

Jean Oscard Domguia Teto

mts
&



Isolation of a measurement platform for UAV applications

Berichte aus dem Lehrstuhl für Messtechnik und Sensorik
Band 13

Herausgeber: Prof. Dr.-Ing. Jörg Seewig

Jean Oscard Domguia Teto

mts



Isolation of a measurement platform for UAV applications

Berichte aus dem Lehrstuhl für Messtechnik und Sensorik
Band 13

Herausgeber: Prof. Dr.-Ing. Jörg Seewig

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Berichte aus dem Lehrstuhl für Messtechnik und Sensorik - Band 13

Herausgeber: Prof. Dr.-Ing. Jörg Seewig
Lehrstuhl für Messtechnik und Sensorik
Fachbereich Maschinenbau und Verfahrenstechnik
Technische Universität Kaiserslautern
Gottlieb-Daimler-Straße
67663 Kaiserslautern

Verfasser: Jean Oscard Domguia Teto
Verlag: Technische Universität Kaiserslautern

Druck: Technische Universität Kaiserslautern
Hauptabteilung 5 | Abteilung 5.6 Foto-Repro-Druck
D-386

© Jean Oscard Domguia Teto · Kaiserslautern 2022

Alle Rechte vorbehalten, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe (Photographie, Mikroskopie), der Speicherung in Datenverarbeitungsanlagen und das der Übersetzung.

Als Manuskript gedruckt. Printed in Germany.

ISSN 2365-9742
ISBN 978-3-95974-175-0

Isolation of a measurement platform for UAV applications

Vom Fachbereich Maschinenbau und Verfahrenstechnik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

verlegt von

M.Sc. Jean Oscard Domguia Teto

aus Jaunde

Kaiserslautern 2022

D386

Dekan:	Prof. Dr.-Ing. Tilmann Beck
Vorsitzender:	Prof. Dr.-Ing. Martin Böhle
Berichterstatter:	Prof. Dr.-Ing. Jörg Seewig Prof. Dr.-Ing. Eduard Reithmeier
Tag der mündlichen Prüfung:	14.02.2022

Preface and Acknowledgments

This thesis was written during my time as scholarship student and research assistant at the Institute of Measurement and Sensor Technology at the University of Kaiserslautern.

In the first place, I would like to express my sincere thanks to my doctoral advisor Prof. Dr.-Ing. Jörg Seewig for giving the opportunity to work on this thesis and for the advices during my research. I also thank Prof. Dr.-Ing. Eduard Reithmeier for accepting reviewing my work.

I would also like to thank my colleagues from the Institute of Measurement and Sensor Technology of the University of Kaiserslautern for the good coworking and the great atmosphere. A special thank to Dipl.-Ing. Viktor Follmann for the moral support and the helpful expert talks during my research. I also thank Dipl.-Ing. Tycho Groche for the discussions, which help me a lot during my research.

I would also like to thank the students who supported me during my research. Especially Dipl.-Ing. Björn Diederichs and B.Sc. Abdullah Farrukh were a great help to me.

A special thank to Dr.-Ing. Marcus Ripp. I am deeply grateful for his helpful advices ever since I started studying at the University of Kaiserslautern.

I also want to thank my family and my friends for the unconditional support and the encouragement. I dedicate this thesis to my father Emmanuel Tetok, who always supports me and gives me great advices. I am very proud, because he always wanted me to become a doctor. So, for me this is an accomplishment of my father's dream. I also dedicate this thesis to my big brother Roméo Kamdem Teto, who has been a great support and excellent mentor to me since my arrival in Germany.

Abstract

In the field of measurement technology, the use of unmanned aerial vehicles is becoming more and more popular. For many measurement tasks, the use of such devices offers many advantages in terms of cost and measurement effort. However, the occurring vibrations and disturbances are a significant disadvantage for the application of these devices for several measurement tasks. Within the scope of this work, a platform for measurement devices is developed. The platform is designed specifically for use on drones. The task of the platform is to isolate measurement equipments mounted on it from the drone disturbances. For this purpose we go through the product development process according to VDI 2221 to design a mechanical model of the platform. Then, control strategies are applied to isolate the platform. Since the disturbances acting on a drone are not always stationary, two control strategies known for their ability to handle uncertain systems are used. One of them comes from the field of acoustic.

Kurzfassung

Im Bereich Messtechnik kommt der Einsatz von unbemannten Fluggeräten immer häufiger in Frage. Für zahlreiche Messaufgaben bietet der Einsatz solcher Geräte viele Vorteile bezüglich der Kosten und der zu treibender Messaufwand. Jedoch sind die auftretenden Vibrationen und Störungen ein wesentlicher Nachteil für die Anwendung von diesen Geräten bei vielen Messaufgaben. Im Rahmen dieser Arbeit wird eine Plattform für Messgeräte entwickelt. Die Plattform wird speziell für den Einsatz auf Drohnen entworfen. Die Aufgabe der Plattform ist die darauf angebrachte Messeinrichtung von den Drohnenstörungen zu isolieren. Zu diesem Zweck gehen wir durch das Produktentwicklungsprozess nach VDI 2221, um ein mechanisches Modell der Plattform zu entwickeln. Dann werden Regelstrategien zur Isolierung der Plattform entworfen. Da die auf einer Drohne wirkenden

Störungen nicht immer stationär sind, werden zwei Regelstrategien angewendet, die für ihren Umgang mit unsicheren Systemen bekannt sind, wobei eine von den beiden Methoden aus dem Bereich Akustik kommt.

Summary

Due to the wide range of possibilities they can offer, unmanned aerial vehicles are being used more and more in various fields. Whether for inspections of difficult-to-access elements, for agriculture, or for surveillance, to name just a few, unmanned aerial vehicles offer an efficient and cost-effective solution for accomplishing multiple tasks.

Depending on the mission and the tasks, there are several requirements on the UAVs. From missions such as fire detection or gas leak detection to missions such as precise 3D reconstruction through missions such as monitoring of photovoltaic plants or monitoring of structures and buildings, the requirements on the stability of UAVs are becoming always higher. Especially in the field of metrology and photography, the requirements for stability are very high, since it has a significant impact on the result.

To compensate the effect of these disturbances, various stabilization strategies have been developed, which can be sorted into four categories. One of them, which is very application-specific, is the use of post-processing algorithms. This strategy does not improve the collected data per se, but the post-processing algorithms are robust enough to produce acceptable result-quality based on collected data. The second strategy, which is also application-specific, is to adapt the measurement principle and the measurement components so that they can compensate disturbances on their own. A variant of this strategy is the optical image stabilization in the field of photography. Another strategy is inertia-based stabilization, which is usually implemented as a gimbal. The fourth strategy is mechanical vibration damping. This strategy consists in incorporating mechanical elements between the UAV and the measurement devices to isolate the measurement devices from the disturbances occurring on the UAV. The last strategy has the advantage of being application-independent. Moreover, it can be used as a complement to the other strategies.

The aim of this work is to design a mechanical vibration isolation structure. For this purpose, a mechanical model for the isolation platform is designed. The platform is

to be mounted on a drone and should be usable for many applications. The design process of the mechanical model of this platform with actuators is carried out according to VDI 2221.

The platform should be actively isolated. For this purpose, controller strategies for the isolation of the platform are designed. Since the disturbances occurring on the drone are not always predictable, these strategies should be able to deal with uncertainties. Two control strategies are used therefore. One is the principle of Active Noise Control (ANC), which is often used for acoustic systems. ANC belongs to the class of adaptive controllers. With its application, online identification and disturbance reduction are aimed simultaneously. In this work, an algorithm for ANC is proposed. This algorithm aims to achieve better results by combining equation-error formulation and output-error formulation for adaptive IIR filters. An Optimization App for simulations is also developed to help find good parameters for the ANC algorithm. The second strategy used for vibration isolation is robust control. A simulation environment and test benches are designed for testing purposes.

Zusammenfassung

Aufgrund des breiten Spektrums an Möglichkeiten, dass sie anbieten, werden unbemannte Fluggeräte mehr und mehr in verschiedenen Bereichen eingesetzt. Sei es für Inspektionen von schwierig zugänglichen Elementen, für die Überwachung, oder für die Landwirtschaft, um nur diese zu nennen, bieten sich unbemannte Fluggeräte als eine effiziente und kostengünstige Lösung zur Erledigung vielfacher Aufgaben.

Abhängig von der Mission und von den zu erledigenden Aufgaben ändern sich die Anforderungen an den Fluggeräten. Ausgehend von Missionen wie Branderkennung oder Gasleckererkennung zu Missionen wie präzises 3D-Rekonstruktion durch Missionen wie Monitoring von Fotovoltaikanlagen oder Monitoring von Bauwerken und Gebäuden werden die Anforderungen an der Stabilität von Fluggeräten immer höher. Vor allem in dem Messtechnikbereich und in der Photographie sind die Anforderungen an der Stabilität sehr hoch, da sie einen großen Einfluss auf die Ergebnisse hat.

Um den Effekt diese Störungen auszugleichen, wurden verschiedene Strategien für Stabilisierung entwickelt, die in vier Kategorien sortiert werden können. Eine davon, was sehr anwendungsspezifisch ist, ist die Anwendung von Postprocessing-Algorithmen. Da werden die gesammelten Daten an sich nicht verbessert. Aber die Postprocessing Algorithmen sind robust genug, um daraus eine akzeptable Qualität der Ergebnisse zu erzeugen. Die zweite Strategie, die ebenfalls anwendungsspezifisch ist, ist die Anpassung des Messprinzips und der Messkomponenten, sodass sie selbstständig Störungen kompensieren. Eine Ausführung davon ist die optische Bildstabilisierung im Bereich der Photographie. Eine andere Strategie ist die trägheitsbasierte Stabilisierung, die meistens als Gimbal implementiert wird. Die vierte Strategie ist die mechanische Vibrationsdämpfung. Diese Strategie besteht darin, mechanische Elemente zwischen dem Fluggerät und der Messeinrichtung einzubauen, die die Messeinrichtung von den im Fluggerät auftretenden Störungen isolieren sollen. Die letzte Variante hat den Vorteil, anwendungsunabhängig zu sein. Darüber hinaus, kann sie komplementär zu den anderen Strategien verwendet werden.

Ziel dieser Arbeit ist eine mechanische Vibrationsisolierungseinrichtung zu entwerfen. Dafür wird ein mechanisches Modell für eine Isolierungsplattform entworfen. Die Plattform ist an einer Drohne zu montieren, und soll für möglichst viele Anwendungen einsetzbar sein. Der Entwurfprozess des mechanisches Modell dieser Plattform mit Aktuatoren wird nach VDI 2221 durchgeführt.

Die Plattform soll aktiv isoliert werden. Zu diesem Zweck werden Reglerstrategien für die Isolierung der Plattform entworfen. Da die auf der Drohne auftretenden Störungen nicht immer vorhersehbar sind, sollen diese Strategien mit Unsicherheiten umgehen können. Zwei Arten von Reglern werden hierfür entworfen. Die eine basiert auf dem Prinzip der Active Noise Control (ANC), das häufig für akustische Systeme verwendet wird. ANC gehört zu der Klasse der adaptiven Regler. Durch seine Anwendung erfolgt gleichzeitig eine online Identifikation und eine Unterdrückung der Störungen. Im Rahmen dieser Arbeit wird auch ein Algorithmus für ANC vorgeschlagen. Dieser Algorithmus soll durch eine Kombination von Equation-Error Formulierung und Output-Error Formulierung für adaptive IIR Filter bessere Ergebnisse anstreben. Es wird auch eine Optimization App für Simulationen entwickelt, die dabei helfen soll, gute Parameter für den ANC Algorithmus zu finden. Die zweite angewendete Strategie für die Vibrationsisolierung ist die robuste Regelung. Zu Testzwecken werden eine Simulationsumgebung und Prüfstände entworfen.

Contents

Preface and Acknowledgments	III
Abstract	V
Summary	VII
1 Introduction	1
2 States of the Art	7
2.1 Vibration sources in UAVs	7
2.2 Vibration isolation	8
2.2.1 Optical image stabilization	9
2.2.2 Digital image stabilization	10
2.2.3 Inertia-based stabilization	11
2.2.4 Mechanical vibration damping	12
2.3 Summary	14
3 Theoretical background	17
3.1 Design procedure of technical products	17
3.1.1 Product development process	18
3.1.2 Development process for control system	20
3.2 Vibration control	21
3.2.1 Types of vibration control	22
3.2.2 Actuators for active vibration control	23
3.3 Control Strategies	25
3.3.1 Robust control	25
3.3.2 Adaptive control	35
3.3.3 Central and decentral control	43
3.4 Optimization algorithms	45
3.4.1 Biologically-inspired algorithms	47
3.4.2 Generalized pattern search optimization	51
3.5 Summary	52
4 Isolation platform design	53
4.1 Specification phase	53
4.1.1 Load requirements	55
4.1.2 Dynamic requirements	55
4.1.3 Energy consumption requirements	56
4.1.4 Other requirements	56
4.2 Concept phase	56
4.3 Design phase	66
4.3.1 Test bench for single suspension	68
4.3.2 Test bench for isolation platform	68
4.4 Summary	70

5	Simulation design	71
5.1	Actuator dynamic	71
5.1.1	Single suspension	72
5.1.2	Isolation platform	84
5.2	Simulation design	86
5.3	Summary	88
6	Controller design	91
6.1	Adaptive Control	91
6.1.1	Identification algorithm	92
6.1.2	Proposed algorithm	97
6.1.3	Application on ANC algorithms	102
6.1.4	Determination of optimal parameters for ANC algorithms	108
6.1.5	SISO-system	111
6.1.6	MIMO-system	117
6.2	Robust control	125
6.2.1	SISO-system	125
6.2.2	MIMO-system	132
6.3	Summary	139
7	Hardware implementation	141
7.1	Components	141
7.2	SISO-system	144
7.3	MIMO-system	146
7.4	Summary	147
8	Conclusion and outlook	149
	List of Figures	153
	List of Tables	157
	Bibliography	159
A	Appendix	165
A.1	Matlab application designed with AppDesigner	165
B	Lebenslauf des Verfassers	173

1 Introduction

Nowadays, the application fields of modern Unmanned Aerial Vehicles (UAVs), commonly known as drones, are gradually moving from military fields to a wide range of civil applications [23], and their use in civilian market is continually growing [21]. This trend is due to the continuous integration of additional systems composed of multiple sensors, actuators and computing units on UAV platforms, that allow the accomplishment of complex assignments with higher efficiency than the techniques used so far. Depending on the types of sensors attached to UAV platforms, they can be used in many different areas like agriculture [31], surveillance [41], aerial photography [42], gas leak detection [65], fire detection mission [30], environmental monitoring [35], archaeology [48], monitoring of photovoltaic systems [63], monitoring of structure and building [74], inspection of power lines [38] etc. (Figure 1.1).

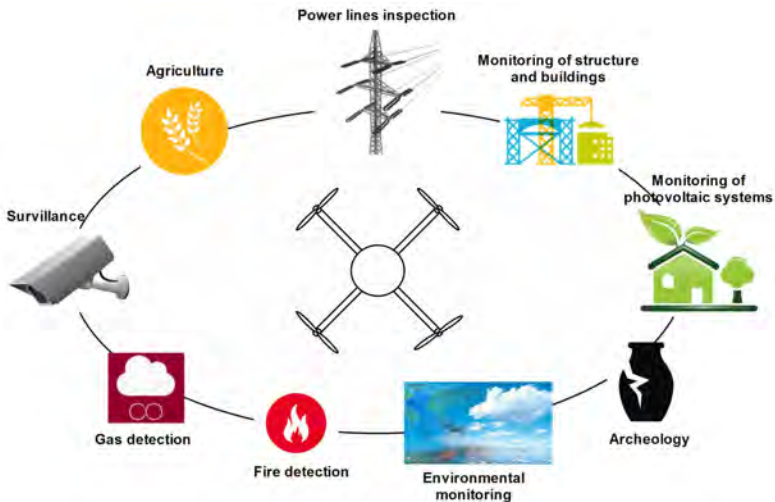


Figure 1.1: UAV for civil applications [23]

In agriculture, UAVs are used for monitoring biomass, crop growth and food quality, harvest and logistic optimization, and precision farming [31]. There is a camera

mounted on the drone to collect images from the ground. However the problem of platform stability and the wind influence during the image acquisition occurs.

To localize and map gas leaks, drones can be equipped with gas sensing systems based on a MEMS gas sensor array, USM-VOC 3.0 from Unitronic [65], that allows measurements of CO₂, NO₂ and SO₂ concentrations.

In environmental monitoring applications, UAVs can be used to detect dangerous situations based on different environmental quantities measurements of hardly accessible place (e.g. volcano). They can also be used for water pollution monitoring to quantify the presence of living vegetation, etc.

For the monitoring of photovoltaic systems, a video camera, a thermal camera and a GPS are mounted on the UAV platform. The goal is to detect crack, misalignment, bubbles, snail trails and burnt cells [63].

In industrial plants and buildings, UAVs are used for inspection. The materials usually deteriorates with time and should be change at the right time to prevent eventual damages. Drones are a cheap and efficient solution for inspection of the state of the material, especially for high and large plants.

For power line inspection the foot patrol and the helicopter-assisted inspection are the two well-known methods. UAV platforms allow faster inspection than the foot patrol, and they have a better accuracy and are cheaper than inspections with helicopter.

In this work, the area of interest is the Large-Scale Dense 3D Reconstruction from Multi-View Stereo Airbone Cameras. This is related to the research project of the Institute of Measurement and Automatic Control of the Leibniz University of Hanover entitled "Simultaneous Localization and Mapping System using Aerial Camera" [27]. There are some available high-quality 3D reconstruction instruments. One of them is the Terrestrial Laser Scanning system. It uses the time-of-flight principle and the waveform, but it has limited view directions, and is cumbersome and costly. There is also the structured-illumination system, which is based on triangulation measurement principle. This system works only for small volumes and is intended for indoor applications. Another available system for 3D reconstruction is the industrial photogrammetry system. It uses two-cameras stereo configuration and requires fiducial markers, control points and well-trained human operator. The drawbacks of this system is that it is time-consuming, labour intensive and technically complex. The target is to have an affordable, practical and effective solution that enables a complete and highly-detailed 3D reconstruction of relatively large-scale objects and scenes. The solution proposed for the imaged-based 3D reconstruction is light-weight digital cameras with low

power-consumption and high-resolution. Since computer vision algorithms for 3D reconstruction are already available, this solution is cost effective and efficient for large-scale reconstruction tasks. To acquire the necessary data, the measurement system should be moved around. Therefore, an unmanned aerial vehicle is used as data acquisition platform (Figure 1.2). UAVs provide high maneuverability,



Figure 1.2: Measurement platform for 3D reconstruction attached to an UAV [68]

relatively low cost and good portability. However, as for every application of UAVs with data acquisition platform, one the main problem is the vibrations caused by the motors, the propellers and the wind disturbances. According to Windau and Itti [79], these vibration have five main sources: the engine, the unexpected strong sources (e.g. wind gusts), the stabilizing task, the fly manoevers, and the aerodynamic sources (e.g. headwind). These vibrations can be compensated using different techniques. Two of them are very popular. The first way to compensate vibrations is to improve the robustness of the processing algorithm in order to handle the imperfections caused by the vibrations. This method can be classified as digital stabilization method. It usually use software-based post-processing algorithms, whose task is to filter out the effects of vibrations on collected data. This method is often used for image-based applications, since many algorithms for image processing already exists. However, these algorithms are effective for small and medium amplitudes. Moreover, depending on the application or on the nature of collected data, such post-processing algorithms for stabilization may not exist. The second and the certainly most popular way is to isolate the data acquisition platform from UAV vibrations. This is known as mechanical stabilization. According to Verma et al. [75], it is the only stabilization method which can perform well in the entire frequency range for all amplitudes. As the name says, this method consists of changing or adding a mechanical construction or mechanical components to the

system in order to reduce the effect of vibration on the data collection system. There are different varieties of this method, which can be classified in three main categories: passive isolation, semi-active isolation, and active isolation. Many types materials such as rubber elements, wire ropes (Maës, Binczak, and Lhenry [51]), viscoelastic elements (Webster and Semke [76]), etc., have been employed for passive isolation on UAVs. According to Verma et al. [75], the passive isolation structures are effective near the resonance frequency, but they amplify high-frequency vibrations. These problems can be avoided by using active isolation systems. These systems use actuators to compensate unwanted vibration noises. Different types of actuators have been used to design such systems. The most popular are piezo stack actuators (Stuckel and Semke [72]) and voice-coil actuators (Verma et al. [75]).

