"Singular Positions"* is fulfilled twice by two independent methodologies. As sketched in Figure 5.1 within Section 5.1, singularity control is enabled by the possibility of applying direct joint control within the singular-free Configuration Space (CS). This possibility is similar to all reviewed singularity-capable approaches, which operate in either the CS or the Actuation Space (AS).

However, in contrast to the reviewed control approaches, the herein-defined PBN control also allows abstract Work Space (WS) control within a singular position to be smoothly reverse-projected into the robot's CS. Such a singular mapping is enabled by using second-derivative optimization gradients to approximate controls represented in their first-derivative. The appropriate quartic (4th-degree) optimization algorithm is outlined in Section 4.7.2 and derived and described in detail in [*Vonwirth 24b*].

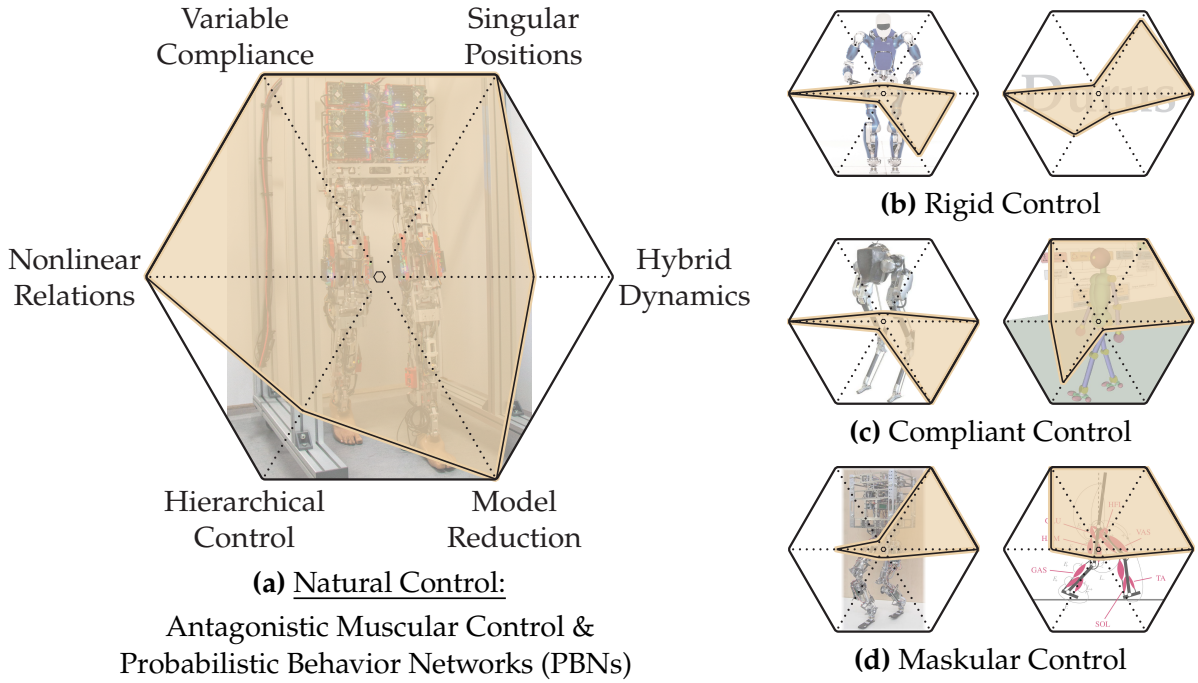


Figure 6.1: Non-empiric illustration of the individual compliance with the natural control requirements derived from the six essential human gait characteristics for the entire natural control system deployed on (a) CARL. For better comparison, the six prior state-of-the-art analysis visualizations are reprinted on the right. These are the two representative plots each for (b) rigid control approaches of Figure 2.11, (c) compliant control approaches of Figure 2.12, and (d) muscular control approaches of Figure 2.13 within Sections 2.3.1, 2.3.2, and 2.3.3, respectively. The visualization highlights the general compliance of the novel natural control system with all demanded requirements. However, more advanced experiments are necessary to declare complete satisfaction with the Requirements of *"Hybrid Dynamics"* and *"Hierarchical Control"*.

Among the six reviewed control approaches, Requirement 2.3: *"Variable Compliance"* is realized through two different principles. The compliant control approach of the B4LC[9] [Luksch 10a, Zhao 16] robot simulation, analyzed in Figure 2.12b, uses actuator-level pure virtual stiffness rendering. In contrast, the muscular reflex control simulation, analyzed in Figure 2.13b, modulates compliance through the pretension of constant, nonlinear stiffness properties built into their nature-like simulated muscles. Notably, both those approaches are simulations.

Thus, the herein-developed bio-inspired SEA control applied to the robot CARL realizes the only functional hardware implementation of actuator-level variable compliance among the herein-compared robots. To achieve this, the novel antagonistic muscular control principle relies on a SEA hardware setup with a fixed mechanical elasticity, virtual rendered damping, and high mechanical transparency, as developed and described in Section 3.2.3.

In Section 2.1.4, the natural control Requirement 2.4: *"Nonlinear Relations"* is derived from the observable linear spring leg behavior, illustrated in Figure 2.8. In the herein-

[9] B4LC is a scientific humanoid robot simulation by the RPTU developed to study compliant, natural dynamics-exploiting gait.

demonstrated natural control system, as sketched in Figure 5.12, this exact spring leg behavior is reproduced by the natural squat behavior, as sketched in Figures 5.10 and 5.11a within Section 5.3.

Its real-world capability is demonstrated by the successful natural squat experiment with the robot CARL, as presented in Section 5.3.2. Within the applied high-level PBN control architecture, this virtual spring leg's nonlinear relation to the primarily affected knee joint is achieved through control linearization by a first-derivative data representation. Details on the PBN control data representation and its consequent beneficial mathematical properties are explained in Section 4.4.5.

Requirement 2.5: *"Hierarchical Control"* is one of the most significant derivations from the human CNS. The hierarchical control characteristic of the CNS identified throughout Section 2.1.5 underlies the foundational bio-inspired requirements for both novel low- and high-level control strategies developed herein.

At the actuation level, this is realized as a partially distributed hybrid impedance control via the novel antagonistic muscular control principle developed in Section 3.2.3. At the higher control level, the hierarchical CNS structure most significantly motivates the functional requirements of the novel PBN architecture, as defined in Section 4.1 and visualized within Figure 4.1.

Functional hierarchical control between centralized PBN behaviors and distributed actuator-intrinsic impedance has been demonstrated successfully on the robot CARL. However, the internal topology of the high-level PBN control graphs in the executed experiments is primarily flat. Thus, similar to Requirement 2.1: *"Hybrid Dynamics"*, it is necessary to demonstrate more complex hierarchical high-level control topologies to demonstrate CARL's compliance with the hierarchical control requirement fully.

Similar to hierarchical control, Requirement 2.6: *"Model Reduction"* is embedded within the control decomposition principles of the BB high-level PBN architecture. Derived and described throughout Section 4.7, Figures 4.10 and 4.12 illustrate the linearized transformations into and back out of arbitrary, lower-dimensional projection spaces.

A real-world evaluation of this architectural feature is demonstrated through the chosen realization of the upright *ankle strategy* balancing behavior. As shown in the applied control's sketch in Figure 5.6, the actual *ankle strategy* behavior is embedded within an abstract, one-dimensional support WS, described in Section 5.2.1.

The entire PBN subgraph topology that enables CARL to balance autonomously is sketched in Figure 5.4. As designed by the PBN architecture, the topology is a sequence of three behaviors. A projection behavior first reduces the robot's entire state space to an equivalent, one-dimensional support WS representation. It is followed by the actual *ankle strategy* behavior that operates purely on this reduced, one-dimensional model. A third and final transformation behavior projects the abstract support WS control data back into CS joint controls. The successful execution of the complete natural squat control system, as sketched in Figure 5.12, eventually demonstrated the fluent integration of PBN behaviors across different levels of abstraction.

6.4 Discussion and Future Work Recommendations

Despite the addressed objectives and achieved contributions to more natural robot control, Figure 1.3 and Section 1.1 illustrate and describe the general process of bio-inspired robotic development as a continuous, iterative cycle. The work at hand contributes fundamental iterations of this process but, of course, does not define its end. Thus, while this thesis has solved several essential problems, it has also naturally uncovered new, consecutive follow-up issues that require further investigations.

The paragraphs in the following outlook entitle and summarize the three most significant of all identified open issues and offer briefly reasoned recommendations on promising future research directions:

Neuromuscular Hybrid Impedance Control **Chapter 3**

Inspired by a human's neuromuscular system, Theorem 1.1 hypothesizes natural control as a combination of motion and motion precision. The herein-derived antagonistic muscular control implements this principle on the actuator level by combined force and damping control. However, within the higher-level PBN architecture, this is interchanged into velocity and responsiveness, respectively.

The concrete implementation of the two motion and motion precision controls is irrelevant to the overall effectiveness of the system. Hence, nonlinear or energy-based realizations of these two controls may be beneficial to consider in future studies.

Unified and Linearized Representation **Chapter 4**

Throughout Section 4.4, the high-level control architecture defined herein, i.e., the PBNs, offers a single consistent calculus for all types of natural control-relevant data. Unfortunately, the chosen traditional representation enforces three entirely different necessary data formats: Cartesian Vectors, Matrices, and Rotors. Consequently, a significant amount of format-specific calculation is inevitable. However, as discussed in Section 4.4.2 and recommended in Section 4.4.4, a higher-dimensional, projective representation, such as in Projective Geometric Algebra (PGA) or even Conformal Geometric Algebra (CGA), can unify all those individual data formats into algebraically identical multivectors.

With such a higher-dimensional representation, infinity becomes representable, and geometric operations, including rigid-body dynamics, are linearized and calculable without exceptions, i.e., without the need for dedicated exception handling. Furthermore, the robot's current minimal DH state representation may be replaced by a unified and generalized robot representation via constrained maximal coordinates.

Probabilistic Modeling and Continuous Online Learning **Chapters 4 and 5**

Every floating-base robot, by definition, features passive DoFs. Thus, to enable more sophisticated and higher-level abstract control goals for Carl in the future, some form of Model Predictive Control (MPC) has to be integrated into the robot control system. In the herein-developed architecture, this is intended to be realized as the interface between the high-level PBN control and the low-level antagonistic muscular control.

As discussed throughout Section 5.1, this modeling is a non-trivial task and has been left open due to the limited scope of this thesis. However, the following discussion provides a brief impression of this problem's underlying complexity and its strong interconnection with the two recommendations above. Additionally, basic directional guidelines for future studies are outlined.

Firstly, the chosen representation format must match the above-discussed high-level data representation and, at the same time, has to serve the two control commands required by the low-level antagonistic muscular control principle.

Second, the chosen model representation must be trainable and should be self-adapting. It is state-of-the-art common knowledge that hand-tuned dynamic models are not sufficient for realistic motion prediction or simulation. Thus, the chosen representation must be suitable for applying Machine Learning (ML) techniques to its data structure.

To automatically react to changing intrinsic and extrinsic dynamics, those techniques should be capable of unsupervised online adaptation via self-observation. Since the PBN design is already based on probability theory, techniques such as Gaussian Process Regression (GPR), which natively embed precision information within their regression models, may be beneficial. Successful robotic application examples of GPR can be found in [Umlauft 20, Beckers 22].

Finally, with the robot dynamics modeled as a stochastic Gaussian process, the same view can be applied to the PBN control series described in Section 4.4.6. As visualized in Figure 4.4c, the herein-defined PBN control series already encodes a continuously diverging Gaussian process, i.e., featuring strictly monotone decreasing precision. However, the direct control commands within those series might be replaceable with more abstract control policies that the above-mentioned MPC can subsequently interpret. This way, both more sophisticated control series and control goals further in the future may become achievable.

6.5 Thesis Conclusion

It is simple to do something complex,
yet it is complex to do the same simple.

With the neuromuscular-inspired concept of *antagonistic muscular control*, this thesis significantly simplifies the general complexity of stable and reliable distributed impedance control. Thus, it enables and empowers controllability of over-actuated and entirely compliant hardware setups, as present in nature and rebuilt with the human-like morphology of the bipedal robot CARL.

To stably control the inherent motion uncertainty of such a hybrid low-level impedance control approach, the novel high-level *Probabilistic Behavior Networks* (PBNs) provide a fully functional, though yet complex, probabilistic implementation of BB robot control.

Simplification of this novel architectural control theory, consequently, yet has to follow.

Bibliography

Literature With Contribution of the Author

[*Giebler 24*] F. A. Giebler, "Modeling and Compensation of Gravitational Forces on the Bipedal Robot CARL", Bachelor's Thesis, RPTU Kaiserslautern-Landau, Kaiserslautern, Germany, 2024. unpublished, supervised by **P. Vonwirth**.

[*Hackenberg 23*] A. Hackenberg, "Gravity Compensation for Compliant Locomotion on the Bipedal Robot CARL", Bachelor's Thesis, RPTU Kaiserslautern-Landau, Kaiserslautern, Germany, 2023. unpublished, supervised by **P. Vonwirth**.

[*Mianowski 24*] K. Mianowski, K. Berns, **P. Vonwirth**, O. Sivak, "Modern Methods of Designing and Producing Elements for Walking Robots Working in Highly Dynamic Conditions", in *Walking Robots into Real World*, ser. Lecture Notes in Networks and Systems, vol. 2. Kaiserslautern, Germany: Springer Nature Switzerland, 2024, pp. 107–118. doi:10.1007/978-3-031-71301-9_11, [online] `https://link.springer.com/chapter/10.1007/978-3-031-71301-9_11`.

[*Nejadfard 18a*] A. Nejadfard, S. Schütz, **P. Vonwirth**, K. Mianowski, K. Berns, "Coordination of the Biarticular Actuators Based on Instant Power in an Explosive Jump Experiment", in *IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Auckland, New Zealand, 2018.

[*Nejadfard 18b*] A. Nejadfard, S. Schütz, **P. Vonwirth**, K. Mianowski, B. Karsten, "Moment Arm Analysis of the Biarticular Actuators in Compliant Robotic Leg CARL", in *Conference on Biomimetic and Biohybrid Systems*. Paris, France: Springer International Publishing, 2018, pp. 348–360. doi:10.1007/978-3-319-95972-6_37.

[*Nejadfard 19*] A. Nejadfard, S. Schütz, K. Mianowski, **P. Vonwirth**, K. Berns, "Design of the Musculoskeletal Leg CARL Based on the Physiology of Mono-Articular and Biarticular Muscles in the Human Leg", *Bioinspiration & Biomimetics*, vol. 14, no. 6, p. 066002, 2019. doi:10.1088/1748-3190/ab3896, [online] `https://iopscience.iop.org/article/10.1088/1748-3190/ab3896` Accessed: 2022-05-03.

[*Nejadfard 20*] A. Nejadfard, K. Berns, **P. Vonwirth**, "Technical Advantages and Disadvantages of Biarticular Actuators in Bipedal Robots", in *Robots in Human Life – Proceedings of the 23rd International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. Moscow, Russian Federation: CLAWAR Association Ltd., 2020, pp. 166–174. doi:10.13180/clawar.2020.24-26.08.48, [online] `https://clawar.org/wp-content/uploads/2021/02/Clawar2020_Paper_48.pdf` Accessed: 2022-05-30.

[*Schütz 17*] S. Schütz, A. Nejadfard, K. Mianowski, **P. Vonwirth**, K. Berns, "CARL – A Compliant Robotic Leg Featuring Mono- and Biarticular Actuation", in *IEEE-RAS International Conference on Humanoid Robots*. Birmingham, UK, 2017.

[*Sivak 22*] O. Sivak, "FPGA-Based Emulation of Natural Muscle Behavior on an Electric Series Elastic Actuator", Master thesis, University of Kaiserslautern, Kaiserslautern, Germany, 2022. unpublished, supervised by **P. Vonwirth**.

[*Sivak 24a*] O. Sivak, K. Mianowski, **P. Vonwirth**, B. Karsten, "Lightweight Human-Like Robotic Leg with Four-Bar Mechanism Joints", in *Walking Robots into Real World*, ser. Lecture Notes in Networks and Systems, vol. 1. Kaiserslautern, Germany: Springer Nature Switzerland, 2024, pp. 255–265. doi:10.1007/978-3-031-70722-3_26, [online] `https://link.springer.com/chapter/10.1007/978-3-031-70722-3_26`.