This thesis aims to design an application-independent isolation system for UAVs. Usually, most of the isolation systems designed for UAV applications are designed in the form of gimbal systems. The structure of gimbal systems is optimal for camera and lightweight payloads, but it may limit the usability of isolation systems for other applications. The isolation system to be designed in this thesis should have a platform that is adaptable for various applications.

The structure of the thesis is organized as follows:

Chapter 2 deals with the state of the art. This chapter reports on the existing scientific advances in the field of vibration isolation in UAVs. Different materials used for passive and semi-active vibration isolation in UAVs are overviewed. Then, different control strategies and isolation platform structures developed so far are reviewed.

Chapter 3 deals with the theoretical basics that are necessary for a good understanding of this thesis. In this chapter, the basics of mechanics for establishing the dynamics of a rigid body are covered. Then different types of vibration isolation systems and control strategies are presented. After that, optimization algorithms are presented, with a focus on gradient-free algorithms.

Chapter 4 focuses on the design of the isolation platform. Here, a construction design of the platform is made according to the product development process conform to VDI 2221. For testing purposes, two test benches are also developed.

Before the manufactured physical equipment is tested, a simulation environment is developed for it. It is used to test the algorithms to be designed without the risk of damaging the physical system. This is the subject of Chapter 5. In addition, Chapter 5 also deals with the mathematical analysis of the isolation system.

Chapter 6 is dedicated to the design of control strategies for the isolation platform.

Because of their ability to deal with imprecise systems, the focus is put on two methods: robust control and adaptive control. The designed controllers are tested in the simulation environment and the result is presented.

Chapter 7 deals with the implementation of the designed algorithms on the test benches. The components and the structure of the test benches are presented.

In the last section, a summary of the results of the work is given, followed by an outlook on possible research approaches.

2 States of the Art

In this chapter, the state of the art regarding vibrations in UAVs is reported. Materials used so far to decouple the weight are presented. Control strategies developed for this purpose are also presented.

2.1 Vibration sources in UAVs

Due to their flexibility and their low cost, UAVs offer a wide range of applications in different domains. They are a cheaper and efficient alternative solution for several Tasks. For some applications, the sensitivity of the equipment is relatively high. In [40] it is mentioned that UAV as a measurement system is restricted through vibrations during the flight and the camera gimbal configuration.

Vibrations occurring in UAVs have many sources. They can be mechanical, aerodynamic and from normal flight motion with broad amplitude and frequency spectrum [62]. Windau and Itti [79] give an overview of the different vibration sources in UAVs in a frequency-amplitude diagram (Figure 2.1). They are:

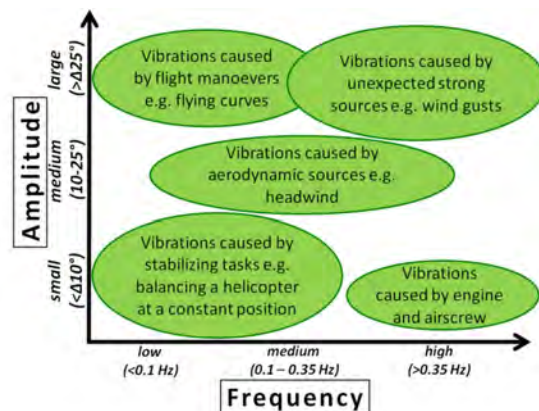


Figure 2.1: Source of vibrations in aerial robotic [79]

UAV Motors: the engine vibrations are usually high-frequency and have low amplitudes. They do not directly affect the payload, since they must be propagated through the UAV frame to reach it. So, the effects of the engine vibrations on the

payload do not only depend on engine vibrations themselves, but also on the propagation path to the payload.

Unexpected strong sources: it makes reference to all sources that cannot be predict. Most of these sources are related to weather conditions (e.g. wind gusts). But they can also refer to all noise related to the environment surrounding the system (e.g. traffic around). These sources can generates vibration with large amplitudes in the range of medium up to high frequencies.

Stabilizing tasks: the stabilizing task in UAV consists in maintaining the position of the UAV constant. Since UAV are not inherently stable system, the control algorithm has to be continuously active to react to disturbances, however small they may be. This stabilization maneuvers generate small amplitude vibrations with low and medium frequencies.

Flight maneuvers: as flight maneuvers we denotes all necessary operations to flight the drone from a point to another following a specified course. Flight maneuvers generates produce vibrations with large amplitudes and low frequencies.

Aerodynamic sources: this refers to the shape of the UAV. The shape of the UAV and the materials used influence the its interactions with the environment. The basics for this are provided by the fluid mechanics. The vibrations caused by these sources have usually medium amplitudes and medium frequencies.

Due to these vibration sources, data obtained from onboard equipment are distorted [1]. According to Li et al. [46], mechanical vibrations of UAV, which greatly hinder the accuracy of on-board sensors, become an increasingly important issue. Therefore, the equipment should be decoupled from all vibration sources as well as possible.

2.2 Vibration isolation

Vibrations in UAVs have always been an important topic of research. Most of previous researches about it focused on the aspects of vibration damping design, attitude control and post-processing algorithm for non-real-time applications [55], [44]. According to Windau and Itti [79], there are four main types of stabilization techniques. They are represented in Figure 2.2 in a frequency-amplitude diagram.

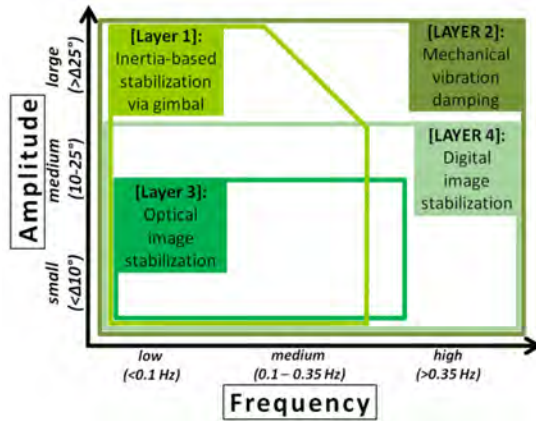


Figure 2.2: Stabilization method [79]

2.2.1 Optical image stabilization

Optical image stabilization is an effective solution to increase the quality of an image taken in a noisy environment. This noisy environment is responsible for small amplitude vibrations which cause blurriness and defocussing. It was first introduced to the consumer market in 1995, with the introduction of stabilized zoom lens by canon [66].

The basic principle of optical image stabilization is to compensate the effect of the movement caused by the object holding the camera by adjusting optical elements in the system (Figure 2.3). The adjustment of optical elements is done by a control

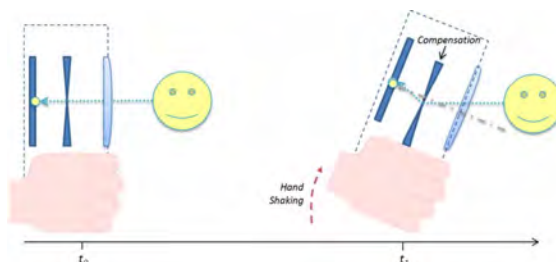


Figure 2.3: Optical image stabilization compensation [43]

system. Gyroscopes are usually employed to sense the movement of the holding object. As actuators, the common types are voice-coil actuators (Song et al. [70],

Chiu et al. [19], Yu and Liu [81]) and piezo actuators (Xu et al. [80]).

La Rosa et al. [43] give an overview of the basic structure of the control system for optical image stabilization. Sachs, Nasiri, and Goehl [66] present requirements on it, and they illustrate it with PID controllers. Various other control models have also been developed, including adaptive algorithms (Yu and Liu [81]) and genetic algorithms (Song et al. [70], Chiu et al. [19]).

Optical image stabilization mostly finds use in smartphones and professional camera, and it is effective for relatively small amplitudes in low up to medium frequency range.

2.2.2 Digital image stabilization

The aim of digital image stabilization is to remove the global motion effects from an image sequence in order to obtain a compensated sequence that displays smooth camera movements only [26]. It is a software-based stabilization methods which can compensate a wide range vibrations, except those with high amplitudes. According to Peng et al. [60], in contrast to optical image stabilization which uses gyro sensors to detect hand motion and shifts a corrective lens inside the lens system (or alternatively shifts the image sensor while keeping the lens fixed), digital image stabilization detects the induced image motion based on the video data and shifts the image display window accordingly to compensate the hand motion. This statement acknowledges that digital image stabilization is a post-processing procedure, since the data have to be captured first before being processed. Figure

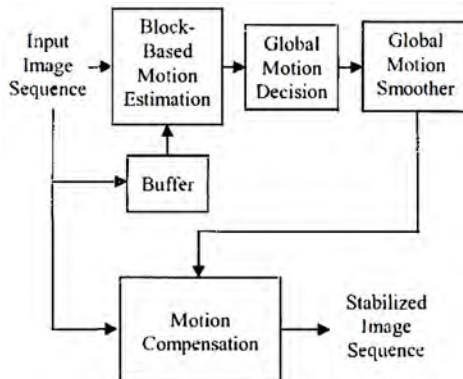


Figure 2.4: Digital image stabilization compensation [47]

2.4 show the architecture of the digital image stabilizer represented by Liang et al. [47].

The three main stages of digital image stabilization are motion estimation, motion compensation, and image warping [64]. Rawat and Singhai [64] give an overview of the developed methods for digital image stabilization.

2.2.3 Inertia-based stabilization

An inertial stabilization platform is used to stabilize the line of sight of an object or device that is tracking another object with stationary or moving targets or targets moving forward [20]. It finds use in several applications including stabilizing cameras, array sensors, and weapons mounted on a moving base [10].

Such stabilization systems are usually realized in form of gimbals. There are two

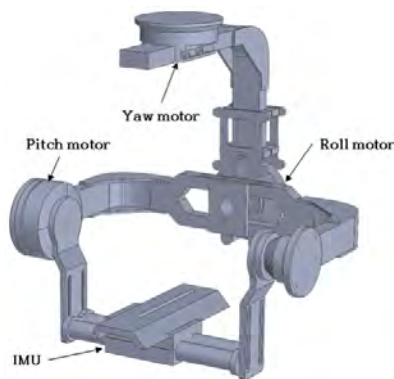


Figure 2.5: Gimbal system architecture [39]

gimbals structures: two-axis gimbals, and three axis gimbals. Three-axis gimbals (Figure 2.5) are more suitable to compensate disturbances in all space directions. The compensation can be done passively or actively. In case of passive stabilization, low friction joints and high inner axis inertia can passively contribute to maintaining the desired attitude [22]. Active compensation uses sensors such as accelerometers, gyroscopes, and magnetometers to estimate the position of the hosting system. These sensors are usually integrated in the inertial measurement unit (IMU) system of the hosting system. Based on the estimated data of the IMU system and the control algorithm, commands are sent to the actuators which are usually motors mounted in the joints. A feedforward approach for inertia-based

stabilization is proposed by DeBruin et al. [24]. However, most of the control algorithms used for gimbals systems are feedback designs. While Baskin and LEBLEBİCİOĞLU [9] used robust control design to stabilize a two-axis gimbal system, Battistel and Oliveira [10] used sliding mode based controllers for three-axis gimbals, and Cong Danh [20] designed adaptive controller for two-axis gimbals.

As shown in Figure 2.2, this stabilization technique is effective for a large range of vibration, but not for those with high frequency.

2.2.4 Mechanical vibration damping

Similar to inertia-based stabilization, mechanical vibration damping use mechanical elements to compensate vibration. However, its implementation is not in form of gimbal systems, but usually in form of payload attached to the hosting system through isolating mechanical components. These components are often designed to allow translational motion. Through a clever combination of the components, rotational movements of the payload can also be performed.

There are different models of mechanical vibration damping design. Most of them can be categorized as passive design, or as active design. The passive design only used passive mechanical components to damp vibrations, without requiring any energy. Well-known materials purposes are springs, dampers, rubber mounts, viscoelastic material. Gjika and Dufour [28] used hysteric suspensions to improve the dynamic behavior of a body attached to another one. Experiments on a camera mounted on an helicopter show a reduced vibration level. Viscoelastic elements have the advantage to act like compact spring-damper systems.

Webster and Semke [76] used these elements in a rotational vibration absorption system for remote sensing application. With viscoelastic elements, they achieved to design a broad frequency band passive system. Maj [52] proposed foam-type isolators to reduce the effect of vibrations on on-board sensors. Compared to common damping materials like neoprene and rubber, tempur-pedic has a higher damping coefficient. This make it better for low-frequency vibrations. Besides the advantage of not needing power, the tempur-pedic is also very lightweight. Tests showed an amelioration around 50% in the measurement error.

The use of wire-rope isolators and rubbers dampers to minimize vibration effects towards data logging during flight has been discussed in [16]. Maës, Binczak, and Lhenry [51] integrated wire-rope isolators (Figure 2.6) in a camera system mounted on an aircraft as passive element to reduce the vibrations affecting the

camera. Wire-rope isolator achieved to reduce the vibration, but not enough to have



Figure 2.6: Wire-rope isolator [51]

a good video quality.

Pete et al. [61] used a spring-damper system on bird-inspired drones to model the neck of a bird. The goal was to decouple the payload (bird head) to the drone itself (bird body) by modeling the neck using passive elements.

As damping system, kyosho Zeal and its performances have also been briefly studied in [46].

For active isolation, some passive elements are used. But additionally, sensors, actuators, a controller board, and an energy source are needed to implement the isolation strategy. Hence, active isolation systems are usually heavier than the passive ones.

Piezo actuators (Stuckel et al. [73], Oh, Han, and Choi [59]) and voice-coil actuators (Verma et al. [75]) are the most used actuators for active isolation systems on UAVs. In [75], a voice-coil actuator is combined with springs and rubber element to form the suspension structure between the gimbal system and the UAV. Through this combination, the isolation system could reach a better performance, and also had a fail-safe behavior.

Oh, Han, and Choi [59] designed an isolation platform for UAVs with piezostack actuators integrated in rubber mounts. The actuators were controlled by a sliding mode controller which is known for its robust behavior. For a camera mount, four active rubber mounts were needed. To evaluate the performance of the suspension system, a hardware-in-the-loop simulation (HILS) was designed (Figure 2.7).

Most of the controllers used are of the PID type. But the trend of research nowadays

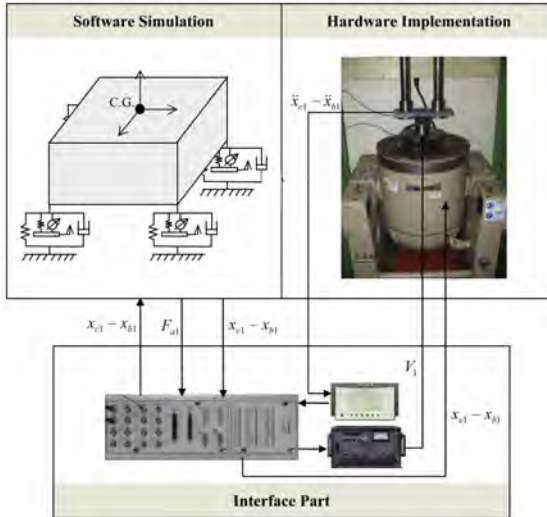


Figure 2.7: Configuration of HILS system for vibration damping in UAV [59]

is towards more robust, and adaptive controllers. Mat Darus and Tokhi [54] proposed an adaptive controller based on genetic algorithm to control the vibration of flexible structure, and a neuro-fuzzy approach was proposed by Marichal et al. [53]. Oh, Han, and Choi [59] proposed an approach with a sliding mode controller. Mechanical vibration damping is the only damping type which is capable of damping vibrations, no matter which amplitudes and frequencies occurs. The effectiveness of the vibration damping depends on the system itself, the control strategy, and the actuators used.

2.3 Summary

In this Chapter, the state of the art of vibration in UAVs was presented. The main sources of vibrations and their effects on UAVs were represented in a frequency-amplitude diagram. The four main sources presented are: unexpected strong sources, stabilizing task, flight maneuvers, and aerodynamic sources.

To damp the induced vibration, there are different classes of compensation strategies. The four well-known ones are: optical image stabilization, digital image stabilization, inertia-based stabilization, and mechanical vibration damping. Each

of these methods and their damping areas were presented in a frequency-amplitude diagram. The mechanical vibration damping appears to be the most effective method to damped vibration in a wider range. But according to Windau and Itti [79], the combination of the four stabilization methods in one application may give better result. Design an isolation platform using a robust strategy for mechanical vibration damping to combine with other application-dependent stabilization methods to achieve a better stabilization of the payload is what we aim.

3 Theoretical background

In this section, the basics that are used in this thesis are presented. First, the design procedure of technical products is presented, then equation of motion of dynamic systems is discussed. After that, the basics of vibration control and optimization algorithms are introduced.

3.1 Design procedure of technical products

The procedure for developing products is characterized by the integration of a multitude of partly competing goals and requirements, which have to be implemented in a task-sharing and discipline-specific manner [12]. According to Bender et al. [13], a methodical approach helps inexperienced product developers in particular, to avoid fundamental mistakes and to follow up competing requirements in parallel.

According to VDI 2221 [36], the technical product life cycle (Figure 3.1) has five

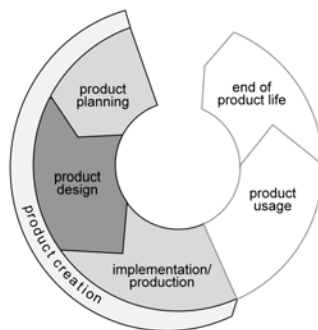


Figure 3.1: Lifecycle of technical product [36]

main phases, which are:

Product planning: in this phase, the framework conditions on which the later design of the product is based, are set [12].

Product design: during this phase, the requirements on the product are set. The requirements are handled and at the end a prototype for the product is created, which is ready for production or implementation.

Implementation/production: usually, the prototype resulting from product design is a digital model or a set of technical drawings and documents. In this phase, these documents are sent to the production so that the physical prototype of the product can be manufactured.

Product usage: if the physical prototype fulfills the requirements on the product, the design phase is done, and the product can be manufactured and used. To the product usage belong also the installation of the product and the maintenance if needed.

End of product life: this phase marks the end of the product use. After its use, the product is either disposed of or recycled.

The focus here is on the second phase, the product design.

3.1.1 Product development process

According to [12], the product design phase is divided in four subphases represented in Figure 3.2.



Figure 3.2: Product Development Process (PDP)

Specification phase

During the specification phase, all information about the product that should be designed is collected. It helps to clarify the purpose and the function of the product. With this information, a list of requirements can be created. At the end of this phase, functions and technical specifications are established, as well as the requirement list.

Concept phase

This phase is the part of the design process which after clarifying the requirements by means of abstracting to the essential problems, setting up functional structures and by searching for suitable working principles and their combination in an

structure, determines the principle solutions [12]. In this phase, the questions **what** and **how** should be answered. This phase is usually divided into different steps:

Elaboration of the black box: the black box is a tool for an abstract and solution-neutral description of the functions of a technical system. It provides a summary of the system inputs and outputs, and an abstraction and delimitation of the tasks of the system.

Creation of the functional structure: the purpose of the functional structure is to break down a complex problem into clearer subproblems, and to represent logical connections. It combines subfunctions into an overall function.

Solution finding and creation of morphological box: here, different solutions are proposed for each subfunction. Then, the proposed solutions are combined in a morphological box. The aim of the morphological box is the systematic combination of subsolutions and the derivation of different suggestions for the overall solution. From the concept phase, there are usually more than one concept resulting. However, only the best concept should be chosen. Therefore, an evaluation process must be carried out to find the best concept methodically. According to DIN EN ISO 9000 ff, an evaluation describes an activity to determine the suitability, the appropriateness and the effectiveness of the unit of consideration to achieve the defined objective [12]. The evaluation process consists of establishing common evaluation criteria for all concepts, which can be evaluated with uniform values for each concept, and comparing the results. The steps of the evaluation process are usually: definition and weighting of the evaluation criteria, quantification of the criteria and individual evaluation of each criterion, and overall evaluation of each concept and formation of the ranking order. More details to the evaluation process are given in Chapter 4.

Design phase

According to [12], design is a clear and complete elaboration of the building structure of a technical system according to technical and economic aspects, including verification. Based on the qualitative ideas, the quantitative design solution is determined. In contrast to the concept phase, the design phase is more iterative than creative. Through the iterations, adjustments can be made to correct some imperfections in the system.

Finishing phase

In short, this phase serves to prepare the entire documentation for production and operation [56]. All dimensions, tolerances, performance data, safety instructions and operating instructions must be defined [56]. Documents such as CAD models, technical drawings, production specifications, etc. result from this phase.

3.1.2 Development process for control system

Technical constructions and control systems are in many cases complementary, however, their development process are not the same. The development process of a control system can be divided in five main phases: establishment of control goals, plant modeling and analysis, controller design and analysis, controller simulation, controller implementation, and validation and verification.

Establishment of control goals

In this phase, the goals of control system should be defined. Under the consideration of the defined goals, requirements on the control system can be set. This helps to choose which kind of control techniques are to use to successfully accomplish the goals defined for the control system. Its also helps for the choice of actuators, sensors, and control board.

Plant modeling and analysis

In order to design a controller, the model of the system should be defined. This can be done by determining the dynamic equation of the plant using different modeling law like Lagrange equation, mesh rule, etc. For more complex systems, an identification can be carried out to approximate the plant dynamic. Depending on the first phase and the strategy used, the identification can be done offline or online. After the plant dynamic is derived, an analysis of it allows the designer to identify system properties that can be helpful for the controller design.

Controller design, simulation, and analysis

Based on the dynamic model of the plant and the requirements on the control system, a controller should be designed. The closed control-loop should then be analyzed in term of stability, performance, reference tracking, etc. Based on the dynamic model of the plant, a simulation environment for the controller can be created. This helps for a further analysis of the controller; and a first verification of the property for the closed-loop system. This concept is know as Model-in-the-Loop (MIL).

Controller implementation

In this phase, the designed controller should be implemented in form of physical system. The controller is flashed on a controller board. The controller board is responsible for reading sensor information. This information is then processed based on the implemented control algorithm, and commands are sent to actuators. This phase may require further steps in controller design depending on how the implementation is done. Usually a digitalization of continuous controllers is required. The digitalization may deteriorate the predicted properties of the closed-loop system, and thus requires an adjustment of the control design.

Validation and verification

This phase consists in testing the controller on the physical system to validate the predicted behavior. Some adjustments can still be made to refine the system behavior.

3.2 Vibration control

Nowadays, vibration control has become a very important topic in industries. Whether in the in automobile for passenger comfort, in the photography for better images. or in the production for better precision, vibration control gets a big attention.

3.2.1 Types of vibration control

Roughly spoken, all undesirable recurrent effects in a system can be described as vibrations. There are usually divided in three main categories: ground vibrations, acoustic vibrations and forces acting directly on the system. Ground vibrations refer to any source around the system that can cause mechanical vibrations like traffic around the system, wind etc. Acoustic vibrations are mostly related to sound or pressure fluctuation caused by any sources around the system. The last category is the forces acting directly on the system. There are usually inherent to the system and can come from motors or any rotating element of the system.

The main task of vibration control is to minimize the effect of vibration sources on sensitive equipments. Vibration control systems are categorized in three classes: passive, semi-active, and active vibration control.

Passive vibration control strategies consist of mounting passive elements on the system considered in order to intentionally influence some properties of the system. Usually, the passive elements are springs, damper, and additional masses (Figure 3.3). They are often limited to tackle high frequencies, single structural

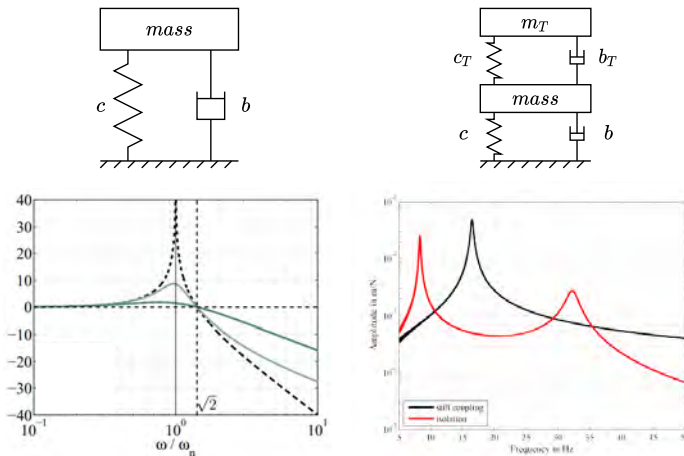


Figure 3.3: Example of passive vibration control configurations [37]

resonance or specific disturbance frequencies [11].

Semi-active vibration control systems are very similar to passive vibration control system in the sense that they both consists in adding passive elements in the structure. However, semi-active systems use passive elements with variable

properties (spring with variable stiffness, damper with variable damping ratio, etc.) and eventually different configurations (Figure 3.4). Some active elements are integrated to switch between the different system configurations and component properties. The switch mechanism between different configurations requires some

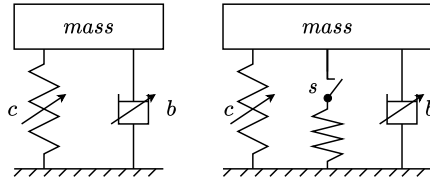


Figure 3.4: Example of semi-active vibration control configurations

energy. Therefore, an external energy source may be needed for semi-active control systems.

In active vibration control systems, sensors and actuators are integrated to the system to actively influence the system behavior (Figure 3.5). For this purpose, controllers are usually designed to generate control signals based on sensor signals. Depending on the design of the controller, we can distinguish between feedback and feedforward controllers. While feedback controllers use signals of sensors

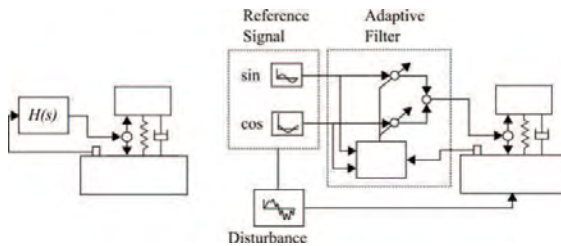


Figure 3.5: Feedback and feedforward control for active vibration control [11]

mounted on the surface to generate the control signal, feedforward controllers use signals from the noise sources and try to generate an opposite signal to cancel the effect of the noise sources on the surface to isolate. Active control strategies are usually applied for low frequencies and wide-band frequencies control.

3.2.2 Actuators for active vibration control

There are many types of actuators used to control vibrations. Depending on the application, three kinds of actuators are often employed: pneumatic,

electrodynamic and piezoelectric actuators.

Pneumatic actuators usually consist of two chambers and a valve that regulates the

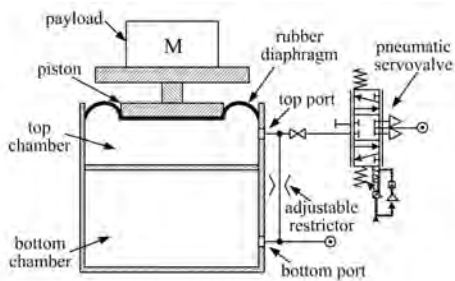


Figure 3.6: Pneumatic actuator [17]

air pressure in the chambers (Figure 3.6). By controlling the valve, the forces generated by the actuator can be specifically influenced. Pneumatic actuators can withstand large forces and displacements. They are often used in laboratories to isolate measurement equipments and installations.

Electrodynamic actuators (Figure 3.7) exploit the principle of Lorentz force. They are composed of a permanent magnet and a moving coil. That is why they are also known as voice-coil motors. When the coil is connected to a voltage source, a

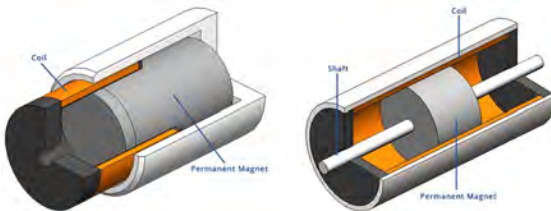


Figure 3.7: Electrodynamic actuator [77]

Lorentz force is generated, which is proportional to the flowing electrical current.

Piezoelectric transducers exploit the piezoelectric effect. Piezoelectric elements have the property that they deform when they are submitted to an electrical potential difference. Reciprocally, an electrical voltage is also induced when their body is deformed. The induced voltage is function of the deformation, and vice versa. Piezoelectric transducers can be deployed as actuators as well as sensors. Piezoelectric actuators (Figure 3.8) can generate very large forces and have very high dynamics, but the displacements generated are very small. Therefore, they are



Figure 3.8: Piezo actuator [29]

only used in applications where a few micrometers are sufficient. A well-known field of application for such sensors is optical precision measurement.

There are many other kinds of actuators for active vibration control like dielectric actuators, magnetostrictiv actuators, electromagnetic actuators etc. In his dissertation, Karsten [37] gave an overview of these actuators with their characteristics, their advantages and their drawbacks.

3.3 Control Strategies

There are several kinds of controllers used for vibration control. Due to their ability to react to non-predictable behavior, the focus is set on robust control and adaptive control strategies.

3.3.1 Robust control

The robustness of a controller describes its ability to deal with uncertainties in the system. Let's consider the classical feedback control loop shown in Figure 3.9. There are four main requirements on such a system. The most important is the stability, which protects different system components from potential damages. The second requirement is the good tracking of the reference signal, which is what the controller is basically made for. The requirement of good tracking performance leads to other requirements, which are the attenuation of process disturbances and the reduction

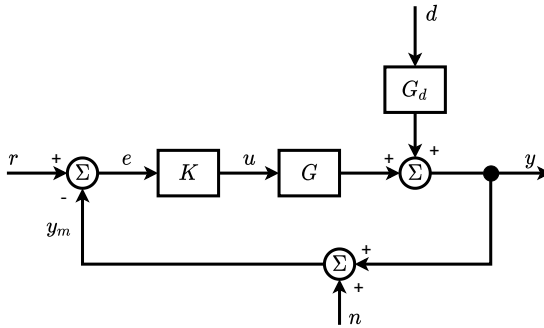


Figure 3.9: Classical feedback control system

of the effects of the measurement noise. Referring to the block diagram in Figure 3.9, the output y can be formulated as

$$\begin{aligned} y &= \frac{GK}{1+GK}r + \frac{G_d}{1+GK}d - \frac{GK}{1+GK}n \\ &= \frac{L}{1+L}r + \frac{G_d}{1+L}d - \frac{L}{1+L}n \\ &= T(s)r + S(s)G_d(s)d - T(s)n, \end{aligned}$$

where $L(s)$, $T(s)$ and $S(s)$ are respectively defined as loop transfer function, complementary sensitivity function, and sensitivity function. The error ε is formulated as

$$\varepsilon = r - y = S(s)r - S(s)G_d(s)d + T(s)n.$$

Ideally, the error canceled for $S(s) = 0$ and $T(s) = 0$. However, this is not possible due to the relation

$$T(s) + S(s) = \frac{L(s)}{1+L(s)} + \frac{1}{1+L(s)} = 1 \quad \Rightarrow \quad T(s) = 1 - S(s).$$

Now, let us considered the different input signals r , d and n . The reference signal r does not usually contains high frequencies respectively frequency below the so called crossover frequency. So, to cancel the term $S(s)r$, $S(s)$ only need to be very low for low frequencies. The measurement noise n usually comes from sensors and mainly contains high frequencies. For this reason, $T(s)$ only need to be low for high frequencies. The process noise d can contain low as well as high frequencies. The high frequencies of d are damped through the eigendynamic and the physical

limitation of the system. The low frequencies can be damped if $S(s)$ fulfills the requirements posed for $S(s)r$. These requirements on $S(s)$ and $T(s)$ can also be deduced from the formulation of the output signal y , which should ideally correspond to the reference signal r . The requirements on $T(s)$ and $S(s)$ can be

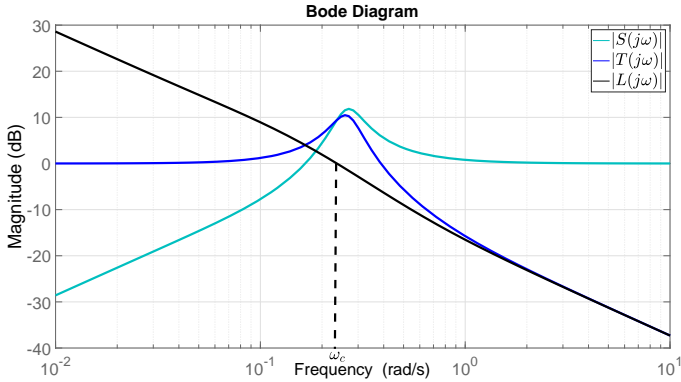


Figure 3.10: Requirements on $T(s)$ and $S(s)$

represented in a bode diagram as shown in Figure 3.10, and they lead to the representation of the desired loop transfer function $L(s)$ as shown in this Figure. While the low frequency design of $L(s)$ is responsible for the good performance and reference tracking, the high frequency design is responsible for the robustness of the controller. The transition interval from low to high frequencies contains the crossover frequency ω_c , and is a critical range, because $S(s)$ and $T(s)$ usually have their maximum magnitudes defined as

$$M_S = \max |S(j\omega)|, \quad M_T = \max |T(j\omega)|$$

in this interval. The higher M_S and M_T are, the worse their effects on the system are. This is well explained by Skogestad and Postlethwaite [69]. A geometric analysis of the Nyquist plot presented in [69] shows that

$$\begin{aligned} \text{GM} &\geq \frac{M_S}{M_S - 1}, & \text{PM} &\geq 2 \arcsin \left(\frac{1}{2M_S} \right) \geq \frac{1}{M_S} \quad \text{and} \\ \text{GM} &\geq 1 + \frac{1}{M_T}, & \text{PM} &\geq 2 \arcsin \left(\frac{1}{2M_T} \right) \geq \frac{1}{M_T}, \end{aligned}$$

where GM and PM are respectively the gain margin and the phase margin. So, M_T and M_S directly affect the lower bounds of stability margins. Moreover, the relationship

$$\frac{dT}{T} = S \frac{dG}{G}$$

holds, and shows that the effect the plant uncertainties on the whole system is amplified by the sensitivity function S . One of the classical methods used to find a controller G_c based on bode diagram so that the requirements on L are fulfilled is the loop shaping.

An approach of the loop shaping is to reshape the sensitivity function considering the so-called performance weighting function. This function is defined based on the requirements on the system. There are:

- Minimum bandwidth frequency ω_B^* ,
- Maximum tracking error at selected frequencies,
- Maximum steady-state tracking error A ,
- Shape of S over selected frequency ranges,
- Maximum peak magnitude M of S

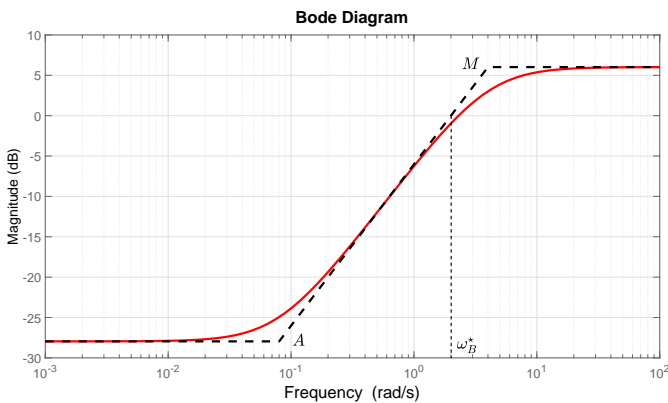


Figure 3.11: Bode plot of the inverse performance weight

[69]. A proposition of the performance weighting function is

$$W_p(s) = \frac{s/M + \omega_B}{s + \omega_B A}$$

and Figure 3.11 shows the bode plot of $1/|W_p(s)|$. The condition required for the nominal performance is that the magnitude of the weighted sensitivity function should be lower than one:

$$|W_p S| < 1 \quad \Leftrightarrow \quad |S| < \frac{1}{|W_p|}.$$

This means that the nominal performance is ensured if the loop transfer function $L(s)$ is reshaped such that the curve of the sensitivity function $S(s)$ is always below the curve of the inverse performance weight in the bode magnitude diagram.