[*Sivak 24b*] O. Sivak, **P. Vonwirth**, K. Berns, "FPGA-Based Emulation of a Muscle Stretch Reflex on an Electric Series Elastic Actuator", in *Synergetic Cooperation between Robots and Humans*, ser. Lecture Notes in Networks and Systems, vol. 811. Springer Nature Switzerland, 2024, pp. 299–310. doi:10.1007/978-3-031-47272-5_25, [online] `https://link.springer.com/10.1007/978-3-031-47272-5_25` Accessed: 2024-01-11.

[*Vonwirth 17*] **P. Vonwirth**, "Modular Control Architecture for Bipedal Walking on a Single Compliant Leg", Master thesis, University of Kaiserslautern, Kaiserslautern, Germany, 2017. unpublished, supervised by Steffen Schütz.

[*Vonwirth 20a*] **P. Vonwirth**, A. Nejadfard, K. Berns, "Biologically Inspired Bipedal Locomotion—From Control Concept to Human-Like Biped", in *Proceedings of 14th International Conference on Electromechanics and Robotics "Zavalishin's Readings"*, A. Ronzhin, V. Shishlakov, Eds., vol. 154. Singapore: Springer Singapore, 2020, pp. 3–14. doi:10.1007/978-981-13-9267-2_1, [online] `http://link.springer.com/10.1007/978-981-13-9267-2_1` Accessed: 2019-10-09.

[*Vonwirth 20b*] **P. Vonwirth**, A. Nejadfard, K. Mianowski, K. Berns, "SLIP-Based Concept of Combined Limb and Body Control of Force-Driven Robots", in *Advances in Service and Industrial Robotics*, vol. 84. Cham: Springer International Publishing, 2020, pp. 547–556. doi:10.1007/978-3-030-48989-2_58, [online] `http://link.springer.com/10.1007/978-3-030-48989-2_58` Accessed: 2020-07-27.

[*Vonwirth 22*] **P. Vonwirth**, K. Berns, "Continuous Inverse Kinematics in Singular Position", in *Robotics for Sustainable Future*, D. Chugo, M. O. Tokhi, M. F. Silva, T. Nakamura, K. Goher, Eds., vol. 324. Cham: Springer International Publishing, 2022, pp. 24–36. doi:10.1007/978-3-030-86294-7_3, [online] https://link.springer.com/10.1007/978-3-030-86294-7_3 Accessed: 2021-09-21.

[*Vonwirth 23a*] **P. Vonwirth**, K. Berns, "Bio-Inspired Imprecise Impedance Control of Muscle-Driven Robotic Limbs", in *Robotics in Natural Settings, CLAWAR 2022*, ser. Lecture Notes in Networks and Systems, J. M. Cascalho, M. O. Tokhi, M. F. Silva, A. Mendes, K. Goher, M. Funk, Eds., vol. 530. Ponta Delgada, Portugal: Springer International Publishing, 2023, pp. 42–53. doi:10.1007/978-3-031-15226-9_7, [online] https://link.springer.com/10.1007/978-3-031-15226-9_7 Accessed: 2022-10-17.

[*Vonwirth 23b*] **P. Vonwirth**, K. Berns, "Muscular Damping Distribution Strategy for Bio-Inspired, Soft Motion Control at Variable Precision", *Sensors*, vol. 23, no. 5, 2023. doi:10.3390/s23052428, [online] https://www.mdpi.com/1424-8220/23/5/2428.

[*Vonwirth 24a*] **P. Vonwirth**, O. Sivak, K. Berns, "Foundations of Probabilistic Behavior Networks Aiming for Structured, Distributed Control of Complex Systems like Legged Robots", in *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. Heidelberg, Germany: IEEE, 2024, pp. 593–598. doi:10.1109/BioRob60516.2024.10719847, [online] https://ieeexplore.ieee.org/document/10719847.

[*Vonwirth 24b*] **P. Vonwirth**, A. Vierling, O. Sivak, K. Berns, "Intuitive Motion : Acceleration-based Inverse Kinematics on Arbitrary Coordinates", in *Walking Robots into Real World*, ser. Lecture Notes in Networks and Systems, vol. 1. Kaiserslautern, Germany: Springer Nature Switzerland, 2024, pp. 239–251. doi:10.1007/978-3-031-70722-3_24, [online] https://link.springer.com/chapter/10.1007/978-3-031-70722-3_24.

Regular Literature

[Aceituno-Cabezas 18] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernandez-Lopez, C. Semini, "Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization", *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2018. doi:10.1109/LRA.2017.2779821, [online] http://ieeexplore.ieee.org/document/8141917/ Accessed: 2022-05-23.

[Agility Robotics 22] Agility Robotics, "Does Our Robot Have Bird Legs?", oct. 2022. [online] https://www.youtube.com/watch?v=9B2CVZe0jFE Accessed: 2024-09-01.

[Al-Shuka 14] H. F. N. Al-Shuka, F. Allmendinger, B. Corves, W.-H. Zhu, "Modeling, Stability and Walking Pattern Generators of Biped Robots: A Review", *Robotica*, vol. 32, no. 6, pp. 907–934, 2014. doi:10.1017/S0263574713001124, [online] https://www.cambridge.org/core/product/identifier/S0263574713001124/type/journal_article Accessed: 2024-09-09.

[Alexander 76] R. McN. Alexander, "Mechanics of Bipedal Locomotion", in *Zoology*, 1976, pp. 493–504. doi:10.1016/B978-0-08-018767-9.50047-0, [online] https://www.sciencedirect.com/science/article/pii/B9780080187679500470 Accessed: 2024-07-11.

[Ames 14] A. D. Ames, "Human-Inspired Control of Bipedal Walking Robots", *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1115–1130, 2014.

[Antoniak 19] G. Antoniak, T. Biswas, N. Cortes, S. Sikdar, C. Chun, V. Bhandawat, "Spring-Loaded Inverted Pendulum Goes through Two Contraction-Extension Cycles during the Single-Support Phase of Walking", *Biology Open*, vol. 8, no. 6, p. bio043695, june 2019. doi:10.1242/bio.043695, [online] http://bio.biologists.org/lookup/doi/10.1242/bio.043695 Accessed: 2019-12-18.

[Armbrust 15] C. Armbrust, *Design and Verification of Behaviour-Based Systems Realising Task Sequences*, ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut, 2015. [online] https://www.dr.hut-verlag.de/978-3-8439-2261-6.html.

[Asada 87] H. Asada, K. Youcef-Toumi, *Direct-Drive Robots Theory and Practice*, Cambridge, Mass.: MIT Press, 1987. [online] https://ieeexplore.ieee.org/servlet/opac?bknumber=6267213.

[Asfour 06] T. Asfour, D. N. Ly, K. Regenstein, R. Dillmann, "Coordinated Task Execution for Humanoid Robots", in *Experimental Robotics IX*, B. Siciliano, O. Khatib, F. Groen, M. H. Ang, O. Khatib, Eds., vol. 21. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 259–267. doi:10.1007/11552246, [online] http://link.springer.com/10.1007/11552246_25 Accessed: 2023-05-24.

[Attwell 01] D. Attwell, S. B. Laughlin, "An Energy Budget for Signaling in the Grey Matter of the Brain", *Journal of Cerebral Blood Flow & Metabolism*, vol. 21, no. 10, pp. 1133–1145, 2001. doi:10.1097/00004647-200110000-00001, [online] http://journals.sagepub.com/doi/10.1097/00004647-200110000-00001 Accessed: 2024-05-18.

[Ball 76] R. S. Ball, "The Theory of Screws: A Study in the Dynamics of a Rigid Body", *Math. Ann.*, vol. 9, pp. 541–553, 1876. doi:10.1007/BF01442479, [online] https://link.springer.com/article/10.1007/BF01442479.

[Barasuol 18] V. Barasuol, O. A. Villarreal-Magaña, D. Sangiah, M. Frigerio, M. Baker, R. Morgan, G. A. Medrano-Cerda, D. G. Caldwell, C. Semini, "Highly-Integrated Hydraulic Smart Actuators and Smart Manifolds for High-Bandwidth Force Control", *Frontiers in Robotics and AI*, vol. 5, p.5̃1, 2018. doi:10.3389/frobt.2018.00051, [online] https://www.frontiersin.org/article/10.3389/frobt.2018.00051/full Accessed: 2022-10-28.

[Batts 15] Z. Batts, S. Song, H. Geyer, "Toward a Virtual Neuromuscular Control for Robust Walking in Bipedal Robots", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, 2015, pp. 6318–6323. doi:10.1109/IROS.2015.7354279.

[Beckers 22] T. Beckers, S. Hirche, "Prediction With Approximated Gaussian Process Dynamical Models", *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6460–6473, 2022. doi:10.1109/TAC.2021.3131988, [online] https://ieeexplore.ieee.org/document/9632353/ Accessed: 2025-02-07.

[Bekey 94] G. A. Bekey, M. A. Lewis, "Biologically Inspired Robot Control", in *IFAC Proceedings Volumes*, ser. 11, vol. 27. Smolenice, Slovak Republic, 1994, pp. 379–383. doi:10.1016/S1474-6670(17)47679-8, [online] https://linkinghub.elsevier.com/retrieve/pii/S1474667017476798 Accessed: 2024-01-07.

[Ber 14] *Constrained Optimization and Lagrange Multiplier Methods*, ser. Computer Science and Applied Mathematics, Burlington: Elsevier Science, 2014. [online] https://books.google.de/books?id=j6LiBQAAQBAJ. OCLC: 1222666612.

[Best 15] C. M. Best, J. P. Wilson, M. D. Killpack, "Control of a Pneumatically Actuated, Fully Inflatable, Fabric-Based, Humanoid Robot", in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. Seoul, Republic of Korea: IEEE, 2015, pp. 1133–1140. doi:10.1109/HUMANOIDS.2015.7363495, [online] http://ieeexplore.ieee.org/document/7363495/ Accessed: 2022-10-28.

[Bingham 74] C. Bingham, "An Antipodally Symmetric Distribution on the Sphere", *The Annals of Statistics*, vol. 2, no. 6, pp. 1201–1225, 1974. [online] http://www.jstor.org/stable/2958339 Accessed: 2025-01-19.

[Blank 09] S. Blank, T. Wahl, T. Luksch, K. Berns, "Biologically Inspired Compliant Control of a Monopod Designed for Highly Dynamic Applications", in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. St. Louis, MO, USA: IEEE, 2009, pp. 148–153. doi:10.1109/IROS.2009.5354404, [online] http://ieeexplore.ieee.org/document/5354404/ Accessed: 2024-11-01.

[Blickhan 89] R. Blickhan, "The Spring-Mass Model for Running and Hopping", *Journal of biomechanics*, vol. 22, no. 11–12, pp. 1217–1227, 1989.

[Borelli 85] G. A. Borelli, *De motu animalium*, 2 ed., Leiden, The Netherlands: Cornelis Boutesteyn, 1685, vol. 1. [online] https://www.deutsches-museum.de/forschung/bibliothek/unsere-schaetze/biologie-medizin/motu-animalium. German translation 1706 titled 'Von der wundersamen Macht der Muskuln'.

[Bouman 20] A. Bouman, M. F. Ginting, N. Alatur, M. Palieri, D. D. Fan, T. Touma, T. Pailevanian, S.-K. Kim, K. Otsu, J. Burdick, A.-a. Agha-Mohammadi, "Autonomous Spot: Long-Range Autonomous Exploration of Extreme Environments with Legged Locomotion", in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Las Vegas, NV, USA: IEEE, 2020, pp. 2518–2525. doi:10.1109/IROS45743.2020.9341361, [online] https://ieeexplore.ieee.org/document/9341361/ Accessed: 2022-03-10.

[Boyd 07] S. P. Boyd, "Least-Norm Solutions of Underdetermined Equations", Stanford Engineering Everywhere, 2007. [online] https://see.stanford.edu/materials/lsoeldsee263/08-min-norm.pdf Accessed: 2025-03-15.

[Brooks 86] R. Brooks, "A Robust Layered Control System for a Mobile Robot", *IEEE Journal on Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986. doi:10.1109/JRA.1986.1087032, [online] http://ieeexplore.ieee.org/document/1087032/ Accessed: 2023-05-24.

[Brown 00] I. E. Brown, G. E. Loeb, "A Reductionist Approach to Creating and Using Neuromusculoskeletal Models", in *Biomechanics and Neural Control of Posture and Movement*, J. M. Winters, P. E. Crago, Eds., New York, NY: Springer New York, 2000, pp. 148–163. doi:10.1007/978-1-4612-2104-3_10, [online] http://link.springer.com/10.1007/978-1-4612-2104-3_10 Accessed: 2022-03-15.

[Bunz 23] E. K. Bunz, D. F. B. Haeufle, C. D. Remy, S. Schmitt, "Bioinspired Preactivation Reflex Increases Robustness of Walking on Rough Terrain", *Scientific Reports*, vol. 13, no. 13219, 2023. doi:10.1038/s41598-023-39364-3, [online] https://www.nature.com/articles/s41598-023-39364-3 Accessed: 2024-09-25.

[Bunz 24] E. K. Bunz, L. H. Pawusch, S. Schmitt, "Optimizing Reflex-Based Neuromusculoskeletal Walking Model on Rough Terrain Reveals Increased Robustness and Key Stabilizing Reflexes", in *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. Heidelberg, Germany: IEEE, 2024, pp. 477–482. doi:10.1109/BioRob60516.2024.10719737, [online] https://ieeexplore.ieee.org/document/10719737/ Accessed: 2025-01-07.

[Buschmann 09] T. Buschmann, S. Lohmeier, H. Ulbrich, "Humanoid Robot Lola: Design and Walking Control", *Journal of Physiology-Paris*, vol. 103, no. 3, pp. 141–148, 2009. doi:10.1016/j.jphysparis.2009.07.008, [online] https://linkinghub.elsevier.com/retrieve/pii/S0928425709000400 Accessed: 2024-01-04.

[Caccavale 98a] F. Caccavale, C. Natale, B. Siciliano, L. Villani, "Experiments of Spatial Impedance Control", in *Experimental Robotics V*, vol. 232. Barcelona, Spain: Springer, Berlin, Heidelberg, 1998, pp. 91–104. doi:10.1007/BFb0112953, [online] `https://link.springer.com/chapter/10.1007/BFb0112953` Accessed: 2025-01-16.

[Caccavale 98b] F. Caccavale, B. Siciliano, L. Villani, "Quaternion-Based Impedance with Nondiagonal Stiffness for Robot Manipulators", in *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, vol. 1. Philadelphia, PA, USA: IEEE, 1998, pp. 468–472. doi:10.1109/ACC.1998.694712, [online] `http://ieeexplore.ieee.org/document/694712/` Accessed: 2024-10-11.

[Caccavale 99] F. Caccavale, B. Siciliano, L. Villani, "Robot Impedance Control with Nondiagonal Stiffness", *IEEE Transactions on Automatic Control*, vol. 44, no. 10, pp. 1943–1946, 1999. doi:10.1109/9.793782, [online] `http://ieeexplore.ieee.org/document/793782/` Accessed: 2024-10-11.

[Campbell 98] K. S. Campbell, M. Lakie, "A Cross-Bridge Mechanism Can Explain the Thixotropic Short-Range Elastic Component of Relaxed Frog Skeletal Muscle", *The Journal of Physiology*, vol. 510, no. 3, pp. 941–962, aug. 1998. doi:10.1111/j.1469-7793.1998.941bj.x.

[Caron 18] S. Caron, B. Mallein, "Balance Control Using Both ZMP and COM Height Variations: A Convex Boundedness Approach", in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, QLD: IEEE, may 2018, pp. 1779–1784. doi:10.1109/ICRA.2018.8460942, [online] `https://ieeexplore.ieee.org/document/8460942/` Accessed: 2019-11-25.

[Cashaback 15] J. G. A. Cashaback, T. Cluff, "Increase in Joint Stability at the Expense of Energy Efficiency Correlates with Force Variability during a Fatiguing Task", *Journal of Biomechanics*, vol. 48, no. 4, pp. 621–626, 2015. doi:10.1016/j.jbiomech.2014.12.053.

[Chellapurath 23] M. Chellapurath, P. C. Khandelwal, A. K. Schulz, "Bioinspired Robots Can Foster Nature Conservation", *Frontiers in Robotics and AI*, vol. 10, no. 1145798, 2023. doi:10.3389/frobt.2023.1145798, [online] `https://www.frontiersin.org/articles/10.3389/frobt.2023.1145798/full` Accessed: 2024-04-25.

[Collins 05] S. Collins, A. Ruina, R. Tedrake, M. Wisse, "Efficient Bipedal Robots Based on Passive-Dynamic Walkers", *American Association for the Advancement of Science*, vol. 307, no. 5712, pp. 1082–1085, 2005. doi:10.1126/science.1107799, [online] `https://www.science.org/doi/10.1126/science.1107799` Accessed: 2024-07-09.

[Cousineau 15] E. Cousineau, A. D. Ames, "Realizing Underactuated Bipedal Walking with Torque Controllers via the Ideal Model Resolved Motion Method", in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 5747–5753. doi:10.1109/ICRA.2015.7140004.

[Creath 05] R. Creath, T. Kiemel, F. Horak, R. Peterka, J. Jeka, "A Unified View of Quiet and Perturbed Stance: Simultaneous Co-Existing Excitable Modes", *Neuroscience Letters*, vol. 377, no. 2, pp. 75–80, march 2005. doi:10.1016/j.neulet.2004.11.071, [online] https://linkinghub.elsevier.com/retrieve/pii/S0304394004014867 Accessed: 2020-07-01.

[Dederick 68] Z. P. Dederick, I. Grass, "Improvement in Steam-Carriage", US Patent Nr. 75874, 1868.

[Demtröder 17] W. Demtröder, *Mechanics and Thermodynamics*, 1 ed., ser. Undergraduate Lecture Notes in Physics, Springer Cham, 2017. doi:10.1007/978-3-319-27877-3, [online] http://link.springer.com/10.1007/978-3-319-27877-3 Accessed: 2025-06-14.

[Denavit 55] J. Denavit, R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215–221, 1955. doi:10.1115/1.4011045, [online] https://asmedigitalcollection.asme.org/appliedmechanics/article-abstract/22/2/215/1110292/A-Kinematic-Notation-for-Lower-Pair-Mechanisms.

[Drama 19] Ö. Drama, A. Badri-Spröwitz, "Trunk Pitch Oscillations for Joint Load Redistribution in Humans and Humanoid Robots", *arXiv:1909.03687 [cs]*, sept. 2019. [online] http://arxiv.org/abs/1909.03687 Accessed: 2019-12-18.

[Dyer 10] F. L. Dyer, T. C. Martin, *Edison, His Life and Inventions*, 1910. [online] https://www.gutenberg.org/ebooks/820.

[Eilenberg 10] M. F. Eilenberg, H. Geyer, H. Herr, "Control of a Powered Ankle–Foot Prosthesis Based on a Neuromuscular Model", *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 2, pp. 164–173, 2010. doi:10.1109/TNSRE.2009.2039620, [online] https://ieeexplore.ieee.org/document/5378631/ Accessed: 2024-11-15.

[Elmenreich 07] W. Elmenreich, "Fusion of Continuous-valued Sensor Measurements Using Confidence-weighted Averaging", *Journal of Vibration and Control*, vol. 13, no. 9–10, pp. 1303–1312, 2007. doi:10.1177/1077546307077457, [online] https://journals.sagepub.com/doi/abs/10.1177/1077546307077457 Accessed: 2023-05-31.

[Englsberger 11] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, G. Hirzinger, "Bipedal Walking Control Based on Capture Point Dynamics", in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sept. 2011, pp. 4420–4427. doi:10.1109/IROS.2011.6094435.

[Englsberger 13] J. Englsberger, C. Ott, A. Albu-Schäffer, "Three-Dimensional Bipedal Walking Control Using Divergent Component of Motion", in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Tokyo, Japan: IEEE, 2013, pp. 2600–2607.

[Englsberger 14] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid et al, "Overview of the Torque-Controlled Humanoid Robot TORO", in *IEEE-RAS International Conference on Humanoid Robots 2014*. nov. 2014, pp. 916–923. doi:10.1109/HUMANOIDS.2014.7041473.

[Englsberger 15] J. Englsberger, C. Ott, A. Albu-Schäffer, "Three-Dimensional Bipedal Walking Control Based on Divergent Component of Motion", *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015. doi:10.1109/TRO.2015.2405592.

[Fankhauser 18] P. Fankhauser, M. Hutter, "ANYmal: A Unique Quadruped Robot Conquering Harsh Environments", *Research Features*, vol. 126, pp. 54–57, 2018. doi:10.3929/ETHZ-B-000262484, [online] `http://hdl.handle.net/20.500.11850/262484` Accessed: 2022-10-28.

[Farina 13] M. Farina, L. Giulioni, L. Magni, R. Scattolini, "A Probabilistic Approach to Model Predictive Control", in *52nd IEEE Conference on Decision and Control*. Florence, Italy: IEEE, 2013, pp. 7734–7739. doi:10.1109/CDC.2013.6761117, [online] `http://ieeexplore.ieee.org/document/6761117/` Accessed: 2025-04-01.

[Featherstone 08] R. Featherstone, *Rigid Body Dynamics Algorithms*, New York: Springer, 2008. OCLC: ocn190774140.

[Felis 16] M. L. Felis, K. Mombaur, "Synthesis of Full-Body 3-D Human Gait Using Optimal Control Methods", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, 2016, pp. 1560–1566. doi:10.1109/ICRA.2016.7487294, [online] `https://ieeexplore.ieee.org/abstract/document/7487294`.

[Ficht 21] G. Ficht, S. Behnke, "Bipedal Humanoid Hardware Design: A Technology Review", *Current Robotics Reports*, vol. 2, no. 2, pp. 201–210, 2021. doi:10.1007/s43154-021-00050-9, [online] `https://link.springer.com/10.1007/s43154-021-00050-9` Accessed: 2022-10-28.

[Folgheraiter 23] M. Folgheraiter, S. Yessirkepov, T. Umurzakov, "NU-Biped-4.5: A Lightweight and Low-Prototyping-Cost Full-Size Bipedal Robot", *Robotics*, vol. 13, no. 9, 2023. doi:10.3390/robotics13010009, [online] `https://www.mdpi.com/2218-6581/13/1/9` Accessed: 2024-09-25.

[Franklin 18] G. F. Franklin, J. D. Powell, A. Emami-Naeini, *Feedback Control of Dynamic Systems, 8th Edition*, 8 ed., Pearson, 2018.

[Garrett 83] M. Garrett, R. G. Luckwill, "Role of Reflex Responses of Knee Musculature during the Swing Phase of Walking in Man", *European Journal of Applied Physiology and Occupational Physiology*, vol. 52, pp. 36–41, 1983. doi:10.1007/BF00429022, [online] `http://link.springer.com/10.1007/BF00429022` Accessed: 2024-07-13.

[Gebicke-Haerter 23] P. J. Gebicke-Haerter, "The Computational Power of the Human Brain", *Frontiers in Cellular Neuroscience*, vol. 17, no. 1220030, p.2̃6, 2023. doi:10.3389/fncel.2023.1220030, [online] `https://www.frontiersin.org/articles/10.3389/fncel.2023.1220030/full` Accessed: 2024-04-26.

[Geijtenbeek 13] T. Geijtenbeek, M. van de Panne, A. F. van der Stappen, "Flexible Muscle-based Locomotion for Bipedal Creatures", *ACM Trans. Graph.*, vol. 32, no. 6, pp. 206:1–206:11, 2013. doi:10.1145/2508363.2508399, [online] http://doi.acm.org/10.1145/2508363.2508399.

[Geyer 06] H. Geyer, A. Seyfarth, R. Blickhan, "Compliant Leg Behaviour Explains Basic Dynamics of Walking and Running", *Proceedings of the Royal Society B: Biological Sciences*, vol. 273, no. 1603, pp. 2861–2867, 2006. doi:10.1098/rspb.2006.3637, [online] https://royalsocietypublishing.org/doi/10.1098/rspb.2006.3637 Accessed: 2019-12-18.

[Geyer 10] H. Geyer, H. Herr, "A Muscle-Reflex Model That Encodes Principles of Legged Mechanics Produces Human Walking Dynamics and Muscle Activities", *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 18, no. 3, pp. 263–273, 2010. doi:10.1109/TNSRE.2010.2047592.

[Ghansah 23] A. B. Ghansah, J. Kim, M. Tucker, A. D. Ames, "Humanoid Robot Co-Design: Coupling Hardware Design with Gait Generation via Hybrid Zero Dynamics", in *2023 62nd IEEE Conference on Decision and Control (CDC)*. Singapore, Singapore: IEEE, 2023, pp. 1879–1885. doi:10.1109/CDC49753.2023.10383745, [online] https://ieeexplore.ieee.org/document/10383745/ Accessed: 2024-06-11.

[Gilitschenski 14] I. Gilitschenski, G. Kurz, S. J. Julier, U. D. Hanebeck, "A New Probability Distribution for Simultaneous Representation of Uncertain Position and Orientation", in *17th International Conference on Information Fusion (FUSION)*. Salamanca, Spain: IEEE, 2014, pp. 1–7.

[Granata 01] K. P. Granata, K. F. Orishimo, "Response of Trunk Muscle Coactivation to Changes in Spinal Stability", *Journal of biomechanics*, vol. 34, no. 9, pp. 1117–1123, 2001.

[Gribble 03] P. L. Gribble, L. I. Mullin, N. Cothros, A. Mattar, "Role of Cocontraction in Arm Movement Accuracy", *Journal of Neurophysiology*, vol. 89, no. 5, pp. 2396–2405, 2003. doi:10.1152/jn.01020.2002.

[Grizzle 14] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, A. D. Ames, "Models, Feedback Control, and Open Problems of 3D Bipedal Robotic Walking", *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.

[Groote 17] F. D. Groote, J. L. Allen, L. H. Ting, "Contribution of Muscle Short-Range Stiffness to Initial Changes in Joint Kinetics and Kinematics during Perturbations to Standing Balance: A Simulation Study", *Journal of Biomechanics*, vol. 55, pp. 71–77, april 2017. doi:10.1016/j.jbiomech.2017.02.008, [online] http://www.sciencedirect.com/science/article/pii/S0021929017301045.

[Haeufle 12] D. F. B. Haeufle, M. D. Taylor, S. Schmitt, H. Geyer, "A Clutched Parallel Elastic Actuator Concept: Towards Energy Efficient Powered Legs in Prosthetics and Robotics", in *2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. Rome, Italy: IEEE, 2012, pp. 1614–1619. doi:10.1109/BioRob.2012.6290722, [online] http://ieeexplore.ieee.org/document/6290722/ Accessed: 2024-07-10.

[Haeufle 20] D. F. B. Haeufle, I. Wochner, D. Holzmüller, D. Driess, M. Günther, S. Schmitt, "Muscles Reduce Neuronal Information Load: Quantification of Control Effort in Biological vs. Robotic Pointing and Walking", *Frontiers in Robotics and AI*, vol. 7, p.77̃, 2020. doi:10.3389/frobt.2020.00077, [online] `https://www.frontiersin.org/article/10.3389/frobt.2020.00077/full` Accessed: 2021-10-25.

[Hamill 14] J. Hamill, K. M. Knutzen, T. R. Derrick, *Biomechanical Basis of Human Movement, 4th Edition*, 4 ed., Wolters Kluwer, 2014.

[Handharu 08] N. Handharu, J. Yoon, G. Kim, "Gait Pattern Generation with Knee Stretch Motion for Biped Robot Using Toe and Heel Joints", in *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*. Daejeon, Korea: IEEE, 2008, pp. 265–270. doi:10.1109/ICHR.2008.4755956, [online] `http://ieeexplore.ieee.org/document/4755956/` Accessed: 2024-06-30.

[Henze 14] B. Henze, C. Ott, M. A. Roa, "Posture and Balance Control for Humanoid Robots in Multi-Contact Scenarios Based on Model Predictive Control", in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. sept. 2014. doi:10.1109/IROS.2014.6943014.

[Henze 16] B. Henze, M. A. Roa, C. Ott, "Passivity-Based Whole-Body Balancing for Torque-Controlled Humanoid Robots in Multi-Contact Scenarios", *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016. doi:10.1177/0278364916653815.

[Hereid 15] A. Hereid, C. M. Hubicki, E. A. Cousineau, J. W. Hurst, A. D. Ames, "Hybrid Zero Dynamics Based Multiple Shooting Optimization with Applications to Robotic Walking", in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5734–5740.

[Hereid 16] A. Hereid, E. A. Cousineau, C. M. Hubicki, A. D. Ames, "3D Dynamic Walking with Underactuated Humanoid Robots: A Direct Collocation Framework for Optimizing Hybrid Zero Dynamics", in *IEEE International Conference on Robotics and Automation (ICRA). IEEE*. 2016, pp. 1447–1454. doi:10.1109/ICRA.2016.7487279.

[Hill 38] A. V. Hill, "The Heat of Shortening and Dynamic Constants of Muscles", *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 126, no. 843, pp. 136–195, 1938.

[Hobbelen 08] D. Hobbelen, T. De Boer, M. Wisse, "System Overview of Bipedal Robots Flame and TUlip: Tailor-made for Limit Cycle Walking", in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Nice, France: IEEE, 2008, pp. 2486–2491. doi:10.1109/IROS.2008.4650728, [online] `http://ieeexplore.ieee.org/document/4650728/` Accessed: 2024-07-09.

[Hof 08] A. L. Hof, "The 'Extrapolated Center of Mass' Concept Suggests a Simple Control of Balance in Walking", *Human movement science*, vol. 27, no. 1, pp. 112–125, 2008.

[Holzinger 20] A. Holzinger, P. Kieseberg, A. M. Tjoa, E. Weippl, Eds., *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings*, ser. Lecture Notes in Computer Science, Springer Cham, 2020, vol. 12279. doi:10.1007/978-3-030-57321-8, [online] https://link.springer.com/10.1007/978-3-030-57321-8 Accessed: 2024-05-18.

[Holzinger 22] A. Holzinger, R. Goebel, R. Fong, T. Moon, K.-R. Müller, W. Samek, Eds., *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, ser. Lecture Notes in Computer Science, Springer Cham, 2022, vol. 13200. doi:10.1007/978-3-031-04083-2, [online] https://link.springer.com/10.1007/978-3-031-04083-2 Accessed: 2024-05-18.

[Hosokawa 18] M. Hosokawa, D. N. Nenchev, T. Hamano, "The DCM Generalized Inverse: Efficient Body-Wrench Distribution in Multi-Contact Balance Control", *Advanced Robotics*, vol. 32, no. 14, pp. 778–792, 2018. doi:10.1080/01691864.2018.1503095, [online] https://www.tandfonline.com/doi/full/10.1080/01691864.2018.1503095 Accessed: 2021-01-11.

[Hubicki 16] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, J. Hurst, "ATRIAS: Design and Validation of a Tether-Free 3D-capable Spring-Mass Bipedal Robot", *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.

[Hufschmidt 87] A. Hufschmidt, I. Schwaller, "Short-Range Elasticity and Resting Tension of Relaxed Human Lower Leg Muscles", *The Journal of Physiology*, vol. 391, no. 1, pp. 451–465, 1987.

[Hutter 16] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, M. Hoepflinger, "ANYmal - a Highly Mobile and Dynamic Quadrupedal Robot", in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea: IEEE, 2016, pp. 38–44. doi:10.1109/IROS.2016.7758092, [online] http://ieeexplore.ieee.org/document/7758092/ Accessed: 2022-03-10.

[Hutter 17] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Bloesch, H. Kolvenbach, M. Bjelonic, L. Isler, K. Meyer, "ANYmal - toward Legged Robots for Harsh Environments", *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017. doi:10.1080/01691864.2017.1378591, [online] https://www.tandfonline.com/doi/full/10.1080/01691864.2017.1378591 Accessed: 2022-10-28.

[Hyon 09] S.-H. Hyon, R. Osu, Y. Otaka, "Integration of Multi-Level Postural Balancing on Humanoid Robots", in *2009 IEEE International Conference on Robotics and Automation*. 2009. doi:10.1109/ROBOT.2009.5152434.

[Izzi 23] F. Izzi, A. Mo, S. Schmitt, A. Badri-Spröwitz, D. F. B. Haeufle, "Muscle Prestimulation Tunes Velocity Preflex in Simulated Perturbed Hopping", *Scientific Reports*, vol. 13, no. 4559, 2023. doi:10.1038/s41598-023-31179-6, [online] https://www.nature.com/articles/s41598-023-31179-6 Accessed: 2023-06-19.

[Kalman 60] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", *ASME Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960. doi:10.1115/1.3662552, [online] https://asmedigitalcollection.asme.org/fluidsengineering/article-abstract/82/1/35/397706 Accessed: 2024-07-21.

[Karniel 13] A. Karniel, "The Minimum Transition Hypothesis for Intermittent Hierarchical Motor Control", *Frontiers in Computational Neuroscience*, vol. 7, no. 12, 2013. doi:10.3389/fncom.2013.00012, [online] http://journal.frontiersin.org/article/10.3389/fncom.2013.00012/abstract Accessed: 2024-01-17.