Furthermore, the shape of the complementary sensitivity function can be specified using a weight W_t , and some requirements can be made on u with a specified weight W_u . All these specifications can be resumed as

$$\|N\|_\infty = \max_{\omega} \bar{\sigma}(N(j\omega)) < 1, \quad \text{with} \quad N = \begin{bmatrix} W_p S \\ W_t T \\ W_u K S \end{bmatrix},$$

where $\bar{\sigma}(N)$ is the maximum singular value of N , which can be expressed as

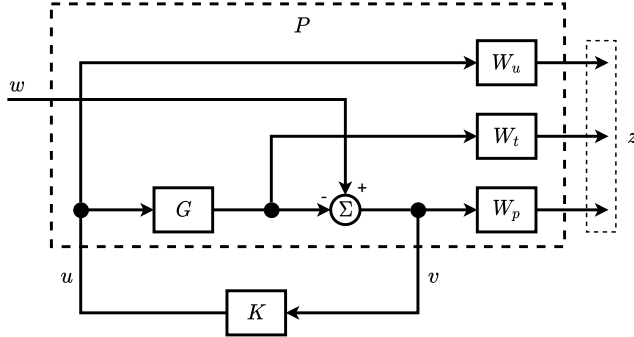
$$\bar{\sigma}(N) = \sqrt{|W_p S|^2 + |W_t T|^2 + |W_u K S|^2}$$

for SISO systems. From these requirements, a desired shape for the loop transfer function can be derived (Figure 3.12). The task of the \mathcal{H}_∞ control design is to find a controller K^* that minimizes $\bar{\sigma}(N)$:

$$N(K^*) = \min_K \|N(K)\|_\infty. \quad (3.1)$$

This type of \mathcal{H}_∞ design based on transfer function shaping is known as mixed-sensitivity \mathcal{H}_∞ control. This structure of a classical feedback system can be remodeled to a general control configuration with exogenous inputs and exogenous outputs as shown in Figure 3.13 with the formulation

$$\begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \begin{bmatrix} w \\ u \end{bmatrix}, \quad \text{with} \quad u = Kv.$$

Figure 3.13: Control structure for mixed-sensitivity \mathcal{H}_∞ [69]

Now let us consider an uncertain system $G(s)$. System uncertainties can be classified in two main categories: dynamic uncertainties and parametric uncertainties. While parametric uncertainties are related to uncertain parameters in a system, which structure is known, dynamic uncertainties are related to missing dynamic in the mathematical description of the system. Multiplicative dynamic uncertainties are one of the often used formulations to describe uncertain dynamics. With this class of uncertainties, the uncertain system is defined as

$$\mathcal{G} = \{G(s) : G(s) = G_0(s) [1 + \Delta W(s)], \quad \Delta \in \mathbb{C}, |\Delta| \leq 1\},$$

where $G_0(s)$ is the nominal plant. Assumed that a controller K stabilizes the nominal plant G_0 . The system is robust stable if K stabilizes all members of \mathcal{G} . This is the case if

$$|T(j\omega)W(j\omega)| < 1, \quad \forall \omega \quad \Leftrightarrow \quad |T(j\omega)| < \frac{1}{|W(j\omega)|},$$

$$\text{with } T(s) = G_0(s)K(s) [1 + G_0(s)K(s)]^{-1}.$$

Mackenroth [50] gives more details on the interpretation of the formulation in the Nyquist plot (Figure 3.14). The robust performance of the uncertain system controlled by K is guaranteed if the performance is guaranteed for all members of \mathcal{G}

$$|W_p(j\omega)S_p(j\omega)| < 1 \quad \Leftrightarrow \quad |S_p(j\omega)| < \frac{1}{|W_p(j\omega)|},$$

$$\text{with } S_p(s) = \frac{1}{1 + L_p(s)} = \frac{1}{1 + K(s)G_0(s) [1 + \Delta W(s)]}, \quad |\Delta| \leq 1.$$

According to Mackenroth [50], a necessary and sufficient theorem for the robust performance is

$$|W_p S| + |WT| < 1.$$

This theorem is graphically interpreted in Figure 3.14.

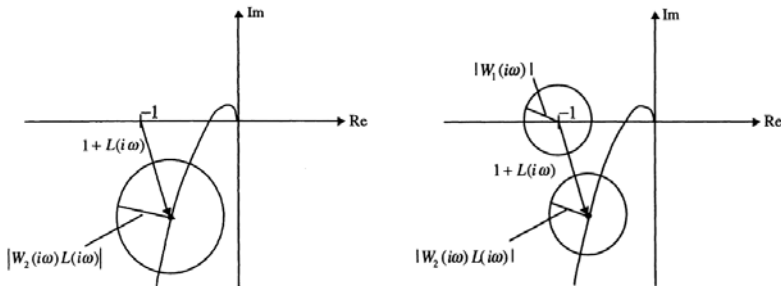


Figure 3.14: Robust stability (left) and robust performance (right) [50]

To define the concept of robust performance for MIMO systems, the so called M/Δ -structure (Figure 3.15) is required with

$$N = F_l(P, K) = P_{11} + P_{12}K(I - P_{22}K)^{-1}P_{21}, \quad \text{and} \quad M = N_{11}.$$

Assuming that the nominal system $M(s)$ and $\Delta(s)$ are stable, and the perturbations

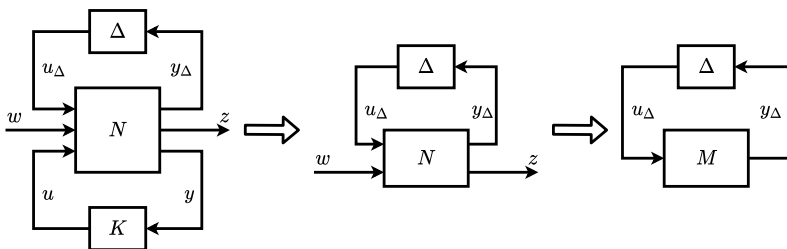


Figure 3.15: M/Δ -structure [69]

$c\Delta$ are allowed, with c defined as complex scalar with a norm lower than 1. Then, it can be derived that the M/Δ -structure is robust stable against perturbations if

$$\sigma_{\max}(M(j\omega)) < 1 \quad \Leftrightarrow \quad \|M\|_{\infty} < 1$$

[69]. By adding some blocks on the M/Δ -structure without modifying the structure itself (Figure 3.16), a least conservative robust stability condition is obtained as

$$\min_{D(\omega) \in \mathcal{D}} \sigma_{\max} (D(\omega)M(j\omega)D^{-1}(\omega)) < 1 \quad \forall \omega.$$

Another tool used to defined the robust stability of MIMO systems is the structured

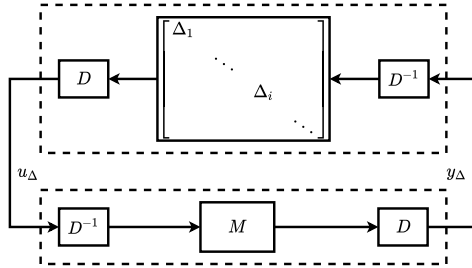


Figure 3.16: M/Δ -structure with D [69]

singular value. Denoted by μ , the structured singular value is defined as

$$\mu_{\underline{\Delta}}(M) = \frac{1}{\min\{\sigma_{\max} | \Delta \in \underline{\Delta}, \det(I - M\Delta) = 0\}},$$

if there is a perturbation $\Delta \in \underline{\Delta}$ such that $\det(I - M\Delta) = 0$. Otherwise $\mu_{\underline{\Delta}}(M) = 0$. [50]

The robust stability of a M/Δ -system is guaranteed if

$$\mu_{\underline{\Delta}}(M(j\omega)) < 1, \quad \forall \omega.$$

To define the robust performance, an additional fictitious uncertainty block Δ_p is added to the system (Figure 3.17). The system is said to be robust performant if the

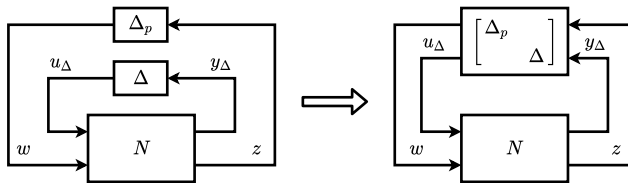


Figure 3.17: N/Δ -structure for robust performance analysis [69]

N/Δ' structure in Figure 3.17 is robust stable. This means

$$\mu_{\underline{\Delta}'}(M(j\omega)) < 1, \quad \text{with} \quad \Delta' = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta' \end{bmatrix}.$$

\mathcal{H}_∞ controller may fulfill the requirements for robust stability and nominal performance, however the robust performance is not guaranteed. In order to design a \mathcal{H}_∞ controller considering the requirements for robust performance, iterative methods known as μ -synthesis were developed.

The first iterative method is the D - K iteration method. In this method, the controller is designed by iteratively recomputing $D(j\omega)$ in order to get a better controller K . Gu, Petkov, and Konstantinov [33] presented this algorithm in four steps:

- **Step 1:** start with an initial guess for D ;
- **Step 2:** fix D and solve the \mathcal{H}_∞ optimization

$$K = \arg \inf_K \left\| F_l(\tilde{P}, K) \right\|_\infty \quad \text{with} \quad \tilde{P} = \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} D^{-1} & 0 \\ 0 & I \end{bmatrix};$$

- **Step 3:** fix K and solve the following convex optimization problem for D at each frequency over a selected frequency range

$$D(j\omega) = \arg \inf_{D \in \mathcal{D}} \sigma_{\max} [DF_l(P, K)D^{-1}(j\omega)];$$

- **Step 4:** Curve fit $D(j\omega)$ to get a stable, minimum-phase $D(s)$; go to Step 2 and repeat, until a prespecified convergence tolerance is achieved, or a prespecified maximum iteration number is reached

[33].

The second method is the μ - K iteration method. This method rather fits μ to increase the performance. In [33], this method is declined in five steps:

- **Step 1:** solve the \mathcal{H}_∞ optimization problem for K_0 ,

$$K_0 := \arg \inf_K \left\| F_l(\tilde{P}, K) \right\|_\infty;$$

- **Step 2:** compute the μ curve corresponding to K_0 over a chosen frequency

range,

$$\mu_0(j\omega) := \mu [F_l(P, K_0)(j\omega)];$$

- **Step 3:** normalize μ_0 by its maximum value, i.e.

$$\tilde{\mu}_0 := \frac{\mu_0}{\max_{\omega} \mu_0};$$

- **Step 4:** curve fit $\tilde{\mu}_0(j\omega)$ to get a stable, minimum-phase, rational $\tilde{\mu}_0(s)$;
- **Step 5:** solve for the \mathcal{H}_{∞} optimal controller $K_1(s)$,

$$K_1 := \arg \inf_K \left\| \tilde{\mu}_0 F_l(\tilde{P}, K) \right\|_{\infty}$$

go to Step 2, multiply the newly obtained μ curve function onto the previous cost function in last equation (e.g. $\left\| \tilde{\mu}_1 \tilde{\mu}_0 F_l(\tilde{P}, K) \right\|_{\infty}$); repeat until the μ curve is sufficiently flat or until the desired level of performance has been reached

[33].

3.3.2 Adaptive control

While robust control strategies are made to design controllers, which remain effective even if the system properties are not well-known, adaptive control strategies are made to design controllers, which can adapt their parameters to the changing system behavior. One of the very popular and most effective strategies used in adaptive control is the active noise canceling (ANC) strategy. ANC is a class of control that is very popular in acoustic applications. It consists in generating a secondary noise, which should destructively interfere with the primary noise in order to cancel it (Figure 3.18). ANC is usually applied in form of feedforward control, but in some applications it can also be designed as feedback control (Figure 3.19). While feedback control is widely applied for vibration control, it is not very suitable for some kind of systems like those with persistent excitations. Since the control signal strongly depends on the error signal, it may decrease if the error decreases. However, the persistent excitation has not changed and a high control signal may still be needed to decrease the error signal. Thus, a contradiction appears, and this may lead to instability or performance limitations in the system.

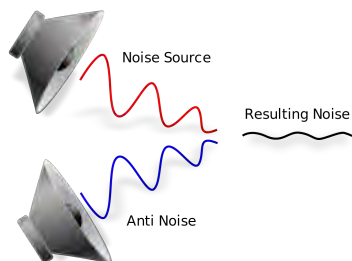


Figure 3.18: Destructive interference [2]

ANC controllers can be categorized as fixed or adaptive. Fixed ANC controllers are like their name indicates static controllers. They are designed based on the system model and the environment in which they will be deployed. Fixed ANC controllers have the advantages of having simple algorithms, an easy and compact implementation, a low power consumption, etc. These characteristics made ANC-headsets as one of the main applications of fixed ANC controllers. However, fixed ANC controllers are only designed for linear time-invariant systems and are not robust against uncertainties and changes in the system and in the environment. In order to handle system uncertainties and time-variant processes, adaptive ANC controllers were adopted.

Adaptive ANC controller design is done with adaptive filters. The primary noise goes through the primary path and generates the signal to cancel. This primary noise is also fed into an adaptive filter which generates a control signal that goes through the secondary path to generate the signal that should cancel the one coming from the primary path (Figure 3.20).

Adaptive FIR-filter

The finite impulse response (FIR) is one of the most used filter architectures in adaptive digital signal processing. As shown in Figure 3.21, it is commonly realized as tapped delay line. [34]

A discrete FIR filter of order N is defined such that the output of the filter is a

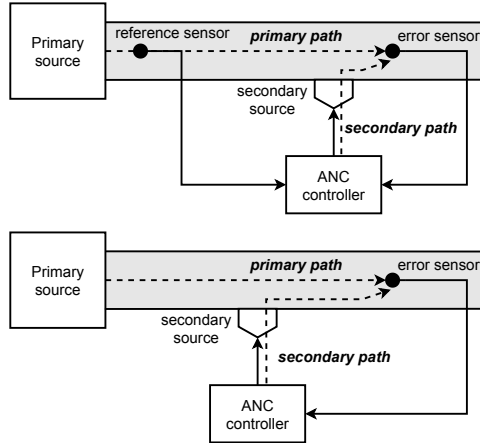


Figure 3.19: ANC feedforward (above) and feedback (below) [14]

weighted sum of the actual input and the N most recent inputs.

$$\overline{y(k)} = \sum_{n=0}^{N-1} w_n(k)x(k-n) = \mathbf{w}^T(k)\mathbf{x}(k) = \mathbf{x}^T(k)\mathbf{w}(k) \quad \text{with}$$

$$\mathbf{x}(k) = \begin{bmatrix} x(k) & x(k-1) & \dots & x(k-N+1) \end{bmatrix}^T \quad \text{and}$$

$$\mathbf{w}(k) = \begin{bmatrix} w_0(k) & w_1(k) & \dots & w_{N-1}(k) \end{bmatrix}^T$$

After applying the \mathcal{Z} -transform, discrete FIR filters provide the following transfer function formulation

$$\frac{Y(z)}{X(z)} = w_0 + w_1 z^{-1} + \dots + w_{N-1} z^{-(N-1)}.$$

Transfer functions of FIR filters have only zeros and no poles. From this follows the most important characteristic of FIR filters, namely the inherent stability.

The adaptive filtering found its main application is system identification. The goal is to find the optimal weight vector $\mathbf{w}(k)$ minimizing the error $e(k)$ between the output $d(k)$ of an unknown system and the output of the adaptive filter $y(k)$ at each time step k .

$$e(k) = d(k) - y(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$$

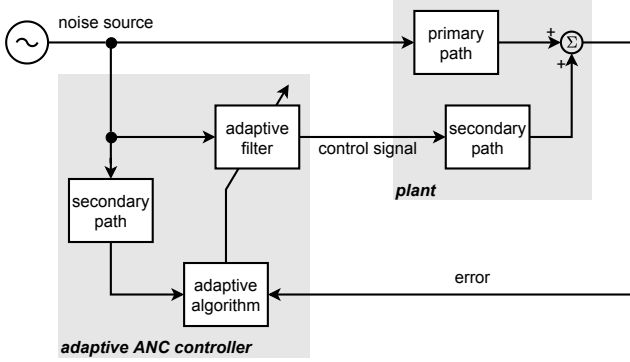


Figure 3.20: ANC with adaptive filtering [34]

For this purpose, the mean square error $\xi(k)$ defined as the expected value of the squared error $e(k)$ is defined as the cost function to minimize:

$$\begin{aligned}
 J(k) &= \xi(k) = \mathbb{E} \{ e^2(k) \} \\
 &= \mathbb{E} \{ d^2(k) \} + \mathbf{w}(k)^T \mathbb{E} \{ \mathbf{x}(k) \mathbf{x}^T(k) \} \mathbf{w}(k) - 2d(k) \mathbf{w}^T(k) \mathbf{x}(k) \\
 &= \sigma_d^2 + \mathbf{w}^T(k) \mathbf{R}_{xx} \mathbf{w}(k) - 2\mathbf{w}^T \mathbf{p}_{du}, \tag{3.2}
 \end{aligned}$$

with $\mathbf{R}_{xx} = \mathbf{R}$ as input signal auto-correlation and $\mathbf{p}_{du} = \mathbf{p}$ as cross-correlation between the input and the output to estimate. Making the assumption that variables are equal to their expected values changes the cost function into a deterministic function [34]. Equation 3.2 shows that the cost function is linear quadratic. The most important property of linear quadratic functions in optimization problems is that they are unimodal. This means that they only have one extremum which is the global minimum in this case. So, the optimal weight vector can be computed by a simple differentiation of the cost function with respect to the weight vector, and its formulation is known as discrete form of the solution to the Wiener-Hopf integral equation.

$$\begin{aligned}
 \frac{\partial J}{\partial \mathbf{w}} &= 2\mathbf{R}\mathbf{w} - 2\mathbf{p} \stackrel{!}{=} 0 \quad \Rightarrow \quad \mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p} \\
 \frac{\partial^2 J}{\partial \mathbf{w}^2} &= 2\mathbf{R} > 0
 \end{aligned}$$

The minimal cost is then defined as

$$J(\mathbf{w} = \mathbf{w}_{opt}) = \sigma_d^2 + \mathbf{p}^T \mathbf{R}^{-1} \mathbf{R} \mathbf{R}^{-1} \mathbf{p} - 2\mathbf{p}^T \mathbf{R}^{-1} \mathbf{p}$$

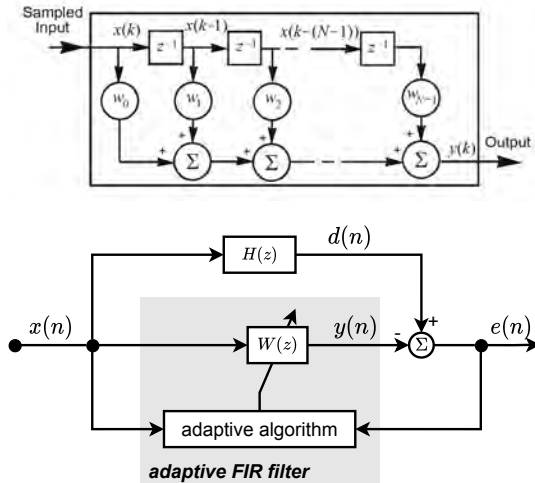


Figure 3.21: Tapped-delay structure of FIR filter (above) and adaptive FIR-filter (below) [34]

$$= \sigma_d^2 - \mathbf{p}^T \mathbf{w}_{opt}.$$

This minimal cost describes the optimal squared error that the filter may achieve (Figure 3.22). It is usually a non-zero value and may results from some non-correlations in the system or non-sufficient filter length. The implementation

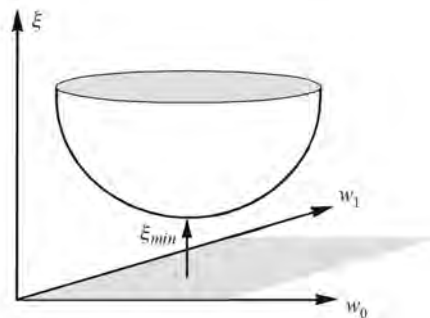


Figure 3.22: Cost function [34]

of the optimal weight vector as defined as solution of the Wiener-Hopf integral equation faces many problems such as the numerical difficulty of inverting a matrix,

the changes in systems, and the average values required. To overcome this problem, numerical algorithms are considered to find the optimal weight vector \mathbf{w}_{opt} . Since the cost function is linear quadratic, a simple gradient descent algorithm is sufficient to find the global minimum.