[Katayama 23] S. Katayama, M. Murooka, Y. Tazaki, "Model Predictive Control of Legged and Humanoid Robots: Models and Algorithms", *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023. doi:10.1080/01691864.2023.2168134, [online] https://www.tandfonline.com/doi/full/10.1080/01691864.2023.2168134 Accessed: 2024-04-26.

[Kawakami 15] T. Kawakami, K. Hosoda, "Bipedal Walking with Oblique Mid-Foot Joint in Foot", in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, dez. 2015, pp. 535–540.

[Khadke 21] A. Khadke, H. Geyer, "Policy Decomposition: Approximate Optimal Control with Suboptimality Estimates", in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. Munich, Germany: IEEE, 2021, pp. 185–192. doi:10.1109/HUMANOIDS47582.2021.9555796, [online] https://ieeexplore.ieee.org/document/9555796/ Accessed: 2024-01-25.

[Khatri 77] C. G. Khatri, K. V. Mardia, "The von Mises–Fisher Matrix Distribution in Orientation Statistics", *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 39, no. 1, pp. 95–106, 1977. doi:10.1111/j.2517-6161.1977.tb01610.x, [online] https://academic.oup.com/jrsssb/article/39/1/95/7027450 Accessed: 2024-10-28.

[Kiekbusch 17] L. Kiekbusch, "Analysis and Verification of Complex Robot Systems Using Behaviour-Based Control", Doctoralthesis, University of Kaiserslautern, Kaiserslautern, Germany, 2017. [online] https://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-46162 Accessed: 2025-01-19.

[Kurazume 05] R. Kurazume, S. Tanaka, M. Yamashita, T. Hasegawa, K. Yoneda, "Straight Legged Walking of a Biped Robot", in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Edmonton, AB, Canada: IEEE, 2005, pp. 337–343. doi:10.1109/IROS.2005.1545447, [online] http://ieeexplore.ieee.org/document/1545447/ Accessed: 2024-06-30.

[Lakatos 18] D. Lakatos, K. Ploeger, F. Loeffl, D. Seidel, F. Schmidt, T. Gumpert, F. John, T. Bertram, A. Albu-Schaffer, "Dynamic Locomotion Gaits of a Compliantly Actuated Quadruped With SLIP-Like Articulated Legs Embodied in the Mechanical Design", *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3908–3915, 2018. doi:10.1109/LRA.2018.2857511, [online] https://ieeexplore.ieee.org/document/8413123/ Accessed: 2024-09-12.

[Lance 81] J. W. Lance, J. G. McLeod, *A Physiological Approach to Clinical Neurology*, 3 ed., London Boston Sydney Wellington Durban Toront: Butterworths, 1981. doi:10.1016/C2013-0-03931-X, [online] https://www.sciencedirect.com/book/9780407001961/a-physiological-approach-to-clinical-neurology.

[Landgraf 23] D. Landgraf, A. Völz, F. Berkel, K. Schmidt, T. Specker, K. Graichen, "Probabilistic Prediction Methods for Nonlinear Systems with Application to Stochastic Model Predictive Control", *Annual Reviews in Control*, vol. 56, p. 100905, 2023. doi:10.1016/j.arcontrol.2023.100905, [online] https://linkinghub.elsevier.com/retrieve/pii/S136757882300069X Accessed: 2025-04-01.

[Lee 17] J. Lee, M. N. Vu, Y. Oh, "A Control Method for Bipedal Trunk Spring Loaded Inverted Pendulum Model", in *The Thirteenth International Conference on Autonomic and Autonomous Systems*. 2017, pp. 24–29.

[Li 21] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, K. Sreenath, "Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots", 2021. [online] http://arxiv.org/abs/2103.14295v1 Accessed: 2021-10-26.

[Lin 15] PH. Lin, TK. Shen, PC. Lin, YR. Zeng, HC. Wu, LC. Wu, WP. Shih, "Designing Synergistic Walking of a Humanoid Single Leg Driven by Pneumatic Artificial Muscles: An Empirical Study", in *Proceedings of the 14th IFToMM World Congress*. oct. 2015, pp. 478–483. doi:10.6567/IFToMM.14TH.WC.OS13.090.

[Liu 15] Q. Liu, J. Zhao, S. Schütz, K. Berns, "Adaptive Motor Patterns and Reflexes for Bipedal Locomotion on Rough Terrain", in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany, 2015.

[Liu 16] Q. Liu, J. Zhao, K. Berns, "Dynamically Stepping Over Large Obstacle Utilizing PSO Optimization in the B4LC System", in *Advances in Cooperative Robotics: Proceedings of the 19th International Conference on Clawar 2016*. London, UK, 2016, p. 428.

[Liu 18] X. Liu, A. Rosendo, S. Ikemoto, M. Shimizu, K. Hosoda, "Robotic Investigation on Effect of Stretch Reflex and Crossed Inhibitory Response on Bipedal Hopping", *Journal of The Royal Society Interface*, vol. 15, no. 140, 2018. doi:10.1098/rsif.2018.0024, [online] https://royalsocietypublishing.org/doi/10.1098/rsif.2018.0024 Accessed: 2024-07-14.

[Lohmeier 09] S. Lohmeier, T. Buschmann, H. Ulbrich, "Humanoid Robot LOLA", in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 775–780. doi:10.1109/ROBOT.2009.5152578, [online] http://ieeexplore.ieee.org/document/5152578/ Accessed: 2024-01-04.

[Loram 02] I. D. Loram, M. Lakie, "Direct Measurement of Human Ankle Stiffness during Quiet Standing: The Intrinsic Mechanical Stiffness Is Insufficient for Stability", *The Journal of Physiology*, vol. 545, no. 3, pp. 1041–1053, 2002. doi:10.1113/jphysiol.2002.025049.

[Luksch 10a] T. Luksch, *Human-like Control of Dynamically Walking Bipedal Robots*, ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut, 2010.

[Luksch 10b] T. Luksch, K. Berns, "Control of Bipedal Walking Exploiting Postural Reflexes and Passive Dynamics", in *IEEE International Conference on Applied Bionics and Biomechanics (ICABB)*. Venice, Italy: Citeseer, 2010.

[Lyapunov 92] A. M. Lyapunov, "The General Problem of the Stability of Motion", *Taylor \& Francis*, vol. 55, no. 3, pp. 531–534, 1992. doi:10.1080/00207179208934253, [online] https://www.tandfonline.com/doi/abs/10.1080/00207179208934253?src=recsys. Translated from French to Englishby A.T. Fuller, Translated from Russian to French by Edouard Davaux from the original doctoral thesis 1892, Univ. Kharkiv.

[Mandery 16] C. Mandery, O. Terlemez, M. Do, N. Vahrenkamp, T. Asfour, "Unifying Representations and Large-Scale Whole-Body Motion Databases for Studying Human Motion", *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 796–809, aug. 2016. doi:10.1109/TRO.2016.2572685, [online] http://ieeexplore.ieee.org/document/7506114/ Accessed: 2021-03-18.

[McGeer 90] T. McGeer, "Passive Dynamic Walking", *The International Journal of Robotics Research*, vol. 9, no. 2, pp. 62–82, 1990. doi:10.1177/027836499000900206, [online] https://journals.sagepub.com/doi/10.1177/027836499000900206.

[McGhee 68] R. B. McGhee, A. A. Frank, "On the Stability Properties of Quadruped Creeping Gaits", *Mathematical Biosciences*, vol. 3, pp. 331–351, 1968. doi:10.1016/0025-5564(68)90090-4, [online] https://linkinghub.elsevier.com/retrieve/pii/0025556468900904 Accessed: 2024-01-21.

[McMahon 90] T. A. McMahon, G. C. Cheng, "The Mechanics of Running: How Does Stiffness Couple with Speed?", *Journal of Biomechanics*, vol. 23, pp. 65–78, 1990. doi:10.1016/0021-9290(90)90042-2, [online] https://www.sciencedirect.com/science/article/pii/0021929090900422 Accessed: 2024-07-11.

[Mergner 10] T. Mergner, "A Neurological View on Reactive Human Stance Control", *Annual Reviews in Control*, vol. 34, no. 2, pp. 177–198, dez. 2010. doi:10.1016/j.arcontrol.2010.08.001, [online] https://linkinghub.elsevier.com/retrieve/pii/S1367578810000404 Accessed: 2025-04-04.

[Mesbah 16] A. Mesbah, "Stochastic Model Predictive Control: An Overview and Perspectives for Future Research", *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016. doi:10.1109/MCS.2016.2602087, [online] https://ieeexplore.ieee.org/document/7740982/ Accessed: 2025-04-01.

[Mihailoff 18] G. A. Mihailoff, D. E. Haines, "Chapter 24 - Motor System I: Peripheral Sensory, Brainstem, and Spinal Influence on Anterior Horn Neurons", in *Fundamental Neuroscience for Basic and Clinical Applications (Fifth Edition)*, Fifth Edition ed., Elsevier, 2018, pp. 346–359.e1. doi:10.1016/B978-0-323-39632-5.00024-4, [online] https://linkinghub.elsevier.com/retrieve/pii/B9780323396325000244 Accessed: 2022-03-15.

[Mikolajczyk 22] T. Mikolajczyk, E. Mikołajewska, H. F. N. Al-Shuka, T. Malinowski, A. Kłodowski, D. Y. Pimenov, T. Paczkowski, F. Hu, K. Giasin, D. Mikołajewski, M. Macko, "Recent Advances in Bipedal Walking Robots: Review of Gait, Drive, Sensors and Control Systems", *Sensors*, vol. 22, no. 12, 2022. doi:10.3390/s22124440, [online] https://www.mdpi.com/1424-8220/22/12/4440 Accessed: 2024-01-03.

[Milner 02] T. E. Milner, "Adaptation to Destabilizing Dynamics by Means of Muscle Cocontraction", *Experimental Brain Research*, vol. 143, no. 4, pp. 406–416, 2002. doi:10.1007/s00221-002-1001-4.

[Milner 98] T. E. Milner, C. Cloutier, "Damping of the Wrist Joint during Voluntary Movement", *Experimental Brain Research*, vol. 122, no. 3, pp. 309–317, 1998. doi:10.1007/s002210050519.

[Ministere du commerce, de l'industrue des postes et des telegraphes 01] Ministere du commerce, de l'industrue des postes et des telegraphes, *Concours Internationaux d'exercices physiques et de sports (Official Report of the 1900 Olympic Games)*, Paris, France: Digitally published by the LA84 Foundation, 1901, vol. 1\&2. [online] https://digital.la84.org/digital/collection/p17103coll8/id/7199/rec/3. Today, this work is within the Public Domain since more than 100 years have passed.

[Mirvakili 18] S. M. Mirvakili, I. W. Hunter, "Artificial Muscles: Mechanisms, Applications, and Challenges", *Advanced Materials*, vol. 30, no. 6, p. 1704407, 2018. doi:10.1002/adma.201704407, [online] https://onlinelibrary.wiley.com/doi/10.1002/adma.201704407 Accessed: 2022-10-28.

[Miura 11] K. Miura, M. Morisawa, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, "Human-like Walking with Toe Supporting for Humanoids", in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA, USA: IEEE, 2011, pp. 4428–4435. [online] https://ieeexplore.ieee.org/abstract/document/6094838.

[Mochon 80] S. Mochon, T. A. McMahon, "Ballistic Walking", *Journal of Biomechanics*, vol. 13, no. 1, pp. 49–57, 1980. doi:10.1016/0021-9290(80)90007-X, [online] https://www.sciencedirect.com/science/article/pii/002192908090007X Accessed: 2024-07-11.

[Mohseni 22] O. Mohseni, A. M. N. Rashty, A. Seyfarth, K. Hosoda, M. A. Sharbafi, "Bioinspired Legged Robot Design via Blended Physical and Virtual Impedance Control", *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, p.22, 2022. doi:10.1007/s10846-022-01631-2, [online] https://link.springer.com/10.1007/s10846-022-01631-2 Accessed: 2022-05-16.

[Moini 20] J. Moini, P. Piran, *Functional and Clinical Neuroanatomy – A Guide for Health Care Professionals*, 1 ed., London, UK: Academic Press (imprint of Elsevier), 2020. doi:10.1016/C2018-0-01786-7, [online] https://www.sciencedirect.com/book/9780128174241/functional-and-clinical-neuroanatomy.

[Mussa-Ivaldi 85] F. Mussa-Ivaldi, N. Hogan, E. Bizzi, "Neural, Mechanical, and Geometric Factors Subserving Arm Posture in Humans", *The Journal of Neuroscience*, vol. 5, no. 10, pp. 2732–2743, 1985. doi:10.1523/JNEUROSCI.05-10-02732.1985, [online] http://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.05-10-02732.1985 Accessed: 2019-11-25.

[Muybridge 78] E. Muybridge, "The Horse in Motion. "Sallie Gardner", Owned by Leland Stanford; Running at a 1:40 Gait over the Palo Alto Track", june 1878. [online] https://www.loc.gov/item/97502309/ Accessed: 2024-10-04.

[Muybridge 87] E. Muybridge, "Animal Locomotion", 1887. [online] https://lccn.loc.gov/92502812 Accessed: 2024-10-04.

[Na 23] G. Na, H. Nabae, K. Suzumori, "Braided Thin McKibben Muscles for Musculoskeletal Robots", *Sensors and Actuators A: Physical*, vol. 357, no. 114381, p.1̃1, 2023. doi:10.1016/j.sna.2023.114381, [online] https://linkinghub.elsevier.com/retrieve/pii/S0924424723002303 Accessed: 2024-09-23.

[Nezhadfard 20] A. Nezhadfard, *Integration and Design of Actuation Redundancy in Robotic Leg CARL Based on the Physiology of Biarticular Muscles*, ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut GmbH, 2020. [online] https://www.dr.hut-verlag.de/978-3-8439-4598-1.html.

[Nguyen 16] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, K. Sreenath, "3D Dynamic Walking on Stepping Stones with Control Barrier Functions", in *IEEE International Conference on Decision and Control (CDC)*, vol. 1001. Las Vegas, NV, USA, 2016.

[Nicolescu 02] M. N. Nicolescu, M. J. Mataric, "A Hierarchical Architecture for Behavior-Based Robots", in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*, ser. AAMAS '02. Bologna, Italy: Association for Computing Machinery, 2002, pp. 227–233. doi:10.1145/544741.544798, [online] https://doi.org/10.1145/544741.544798.

[Ogura 06a] Y. Ogura, H. Aikawa, K. Shimomura, H. Kondo, A. Morishima, H.-o. Lim, A. Takanishi, "Development of a New Humanoid Robot WABIAN-2", in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* Orlando, FL, USA: IEEE, 2006, pp. 76–81. doi:10.1109/ROBOT.2006.1641164, [online] http://ieeexplore.ieee.org/document/1641164/ Accessed: 2024-06-30.

[Ogura 06b] Y. Ogura, K. Shimomura, H. Kondo, A. Morishima, T. Okubo, S. Momoki, H.-o. Lim, A. Takanishi, "Human-like Walking with Knee Stretched, Heel-Contact and Toe-off Motion by a Humanoid Robot", in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China: IEEE, 2006, pp. 3976–3981. doi:10.1109/IROS.2006.281834, [online] http://ieeexplore.ieee.org/document/4059029/ Accessed: 2024-06-30.

[Ott 16] C. Ott, B. Henze, G. Hettich, T. N. Seyde, M. A. Roa, V. Lippi, T. Mergner, "Good Posture, Good Balance: Comparison of Bioinspired and Model-Based Approaches for Posture Control of Humanoid Robots", *IEEE Robotics Automation Magazine*, vol. 23, no. 1, pp. 22–33, 2016. doi:10.1109/MRA.2015.2507098.

[Paine 14] N. A. Paine, "High-Performance Series Elastic Actuation", Dissertation, The University of Texas, Austin, aug. 2014.

[Paine 15] N. Paine, J. S. Mehling, J. Holley, N. A. Radford, G. Johnson, C.-L. Fok, L. Sentis, "Actuator Control for the NASA-JSC Valkyrie Humanoid Robot: A Decoupled Dynamics Approach for Torque Control of Series Elastic Robots", *Journal of Field Robotics*, vol. 32, no. 3, pp. 378–396, 2015. doi:10.1002/rob.21556, [online] `https://onlinelibrary.wiley.com/doi/10.1002/rob.21556` Accessed: 2021-10-27.

[Perry 10] J. Perry, J. Burnfield, *Gait Analysis: Normal and Pathological Function. 2nd Ed.*, 2 ed., Boca Raton: CRC Press, 2010. [online] `https://doi.org/10.1201/9781003525592`.

[Perry 92] J. Perry, *Gait Analysis: Normal and Pathological Function*, Slack Incorporated, 1992.

[Peterka 02] R. J. Peterka, "Sensorimotor Integration in Human Postural Control", *Journal of Neurophysiology*, vol. 88, no. 3, pp. 1097–1118, may 2002. doi:10.1152/jn.2002.88.3.1097.

[Petri 62] C. A. Petri, "Kommunikation mit Automaten", doctoralthesis, Technische Hochschule Darmstadt, Darmstadt, 1962. [online] `http://edoc.sub.uni-hamburg.de/informatik/volltexte/2011/160/` Accessed: 2025-01-11.

[Plooij 17] M. Plooij, W. Wolfslag, M. Wisse, "Clutched Elastic Actuators", *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 2, pp. 739–750, 2017. doi:10.1109/TMECH.2017.2657224, [online] `http://ieeexplore.ieee.org/document/7829390/` Accessed: 2024-07-10.

[Pollayil 22] M. J. Pollayil, C. D. Santina, G. Mesesan, J. Englsberger, D. Seidel, M. Garabini, C. Ott, A. Bicchi, A. Albu-Schaffer, "Planning Natural Locomotion for Articulated Soft Quadrupeds", in *2022 International Conference on Robotics and Automation (ICRA)*. Philadelphia, PA, USA: IEEE, may 2022, pp. 6593–6599. doi:10.1109/ICRA46639.2022.9812416, [online] `https://ieeexplore.ieee.org/document/9812416/` Accessed: 2024-09-11.

[Poulakakis 09] I. Poulakakis, J. W. Grizzle, "The Spring Loaded Inverted Pendulum as the Hybrid Zero Dynamics of an Asymmetric Hopper", *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1779–1793, 2009. doi:10.1109/TAC.2009.2024565, [online] `http://ieeexplore.ieee.org/document/5175424/` Accessed: 2024-07-26.

[Pratt 06] J. Pratt, J. Carff, S. Drakunov, A. Goswami, "Capture Point: A Step toward Humanoid Push Recovery", in *2006 6th IEEE-RAS International Conference on Humanoid Robots*. Genova, Italy: IEEE, 2006, pp. 200–207. doi:10.1109/ICHR.2006.321385.

[Proetzsch 07] M. Proetzsch, T. Luksch, K. Berns, "The Behaviour-Based Control Architecture iB2C for Complex Robotic Systems", in *Proceedings of the 30th Annual German Conference on Artificial Intelligence (KI)*. sept. 2007, pp. 494–497.

[Proetzsch 10a] M. Proetzsch, *Development Process for Complex Behavior-Based Robot Control Systems*, ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut, 2010.

[Proetzsch 10b] M. Proetzsch, T. Luksch, K. Berns, "Development of Complex Robotic Systems Using the Behavior-Based Control Architecture iB2C", *Robotics and Autonomous Systems*, vol. 58, no. 1, pp. 46–67, 2010. doi:10.1016/j.robot.2009.07.027, [online] `https://www.sciencedirect.com/science/article/pii/S092188900900116X` Accessed: 2023-01-10.

[Radford 15] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, R. Berka, R. Ambrose, M. Myles Markee, N. J. Fraser-Chanpong, C. McQuin, J. D. Yamokoski, S. Hart, R. Guo, A. Parsons, B. Wightman, P. Dinh, B. Ames, C. Blakely, C. Edmondson, B. Sommers, R. Rea, C. Tobler, H. Bibby, B. Howard, L. Niu, A. Lee, M. Conover, L. Truong, R. Reed, D. Chesney, R. Platt, G. Johnson, C.-L. Fok, N. Paine, L. Sentis, E. Cousineau, R. Sinnet, J. Lack, M. Powell, B. Morris, A. Ames, J. Akinyode, "Valkyrie: NASA's First Bipedal Humanoid Robot", *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015. doi:10.1002/rob.21560, [online] `https://onlinelibrary.wiley.com/doi/10.1002/rob.21560` Accessed: 2021-10-26.

[Radkhah 12] K. Radkhah, O. von Stryk, "Human-like Model-based Motion Generation Combining Feedforward and Feedback Control for Musculoskeletal Robots,"", in *Proc. 7th Annual Dynamic Walking Conference*. may 2012, pp. 91–93.

[Reeves 08] N. P. Reeves, J. Cholewicki, T. Milner, A. S. Lee, "Trunk Antagonist Co-Activation Is Associated with Impaired Neuromuscular Performance", *Experimental Brain Research*, vol. 188, no. 3, pp. 457–463, 2008. doi:10.1007/s00221-008-1378-9.

[Reher 16] J. Reher, E. A. Cousineau, A. Hereid, C. M. Hubicki, A. D. Ames, "Realizing Dynamic and Efficient Bipedal Locomotion on the Humanoid Robot DURUS", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden, 2016, pp. 1794–1801. doi:10.1109/ICRA.2016.7487325.

[Reher 19] J. Reher, W.-L. Ma, A. D. Ames, "Dynamic Walking with Compliance on a Cassie Bipedal Robot", in *2019 18th European Control Conference (ECC)*. Napoli, Italy: IEEE, 2019, pp. 2589–2595. doi:10.23919/ECC.2019.8796090, [online] `https://ieeexplore.ieee.org/document/8796090/` Accessed: 2022-04-04.

[Reichardt 12] M. Reichardt, T. Föhst, K. Berns, "Introducing Finroc: A Convenient Real-Time Framework for Robotics Based on a Systematic Design Approach", Robotics Research Lab, Department of Computer Science, University of Kaiserslautern, Technical Report, 2012.

[Reichardt 13] M. Reichardt, T. Föhst, K. Berns, "Design Principles in Robot Control Frameworks.", in *GI-Jahrestagung*, ser. Lecture Notes in Informatics (LNI), M. Horbach, Ed. 2013, pp. 2765–2779.

[Rezazadeh 18] S. Rezazadeh, A. Abate, R. L. Hatton, J. W. Hurst, "Robot Leg Design: A Constructive Framework", *IEEE Access*, vol. 6, pp. 54369–54387, 2018. doi:10.1109/ACCESS.2018.2870291, [online] https://ieeexplore.ieee.org/document/8466587/ Accessed: 2024-06-30.

[Ropertz 17] T. Ropertz, P. Wolf, K. Berns, "Quality-Based Behavior-Based Control for Autonomous Robots in Rough Environments", in *Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics*, vol. 1. SciTePress, 2017, pp. 513–524. doi:10.5220/0006481405130524, [online] https://www.scitepress.org/Link.aspx?doi=10.5220/0006481405130524.

[Ropertz 19] T. Ropertz, *Design and Analysis of Behavior-Based Systems Using Formal Techniques*, ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut, 2019. [online] https://www.dr.hut-verlag.de/978-3-8439-4189-1.html.

[Rose 06] J. Rose, J. G. Gamble, *Human Walking, , 3rd Edition*, 3 ed., Lippincott Williams & Wilkins, 2006.

[Rygg 93] L. A. Rygg, "Mechanical Horse", US Patent Nr. 491 927, feb. 1893.

[Sakagami 02] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, K. Fujimura, "The Intelligent ASIMO: System Overview and Integration", in *IEEE/RSJ International Conference on Intelligent Robots and System*, vol. 3. Lausanne, Switzerland: IEEE, 2002, pp. 2478–2483. doi:10.1109/IRDS.2002.1041641, [online] http://ieeexplore.ieee.org/document/1041641/ Accessed: 2024-10-22.

[Sakanaka 16] T. E. Sakanaka, M. Lakie, R. F. Reynolds, "Sway-dependent Changes in Standing Ankle Stiffness Caused by Muscle Thixotropy", in *The Journal of Physiology*. 2016.

[Samek 19] W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, K.-R. Müller, Eds., *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, ser. Lecture Notes in Computer Science, Springer Cham, 2019, vol. 11700. doi:10.1007/978-3-030-28954-6, [online] http://link.springer.com/10.1007/978-3-030-28954-6 Accessed: 2024-05-18.

[Schaeben 92] H. Schaeben, ""Normal" Orientation Distributions", *Texture, Stress, and Microstructure*, vol. 19, no. 4, pp. 197–202, 1992. doi:10.1155/TSM.19.197, [online] https://onlinelibrary.wiley.com/doi/10.1155/TSM.19.197.

[Scheutz 04] M. Scheutz, V. Andronache, "Architectural Mechanisms for Dynamic Changes of Behavior Selection Strategies in Behavior-Based Systems", *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 6, pp. 2377–2395, 2004. doi:10.1109/TSMCB.2004.837309, [online] http://ieeexplore.ieee.org/document/1356026/ Accessed: 2023-06-02.

[Schneider 10] K. Schneider, "The Synchronous Programming Language Quartz", University of Kaiserslautern, Germany, 2010. [online] https://www.researchgate.net/publication/252064909_The_Synchronous_Programming_Language_Quartz Accessed: 2025-01-28.

[Schneider 17] K. Schneider, J. Brandt, "Quartz: A Synchronous Language for Model-Based Design of Reactive Embedded Systems", in *Handbook of Hardware/Software Codesign*, S. Ha, J. Teich, Eds., Dordrecht: Springer Netherlands, 2017, pp. 1–30. doi:10.1007/978-94-017-7358-4_3-1, [online] `http://link.springer.com/10.1007/978-94-017-7358-4_3-1` Accessed: 2025-01-28.

[Scholz 11] D. Scholz, S. Kurowski, K. Radkhah, O. von Stryk, "Bio-Inspired Motion Control of the Musculoskeletal Biobiped1 Robot Based on a Learned Inverse Dynamics Model", in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference On*. IEEE, 2011, pp. 395–400. doi:10.1109/Humanoids.2011.6100879.

[Schumacher 20] C. Schumacher, M. Sharbafi, A. Seyfarth, C. Rode, "Biarticular Muscles in Light of Template Models, Experiments and Robotics: A Review", *Journal of The Royal Society Interface*, vol. 17, no. 163, p. 20180413, 2020. doi:10.1098/rsif.2018.0413, [online] `https://royalsocietypublishing.org/doi/10.1098/rsif.2018.0413` Accessed: 2020-06-16.

[Schütz 16a] S. Schütz, K. Mianowski, C. Kotting, A. Nejadfard, M. Reichardt, K. Berns, "RRLAB SEA – A Highly Integrated Compliant Actuator with Minimised Reflected Inertia", in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. Banff, AB, Canada: IEEE, 2016, pp. 252–257.

[Schütz 16b] S. Schütz, A. Nejadfard, K. Berns, "Influence of Loads and Design Parameters on the Closed-Loop Performance of Series Elastic Actuators", in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Qingdao, China: IEEE, 2016, pp. 2075–2080. doi:10.1109/ROBIO.2016.7866635.

[Schütz 16c] S. Schütz, A. Nejadfard, C. Kötting, K. Berns, "An Intuitive and Comprehensive Two-Load Model for Series Elastic Actuators", in *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*. Auckland, New Zealand: IEEE, 2016, pp. 573–580.

[Schütz 17] S. Schütz, A. Nejadfard, K. Mianowski, P. Vonwirth, K. Berns, "CARL – A Compliant Robotic Leg Featuring Mono- and Biarticular Actuation", in *IEEE-RAS International Conference on Humanoid Robots*. Birmingham, UK, 2017.

[Schütz 20a] S. Schütz, "CARL - A Compliant Robotic Leg Designed for Human-Like Bipedal Locomotion", Doctoralthesis, University of Kaiserslautern, Kaiserslautern, Germany, 2020. [online] `http://nbn-resolving.de/urn:nbn:de:hbz:386-kluedo-59751`.

[Schütz 20b] S. Schütz, A. Nejadfard, N. Dorosti, K. Berns, "Exploiting the Intrinsic Deformation of a Prosthetic Foot to Estimate the Center of Pressure and Ground Reaction Force", *Bioinspiration & Biomimetics*, vol. 15, no. 5, p. 056002, 2020. doi:10.1088/1748-3190/ab96d7, [online] `https://dx.doi.org/10.1088/1748-3190/ab96d7`.

[Schwenzer 21] M. Schwenzer, M. Ay, T. Bergs, D. Abel, "Review on Model Predictive Control: An Engineering Perspective", *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5, pp. 1327–1349, 2021. doi:10.1007/s00170-021-07682-3, [online] https://link.springer.com/10.1007/s00170-021-07682-3 Accessed: 2024-04-26.

[Seely 70] S. Seely, N. H. Tarnoff, D. Holstein, *Digital Computers in Engineering*, New York: Holt, Rinehart and Winston, 1970. [online] https://archive.org/details/digitalcomputers0000seel/. dIgitalized at archive.org.

[Semini 11] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, D. G. Caldwell, "Design of HyQ – a Hydraulically and Electrically Actuated Quadruped Robot", *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, no. 6, pp. 831–849, 2011. doi:10.1177/0959651811402275, [online] http://journals.sagepub.com/doi/10.1177/0959651811402275 Accessed: 2022-05-23.

[Semini 15] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, M. Focchi, D. G. Caldwell, J. Buchli, "Towards Versatile Legged Robots through Active Impedance Control", *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1003–1020, 2015. doi:10.1177/0278364915578839, [online] http://journals.sagepub.com/doi/10.1177/0278364915578839 Accessed: 2022-03-07.

[Semini 16] C. Semini, V. Barasuol, T. Boaventura, M. Frigerio, J. Buchli, "Is Active Impedance the Key to a Breakthrough for Legged Robots?", in *Robotics Research*, M. Inaba, P. Corke, Eds., Cham: Springer International Publishing, 2016, vol. 114, pp. 3–19. [online] http://link.springer.com/10.1007/978-3-319-28872-7_1 Accessed: 2022-03-03.

[Sengupta 14] B. Sengupta, M. B. Stemmler, "Power Consumption During Neuronal Computation", *Proceedings of the IEEE*, vol. 102, no. 5, pp. 738–750, 2014. doi:10.1109/JPROC.2014.2307755, [online] http://ieeexplore.ieee.org/document/6797884/ Accessed: 2024-05-18.

[Seok 12] S. Seok, A. Wang, D. Otten, S. Kim, "Actuator Design for High Force Proprioceptive Control in Fast Legged Locomotion", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Vilamoura-Algarve, Portugal: IEEE, 2012, pp. 1970–1975. doi:10.1109/IROS.2012.6386252, [online] http://ieeexplore.ieee.org/document/6386252/ Accessed: 2022-03-07.

[Seok 13] S. Seok, A. Wang, Chuah, Meng Yee, D. Otten, J. Lang, S. Kim, "Design Principles for Highly Efficient Quadrupeds and Implementation on the MIT Cheetah Robot", in *2013 IEEE International Conference on Robotics and Automation*. Karlsruhe, Germany: IEEE, 2013, pp. 3307–3312. doi:10.1109/ICRA.2013.6631038, [online] http://ieeexplore.ieee.org/document/6631038/ Accessed: 2022-03-08.

[Seok 15] S. Seok, A. Wang, M. Y. Chuah, D. J. Hyun, J. Lee, D. M. Otten, J. H. Lang, S. Kim, "Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot", *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 3, pp. 1117–1129, 2015. doi:10.1109/TMECH.2014.2339013, [online] https://ieeexplore.ieee.org/document/6880316/ Accessed: 2022-03-10.

[Sharbafi 13] M. A. Sharbafi, C. Maufroy, M. N. Ahmadabadi, M. J. Yazdanpanah, A. Seyfarth, "Robust Hopping Based on Virtual Pendulum Posture Control", *Bioinspiration & Biomimetics*, vol. 8, no. 3, p. 036002, 2013. doi:10.1088/1748-3182/8/3/036002, [online] http://stacks.iop.org/1748-3190/8/i=3/a=036002?key=crossref.3b03ac4af1a871746f8ba449728dfb26 Accessed: 2019-12-18.

[Sharbafi 16] M. A. Sharbafi, C. Rode, S. Kurowski, D. Scholz, R. Möckel, K. Radkhah, G. Zhao, A. M. Rashty, O. von Stryk, A. Seyfarth, "A New Biarticular Actuator Design Facilitates Control of Leg Function in BioBiped3", *Bioinspiration & Biomimetics*, vol. 11, no. 4, p. 046003, 2016. doi:10.1088/1748-3190/11/4/046003, [online] https://iopscience.iop.org/article/10.1088/1748-3190/11/4/046003.

[Sharbafi 17a] M. A. Sharbafi, A. Seyfarth, Eds., *Bioinspired Legged Locomotion: Models, Concepts, Control and Applications*, Oxford: Elsevier, Butterworth-Heinemann, 2017. doi:10.1016/C2014-0-04899-X, [online] https://www.sciencedirect.com/book/9780128037669/bioinspired-legged-locomotion.