The basic gradient descent algorithm is based on the following formulation

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \cdot \Delta \mathbf{w}(k), \quad \text{with} \quad \Delta \mathbf{w}(k) = \frac{\partial J(k)}{\partial \mathbf{w}(k)},$$

where μ is the step-size. Considering the assumption that made the cost function function deterministic, we have:

$$\begin{aligned} J(k) &= \xi(k) = \mathbb{E} \{e^2(k)\} \approx e^2(k) = (d(k) - \mathbf{w}^T(k)\mathbf{x}(k))^2 \\ \Rightarrow \Delta \mathbf{w}(k) &= \frac{\partial J(k)}{\partial \mathbf{w}(k)} = -2 (d(k) - \mathbf{w}^T(k)\mathbf{x}(k)) \mathbf{x}(k) \\ &= -2e(k)\mathbf{x}(k). \end{aligned}$$

Thus, the following rule results for weight vector update

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k),$$

also known as least mean square or LMS algorithm.

Adaptive IIR-Filter

As their name indicates, infinite impulse response (IIR) filters have in contrast to FIR filters infinite impulse response. Their output is usually defined as a weighted sum of the actual and the recent inputs, and the recent outputs in form of

$$\begin{aligned} y(k) &= \sum_{n=0}^{N-1} a_n x(k-n) + \sum_{m=1}^{M-1} b_m y(k-m) \\ &= \mathbf{a}^T(k)\mathbf{x}(k) + \mathbf{b}^T(k)\mathbf{y}(k-1) = \mathbf{w}^T(k)\mathbf{u}(k) \end{aligned}$$

with

$$\begin{aligned} \mathbf{w}^T &= [\mathbf{a}^T \quad \mathbf{b}^T] = [a_0 \quad a_1 \quad \dots \quad a_{N-1} \quad b_1 \quad b_2 \quad \dots \quad b_{M-1}] \quad \text{and} \\ \mathbf{u}(k) &= [\mathbf{x}^T(k) \quad \mathbf{y}^T(k-1)] \\ &= [x(k) \quad x(k-1) \quad \dots \quad x(k-N+1) \quad y(k-1) \quad \dots \quad y(k-M+1)]. \end{aligned}$$

IIR filters can be interpreted as combination of two FIR filters, namely one for the

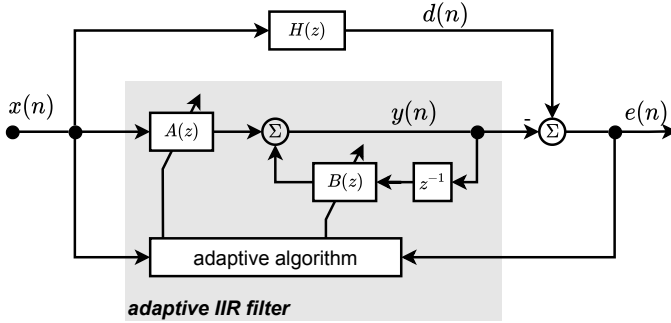


Figure 3.23: Adaptive IIR-filter

input processing and one for the output feedback (Figure 3.23). From \mathcal{Z} -transformation follows the formulation

$$\frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + \dots + a_{N-1} z^{-(N-1)}}{1 - b_1 z^{-1} - b_2 z^{-2} - \dots - b_{M-1} z^{-(M-1)}}$$

as transfer function. Unlike FIR filters, IIR filters have poles. This makes IIR filters more suitable for identification of feedback or resonant systems using a small number of coefficients. However, stability issues may occur during the adaption.

Using the same formulation as for FIR for the cost function, it follows

$$\begin{aligned} e(k) &= d(k) - \mathbf{w}^T(k) \mathbf{u}(k) \\ J(k) &= \sigma_d^2 + \mathbf{w}^T(k) \mathbf{R}_{uu} - 2\mathbf{w}^T \mathbf{p}_{du}, \end{aligned}$$

with

$$\begin{aligned} \mathbf{R}_{uu} &= \begin{bmatrix} \mathbb{E} \{ \mathbf{x}(k) \mathbf{x}^T(k) \} & \mathbb{E} \{ \mathbf{x}(k) \mathbf{y}^T(k-1) \} \\ \mathbb{E} \{ \mathbf{y}(k-1) \mathbf{x}^T(k) \} & \mathbb{E} \{ \mathbf{y}(k-1) \mathbf{y}^T(k-1) \} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{xx} & \mathbf{R}_{xy} \\ \mathbf{R}_{xy}^T & \mathbf{R}_{yy} \end{bmatrix} \quad \text{and} \\ \mathbf{p}_{du} &= \begin{bmatrix} \mathbb{E} \{ d(k) \mathbf{x}(k) \} \\ \mathbb{E} \{ d(k) \mathbf{y}(k-1) \} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{dx} \\ \mathbf{p}_{dy} \end{bmatrix}. \end{aligned}$$

However, the cost function in this case is not linear quadratic with respect to \mathbf{w} . This is due to the term \mathbf{y} in the auto-correlation matrix \mathbf{R}_{uu} , because \mathbf{y} depends on \mathbf{w} , and so introduces nonlinearity in the cost function. This leads to the fact that the existence of only one extremum is not guaranteed anymore, and the cost function may have local minima and so be multinodal (Figure 3.24). Find the optimal solution

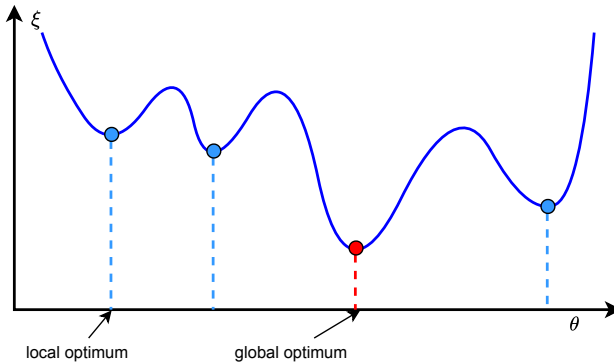


Figure 3.24: Multimodal function

analytically is more difficult than for FIR filter. In fact, the equation

$$\frac{\partial J}{\partial \mathbf{w}} = 0$$

does not have the solution to the Wiener-Hopf integral equation as solution anymore. Therefore, gradient-based techniques are used to find a solution to the problem. The application of the LMS algorithm to adaptive IIR filters gives:

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu \cdot \Delta \mathbf{w}(k) \quad \text{with} \\ \Delta \mathbf{w}(k) &= \frac{\partial J(k)}{\partial \mathbf{w}(k)} \approx \frac{\partial e^2(k)}{\partial \mathbf{w}(k)} = -2e(k) \frac{\partial y(k)}{\partial \mathbf{w}(k)}, \\ \frac{\partial y(k)}{\partial \mathbf{w}(k)} &= \left[\frac{\partial y(k)}{\partial a_0(k)} \quad \cdots \quad \frac{\partial y(k)}{\partial a_{N-1}(k)} \quad \frac{\partial y(k)}{\partial b_1(k)} \quad \cdots \quad \frac{\partial y(k)}{\partial b_{M-1}(k)} \right]^T \\ &= \left[\alpha_0(k) \quad \cdots \quad \alpha_{N-1}(k) \quad \beta_1(k) \quad \cdots \quad \beta_{M-1}(k) \right]^T \quad \text{and} \\ \alpha_n(k) &= \frac{\partial y(k)}{\partial a_n(k)} \approx x(k-n) + \sum_{j=1}^{N-1} b_j(k) \frac{\partial y(k-j)}{\partial a_n(k-j)} \\ \beta_m(k) &= \frac{\partial y(k)}{\partial b_m(k)} \approx y(k-m) + \sum_{j=1}^{M-1} b_j(k) \frac{\partial y(k-j)}{\partial b_m(k-j)}. \end{aligned}$$

This algorithm is known as full gradient IIR LMS. By choosing small step-size, the assumption of slow varying weights can be made for the simplified gradient IIR LMS which leads to

$$B_k(z) \approx B_{k-1}(z) \approx \cdots \approx B_{k-M+1}(z) \approx \cdots \approx B_{k-N+1}(z) \quad \text{with}$$

$$B_k(z) = b_1z^{-1} + b_2z^{-2} + \dots + b_{M-1}z^{-M+1}.$$

The Feintuch's IIR LMS makes further simplifications of the model with the assumption that all derivatives of past outputs with respect to current weights are zero. That leads to

$$\alpha_n(k) \approx x(k-n) \quad \text{and} \quad \beta_m(k) \approx y(k-m).$$

The resulting formulations for the coefficient update of IIR filters are similar to the coefficient update of FIR filters

$$\begin{aligned} \mathbf{a}(k+1) &= \mathbf{a}(k) + 2\mu e(k)\mathbf{x}(k), \\ \mathbf{b}(k+1) &= \mathbf{b}(k) + 2\mu e(k)\mathbf{y}(k-1). \end{aligned}$$

Despite the simplifications and assumptions made while deriving the Feintuch's IIR LMS to reduce the needed computational effort, it appears to be more stable and more efficient except for some rare cases.

Adaptive filters for ANC applications can be designed using these basic principles. More to this will be developed in Chapter 6.

3.3.3 Central and decentral control

Many control algorithms can be used for platform isolation. Depending on how the algorithms are used and applied to the system, one can distinguish between central and decentral control. The concept of central and decentral control cannot be used for every systems. This concept only makes sense when the system is of type MIMO (multi-input multi-output), and each actuator respectively each group of actuator has a predominant influence on one output or a group of outputs.

Central control consists in designing a "central" controller for the whole system. All data needed for control are processed in the control unit as one task. In case of Gimbal i.e., the controller receives the actual roll, pitch and yaw data, and can compute the control inputs for the different motors (Figure 3.25).

In case of decentral control, different "decentral" controllers are designed to run in parallel. Each decentral controller computes the control inputs for a group of actuators to control specific outputs of the system. Considering a gimbal system, three motors are usually integrated to the system and each of these has a major influence on a specific rotation direction (Figure 3.26).

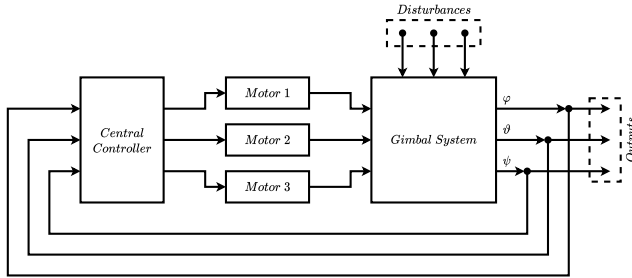


Figure 3.25: Central control for Gimbal

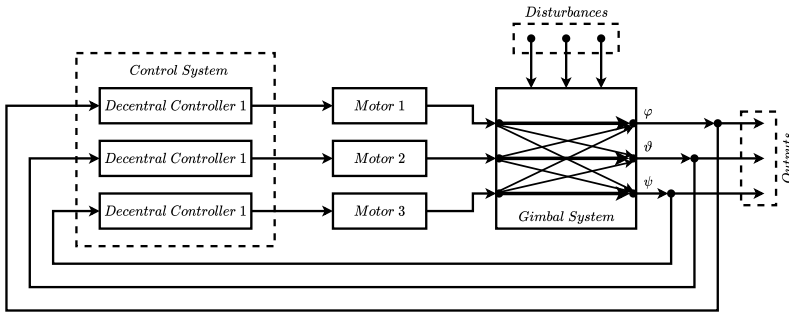


Figure 3.26: Decentral control for Gimbal

Decentral control should be applied on systems whose dynamic can be easily decoupled. The advantage is that the decoupled configuration are easier to handle, and it is well suited for several control algorithms. However, to obtain such a structure in the dynamic, some effects are usually neglected. In some cases, the system is already split in subsystems for decentral control before the dynamic is derived. This is the case for vehicle suspensions, where the quarter vehicle model is exploited to calculate the dynamic of a single suspension and to design a decentral controller for this suspension. But the neglected effects, which was not taken into account while designing the decentral controllers, influence the system and can lead to instability if the designed controllers are not robust enough. This phenomenon has less impact in a central control since the coupling terms between the main channels were considered in the controller design. However, considering these terms limits the possibility of designing certain classes of controllers.

3.4 Optimization algorithms

Whether it is linear or nonlinear, an optimization problem can generally be described as

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad f : \mathbb{X} \rightarrow \mathbb{Y}.$$

In case of constrained optimization, restrictions formulated as equality or inequality constraints are added to the optimization problem:

$$\begin{aligned} h_i(\mathbf{x}) &= 0, \quad i = 1, \dots, m, \\ g_j(\mathbf{x}) &\leq 0, \quad j = 1, \dots, p. \end{aligned}$$

$f(\mathbf{x})$ is usually called cost function. How the cost function is defined depends on the objective of the optimization problem. But, it usually has a quadratic form.

The point $\mathbf{x}^* \in \mathbb{X}$ is called local minimum of $f(\mathbf{x})$ if

$$\exists \varepsilon > 0 \quad / \quad f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \quad \mathbf{x} \in \mathbb{X} \setminus \{\mathbf{x}^*\}, \quad \|\mathbf{x} - \mathbf{x}^*\| < \varepsilon,$$

and global minimum of $f(\mathbf{x})$ if

$$f(\mathbf{x}^*) \leq f(\mathbf{x}) \quad \forall \quad \mathbf{x} \in \mathbb{X} \setminus \{\mathbf{x}^*\}.$$

A cost function can have many local minima but only one global minimum. There are no methods yet, to check if a minimum is global. However, in case of convex optimization problem, there is only one minimum which is the global minimum.

An optimization problem is convex if the feasible set \mathbb{X} is convex and the cost function f is convex on \mathbb{X} .

The set \mathbb{X} is convex if

$$(1 - \mu)\mathbf{x} + \mu\mathbf{y} \in \mathbb{X}, \quad \forall \quad \mathbf{x}, \mathbf{y} \in \mathbb{X}, \quad \mu \in (0,1).$$

The function $f : \mathbb{X} \rightarrow \mathbb{Y}$ is convex on \mathbb{X} if

$$f((1 - \mu)\mathbf{x} + \mu\mathbf{y}) \leq (1 - \mu)f(\mathbf{x}) + \mu f(\mathbf{y}), \quad \forall \quad \mathbf{x}, \mathbf{y} \in \mathbb{X}, \quad \mu \in (0,1).$$

To solve constrained optimization problem analytically, the Lagrangian function

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) + \sum_{j=1}^p \mu_j g_j(x) = f(x) + \boldsymbol{\lambda}^T \mathbf{h}(x) + \boldsymbol{\mu}^T \mathbf{g}(x) \quad [6]$$

is defined. The parameters λ_i are called Lagrange multiplier and μ_j Karush-Kuhn-Tucker (KKT) multiplier. The problem is solved using the KKT optimality conditions for constrained optimization problems.

Karush-Kuhn-Tucker Conditions: $\mathbf{x}^* \in \mathbb{X}$ a local minimizer of the constraint optimization problem, then there exist a Lagrange multiplier $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and a Karush-Kuhn-Tucker multiplier $\boldsymbol{\mu}^* \in \mathbb{R}^p$ such that

$$\begin{aligned} \nabla_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= \nabla f(\mathbf{x}^*) + \mathbf{J}_{\mathbf{h}}^T(\mathbf{x}^*) \boldsymbol{\lambda}^* + \mathbf{J}_{\mathbf{g}}^T(\mathbf{x}^*) \boldsymbol{\mu}^* = 0, \\ \nabla_{\boldsymbol{\lambda}} L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) &= \mathbf{h}(\mathbf{x}^*) = 0, \\ \mathbf{g}(\mathbf{x}^*) &\leq 0, \\ \mathbf{g}^T(\mathbf{x}^*) \boldsymbol{\mu}^* &= 0, \\ \boldsymbol{\mu}^* &\geq 0. \end{aligned}$$

These equations are first-order necessary conditions for a local minimum. The second-order necessary respectively sufficient conditions are that a regular and feasible point \mathbf{x}^* is a local minimizer if the necessary first-order conditions for local minimum are fulfilled, and

$$\mathbf{H}_L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \geq 0 \quad \text{respectively} \quad \mathbf{H}_L(\mathbf{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) > 0$$

subject to the constraints

$$\mathbf{J}_{\mathbf{h}}(\mathbf{x}^*) \delta \mathbf{x} = 0, \quad \mathbf{J}_{\mathbf{g}^a}(\mathbf{x}^*) \delta \mathbf{x} = 0,$$

where \mathbf{g}^a denotes the active inequality constraints.

These optimality conditions are just applicable for problem formulations which have differentiable cost functions. This is not always the case. For problems that do not have differentiable cost functions, there are metaheuristic methods. Metaheuristic methods belong to the class of gradient-free optimization procedures. Some of these algorithms are presented in the next subsection.

3.4.1 Biologically-inspired algorithms

For several years, nature has been a source of inspiration for various fields of engineering. There is even a field of engineering sciences called Bionic, that is mainly dedicated to apply the principles of nature to technical systems. For example, the structure of the skeleton of human and various animals is exploited in robotics (human-like robots, insect-like robot, etc.). The functioning principles of the brain are used as basics of neural networks, which are used in machine learning, deep learning, reinforcement learning, etc. For optimization problems e.g., evolutionary theory has inspired evolutionary algorithms, bird flock has inspired swarm optimization, etc.

Evolutionary algorithms are stochastic biologically inspired optimization approaches to solve optimization problems iteratively (Figure 3.27). The

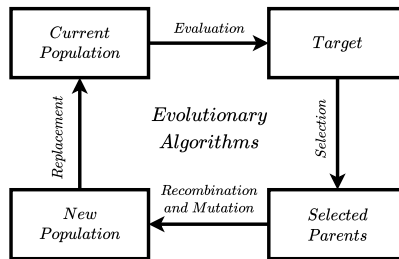


Figure 3.27: Principle of evolutionary algorithms

evolutionary theory has three main principles: the natural selection of the best individuals of the current population, the inheritance to preserve the genes of the best individuals, and the mutation to maintain the diversity in the new population. Evolutionary algorithms are categorized in four classes. One of these classes is genetic algorithm.

Genetic algorithms

Genetic algorithms were developed in the United States under the leadership of John Holland and his students [78]. These algorithms are made of seven steps. The basic procedure is:

Step 1: Population creation

To start a genetic algorithm, a population of N individuals is created for the first

generation. An individual is a candidate solution for the optimization problem. The individuals are chosen randomly in the definition domain. Depending on the problem, an individual can be represented as a binary string or as a vector of real numbers:

$$x = 10001101111, \quad \text{or} \quad x = \left(1 \quad -1.3 \quad 0 \quad 5 \quad \sqrt{2}\right)^T.$$

Step 2: Fitness function evaluation

The goal is to find an individual respectively the parameters, which will fit our application the best. To judge how good an individual fit to the problem, a scalar field called fitness function is defined. This function is evaluated for each individual of the current generation.

Step 3: Selection

A pair of individuals from the current population considered as parents for the next generation are selected based on their fitness values. The idea is to transmit good genes of the current generation to the next generation. Three widely used methods for selection are the roulette wheel method, the rank based method, and the tournament selection method. The common point of these methods is the high probability to have individuals with good performances as parents.

Step 4: Crossover

The Crossover is the most important step of the genetic algorithm. In this Step the chosen parents are combined to generate individuals for the next generation (offsprings). This is done with a specified crossover rate. There are several crossover techniques. The most known are the single-point crossover and the multi-point crossover. They consist in dividing an individual in two or multiple parts and to swap the parts. The resulting individuals are the offsprings.

$$\begin{array}{ll} \text{parent 1: } p_1 = a b c d & \text{offspring 1: } o_1 = a f c h \\ \text{parent 2: } p_2 = e f g h & \text{offspring 2: } o_2 = e b g d \end{array}$$

For real-coded genetic algorithms, The crossover is done by a linear combination of the parents with a randomly chosen real parameter α .

$$\begin{array}{ll} \text{parent 1: } p_1 = \underline{x} & \text{offspring 1: } o_1 = \alpha \underline{x} + (1 - \alpha) \underline{y} \\ \text{parent 2: } p_2 = \underline{y} & \text{offspring 2: } o_2 = (1 - \alpha) \underline{x} + \alpha \underline{y} \end{array}$$

Step 5: Mutation

In this step, one or multiple elements of an offspring are changed with a specified mutation rate. The candidates for the mutation are randomly chosen. In case of binary-string individuals, the selected elements just flipped from 1 to 0 or from 0 to 1. In case of real-vector individuals, the selected elements are replaced by random values.

Step 6: Replacement

The steps 3, 4 and 5 is repeated until the number of offsprings required for the next generation is reached. Then, the current population is replaced by the new generation.

Step 7: Stopping criteria

If one of the stopping criteria is fulfilled, the algorithm ends. Otherwise, the algorithm goes to Step 3.

Particle swarm optimization

Particle swarm optimization (PSO) is one of the most popular population-based algorithms inspired from the social behavior of animals. The swarm optimization is based on the cooperative behavior of animals. Each individual in the swarm communicates with others in such a way to find the food in the shortest time and way [25]. Each individual uses its own knowledge (cognitive part) and the global knowledge of the swarm (social part) to make the next movements.

In the swarm, each individual i at the step k has a position $x_{(i)}(k)$ and a velocity $v_{(i)}(k)$. The goal of the algorithm is to find the best position $x_{(i)}$, minimizer of the fitness function $f(x_{(i)})$. The next position $x_{(i)}(k+1)$ of an individual depends the current position $x_{(i)}(k)$ of the individual and its velocity $v_{(i)}(k+1)$ at the next step:

$$x_{(i)}(k+1) = x_{(i)}(k) + v_{(i)}(k+1).$$

The equation above describes the iteration step needed in many optimization algorithms. The intelligent part of swarm optimization algorithms is in the way the velocity $v_{(i)}(k+1)$ is calculated. The velocity $v_{(i)}(k+1)$ of the an individual at the next step depends on its velocity $v_{(i)}(k)$ at the current step, as well as its best position $p_{(i)}(k)$ reached so far in term of cost function called particular best, and the

best position reached by the swarm so far $g(k)$ called global best. It is defined as

$$v_{(i)}(k+1) = wv_{(i)}(k) + c_1\epsilon_1 (p_{(i)}(k) - x_{(i)}(k)) + c_2\epsilon_2 (g(k) - x_{(i)}(k)),$$

where w is the inertia coefficient, c_1 and c_2 are respectively cognitive and social weighting parameters, and ϵ_1 and ϵ_2 are random numbers in the interval $(0,1)$ (Figure 3.28). As well as for the particles positions, constraints can also be defined

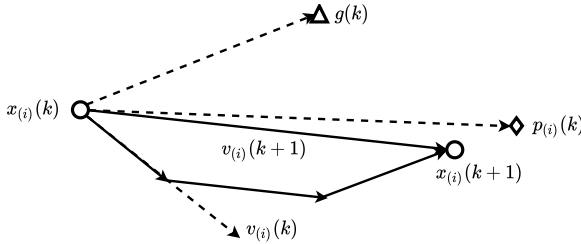


Figure 3.28: Particle swarm optimization: position update

for the velocity. This is called v_{max} method. It prevents from too fast changes of particles positions in the swarm which can lead to instability of the algorithm.

In order to reach a better convergence, the constriction method can also be used. In this method, a so called constriction factor χ defined as

$$\chi = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \quad \varphi = c_1 + c_2,$$

is applied to the velocity

$$v_{(i)}(k+1) = \chi (wv_{(i)}(k) + c_1\epsilon_1 (p_{(i)}(k) - x_{(i)}(k)) + c_2\epsilon_2 (g(k) - x_{(i)}(k))).$$

Analyses of this method presented by Bratton and Kennedy [15] have shown that the convergence has a better behavior for $\varphi > 4$. Usually following values are used:

$$c_1 = 2.05, \quad c_2 = 2.05 \quad \Rightarrow \quad \chi = 4.1.$$

Particle swarm optimization is known to be very powerful. Based on the basic PSO algorithm, many other biologically-inspired algorithms using the swarm intelligence principle were developed. Some of them are firefly algorithms, ant colony algorithms, bee colony algorithms, cuckoo search algorithms etc.

3.4.2 Generalized pattern search optimization

Generalized pattern search optimization (GPSO) algorithms are one of the well known and powerful heuristic optimization algorithms. According to Audet [7], the original algorithm was described by Lewis and Torczon [45].

Algorithm 1 Generalized pattern search algorithm

- 1: randomly choose n candidates $x_{(i)}$ such that $\|x_{(i)} - x_{(0)}^{(k)}\| = \Delta_k$
 - 2: find $x_{(p)}$ such that $f(x_{(p)}) \leq f(x_{(i)}) \forall i = 1 \dots n$
 - 3: **if** $f(x_{(p)}) \leq f(x_{(0)}^{(k)})$ **then**
 - 4: $x_{(0)}^{(k+1)} = x_{(p)}$
 - 5: $\Delta_{k+1} = \Delta_k$
 - 6: **else**
 - 7: $x_{(0)}^{(k+1)} = x_{(0)}^{(k)}$
 - 8: $\Delta_{k+1} = w \cdot \Delta_k, \quad w < 1$
 - 9: **end if**
 - 10: $k = k + 1$
 - 11: **if** break condition fulfilled **then**
 - 12: break
 - 13: **else**
 - 14: goto 1
 - 15: **end if**
-

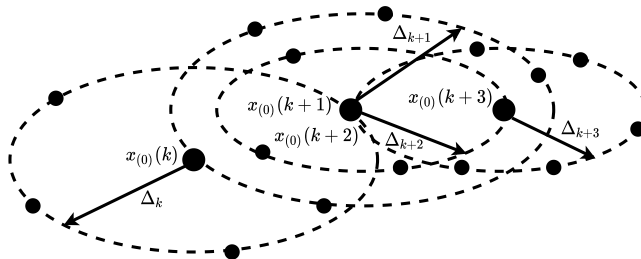


Figure 3.29: Global pattern search optimization

The main idea of GPSO algorithms consists in a random selection N candidates $x_{(i)}$ around the current minimizer $x_{(0)}$ in a radius Δ and evaluating their costs. If the cost function of the candidate with the smallest cost is smaller than that of the current minimizer, then it is chosen as new minimizer. Otherwise, the search radius

Δ is scaled down. Then the process starts against from the beginning until one of the termination conditions is fulfilled. Figure 3.29 shows a graphical interpretation of this algorithm.

The algorithms presented in this section have this point in common that the gradient of the objective function is not needed. This extends the application of these algorithms to many optimization problems that cannot be solved with gradient-based methods. However, the computational cost of these algorithms are high and their convergence behavior is not unique.

3.5 Summary

This Chapter presented the basics for what is handled in the next Chapters. First, development processes were presented. The focus were on technical products and control systems. Then, techniques for deriving the equation of motion of dynamic systems are presented.

After that, the theme vibration control was discussed. Two control strategies were presented: robust control and active noise canceling. They are both known for their robust behavior against uncertainties. However, while robust controllers are fixed controllers that can tolerate uncertainties in the system, active noise controllers adapt their behavior to the changes in system.

The last part of this chapter was about optimization algorithms. The attention was more on heuristic gradient-free optimization methods, because they can cover a wider range of problems. In total three types of algorithms were presented: genetic algorithms, particle swarm optimization algorithms, and general pattern search optimization algorithms.

The next step in this thesis handles the mechanical design of the isolation platform.

4 Isolation platform design

As stated in Chapter 1, the main objective of this thesis is to design an isolation platform for data acquisition on UAVs. The objective of this thesis is to find a good mechanical design for the isolation platform. Therefore, we will go through different steps of the product development process (PDP) as presented in Chapter 3.

4.1 Specification phase

The primary problem was the development of a framework for active 3D reconstruction. Therefore, image-based 3D reconstruction solutions were chosen. They use digital cameras which are lightweight, have low power-consumption and high resolution. Furthermore, there are already computer vision algorithms for 3D-reconstruction available, and they are cost-effective and efficient for large-scale reconstruction tasks. To acquire images to feed in the reconstruction algorithm, UAVs are chosen as platform for the scanning system. They are namely easy to maneuver, have a good stability behavior, and are not expensive.

As UAV for data acquisition the hexacopter "DJI Matrice 600 Pro" was chosen

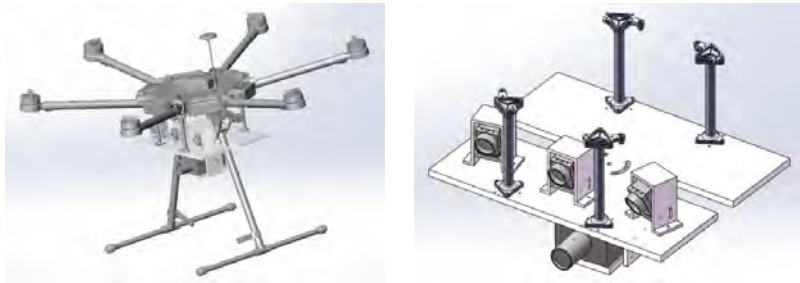


Figure 4.1: Data acquisition platform on DJI Matrice 600 Pro [68]

(Figure 4.1). It can hover a payload up to $6Kg$ for a duration of over a half hour. The data acquisition elements build a subsystem which is attached to the hexacopter through four rigid connections. To collect the data, discrete points in space are defined to which the drone is supposed to fly. When the drone reaches these points,



Figure 4.2: Image of data acquisition [68]

its control is switched to stabilized mode and only then, the images are snapped. Figure 4.2 shows a capture of the data acquisition process. During image capture, the stabilized mode is active so that the flight controller keeps the drone stable, and thus compensating for some disturbances. However, the drone is still submitted to



Figure 4.3: Main vibration sources in UAV

some vibrations. The main sources of these vibrations are on one hand inherent to system like vibrations coming motor-propeller subsystems (Figure 4.3 - red arrow), and on the other hand independent from the system like wind blowing around (Figure 4.3 - blue arrow).

Since the data acquisition platform has a stiff connection to the hexacopter, the vibrations to which the hexacopter is submitted are directly transmitted to the platform. To reduce the vibrations reaching the platform, the stiff connections to the drone should be replace with an appropriate suspension system.

4.1.1 Load requirements

The suspension system should be designed to carry all equipment components for data acquisition. The weight of the data acquisition platform depends on the elements attached to the platform. It may vary if additional components are added. So, as long as the components of the acquisition platform are not chosen, a total mass of the platform cannot be calculated. However, the mass of the whole system (suspension system + data acquisition platform + equipments) must be lower than the maximum allowable payload of the UAV. So, to carry a maximum payload on the platform, the suspension system and the platform itself should be as lightweight as possible.

4.1.2 Dynamic requirements

UAVs belong to high dynamic systems. This high dynamic comes from the motor-propeller subsystems which have to rotate with very high frequency to generate enough torque and thrust to hover the vehicle. In fact, for drone the thrust and torque generated by a motor-propeller system depend on the motor speed:

$$F = c_F \omega^2, \quad M = c_T \omega^2,$$

where c_T and c_M are respectively drag and torque coefficients which depend on the geometry of propellers. Due to imbalances in motor-propeller subsystems, vibrations are generated by the rotational movements. Since the rotational movements of the motors have very high frequencies, the resulting vibrations are high-frequency. These vibrations are transmitted through the drone structure to the platform, since the drone-structure is generally stiff. If the structure were soft, the high-frequency vibrations would not be transmitted, but natural vibrations would occur, which is also undesirable.

In addition to engine vibrations, there are also vibrations from the environment, which are mostly generated outdoors by wind. In contrast to engine vibrations, they do not have much high frequencies, and their effects on the measurement system are not as significant as the effects of engine vibrations.

To compensate motor vibrations as well as wind vibrations, the suspension system to design should be able to absorb high frequencies of motors noises, and to counteract low frequencies of wind noises.

4.1.3 Energy consumption requirements

Active suspension systems need energy to drive embedded actuators. Since the data acquisition platform should be mounted on an UAV and moved around, immobile energy sources cannot be used for the suspensions. Considering the fact that the drone has a maximum payload that it can carry, it is obvious that the energy source should be chosen as lightweight as possible, to save mass for the data acquisition components.

4.1.4 Other requirements

Besides the requirements defined above, there are others requirements that are more implicit and are not necessarily explicitly defined. In case of an isolation platform, they may be the stability of the platform, the degree of freedom of motion, the capacity of the platform to create a motion in the frequency range of the disturbances to compensate them, etc. Table 4.1 shows a shortened list of requirements for the isolation platform.

Table 4.1: Compact requirements

Types of requirements	Description
Mass	lightweight
Dynamic	Motion frequency up to thousand Hz
Energy	Mobile energy source
Cinematic	Translation in the vertical, roll and pitch; or roll, pitch, and yaw
Forces	Carry the payload / measurement system
Cost	Minimal cost
Geometry	Ability to fit for many types of applications

4.2 Concept phase

The first step of the concept phase is the elaboration of a black box. There are three types of inputs and outputs for a black box: material, energy and signal. In our case, as material we have a vibrating ground surface as input which should be decoupled from the platform considered as system output. For the energy, the vibrations of the ground generate mechanical energy while electrical energy will be provided to run the isolation system. The isolation system should absorb the mechanical energy of

the ground vibrations and dissipates it. As disturbance input, there are vibrations coming from UAVs motors, wind, UAVs motion etc. The system will generate acoustic noise and heat coming mainly from the actuation system. Figure 4.4 shows a black box model of the isolation system for the platform we want to design.

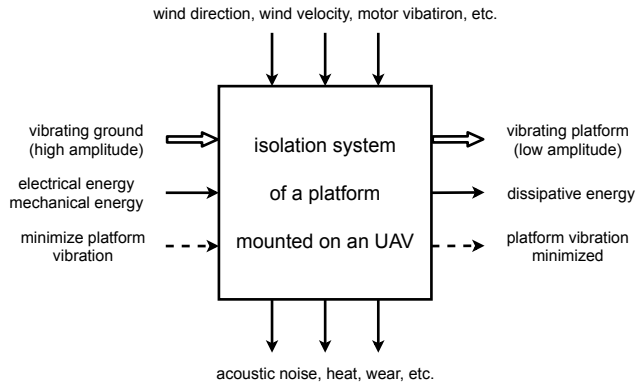


Figure 4.4: Black box model of the isolation platform

After the black box has been made, the next step is the creation of a functional structure. Our system has five main components: the platform, the control board, the sensors, the actuators, and the energy supply. The task of the control board is to execute the algorithms for vibration reduction. Therefore, it communicates with sensors and actuators. The different components are supplied with electrical energy.

Energy is also dissipated in Form of heat coming from electrical component or dissipative elements of the isolation system. Figure 4.5 shows a functional structure of the isolation platform. The main task is split in four main subfunctions. Two of them are responsible for sensing vibrations of the ground and of the platform. The remaining two are responsible of carrying the platform and driving the actuators to reduce the vibrations.

The next step of the PDP is the solution finding and the creation of the morphological box. In total, five subfunctions are identified: type of isolation, energy supply, sense vibration source, carry platform, actuation type, actuator type, sense platform vibration, and control board.

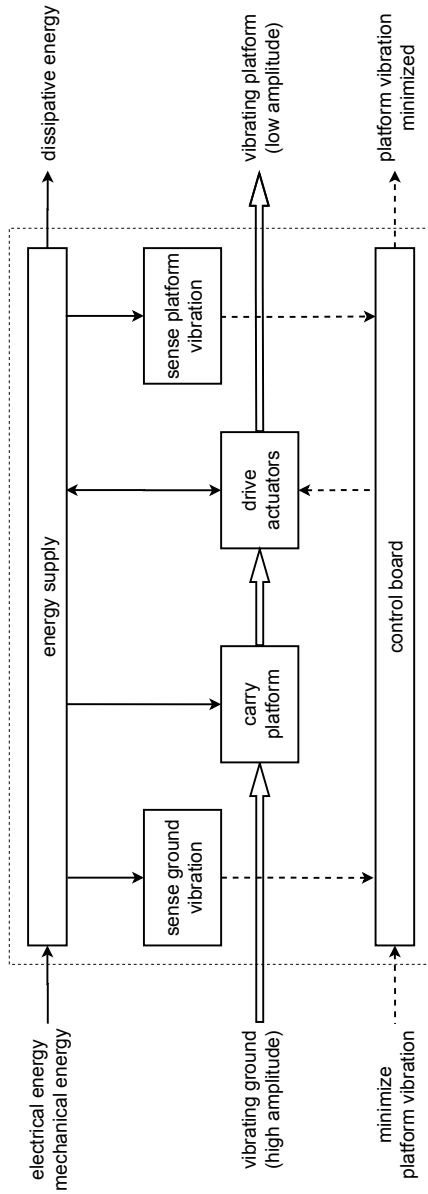


Figure 4.5: Functional structure of isolation platform

For the type of isolation, there are three possible type: passive, semi-active, and active. Due to the advantages listed in the last Chapters, active isolation is the choice made here.

The energy supply for an additional system on a UAV can be done in two manners. The one way is to use the energy supply of the UAV itself, and the second way is to use an additional energy source onboard. While one method shrinks the total energy available for the UAV, the other brings additional mass that increases the total payload of the UAV. Both alternatives lead to a reduction of the time of flight.

Up on the system considered, there are many types of sensors for vibration. There are e.g. position sensor, velocity sensor, acceleration sensor, etc. Before using a sensor, it should be checked if the parameter to measure is at least measurable in the system, and how complex the measurement will be. In our case, the acceleration sensor is the best choice for vibration sensing.

We proposed three solutions to carry the platform. The first one is a rotating platform like in a gimbal. The second one is the platform on top of the suspension system like in laboratory table. The last proposition is a platform suspended on the bottom of suspension system like a swing.

With actuation type, the type of motion of actuators is meant. Rotary actuation can be used to allow roll, pitch, and yaw movements. The other alternative is linear actuation. Since the position of a plane in space can be defined by three points that are not collinear, three linear actuators are sufficient.

As actuator type for vibration control, electric, pneumatic and hydraulic actuators are very common.

For the control board, important requirements are the compatibility with sensors and actuators, the compatibility with computer programs, and the possibility to use it in a Hardware-in-the-Loop (HiL) system. Most used board so such project are Arduino boards and Raspberry Pi.

Table 4.2 presents a morphological box for the isolation platform. In this table, the different suggestions for an overall solution are colored in purple, blue and brown. From this, a total of 42 suggestions for the overall solution are counted.

From the 42 suggestions, four mechanical models shown in Figure 4.6. The design 1 is a platform attached to the drone legs with rotary actuators. The design 2 is a kind of gimbal system. The design 3 is a platform connected like a swing to the drone using linear actuators. The design 4 is a kind of table with three linear actuators as table legs, and the ground plate is connected to drone legs.