[Sharbafi 17b] M. A. Sharbafi, A. M. N. Rashty, C. Rode, A. Seyfarth, "Reconstruction of Human Swing Leg Motion with Passive Biarticular Muscle Models", *Human Movement Science*, vol. 52, pp. 96–107, 2017. doi:10.1016/j.humov.2017.01.008, [online] https://www.sciencedirect.com/science/article/abs/pii/S0167945717300313.

[Shen 15] TK. Shen, PH. Lin, PC. Lin, IF. Lee, CI. Lin, WP. Shih, "Implementation of a PID Controller for a Robotic Leg Actuated by Pneumatic Artificial Muscles", in *Proceedings of the 14th IFToMM World Congress*. oct. 2015, pp. 459–464. doi:10.6567/IFToMM.14TH.WC.OS13.086.

[Shi 21] F. Shi, T. Homberger, J. Lee, T. Miki, M. Zhao, F. Farshidian, K. Okada, M. Inaba, M. Hutter, "Circus ANYmal: A Quadruped Learning Dexterous Manipulation with Its Limbs", in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE, 2021, pp. 2316–2323. doi:10.1109/ICRA48506.2021.9561926, [online] https://ieeexplore.ieee.org/document/9561926/ Accessed: 2022-10-28.

[Shimizu 12] M. Shimizu, K. Suzuki, K. Narioka, K. Hosoda, "Roll Motion Control by Stretch Reflex in a Continuously Jumping Musculoskeletal Biped Robot", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct. 2012, pp. 1264–1269.

[Siciliano 16] B. Siciliano, O. Khatib, Eds., *Springer Handbook of Robotics*, 2 ed., ser. Springer Handbooks, Cham: Springer International Publishing, 2016. [online] https://link.springer.com/10.1007/978-3-319-32552-1 Accessed: 2023-05-24. https://doi.org/10.1007/978-3-319-32552-1.

[Silva 24] A. B. Silva, M. Murcia, O. Mohseni, R. Takahashi, A. Forner-Cordero, A. Seyfarth, K. Hosoda, M. A. Sharbafi, "Design of Low-Cost Modular Bio-Inspired Electric–Pneumatic Actuator (EPA)-Driven Legged Robots", *Biomimetics*, vol. 9, no. 3, p. 164, 2024. doi:10.3390/biomimetics9030164, [online] https://www.mdpi.com/2313-7673/9/3/164 Accessed: 2024-03-11.

[Soch 22] J. Soch, T. J. Faulkenberry, K. Petrykowski, C. Allefeld, "The Book of Statistical Proofs", https://statproofbook.github.io/, oct. 2022. doi:10.5281/ZENODO.4305950, [online] https://statproofbook.github.io/ Accessed: 2023-06-05.

[Song 15] S. Song, H. Geyer, "A Neural Circuitry That Emphasizes Spinal Feedback Generates Diverse Behaviours of Human Locomotion", *The Journal of Physiology*, vol. 593, no. 16, pp. 3493–3511, 2015. doi:10.1113/JP270228, [online] http://dx.doi.org/10.1113/JP270228.

[Song 16] H. Song, H. Park, S. Park, "A Springy Pendulum Could Describe the Swing Leg Kinetics of Human Walking", *Journal of Biomechanics*, vol. 49, no. 9, pp. 1504–1509, 2016.

[Sreenath 11] K. Sreenath, H.-W. Park, I. Poulakakis, J. W. Grizzle, "A Compliant Hybrid Zero Dynamics Controller for Stable, Efficient and Fast Bipedal Walking on MABEL", *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[Stephens 07] B. Stephens, "Humanoid Push Recovery", in *2007 7th IEEE-RAS International Conference on Humanoid Robots*. nov. 2007. doi:10.1109/ICHR.2007.4813931.

[Tamaddoni 10] S. H. Tamaddoni, F. Jafari, A. Meghdari, S. Sohrabpour, "BIPED HOPPING CONTROL BAzSED ON SPRING LOADED INVERTED PENDULUM MODEL", *International Journal of Humanoid Robotics*, vol. 07, no. 02, pp. 263–280, june 2010. doi:10.1142/S0219843610002106, [online] https://www.worldscientific.com/doi/abs/10.1142/S0219843610002106 Accessed: 2019-12-18.

[Terlemez 14] O. Terlemez, S. Ulbrich, C. Mandery, M. Do, N. Vahrenkamp, T. Asfour, "Master Motor Map (MMM) — Framework and Toolkit for Capturing, Representing, and Reproducing Human Motion on Humanoid Robots", in *2014 IEEE-RAS International Conference on Humanoid Robots*. Madrid: IEEE, nov. 2014, pp. 894–901. doi:10.1109/HUMANOIDS.2014.7041470, [online] https://ieeexplore.ieee.org/document/7041470/ Accessed: 2021-03-18.

[Thach 14] W. T. Thach, "Does the Cerebellum Initiate Movement?", *The Cerebellum*, vol. 13, pp. 139–150, 2014. doi:10.1007/s12311-013-0506-7, [online] http://link.springer.com/10.1007/s12311-013-0506-7 Accessed: 2024-01-05.

[Torricelli 16] D. Torricelli, J. Gonzalez, M. Weckx, R. Jiménez-Fabián, B. Vanderborght, M. Sartori, S. Dosen, D. Farina, D. Lefeber, J. L. Pons, "Human-like Compliant Locomotion: State of the Art of Robotic Implementations", *Bioinspiration & Biomimetics*, vol. 11, no. 5, p. 051002, 2016. doi:10.1088/1748-3190/11/5/051002, [online] http://stacks.iop.org/1748-3190/11/i=5/a=051002.

[Umlauft 20] J. Umlauft, S. Hirche, "Learning Stochastically Stable Gaussian Process State–Space Models", *IFAC Journal of Systems and Control*, vol. 12, no. 100079, 2020. doi:10.1016/j.ifacsc.2020.100079, [online] https://linkinghub.elsevier.com/retrieve/pii/S2468601820300055 Accessed: 2025-02-07.

[Vadakkepat 04] P. Vadakkepat, O. C. Miin, X. Peng, T. H. Lee, "Fuzzy Behavior-Based Control of Mobile Robots", *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 559–565, 2004. doi:10.1109/TFUZZ.2004.832536, [online] http://ieeexplore.ieee.org/document/1321084/ Accessed: 2023-06-26.

[Vukobratović 04] M. Vukobratović, B. Borovac, "Zero-Moment Point — Thirty Five Years of Its Life", *International Journal of Humanoid Robotics*, vol. 01, no. 01, pp. 157–173, 2004. doi:10.1142/S0219843604000083, [online] https://www.worldscientific.com/doi/abs/10.1142/S0219843604000083 Accessed: 2021-12-02.

[Vukobratovic 69] M. Vukobratovic, D. Juricic, "Contribution to the Synthesis of Biped Gait", *IEEE Transactions on Biomedical Engineering*, vol. BME-16, no. 1, pp. 1–6, 1969. doi:10.1109/TBME.1969.4502596, [online] http://ieeexplore.ieee.org/document/4502596/ Accessed: 2024-08-03.

[Westervelt 07] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion*, CRC press, 2007, vol. 28.

[White 20] J. White, D. Swart, C. Hubicki, "Force-Based Control of Bipedal Balancing on Dynamic Terrain with the "Tallahassee Cassie" Robotic Platform", in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, 2020, pp. 6618–6624. doi:10.1109/ICRA40945.2020.9196725, [online] https://ieeexplore.ieee.org/document/9196725/ Accessed: 2021-11-12.

[Winter 09] D. A. Winter, *Biomechanics and Motor Control of Human Movement*, 4th Edition ed., Hoboken, NJ, USA and simultaneously in Canada: John Wiley & Sons, 2009.

[Winter 98] D. A. Winter, A. E. Patla, F. Prince, M. Ishac, K. Gielo-Perczak, "Stiffness Control of Balance in Quiet Standing", *Journal of neurophysiology*, vol. 80, pp. 1211–1221, oct. 1998. doi:10.1152/jn.1998.80.3.1211.

[Wolf 11] S. Wolf, O. Eiberger, G. Hirzinger, "The DLR FSJ: Energy Based Design of a Variable Stiffness Joint", in *2011 IEEE International Conference on Robotics and Automation*. Shanghai, China: IEEE, 2011, pp. 5082–5089. doi:10.1109/ICRA.2011.5980303, [online] http://ieeexplore.ieee.org/document/5980303/ Accessed: 2024-09-28.

[Wolf 22] P. Wolf, *Cognitive Processing in Behavior-Based Perception of Autonomous Off-Road Vehicles*, ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut GmbH, 2022. [online] https://www.dr.hut-verlag.de/978-3-8439-5165-4.html.

[Xie 19] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, M. van de Panne, "Iterative Reinforcement Learning Based Design of Dynamic Locomotion Skills for Cassie", 2019. doi:10.48550/arXiv.1903.09537, [online] http://arxiv.org/abs/1903.09537 Accessed: 2024-01-05.

[Yang 08] T. Yang, E. R. Westervelt, J. P. Schmiedeler, R. A. Bockbrader, "Design and Control of a Planar Bipedal Robot ERNIE with Parallel Knee Compliance", *Autonomous Robots*, vol. 25, no. 4, pp. 317–330, 2008. doi:10.1007/s10514-008-9096-5, [online] http://link.springer.com/10.1007/s10514-008-9096-5 Accessed: 2024-07-09.

[Yazdani 18] M. Yazdani, H. Salarieh, M. S. Foumani, "Bio-Inspired Decentralized Architecture for Walking of a 5-Link Biped Robot with Compliant Knee Joints", *International Journal of Control, Automation and Systems*, vol. 16, no. 6, pp. 2935–2947, 2018. doi:10.1007/s12555-017-0578-0, [online] http://link.springer.com/10.1007/s12555-017-0578-0 Accessed: 2024-01-10.

[Ye 98] H. Ye, A. N. Michel, L. Hou, "Stability Theory for Hybrid Dynamical Systems", *IEEE Transactions on Automatic Control*, vol. 43, no. 4, pp. 461–474, 1998. doi:10.1109/9.664149, [online] http://ieeexplore.ieee.org/document/664149/ Accessed: 2025-06-05.

[Yoon 11] J. Yoon, G. Kim, N. Handharu, A. Özer, "A Bio-Robotic Toe & Foot & Heel Model of a Biped Robot for More Natural Walking: Foot Mechanism & Gait Pattern", in *Biped Robots*, IntechOpen, 2011, p. 334. doi:10.5772/578, [online] https://www.intechopen.com/books/46.

[Zhao 13] J. Zhao, S. Schütz, K. Berns, "Biologically Motivated Push Recovery Strategies for a 3D Bipedal Robot Walking in Complex Environments", in *IEEE - International Conference on Robotics and Biomimetics (Robio 2013)*. Shenzhen, China, 2013.

[Zhao 14] J. Zhao, Q. Liu, S. Schütz, K. Berns, "Experimental Verification of an Approach for Disturbance Estimation and Compensation on a Simulated Biped during Perturbed Stance", in *IEEE International Conference on Robotics and Automation (ICRA 2014)*. Hongkong, China, 2014.

[Zhao 15] J. Zhao, Q. Liu, S. Schütz, K. Berns, "A Bio-inspired Behavior Based Bipedal Locomotion Control – B4LC Method for Bipedal Upslope Walking", in *18th International Conference on Climbing and Walking Robots (CLAWAR)*. Hangzhou, China, 2015.

[Zhao 16] J. Zhao, *Advanced Walking Skills for Bipedal Locomotion*, 1 ed., ser. RRLab Dissertations, Munich, Germany: Verlag Dr. Hut, 2016.

[Zhao 19] Y. Zhao, L. Sentis, "Chapter 2 – Distributed Impedance Control of Latency-Prone Robotic Systems with Series Elastic Actuation", in *Stability, Control and Application of Time-delay Systems*, Butterworth-Heinemann, 2019, pp. 23–51. doi:10.1016/B978-0-12-814928-7.00002-0, [online] https://www.sciencedirect.com/science/article/pii/B9780128149287000020 Accessed: 2024-09-25.

Nomenclature

Abbreviations and Names

A_A	Monoarticular ankle actuator, implementing the natural human leg muscles Sol and TA.
AC	Alternating Current
AFD	Actuator Force Distribution
A_H	Monoarticular hip actuator, implementing the natural human leg muscles GMax and IL.
A_{HK}	Biarticular hip-knee actuator, implementing the natural human leg muscles Ham and RF.
AI	Artificial Intelligence
A_K	Monoarticular knee actuator, implementing the natural human leg muscles BFsh and Vas.
A_{KA}	Biarticular knee-ankle actuator, implementing the natural human leg muscle Gas.
ANYmal	ANYmal is a previously scientific, yet commercial quadruped robot by the company *ANYbotics*. URL: `https://www.anybotics.com` (2025) Literature: [Hutter 16, Hutter 17, Fankhauser 18, Shi 21]
AS	Actuation Space
ASIMO	ASIMO (Advanced Step in Innovative Mobility) is a child-size, full-featured humanoid robot designed and built by the company *Honda* in 2000 and 2005 (version 2). Literature: [Sakagami 02]
ATRIAS	ATRIAS is a compliant bipedal robotic research platform at the *Oregon State University* to exploit Spring Loaded Inverted Pendulum (SLIP)-based walking. Literature: [Hubicki 16, Rezazadeh 18]
B4LC	B4LC is a scientific humanoid robot simulation by the RPTU developed to study compliant, natural dynamics-exploiting gait. Literature: [Luksch 10a, Zhao 16]

BB	Behavior-Based
Bert	Bert is a compliant quadruped robotic research platform developed by the *German Aerospace Center*. Literature: [Lakatos 18, Pollayil 22]
BFsh	Biceps Femoris short-head (monoarticular posterior knee muscle)
BioBiped3	BioBiped3 is a planar scientific lower body constructed at the *Technical University of Darmstadt* featuring series elastic actuation. Literature: [Sharbafi 16, Sharbafi 17b]
CAD	Computer Aided Design (usually a digital model of a mechanical design)
Carl	Carl (Compliant Robotic Leg) is a bio-inspired planar lower body research platform at the RPTU. URL: `https://rrlab.cs.rptu.de/en/robots/carl` (2025) Literature: [Schütz 17, Schütz 20a]
Cassie	Cassie is the bipedal predecessor robot of Digit, both produced by the company *Agility Robotics*. URL: `https://agilityrobotics.com/` (2025)
CC BY 3.0	Creative Commons Attribution License 3.0 URL: `https://creativecommons.org/licenses/by/3.0` (2025)
CC BY 4.0	Creative Commons Attribution License 4.0 URL: `https://creativecommons.org/licenses/by/4.0` (2025)
CGA	Conformal Geometric Algebra
CI/CD	Continuous Integration & Continuous Delivery (sometimes also called *Continuous Deployment*)
Close-To-Singular	Close-to-singular describes a configuration that is not yet entirely singular but within a narrow region around a singularity that may already cause numerical instability when digitally processed with finite-resolution representations, such as floating-point numbers.
CNS	Central Nervous System
CoM	Center of Mass
CoP	Center of Pressure
CPG	Central Pattern Generator
CPU	Central Processing Unit
CS	Configuration Space
DC	Direct Current
DCM	Divergent Component of Motion
DH	Denavit-Hartenberg (convention of a minimal set of kinematic chain description parameters [Denavit 55])

Digit	Digit is a commercial humanoid developed and produced by the company *Agility Robotics*. URL: `https://agilityrobotics.com` (2025)
DLS	Damped Least Square
DoF	Degree of Freedom
DURUS	DURUS is a bipedal robot developed and designed at the *Caltech California Institute of Technology*. Literature: [Hereid 16, Reher 16]
EEG	Electroencephalography
EMG	Electromyography
EPA-Hopper II	EPA-Hopper II is a planar, 3-Degree of Freedom (DoF) robotic research leg designed at the *Technical University of Darmstadt*. Literature: [Mohseni 22, Silva 24]
ERNIE	ERNIE is a bipedal robot prototype designed and housed at the *Ohio State University*. Literature: [Yang 08]
ETH	Ethernet
Finroc	Finroc is a modular and concurrent robot control framework developed at the RPTU. URL: `https://www.finroc.org/` (2025) Literature: [Reichardt 12, Reichardt 13]
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array (a programmable digital logic IC)
FSM	Finite State Machine
GA	Geometric Algebra
Gas	Gastrocnemius (biarticular posterior shank muscle)
GMax	Gluteus Maximus (monoarticular posterior hip muscle)
GPR	Gaussian Process Regression
GRF	Ground Reaction Force
Ham	Hamstrings Group (biarticular posterior thigh muscle)
HyQ	HyQ is a hydraulic-electric actuated quadruped robot by the *Italian Institute of Technology*. Literature: [Semini 11, Semini 15, Semini 16, Aceituno-Cabezas 18]
HZD	Hybrid Zero Dynamics
iB2C	Integrated Behavior-Based Control

IC Integrated Circuit

IIR Infinite Impulse Response

IL Iliacus Psoas (monoarticular anterior hip muscle)

Interneuron In contrast to *sensor neurons* and *motor neurons*, *interneurons* only connect between other neurons. Thus, *interneurons* implement pure processing purposes.
Literature: [Moini 20]

IP Inverted Pendulum

LDL Square Root Free Cholesky Decomposition

Lola Lola is a scientific humanoid robot developed and built at the *Technical University of Munich*.
Literature: [Lohmeier 09, Buschmann 09]

LTS Linear Time-Invariant System

Lyapunov Stability: If a dynamic system that is (infinitesimal) close to one of its equilibrium states forever stays that close, it is called *Lyapunov stable*
Literature: [Lyapunov 92]

MIMO Multiple-Input Multiple-Output

MIT Cheetah MIT Cheetah is a lightweight, four-legged robotic research platform developed at the *Massachusetts Institute of Technology*.
Literature: [Seok 12, Seok 13, Seok 15]

ML Machine Learning

Motor Neuron *Motor neurons* carry information from the Central Nervous System (CNS) toward peripheral effectors, such as muscles.
Literature: [Moini 20]

MPC Model Predictive Control

Nadia Nadia is a humanoid commercially developed and produced by the company *Boardwalk Robotics*.
URL: `https://boardwalkrobotics.com` (2025)

NN Neural Network

OC Optimal Control

ODE Ordinary Differential Equation

OS Operating System

PAM Pneumatic Artificial Muscle

PBN Probabilistic Behavior Network

PDF Probability Density Function

PGA	Projective Geometric Algebra
PID	Proportional-Integral-Derivative (control)
Pneumat-BH	Pneumat-BH is a scientific, pneumatically-powered, human-like lower body built at the *Osaka University*. Literature: [Shimizu 12, Kawakami 15]
PS	Projection Space
RF	Rectus Femoris (biarticular anterior thigh muscle)
RK	Runge-Kutta
RL	Reinforcement Learning
RoM	Range of Motion
RPTU	Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau URL: `https://rptu.de` (2025)
RRLab	Robotics Research Lab, Departement of Computer Science, RPTU URL: `https://rrlab.cs.rptu.de` (2025)
SEA	Series Elastic Actuator
Sensor Neuron	*Sensor neurons* transport information from peripheral receptors toward the CNS. Literature: [Moini 20]
SISO	Single-Input Single-Output
SLIP	Spring Loaded Inverted Pendulum
SoC	System on Chip
Sol	Soleus (monoarticular posterior ankle muscle)
SPG	Spinal Pattern Generator (a Finite State Machine)
Spot	Spot is a commercial quadrupedal robotic platform offered by the company *Boston Dynamics*. URL: `https://bostondynamics.com/` (2025)
STO	Safe Torque Off
Strandbeests	Strandbeests by the artist Theo Janson are a mechanical art engaged to create a *"new form of life"*. URL: `https://www.strandbeest.com` (2025)
SVD	Singular Value Decomposition
TA	Tibialis Anterior (monoarticular anterior ankle muscle)
TCP	Tool Center Point
TORO	TORO is a torque-driven humanoid robot developed and operated at the *German Aerospace Center*. Literature: [Englsberger 13, Englsberger 14]

TS	Transformation Space
Twist	Originating from *screw theory*, a twist vector is interpreted as a combined linear and rotational velocity throughout this thesis. Consequently, twist control combines linear and angular velocity control. Literature: [Ball 76]
UDP/IP	User Datagram Protocol via Internet Protocol
Valkyrie	Valkyrie is a humanoid research robot developed and build by the *NASA's Johnson Space Center*. URL: `https://www.nasa.gov/technology/r5` (2025) Literature: [Paine 15, Radford 15]
Vas	Vastus Group (monoarticular anterior knee muscle)
VPP	Virtual Pivot Point
Wrench	Originating from *screw theory*, a wrench vector is interpreted as a combined force and torque vector throughout this thesis. Consequently, wrench control combines force and torque control. Literature: [Ball 76]
WS	Work Space
XCoM	Extrapolated Center of Mass
ZMP	Zero Moment Point

Symbols and Functions

$\mathcal{BN}(\cdot, \cdot)$	*Bingham Distribution*: The Bingham distribution of a continuous Cartesian random unit vector variable, e.g., a rotor
$(\cdot \otimes \cdot)$	*Kronecker Product*: The Kronecker product applied on vector and/or matrix arguments
$\mathrm{col}(\cdot)$	*Column*: Extracts a matrix argument's i-th column as indexed $\mathrm{col}_i(\cdot)$
$\cos(\cdot)$	*Cosine*: The regular trigonometric cosine function
$\mathcal{X}$	Set of probabilistic data vector representations used within a Probabilistic Behavior Network (PBN), following Definition 4.4
$\det(\cdot)$	*Determinant*: The regular determinant function on square matrices
$\mathrm{diag}(\cdot)$	*Diagonal*: Returns either the vector of the diagonal elements of a matrix argument or a diagonal matrix with a vector argument as its diagonal
$\dim(\cdot)$	*Dimensions*: Returns the effective dimensionality of the argument
$\exp(\cdot)$	*Exponential*: The exponential function applied on scalar values or matrices
$\mathbb{E}[\cdot]$	*Expected Value*: The expected value of a stochastic random variable, i.e., its mean
$\mathrm{expm1}(\cdot)$	*Exponential Minus One*: $\mathrm{expm1}(x) := \exp(x) - 1$ – Recommended for high-resolution floating-point results around zero
$\mathbf{G}$	Symmetric, positive semidefinite responsiveness matrix. Following the definition of Seely et al., responsiveness represents the mechanical equivalent of electrical conductance, hence being to the real part of mechanical admittance or the inverse of mechanical damping [Seely 70, p. 200].
$\mathcal{G}$	A directed, acyclic graph describing the data flow computer architecture of a PBN, given in Definition 4.3
$\textsc{id}(\cdot)$	*Identity*: The general identity function mapping any arbitrary input to itself
$\mathbf{J}_{\mathbf{CA}}$	Lever arm (or gear) Jacobian matrix of all actuators w.r.t their spanned joints
$\ln(\cdot)$	*Natural Logarithm*: The logarithm w.r.t.the natural base e
$\mathrm{lnp1}(\cdot)$	*Natural Logarithm Plus One*: $\mathrm{lnp1}(x) := \ln(x + 1)$ – Recommended for high-resolution floating-point results around one
$\mathcal{MN}(\cdot, \cdot, \cdot)$	*Matrix Normal Distribution*: The matrix normal distribution of a continuous Cartesian random matrix variable

$\max\{\cdot\}$	*Maximum*: The maximum element of the given set
$\min\{\cdot\}$	*Minimum*: The minimum element of the given set
$\mathcal{N}(\cdot,\ \cdot)$	*Normal Distribution*: The normal distribution or Gaussian distribution of a continuous Cartesian random vector variable
$\#\text{DoF}(\cdot)$	*Number of DoF*: Returns the effective number of existing Degrees of Freedom of the argument space, according to its indexed property
$\mathbb{E}(\cdot)$	*Probabilistic Annotation*: Some arbitrary function converting a dedicated signal into a probabilistic one featuring expected value and variance, introduced in Section 4.5.1
$\mathcal{B}(\cdot)$	*Behavior*: A PBN behavior, given by its computational function $\mathcal{B}$ as stated in the behavioral calculation Definition 4.6
$\Sigma(\cdot)$	*Composition*: The unique PBN data composition function, according to Definitions 4.16 and 4.17
$\int(\cdot)$	*Probabilistic Integration*: The probabilistic PBN integration to generate continuous precise values out of probabilistic first derivative data, as specified in Definition 4.15
$\mathcal{P}(\cdot)$	*Improper Projection*: An improper, i.e., information-lossy, PBN data projection function, as introduced in Section 4.7.1
$\mathcal{T}(\cdot)$	*Proper Transformation*: A proper, i.e., information-preserving, PBN data transformation function, as introduced in Section 4.7.2
$\langle\cdot,\ \cdot\rangle$	*Probabilistic Data*: Special notation for representing a normal distribution
$\text{SO}(\cdot)$	*Special Orthogonal Group*, also called *Rotation Group*: The group of all (n-dimensional) Cartesian rotations, herein used independent of their concrete representation as, e.g., *Euler* angles, a rotation matrix, or a Projective Geometric Algebra (PGA) rotor
$\text{L}[\cdot]$	*Rotor Left Multiplication Matrix*: The matrix that describes the left multiplication operation in the respective rotor algebra – not a rotation matrix
$\text{R}[\cdot]$	*Rotor Right Multiplication Matrix*: The matrix that describes the right multiplication operation in the respective rotor algebra – not a rotation matrix
$\text{sat}(\cdot)$	*Satisfaction*: The satisfaction of a probabilistic data with an external mean value
$\sin(\cdot)$	*Sine*: The regular trigonometric sine function
$\text{tr}(\cdot)$	*Trace*: Return the trace of a matrix argument
$\text{Var}(\cdot)$	*Variance*: The expected value of the squared distance to a stochastic random variable's expected value: $\text{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$
$\text{vec}(\cdot)$	*Vectorized*: Returns a matrix argument's vectorization, i.e., its stacked columns as a single column vector

Index

A

Actuation Space
(AS), 48, 84, 85, 89, 105, 176, 219, 251
Actuator Force Distribution
(AFD), 90, 182, 251
Acyclic, 120, 146, 257
Admittance, 25, 53, 66
Alternating Current
(AC), 77, 251
Ankle Strategy, 35, 171, 189, 192, 203, 218, 221
Antagonist, 11, 46, 52, 61, 64, 84, 89, 104, 109, 111, 175, 176, 194, 203, 216, 218, 220, 222, 223, k
ANYmal, 41, 54, 251
Approximation, 90, 105, 178
Architecture, 2, 10, 11, 16, 48, 55, 65, 111, 114, 116, 120, 143, 153, 155, 165, 171, 173, 175, 176, 195, 203, 214, 216, 218, 221, 222, 257, k
Subsumption, 116
Artificial Intelligence
(AI), 41, 111, 112, 116, 125, 251, k
Artificial Muscle, 54
ASIMO, 53, 251
Associativity, 123, 145, 157, 173, 217, k
ATRIAS, 44, 251
Axis of Rotation, 131

B

B4LC, 18, 44, 53, 69, 220, 251
Back-Drivable, 51, 53, 72
Backpropagation, 125
Ball Screw, 57, 202
Behavior, 18, 114, 116, 120, 135, 150, 155, 162, 170, 178, 190, 203, 217, 218, 221, 258
Behavior Flexibility, 117, 120
Behavior-Based
(BB), 10, 11, 49, 112, 116, 117, 119, 142, 155, 173, 202, 217, 221, 223, 252, k
Bert, 41, 72, 252
Biarticular, 46, 60, 86, 92, 108, 201, 212
Biarticular hip-knee actuator
(A_{HK}), 62, 95, 108, 251
Biarticular knee-ankle actuator
(A_{KA}), 62, 251
Biceps Femoris short-head
(BFsh), 62, 251, 252
Bingham Distribution, 122, 129, 159, 257
Bio-Inspiration, 1, 2, 5, 10, 50, 114, 216
Bio-Inspired, 1, 2, 6, 10, 13, 16, 37, 41, 48, 50, 60, 65, 84, 109, 111, 114, 175, 186, 189, 203, 215, 216, 220, 222, k
Control, 6
Engineering, 2, 6, 10, 16, 37, 46, 48, 72, 111, 216
Robotic, 3, 39, 222
BioBiped3, 32, 46, 217, 252
Biomimetics, 2
Bionics, 2
Bode Plot, 80, 104
Brushless DC Motor, 57, 77, 218

C

Capture Point, 42
Carl, 3, 11, 40, 48, 52, 55, 65, 84, 89, 104, 109, 133, 170, 173, 175, 203, 214–216, 218,

219, 223, 252, k
Cartesian, 84, 105, 118, 122, 128, 149, 155, 167, 177, 217, 222, 257
Cassie, 44, 217, 252
Center of Mass
(CoM), 35, 41, 107, 176, 189, 204, 252
Center of Pressure
(CoP), 35, 36, 189, 206, 252
Central Nervous System
(CNS), 2, 4, 5, 9, 13, 29, 33, 70, 111, 115, 216, 221, 252, 254
Central Pattern Generator
(CPG), 41, 252
Central Processing Unit
(CPU), 5, 31, 64, 87, 116, 122, 185, 218, 252, k
Centralized, 8, 33, 41, 63, 84, 112, 116, 189, 218, 221
Cerebellum, 29
Cerebrum, 29, 124
Certainty, 122
Characteristic, 6, 10, 13, 16, 38, 40, 48, 50, 51, 54, 72, 109, 111, 114, 124, 130, 149, 175, 181, 203, 215, 216, 219, k, m
Hypothetical, 18, 39
Measurable, 18, 39, 181
Visible, 18, 39
Chronophotography, 13
Close-To-Singular, 23, 252
Closed-Loop, 22, 48, 69, 128, 146, 185, 198
Co-Contraction, 10, 11, 32, 46, 72
Commutativity, 123, 145, 153, 157, 168, 173, 217, k
Complex Number, 84, 129
Dual Complex Number, 139
Composition, 114–116, 120, 155, 206, 217, 258
Computer Aided Design
(CAD), 57, 68, 194, 252
Concentration Matrix, 129
Conclusion, 19, 50, 89, 103, 109, 166, 173, 189, 214, 215, 219, 223
Configuration Space
(CS), 43, 84, 85, 89, 104, 163, 171, 176, 191, 204, 219, 252
Conformal Geometric Algebra
(CGA), 177, 222, 252
Continuous Integration & Continuous Delivery
(CI/CD), 112, 252
Contractile Element, 66
Contribution, 2, 3, 10, 19, 52, 84, 92, 109, 216, 218, 222, 225, o
Control
Responsiveness Control, 9, 181, 193
Admittance Control, 53, 66
Antagonistic Muscular Control, 11, 52, 64, 84, 89, 104, 109, 111, 175, 176, 194, 203, 216, 218, 220, 222, 223, k
Centralized Control, 33, 63
Compliant Pose Control, 185, 195, 205, 207
Current Control, 49, 57, 64, 182, 218
Damping Control, 9, 82, 90, 176, 202
Distributed Control, 33, 65, 189
Force Control, 8, 9, 53, 54, 65, 66, 71, 90, 256
Hybrid Control, 63
Hybrid Impedance Control, 2, 9, 11, 52, 74, 84, 90, 111, 116, 128, 175, 215, 216, 221, n
Impedance Control, 8, 11, 45, 52, 53, 63, 66, 67, 89, 109, 187, 216, 222, 223, k
Muscular Reflex Control, 47, 65, 220
Muscular Rest Length Control, 47, 65
Natural Control, 2, 3, 5, 10, 16, 37, 40, 48, 50, 51, 56, 70, 84, 109, 111, 114, 124, 128, 162, 173, 175, 187, 189, 203, 214–216, 219, 222, k
Neuromuscular Control, 5, 10
Path Control, 8, 49, 53, 66, 217
Pose Control, 8, 53, 70, 171, 197, 206, 217, 218
Position Control, 170, 171
Probabilistic Control, 12
Probabilistic Model Predictive Control, 183
Torque Control, 8, 53, 55, 64, 90, 182, 256
Twist Control, 8, 105, 177, 256

Velocity Control, 8, 78, 181, 193, 205, 256
Wrench Control, 8, 53, 70, 105, 116, 256
Creative Commons Attribution License
(CC BY 3.0), 17, 43, 47, 252
(CC BY 4.0), 45, 47, 252
Critical Damping, 105

D

Damped Least Square
(DLS), 94, 253
Damping, 9, 64, 85, 89, 90, 104, 134, 177, 202
Data Flow, 11, 48, 118, 120, 142, 146, 187, 195, 257
Decompose, 34, 121, 154, 161, 180
Decomposition, 11, 41, 87, 116, 120, 155, 173, 175, 179, 204, 215, 217, 221, k
Degree of Freedom
(DoF), 5, 15, 20, 55, 60, 70, 84, 90, 188, 216, 222, 253, 258
Denavit-Hartenberg
(DH), 222, 252
Denavit-Hartenberg
(DH), 222
Determinant, 257
Digit, 40, 217, 253
Direct Current
(DC), 57, 69, 91, 218, 252
Direct Drive, 11, 53, 66
Dissatisfaction, 123, 152
Distributed, 2, 9, 29, 52, 55, 64, 84, 90, 104, 109, 116, 178, 189, 216, 218, 221, 223
Divergent Component of Motion
(DCM), 41, 43, 53, 252
Divide and Conquer, 19
Double Support, 19
DURUS, 41, 64, 253
Dynamic, 16, 41, 49, 79