An evaluation of the four proposed models must be carried out and the model with the best evaluation will be selected. Therein lies the next step of the PDP. The first

Table 4.2: Morphological box of isolation platform

		Solution			
		1	2	3	4
1	Type of isolation	passive	semi-active	active	
2	Energy supply	UAV battery	Rechargeable battery	Non-rechargeable battery	
3	Sense vibration source	Position sensor	Velocity sensor	Acceleration sensor	Force sensor
4	Carry platform	Rotating platform (gimbal system)	active suspension (table system)	active suspension (suspended plate)	
5	Actuation type	Rotary actuation	Linear actuation		
6	Actuator type	Electric	Pneumatic	Hydraulic	
7	Sense platform vibration	Position sensor	Velocity sensor	Acceleration sensor	Force sensor
8	Control board	Arduino	Raspberry Pi		

step of the evaluation of each concept is the definition of evaluation criteria and their quantifications with weighting coefficients. For that, there are many procedures. The most used procedures are the weighted point evaluation by ranking method, and the weighted point evaluation by preference matrix.

Briefly described, the ranking method is employed to determine the importance of each evaluation criterion based on general judgment. It consists of comparing the individual evaluation criteria in pairs and determining the importance of each criterion in form of weighting factors. In contrast to the preference matrix, two criteria can be equally important. This method is applied to find the weighting factors for the evaluation criteria for the mechanical design. There is a total of six evaluation criteria. As evaluation criteria following are chosen:

- Task fulfillment: this is related to the number of degree of freedom of the design which directly affects the direction in which the vibration can be damped;
- Weight: since the platform is mounted on a UAV, the weight of the design is a

factor that should not be neglected because it may reduce the flight time and decrease the payload;

- Space required: this factor is also important because the space on the UAV is limited;
- Motor load: this is for the load to which the motors in the actuator systems are submitted;
- Stability degree: this is a factor for the inherent stability of the isolation platform;
- Construction effort: this described the complexity that could be faced during the construction.

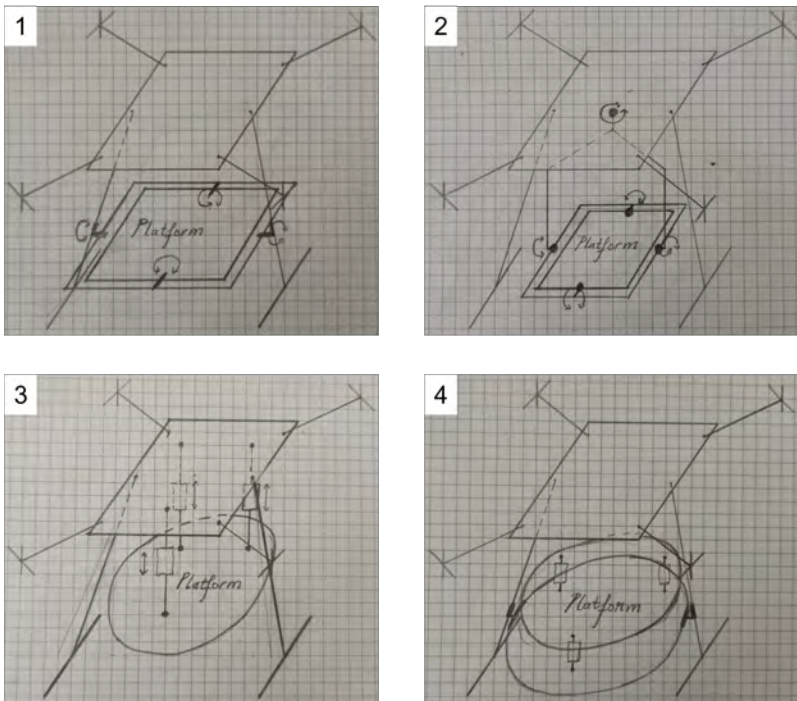


Figure 4.6: Mechanical designs

The result of the ranking method is presented in Table 4.3. Out of this result, it turns out that the task fulfillment is the most important criterion. This is obvious because

Table 4.3: Ranking procedure for mechanical construction

Nr.	Evaluation criteria	1	2	3	4	5	6	Number of "+"	Ranking	Factor
1	Task fulfillment	x	+	+	+	+	+	5	1	45
2	Weight	-	x	+	0	0	+	2	2	18
3	Space required	-	-	x	-	-	+	1	3	9
4	Motor load	-	0	+	x	0	+	2	2	18
5	Stability degree	-	0	+	0	x	+	1	3	9
6	Construction effort	-	-	-	-	-	x	0	4	0 → 1
	Number of "-"	5	2	1	2	1	0	11		

this is what the system is designed for. The second most important criteria are the weight, and the motor load. These criteria directly influence the flight time of the UAVs. The last criterion is the construction effort because it does not impact the effectiveness of the system. It is preceded by the stability degree and the space required.

The second step of the evaluation procedure consists in quantifying the evaluation criteria for all suggestions and choosing the best suggestion by comparing the scores of each suggestion. There exists a rating system from VDI 2225 which evaluates each criterion from 0 to 4, where 0 is the worst evaluation and 4 is the best one. A correspondence table of the rating system of VDI 2225 is shown on Table 4.4.

Table 4.6 shows the application of the second step of the evaluation procedure for the mechanical design. Since the first design only allows roll and pitch motion of the platform, the task can not be fully fulfilled. However, it needs only two rotary actuators. This makes it lighter than the second design, which need three rotating actuators to be able to fulfill the task. To move the platform, rotary actuators in the designs 1 and 2 have to generate high torques, since the system is only supported by one actuator in each direction. The actuators should be big enough to generate the required torques. This is not the case for the designs 3 and 4 where the load is distributed to the three linear actuators. This step of the evaluation procedure shows that the design 3 and 4 are the best mechanical designs, with design 4 having the highest score, and therefore being the best mechanical design.

Table 4.4: Guideline VDI 2225 - Rating System [13]

Rating system			
User analysis		Guideline VDI 2225	
Points	Meaning	Points	Meaning
0	absolutely useless solution	0	unsatisfactory
1	very poor solution		
2	weak solution	1	just acceptable
3	acceptable solution		
4	sufficient solution	2	sufficient
5	satisfactory solution		
6	good solution with minor lacks	3	good
7	good solution		
8	very good solution	4	very good (ideal)
9	solution exceeding the objective		
10	ideal solution		

Table 4.5: Ranking procedure for actuator

Nr.	Evaluation criteria	1	2	3	4	5	6	Number of "+"	Ranking	Factor
1	Dynamic	x	0	0	+	+	+	3	1	27
2	Load	0	x	0	+	+	+	3	1	27
3	Stroke	0	0	x	+	+	+	3	1	27
4	Construction effort	-	-	-	x	0	-	0	3	0 → 0.5
5	Heat	-	-	-	0	x	-	0	3	0 → 0.5
6	Cost	-	-	-	+	+	x	2	2	18
	Number of "-"	5	2	1	2	1	0	11		

Table 4.6: Evaluation of mechanical design

Evaluation criteria	Weighting factor in [%]	Mechanical design							
		Design 1		Design 2		Design 3		Design 4	
		Score	Weighted score	Score	Weighted score	Score	Weighted score	Score	Weighted score
Task fulfillment	45	2	90	4	180	4	180	4	180
Weight	18	2	36	1	18	4	72	3	54
Space required	9	2	18	1	9	4	36	3	27
Motor load	18	2	36	1	18	2	36	4	72
Stability	9	4	36	4	36	4	36	3	27
Construction effort	1	3	3	2	2	2	2	4	4
Valuation			219		263		362		364
Normalized valuation			0.60		0.72		0.99		1.00
Ranking			4		3		2		1

Table 4.7: Evaluation of type of actuator

Evaluation criteria	Weighting factor in [%]	Type of actuator							
		Pneumatic		Electrodynamic		Electromagnetic		Piezoelectric	
		Score	Weighted score	Score	Weighted score	Score	Weighted score	Score	Weighted score
Dynamic	27	2	54	4	108	3	81	4	108
Load	27	1	27	4	108	3	81	1	27
Stroke	27	2	54	3	81	4	108	1	27
Construction effort	0.5	1	0.5	2	1	2	1	4	2
Heat	0.5	3	1.5	1	0.5	1	0.5	4	2
Cost	18	2	36	4	72	4	72	3	54
Valuation			173		370.5		343.5		220
Normalized valuation			0.47		1.00		0.93		0.64
Ranking			4		1		2		3

The same procedure is applied to choose the type of actuators to use for the mechanical design 4. Four suggestions are made: pneumatic actuators, electrodynamic actuators, electromagnetic actuators and piezoelectric actuators. Six evaluation criteria are chosen: the dynamic range, the load, the stroke, the construction complexity, the generated heat and the financial cost. Their ranking procedure is done in Table 4.7. Because of their wide dynamic range, their stroke in millimeter range, and their sufficient load capacity, the best suggestion for the actuators is electrodynamic actuators.

4.3 Design phase

The design phase is one of the last phases of the PDP. In this phase the developed CAD models of the solutions found in the last phase will be presented.

Figure 4.7 shows the mechanical construction for the actuators. As actuators, we have chosen voice-coil motors actuator which are electrodynamic actuators. Electrodynamic actuators require passive elements to be integrated in the construction. Therefore, three springs are integrated to the construction in parallel

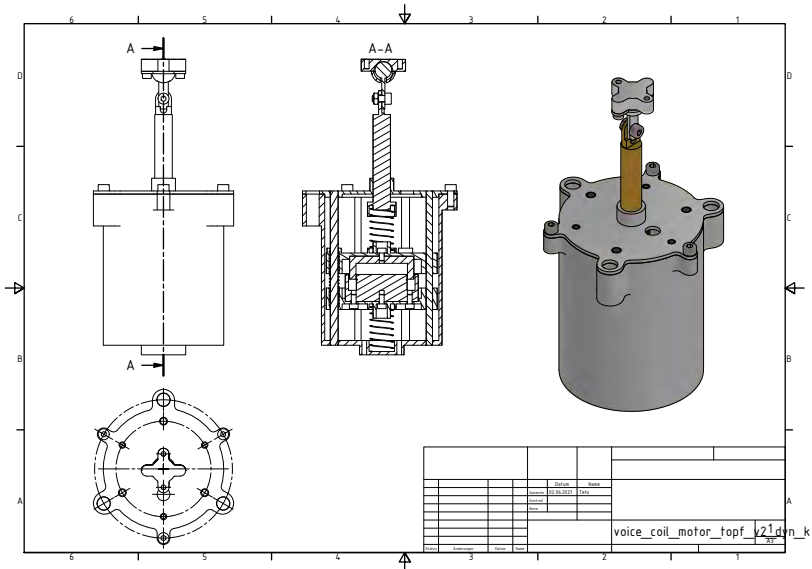


Figure 4.7: Actuator design

to the voice-coil motor itself. Other springs are added to the construction to adjust the actuator properties for control purposes. This will be discussed in detail in

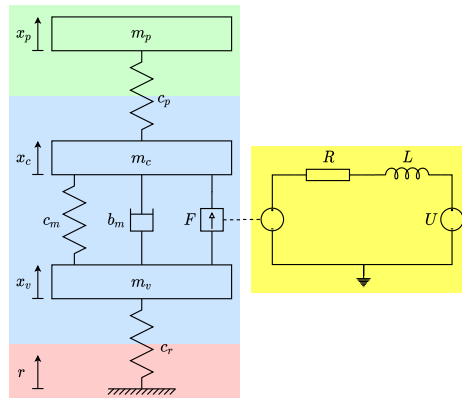


Figure 4.8: Free-body diagram of the actuator

Chapter 5. The actuator has only one degree of freedom which is the vertical direction. This is ensured by the built-in dowels, which are considered as guidance. However, it should be possible to have an inclination angle between the actuator and the payload. Therefore, a kind of spherical joint is build on the top of the actuator. Figure 4.8 shows the free-body diagram of one configuration of the actuators. The electrical circuit represents the motor, and m_p represents the payload of the actuator. The damper stands for the friction in the voice-coil motor. The overall construction of the isolation platform on a quadcopter is shown in Figure 4.10. Using the PDP, this design was chosen as the best suggestion from the four proposed designs. The platform has three support points where it is connected to each actuator. The choice to use three actuators for the platform is because three points are sufficient to clearly define the position and the inclination of a plane in space, and thus the position of the platform. Through the in each actuator integrated spherical joint, an inclination of the platform in each space direction and a translation in vertical direction are possible. The inclination of the platform also induces a small displacement of the support points of the platform. That is reason why there is free space in the spherical joint of each actuator. The free-body diagram of the isolation platform is in Figure 4.9.

The design of the actuator and the isolation platform is now fixed. The next task is the design of a control system for the platform. The various controllers to be designed must be tested on a real system. However, make the first tests of control

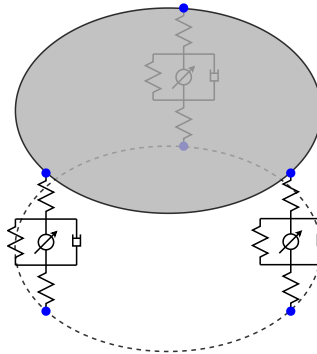


Figure 4.9: Free-body diagram of the isolation platform

algorithms directly on a flying UAV may be very dangerous and expensive. That is the reason why test benches are developed to test different controllers.

4.3.1 Test bench for single suspension

Figure 4.11 shows the technical drawing of the test bench for a single suspension. The aim of this test bench is to test the functionality of single suspensions in different configurations. Briefly described, the test bench is a single suspension excited from the bottom with a motor-propeller system, and to which an isolate load is attached on the top. Because the motor-propeller pair is one of the vibration sources on an UAV, it is used in the test bench with the purpose of trying to replicate the vibrations occurring on the UAV. To ensure that the vibrations only appear in the vertical direction, three dowels are used as guidance system in this direction. To allow the vibrations, the vibrating system is not rigidly connected to the rest of the system, but it is supported by three parallel springs.

4.3.2 Test bench for isolation platform

The test bench for the whole isolation platform is shown in Figure 4.12. It is composed of a drone frame with an isolation platform, the whole supported by an aluminum frame. The drone is connected to the aluminium frame through springs to allow relative motion between the drone and the aluminium frame. This is done to replicate the vibration behavior of a drone.

5 Simulation design

In the last Chapter, a model has been chosen for the mechanical design and the type of actuators. Here, a mathematical modeling of the isolation platform with the chosen components will be made. Then, a simulation environment will be set.

In the design phase, we have decided to use active suspensions with electrodynamic actuators. They can operate at frequencies ranging from a few Hz to kHz , and can support a mass up to $100kg$. As motor in the electrodynamic actuator we have



Figure 5.1: Voice-coil motor

chosen a voice-coil motor (Figure 5.1). According to Zirner and Weikert [83], the voice-coil motor can be represented as shown in Figure 5.2. Electrodynamic

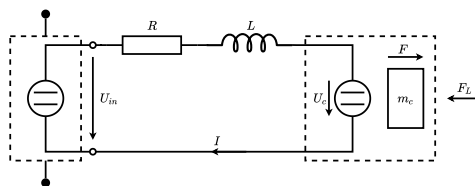


Figure 5.2: electrical circuit of voice-coil motor

actuators need passive elements. Depending on how the passive elements are integrated in the actuator, some properties of the actuator change.

5.1 Actuator dynamic

Concerning the actuator configurations, four main configurations are proposed. There are displayed in Figure 5.4. While configuration 3 is directly inspired

from the well-known single mass damper system, configuration 2 is inspired from the car suspension configuration (Figure 5.3). In this section, two versions of the

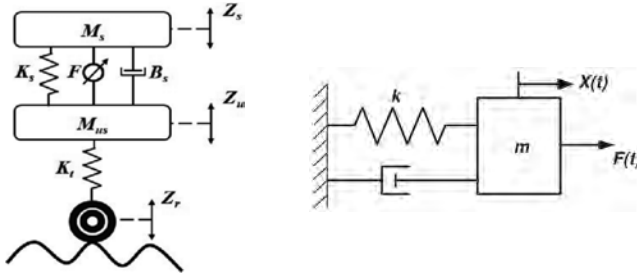


Figure 5.3: Car suspension (left) and single mass damper (right)

dynamic of each actuator will be derived. The reason for this will be explained in the Chapter 6. The main difference between the two versions is the integration of the ground disturbance in the system dynamic. While the ground acceleration is considered as disturbance input in the first version, the ground displacement is considered as disturbance in the second version.

5.1.1 Single suspension

Configuration 1

The suspension configuration 1 is an extension of the configuration of an active car suspension. Instead of attaching the payload directly to the voice-coil motor, a spring is connected between them so that the payload is now only connected to the spring. The damper does not represent an additional passive component, but instead models the friction and energy dissipation in the motor.

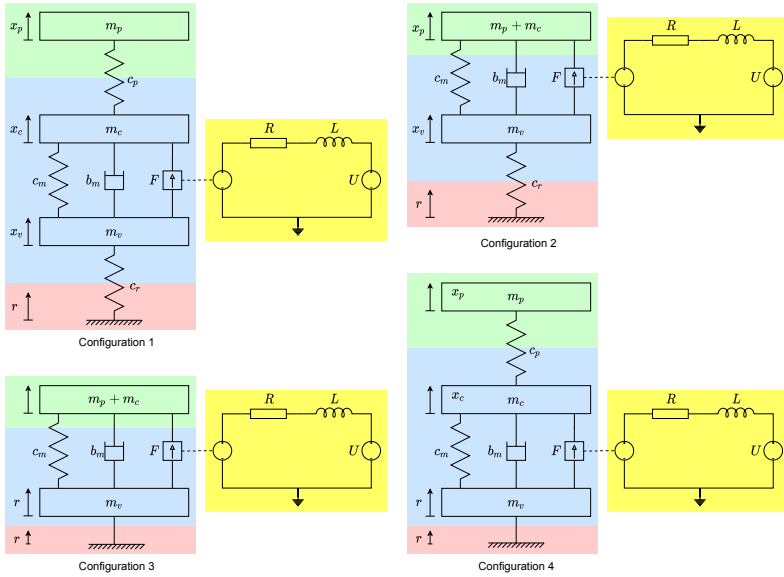


Figure 5.4: Different suspension configurations

Considering the free-body diagram in Figure 5.5, following dynamic can be derived:

$$\begin{cases} m_p \ddot{x}_p = F_{c_p} \\ m_c \ddot{x}_c = -F_{c_p} + F_{c_m} + F_{b_m} + F \\ m_v \ddot{x}_v = -F_{c_m} - F_{b_m} - F + F_{c_r} \\ U = RI + LI + e \end{cases} \quad \text{with} \quad \begin{cases} F_{c_p} = c_p (x_c - x_p) \\ F_{c_m} = c_m (x_v - x_c) \\ F_{b_m} = b_m (\dot{x}_v - \dot{x}_c) \\ F = KI \\ F_{c_r} = c_r (r - x_v) \\ e = K (\dot{x}_c - \dot{x}_v) \end{cases}$$

Out of these equations, following state-space representation results for the first version:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_{ar} \ddot{r} \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_{ar} \ddot{r} \end{cases}, \quad \text{with} \\
 \mathbf{x} = [x_p \quad \dot{x}_p \quad x_c \quad \dot{x}_c \quad x_v \quad \dot{x}_v \quad r \quad \dot{r} \quad I]^T$$

$$\begin{aligned}
 y = \ddot{x}_p &\Rightarrow \mathbf{D}_U = 0, \quad \mathbf{D}_r = 0, \\
 &\Rightarrow \mathbf{C} = \begin{bmatrix} -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.
 \end{aligned}$$

For the second version of the state-space model is represented as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_r r \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_r r \end{cases}, \quad \text{with}$$

$$\mathbf{x} = [x_p \quad \dot{x}_p \quad x_c \quad \dot{x}_c \quad x_v \quad \dot{x}_v \quad I]^T$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{c_p}{m_c} & 0 & -\frac{c_p + c_m}{m_c} & -\frac{b_m}{m_c} & \frac{c_m}{m_c} & \frac{b_m}{m_c} & \frac{K}{m_c} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{c_m}{m_v} & \frac{b_m}{m_v} & -\frac{c_m + c_r}{m_v} & -\frac{b_m}{m_v} & -\frac{K}{m_v} & \\ 0 & 0 & 0 & -\frac{K}{L} & 0 & \frac{K}{L} & -\frac{R}{L} & \end{bmatrix}.$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \frac{c_r}{m_v} & 0 \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{C} = \begin{bmatrix} -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_U = 0, \quad \mathbf{D}_r = 0.$$

The system is modeled with two inputs (the input voltage U of the motor as control input and the ground acceleration \ddot{r} respectively the ground displacement r as noise input) and one output (the acceleration \ddot{x}_p of the payload m_p).