E

Efficiency, 24, 112
Eigenvalue, 132, 159, 179
Eigenvector, 135, 159
Electroencephalography
(EEG), 6, 253
Electromyography
(EMG), 6, 189, 253
EPA-Hopper II, 32, 55, 253
Equilibrium, 20, 69, 105, 193, 203, 254
ERNIE, 24, 253
Ethernet
(ETH), 218, 253
Euclidean, 122, 129
Euler Angle, 258
Euler Integration, 169
Expected Value, 147, 257
Experiment, 4, 10, 11, 18, 41, 52, 58, 69, 94, 104, 109, 170, 175, 194, 204, 214, 215, 217, 220
Experimental Result, 12, 104, 189, 206
Extendability, 113
Extrapolated Center of Mass
(XCoM), 42, 193, 208, 256
Extrinsic, 13, 19, 49, 66, 128, 171, 186, 203, 223

F

Feed-Forward, 31, 44, 55, 70, 105, 185, 201
Feedback, 49, 128
Field Programmable Gate Array
(FPGA), 68, 94, 218, 253, g
Finite Impulse Response
(FIR), 148, 253
Finite State Machine
(FSM), 55, 113, 253
Finroc, 218, 253
Fixed Base, 20
Fixpoint Iteration, 128
Flexibility, 112, 141
Floating-Base, 20, 128, 146, 186, 191, 222
Floating-Point, 12, 23, 122, 153, 180, 252, 257
Four-Bar Linkage, 62
Functional, 144, 161, 166
Fusion, 118
Average, 118
Maximum, 118

Future
Future Investigation, 77, 92, 153
Future Research, 10, 60, 77, 86, 177, 215, 222
Future Studies, 19, 188, 222
Future Work, 12, 19, 78, 92, 116, 121, 140, 176, 178, 194, 222
Fuzzy, 118

G

Gait Cycle, 17
Gait Periods, 19
Gait Phases, 17
Gastrocnemius
(Gas), 61, 62, 251, 253
Gaussian, 122, 147, 197, 209, 223, 258
Distribution, 122, 258
Process, 147, 223
Gaussian Process Regression
(GPR), 158, 223, 253
Generalized Coordinates, 84
Geometric Algebra
(GA), 130, 253
Conformal GA, 177
Projective GA, 129, 258
Gimbal Lock, 129
Gluteus Maximus
(GMax), 62, 251, 253
Golgi Tendon Organ, 71
Gradient, 115, 125, 170, 180
Gradient Descent, 125, 170
Graph, 113, 120, 142, 146, 155, 162, 170, 176, 195, 206, 221, 257
Multigraph, 121
Subgraph, 125, 157, 162, 190, 204, 221
Ground Reaction Force
(GRF), 23, 36, 192, 253

H

Hamstrings Group
(Ham), 62, 251, 253
Hessian, 94, 180
Hill-Type Muscle, 46, 71
Hip Strategy, 35, 189
Hot Starting, 125
Hybrid, 2, 10, 15, 20, 42, 51, 84, 90, 104, 109, 111, 116, 128, 175, 216, 220, 222, 223, m, n
Hybrid Dynamics, 15, 20, 42, 219
Hybrid Zero Dynamics
(HZD), 41, 43, 49, 53, 113, 253
HyQ, 54, 67, 112, 253

I

Iliacus Psoas
(IL), 62, 251, 254
Impedance, 11, 18, 45, 51, 63, 84, 89, 104, 109, 113, 129, 181, 189, 216, 221–223, k, m
Infinite Impulse Response
(IIR), 148, 254
Inhibition, 181
Integrated Behavior-Based Control
(iB2C), 118, 121, 155, 217, 253
Integrated Circuit
(IC), 254
Integrated Smart Actuator, 54
Integration
Explicit Euler, 169
Implicit Euler, 169
Probabilistic, 149, 172, 177
Runge-Kutta, 169
Intellectual Property Transfer, 3, 6, 16, 39, 50
Intelligence, 1, 29, 73, 87, 124
Interface Duality, 117, 120
Interneuron, 29, 77, 254
Intrinsic, 10, 11, 13, 19, 46, 49, 51, 54, 64, 84, 90, 108, 116, 128, 146, 171, 183, 188, 203, 223
Intuition, 29
Invariance, 33, 115, 117, 141, 162
Composition, 116
Decomposition, 116
Frequency, 33, 116, 143
Reorganization, 116
Transformation, 33, 115, 162

Inverted Pendulum
(IP), 34, 44, 188, 206, 254
Double Inverted Pendulum, 37, 189
Single Inverted Pendulum, 37, 44, 189

J

Jacobian, 23, 84, 98, 142, 166, 177, 191, 205, 257
Jansen's Linkage, 15

K

Kalman Filter, 26, 115, 147, 158
Kinematic, 4, 22, 47, 95
Kronecker Product, 136, 257

L

Laplace-Transformation, 34
Learning, 31, 44, 116, 222
Linear Time-Invariant System
(LTS), 34, 254
Lola, 40, 254
Lyapunov Stability, 20, 254

M

Machine Learning
(ML), 53, 113, 114, 146, 158, 223, 254
Maintainability, 112, 141
Maximal Coordinates, 188, 222
McKibben, 46
Mean, 58, 122, 129, 148, 157, 163, 172, 181, 257
Precise Mean, 129, 164, 181
Minimal Coordinates, 85, 177, 222
Minimization, 90, 135, 159, 170, 178
Mises-Fisher Distribution, 129
MIT Cheetah, 54, 67, 254
Mode, 124, 133
Model Predictive Control
(MPC), 2, 5, 16, 104, 112, 125, 146, 183, 222, 254, k
Monoarticular, 46, 60, 61, 86, 95, 108, 201, 212
Monoarticular ankle actuator
(A_A), 62, 251
Monoarticular hip actuator
(A_H), 62, 95, 108, 251
Monoarticular knee actuator
(A_K), 62, 95, 251
Moore-Penrose, 184
Morphology, 1, 3, 10, 23, 40, 49, 51, 59, 84, 95, 108, 109, 111, 147, 175, 180, 200, 213, 216, 219, 223, k
Motor Neuron, 29, 31, 70, 254
Multiarticular, 62, 70, 102, 111, 213
Multiple-Input Multiple-Output
(MIMO), 34, 254
Multitasking, 117, 155
Multivector, 129, 222
Muscle Memory, 29, 124
Muscle Stretch Reflex, 30, 71
Muscle Synergy, 29, 31, 124
Muscle-Reflex Strategies, 53
Musculoskeletal, 3, 6, 32, 50, 61, 65, 103, 104, 128, 213, 217–219
Musculoskeletal Apparatus, 13, 34, 49, 55, 72
Musculoskeletal Control, 51, m
Musculoskeletal Morphology, 61, 84, 186, 200
Musculoskeletal System, 4, 11, 51, 73, 92, 213, 216

N

Nadia, 40, 254
Natural Frequency, 24, 70, 210
Natural Squat, 12, 175, 217, 221, n
Neural Network
(NN), 6, 113, 119, 125, 254
Neuromuscular, 5, 10, 19, 109, 115, 216, 222, 223, k
Newton Iteration, 153
Non-Euclidean, 122
Non-Functional, 28, 114
Non-Real-Time, 200

Normal Distribution, 11, 122, 129, 151, 158, 217, 257
Matrix, 134, 257
Multivariate, 122, 129, 217

O

Open-Loop, 22, 80, 128
Operating System (OS), 200, 254
Optimal Control (OC), 2, 5, 15, 34, 125, 254
Optimization, 23, 43, 74, 90, 104, 124, 135, 152, 156, 166, 177, 219
Ordinary Differential Equation (ODE), 169, 254
Oscillation, 24, 66, 127, 194, 208
Oscillator, 60, 69
Over-Actuation, 1, 3, 8, 60, 70, 84, 90, 111, 182, 213, 215, 216, 223, k
Overdetermined, 92
Oversampling, 150

P

Parallelization, 117
Patellar Reflex, 30
Petri Net, 112
Physical Property Transfer, 3, 5, 6, 39, 49
Pneumat-BH, 46, 255
Pneumatic Artificial Muscle (PAM), 46, 254
Pose, 8, 53, 70, 85, 171, 197, 206, 218
Positive Semi-Definite, 90, 122, 129, 157, 179
Positive-Definite, 90, 132, 150, 160, 180, 191
Postprocessing, 147
Precise Notation, 129, 168, 178
Precision, 9, 12, 26, 38, 51, 66, 86, 104, 131, 147, 157, 168, 173, 186, 193, 217, 222
Precision Matrix, 129, 153, 157, 163, 186
Preflex, 29, 73, 116, 124
Preprocessing, 147, 218
Probabilistic Behavior Network (PBN), 11, 111, 113, 114, 120, 128, 146, 162, 170, 173, 175, 176, 217, 218, 220, 222, 223, 254, 257, k
Probability, 130, 173, 215
Probability Calculus, 124
Probability Distribution, 131
Probability Theory, 11, 217, 223
Probability Density Function (PDF), 122, 131, 151, 163, 254
Projection Space (PS), 162, 255
Projective Geometric Algebra (PGA), 129, 177, 222, 255, 258
Proportional-Integral-Derivative (PID), 34, 68, 118, 125, 206, 255
Pseudo Inverse, 184
Pseudo-Analog, 127

Q

Quantization, 144
Quasi-Direct Drive, 11, 53, 66
Quaternion, 84, 119, 122, 129
Dual Quaternion, 139

R

Range of Motion (RoM), 60, 255
Real-Time, 28, 38, 53, 69, 113, 114, 200
Recommendation, 9, 12, 66, 215, 216, 222, k
Rectus Femoris (RF), 62, 251, 255
Reflected Inertia, 80
Reflex, 1, 3, 29, 47, 53, 65, 116, 124, 220
Reinforcement Learning (RL), 44, 255
Reliability, 46, 194
Reorganization, 117
Requirement, 5, 10, 13, 16, 38, 40, 48, 50, 51, 53, 66, 84, 92, 109, 111, 114, 117, 120, 141, 149, 162, 171, 173, 175, 176, 190, 203, 214–216, 219, m
Architecture Requirement, 114, 143

Non-Functional, 28, 114
Technical Requirement, 29, 39
Responsiveness, 9, 177, 192, 205, 222, 257
Rigid-Body dynamics, 20
Root-Finding, 152
Rotor, 129, 149, 158, 190, 222, 257
RPTU, 3, 18, 40, 52, 53, 133, 175, 215, 218, 220, 251, 255
RRLab, 55, 65, 84, 90, 149, 187, 194, 206, 255, g
RRLab SEA, 56
Runaway Integration, 143, 178, 198, 212
Runge-Kutta
(RK), 169, 255

S

Safe Torque Off
(STO), 69, 200, 255
Sagittal
Sagittal Leg, 61, 62
Sagittal Muscular System, 62
Sagittal Plane, 16, 62
Satisfaction, 122, 129, 150, 156, 258
Screw Theory, 8, 256
Semantic, 114, 118, 129
Sensor Neuron, 29, 31, 254, 255
Series Elastic Actuator
(SEA), 3, 8, 11, 46, 52, 55, 65, 77, 84, 90, 109, 111, 185, 194, 206, 216, 218, 219, 255, k
Short-Range Stiffness, 32, 66
Sigma Notation, 129, 164
Single Support, 19
Single-Input Single-Output
(SISO), 34, 80, 128, 146, 255
Singular Value Decomposition
(SVD), 122, 132, 154, 159, 255
Singularity, 15, 23, 45, 88, 96, 169, 252
Close-to-Singular, 23, 24, 206, 252
Non-Singular, 23, 169, 180
Singular, 10, 16, 43, 56, 88, 96, 114, 141, 169, 187, 206, 216, 219, 252
Singular Knee, 16, 24
Singular Position, 10, 22, 43, 56, 88, 96, 114, 141, 187, 206, 216, 219
Singular-Free, 43, 187, 219
Skill, 113
Smart Actuator, 32, 40, 65
Soleus
(Sol), 62, 251, 255
Spinal Cord, 29
Spinal Pattern Generator
(SPG), 55, 255
Spot, 54, 255
Spring Leg, 27, 176, 203, 218, 220
Spring Loaded Inverted Pendulum
(SLIP), 27, 44, 190, 203, 251, 255, m
Square Root Free Cholesky Decomposition
(LDL), 122, 179, 254
Stance, 18
Standard Deviation, 132, 155, 172, 193
Standardization, 112
State Space, 28, 186, 195, 221
Step Strategy, 189
Stiff Inside, Soft Outside, 66, 67
Stiffness, 65, 85, 105
Stiffness Ellipse, 72
Strandbeests, 15, 255
Stretch Reflex, 29
Subsumption, 116
Support Interval, 191, 206
Support Polygon, 20, 189
Swing, 18
Symmetric Part, 136
System on Chip
(SoC), 218, 255

T

Tendon, 66
Tensor, 170
Tibialis Anterior
(TA), 62, 251, 255
Toes-Off, 21
Tolerance, 12, 26, 122, 142, 149, 155, 171, 183, 200, 204, k
Tool Center Point
(TCP), 30, 72, 84, 104, 255
TORO, 41, 64, 255
Transformation Space
(TS), 162, 256

Transparency, 11, 53, 67, 220
Twist, 8, 84, 104, 166, 177, 256

U

Uncertainty, 13, 18, 106, 114, 142, 178, 217, 223, k
    Uncertain Data, 12, 114, 128
Uncontrollable, 15, 16, 215
Under-Actuation, 11, 60, 90
Underdetermined, 90
Undersampling, 151
Unit
    Circle, 132
    Multivector, 129
    Rotor, 129, 149, 158
    Sphere, 122, 129
    Vector, 130, 257
User Datagram Protocol via Internet Protocol
    (UDP/IP), 200, 218, 256

V

Valkyrie, 34, 40, 256
Variance, 122, 130, 148, 155, 163, 193, 258
    Exponential Variance, 149
Vastus Group
    (Vas), 62, 251, 256
Versatility, 112
Virtual Pivot Point
    (VPP), 37, 256
Virtual Repellent Points, 42

W

Weighted Average, 118, 155, 178
    Multivariate, 150
Weighted Square Error, 93
Work Space
    (WS), 43, 84, 85, 87, 89, 105, 163, 171, 176, 190, 204, 219, 256
Workspace, 108
Wrench, 8, 53, 70, 84, 89, 90, 104, 116, 177, 256

Z

Zero Moment Point
    (ZMP), 35, 36, 189, 256

Patrick Michael Vonwirth

Curriculum Vitae

It is simple to do something complex,
yet it is complex to do the same simple.

Higher Education

12/2017 – 06/2025	**Ph.D. in Computer Science** RPTU University Kaiserslautern-Landau, Campus Kaiserslautern, Robotics Research Lab
Research Field	Neuro- and Muscle-inspired Humanoid Robot Control
Dissertation	*Natural Control* *Compliant Robot Control Featuring Neuromuscular Characteristics*
10/2014 – 09/2017	**M.Sc. in Computer Science** University of Kaiserslautern
Core Subject	Embedded Systems and Robotics
Thesis	*Modular Control Architecture for Bipedal Walking on a Single Compliant Leg*
04/2011 – 02/2015	**B.Sc. in Computer Science** University of Kaiserslautern
Core Subject	Software Engineering
Thesis	*Graphical Interface for Query Generation in Behavior-Based Networks Covering Environmental Influences*

About this Book

Bipedal locomotion is a complex field in both biology and robotics. As modern AI-enhanced control methods continue struggling to manage compliant structures, artificial bipedal locomotion still lacks the natural agility and adaptability of biological systems.

To bridge this gap, a novel natural control paradigm is developed, inspired by the human gait and neuromuscular complex. The approach employs antagonistic muscular control as a low-level hybrid impedance control strategy for series elastic actuators. A probabilistic behavior architecture provides an efficient high-level control framework. Both enable natural adaptability and resilience against disturbed motions in uncertain environments.

Experimental results on the bipedal robot CARL demonstrate the effectiveness of the natural control system in replicating human-like control characteristics.

About the Author

Patrick M. Vonwirth studied computer science at the University of Kaiserslautern, focusing on robotics, embedded systems, and software engineering. During his studies, he specialized in biological control paradigms, which became his main field of research. In 2017, he joined the Robotics Research Lab, where he developed muscle-inspired actuation and neural-inspired control methodologies.