The motor chosen is the voice-coil linear motor AIVCL18-06 from AGICO. Table 5.1 is an extract of the data sheet of this motor. As parameters, following were chosen:

$$\begin{aligned}
 m_p &= 0.75 \text{ Kg}, & m_c &= 0.12 \text{ Kg}, & m_v &= 0.223 \text{ kg}, \\
 b_m &= 0.5 \frac{\text{N}}{\text{m/s}}, & c_m &= 3 \times 0.146 \text{ N/mm}, & c_r &= c_p = 0.174 \text{ N/mm}, \\
 R &= 2.6 \Omega, & L &= 0.95 \text{ mH}, & K &= 8 \text{ N/A} = 8 \frac{\text{V}}{\text{m/s}}.
 \end{aligned}$$

Table 5.1: Extract from the voice-coil motor data sheet

Coil Parameter:	Symbol	Unit	AIVCL18-06
Coil Resistance	R	Ω	2.6
Max. Terminal Voltage	V_p	V	14.3
Peak Current	I_p	A	5.5
Force Constant	K_f	N/A	8
Back EMF Constant	K_b	$\frac{V}{m/s}$	8
Inductance	L	mH	0.95
Motor Parameter:			
Peak Force at 150° within 10 second	F_p	N	44.5
Continuous Stall Force at 150°	F_c	N	17.8
Motor constant	K_a	N/\sqrt{W}	4.96
Electrical time constant	T_e	ms	0.37
Power at peak force	P_p	W	79
Stroke	S	mm	6
Clearance of coil	CL	mm	0.5
Thermal resistance	ϑ_{th}	$^{\circ}C/W$	9.8
Max coil temperature	T_{max}	$^{\circ}C$	150
Coil assembly mass	W_{T_c}	g	45
Core assembly mass	W_{T_f}	g	173

The input-output transfer functions of the dynamic are derived using the formulation

$$G_{yu} = C(sI - A)^{-1}B_U + D_U$$

$$G_{yar} = C(sI - A)^{-1}B_{ar} + D_{ar},$$

and their bode magnitude plots are shown in Figure 5.6.

Configuration 2

The configuration 2 of the actuators is the same as the model of a car suspension whose tires are modeled as springs. The free-body diagram in Figure 5.7 leads to this first version of the state-space representation:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_{ar} \ddot{r} \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_{ar} \ddot{r} \end{cases}, \quad \text{with}$$

$$\mathbf{x} = [x_p \quad \dot{x}_p \quad x_c \quad \dot{x}_c \quad x_v \quad \dot{x}_v \quad r \quad \dot{r} \quad I]^T$$

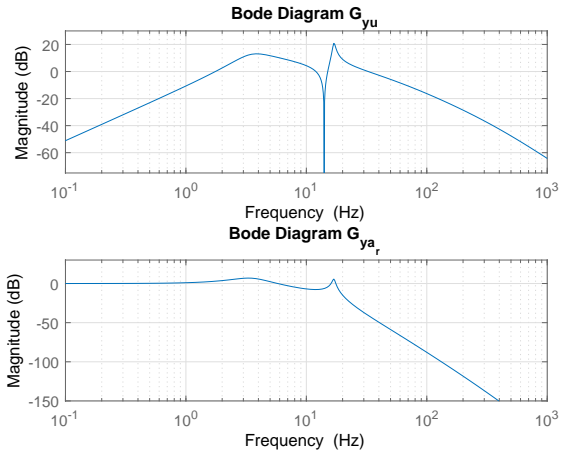


Figure 5.6: Bode Plot of transfer functions - Configuration 1

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{c_m}{m_p + m_c} & \frac{b_m}{m_p + m_c} & 0 & 0 & \frac{K}{m_p + m_c} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{c_m}{m_v} & \frac{b_m}{m_v} & -\frac{c_m + c_r}{m_v} & -\frac{b_m}{m_v} & \frac{c_r}{m_v} & 0 & -\frac{K}{m_v} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{K}{L} & 0 & \frac{K}{L} & 0 & 0 & -\frac{R}{L} \end{bmatrix}$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_{ar} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{D}_U = 0, \quad \mathbf{D}_{ar} = 0,$$

$$\mathbf{C} = \begin{bmatrix} -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{c_m}{m_p + m_c} & \frac{b_m}{m_p + m_c} & 0 & 0 & \frac{K}{m_p + m_c} \end{bmatrix}.$$

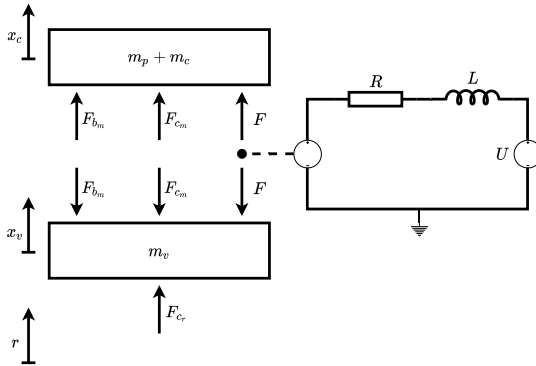


Figure 5.7: Free-body diagram of suspension configuration 2

The second version can be expressed as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_r r \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_r r \end{cases}, \quad \text{with}$$

$$\mathbf{x} = [x_p \quad \dot{x}_p \quad x_c \quad \dot{x}_c \quad x_v \quad \dot{x}_v \quad I]^T$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{c_m}{m_p + m_c} & \frac{b_m}{m_p + m_c} & \frac{K}{m_p + m_c} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{c_m}{m_v} & \frac{b_m}{m_v} & -\frac{c_m + c_r}{m_v} & -\frac{b_m}{m_v} & -\frac{K}{m_v} & 0 \\ 0 & -\frac{K}{L} & 0 & \frac{K}{L} & -\frac{R}{L} & 0 \end{bmatrix}$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_r = \begin{bmatrix} 0 & 0 & 0 & \frac{c_r}{m_v} & 0 \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{D}_U = 0, \quad \mathbf{D}_r = 0,$$

$$\mathbf{C} = \begin{bmatrix} -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{c_m}{m_p + m_c} & \frac{b_m}{m_p + m_c} & \frac{K}{m_p + m_c} \end{bmatrix}.$$

The bode magnitude plots of the input-output transfer functions are shown in Figure 5.8.

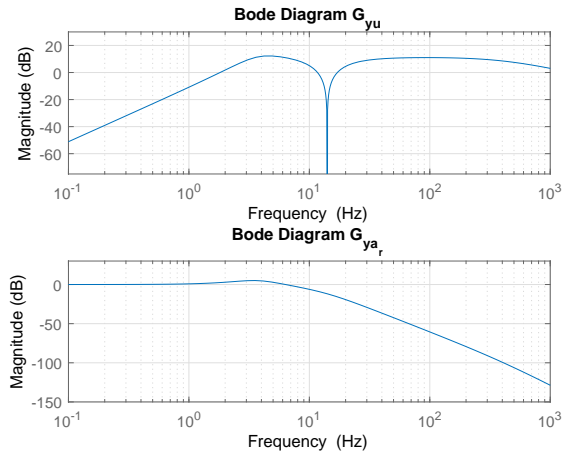


Figure 5.8: Bode Plot of transfer functions - Configuration 2

Configuration 3

The configuration 3 is a simple single mass damper with a motor. Figure 5.9 shows

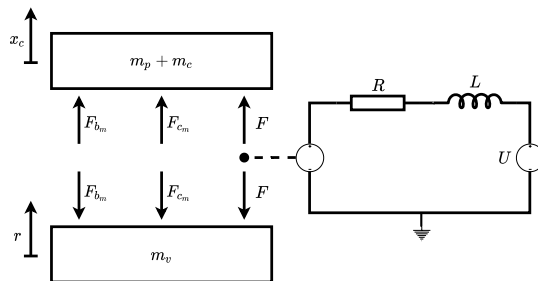


Figure 5.9: Free-body diagram of suspension configuration 3

its free-body diagram. The dynamic of this suspension is represented through this first version of the state-space representation:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_{ar} \ddot{r} \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_{ar} \ddot{r} \end{cases}, \text{ with}$$

$$\mathbf{x} = [x_p \quad \dot{x}_p \quad x_c \quad \dot{x}_c \quad x_v \quad \dot{x}_v \quad r \quad \dot{r} \quad I]^T$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{c_m}{m_p + m_c} & \frac{b_m}{m_p + m_c} & \frac{K}{m_p + m_c} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{K}{L} & 0 & \frac{K}{L} & -\frac{R}{L} \end{bmatrix}$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_{ar} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{D}_U = 0, \quad \mathbf{D}_{ar} = 0,$$

$$\mathbf{C} = \begin{bmatrix} -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{c_m}{m_p + m_c} & \frac{b_m}{m_p + m_c} & \frac{K}{m_p + m_c} \end{bmatrix}.$$

The second version has the following representation:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_r r + \mathbf{B}_{vr} \dot{r} \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_r r + \mathbf{D}_{vr} \dot{r} \end{cases}, \quad \text{with}$$

$$\mathbf{x} = \begin{bmatrix} x_p & \dot{x}_p & x_c & \dot{x}_c & x_v & \dot{x}_v & r & \dot{r} & I \end{bmatrix}^T$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{K}{m_p + m_c} \\ 0 & -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_r = \begin{bmatrix} 0 & \frac{c_m}{m_p + m_c} & 0 \end{bmatrix}^T$$

$$\mathbf{B}_{vr} = \begin{bmatrix} 0 & \frac{b_m}{m_p + m_c} & \frac{K}{L} \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{D}_U = 0, \quad \mathbf{D}_r = 0, \quad \mathbf{D}_{vr} = 0,$$

$$\mathbf{C} = \begin{bmatrix} -\frac{c_m}{m_p + m_c} & -\frac{b_m}{m_p + m_c} & \frac{K}{m_p + m_c} \end{bmatrix}.$$

The bode magnitude plots of the input-output transfer functions are shown in Figure

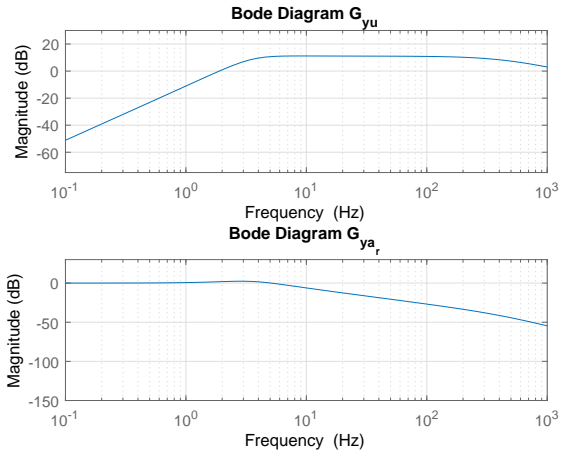


Figure 5.10: Bode Plot of transfer functions - Configuration 3

5.10.

Configuration 4

The configuration 4 is an expanded single mass damper suspension with an additional spring on the top. The free-body diagram of this configuration is illustrated in Figure 5.11. The dynamic of the first state-space representation is derived as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_{ar} \ddot{r} \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_{ar} \ddot{r} \end{cases}, \quad \text{with}$$

$$\mathbf{x} = \begin{bmatrix} x_p & \dot{x}_p & x_c & \dot{x}_c & x_v & \dot{x}_v & r & \dot{r} & I \end{bmatrix}^T$$

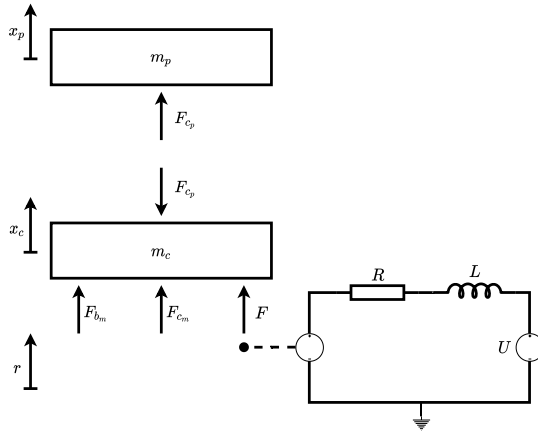


Figure 5.11: Free-body diagram of suspension configuration 4

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{c_p}{m_c} & 0 & -\frac{c_p + c_m}{m_c} & -\frac{b_m}{m_c} & \frac{c_m}{m_c} & \frac{b_m}{m_c} & \frac{K}{m_c} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{K}{L} & 0 & \frac{K}{L} & -\frac{R}{L} \end{bmatrix},$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_{ar} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{C} = \begin{bmatrix} -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_U = 0, \quad \mathbf{D}_{ar} = 0.$$

For the second version of the state-space is represented as:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_U U + \mathbf{B}_r r + \mathbf{B}_{vr} \dot{r} \\ y = \mathbf{C}\mathbf{x} + \mathbf{D}_U U + \mathbf{D}_r r + \mathbf{D}_{vr} \dot{r} \end{cases}, \quad \text{with}$$

$$\mathbf{x} = \begin{bmatrix} x_p & \dot{x}_p & x_c & \dot{x}_c & x_v & \dot{x}_v & r & \dot{r} & I \end{bmatrix}^T$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{c_p}{m_c} & 0 & -\frac{c_p + c_m}{m_c} & -\frac{b_m}{m_c} & \frac{K}{m_c} & 0 \\ 0 & 0 & 0 & -\frac{K}{L} & -\frac{R}{L} & 0 \end{bmatrix}$$

$$\mathbf{B}_U = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{L} \end{bmatrix}^T$$

$$\mathbf{B}_r = \begin{bmatrix} 0 & 0 & 0 & \frac{c_m}{m_c} & 0 \end{bmatrix}^T$$

$$\mathbf{B}_{vr} = \begin{bmatrix} 0 & 0 & 0 & \frac{b_m}{m_c} & \frac{K}{L} \end{bmatrix}^T$$

$$y = \ddot{x}_p \Rightarrow \mathbf{D}_U = 0, \quad \mathbf{D}_r = 0, \quad \mathbf{D}_{vr} = 0,$$

$$\mathbf{C} = \begin{bmatrix} -\frac{c_p}{m_p} & 0 & \frac{c_p}{m_p} & 0 & 0 \end{bmatrix}.$$

The bode magnitude plots of the input-output transfer functions are shown in

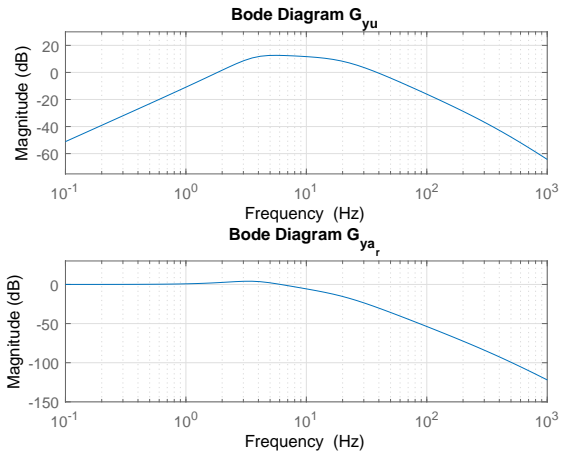


Figure 5.12: Bode Plot of transfer functions - Configuration 4

Figure 5.12.

Figures 5.6, 5.8, 5.10 and 5.12 show on their first subplots the bode magnitude of the transfer function from the noise input (ground acceleration \ddot{r}) to the output (payload acceleration \ddot{x}_p), and on their second subplots the bode magnitude of the transfer function from the control input (voltage U of the voice-coil motor) to the output. The comparison of the bode diagrams shows that the configuration 1 has the best passive damping behavior for frequencies above $20Hz$. The passive isolation of configurations 2 and 4 are not better in this frequency range, but they show a better behavior for frequencies lower than $20Hz$. The configuration 3 is a bit stiffer and has the worst passive isolation behavior. However, it has the best active behavior. It has a wide bandwidth, this means that the actuator can react to noises in a large frequency range. The configuration 4 has also a good active behavior. It is really effective for frequencies between $2Hz$ and $35Hz$. In configuration 2, the actuator has a similar behavior as an inverse notch filter. The actuator in configuration 2 is very effective for noise frequencies between $2Hz$ and $11Hz$, and higher than $17Hz$. The configuration 1 has the same inverse notch characteristic, and it is even worse. In fact, it is the worst configuration for the active behavior. It can only effectively react to noise frequencies between $2Hz$ and $12Hz$, and between $15Hz$ and $33Hz$.

A closer analysis from the impact of the suspension structures related to the bode diagram shows that the lower spring is responsible for a better passive damping behavior, but it is also responsible for the inverse notch effect of actuators on the payload. The upper spring amplifies the passive damping behavior, however, it reduces the frequency bandwidth of actuators.

In summary, while the configuration 1 has the best passive behavior and the worst active behavior, the configuration 3 has the worst passive behavior and the best active behavior. The configurations 2 and 4 are a kind of tradeoff between the configurations 1 and 3, whereby the configuration 4 seems to be the better compromise.

5.1.2 Isolation platform

With the dynamic of single suspension, the behavior of SISO systems have been derived. Now, the question is, what is the effect of the different suspension configurations on the isolation platform? To answer this question, we consider the free-body diagram of the isolation platform as shown in Figure 5.13. The dynamics

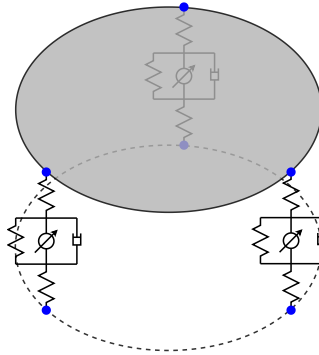


Figure 5.13: Free-body diagram of the isolation platform

derived for each single suspension are coupled by the platform dynamic. Figures 5.14, 5.15, 5.16 and 5.17 show extracts from the bode diagrams of the isolation platform for each configuration. It shows that the response of each suspension to its excitation is very similar to the response of a single suspension. The suspensions also have influences on each other. Although each suspension is mainly influenced by its excitation, the effects of the dynamic of other suspensions on it are not negligible.

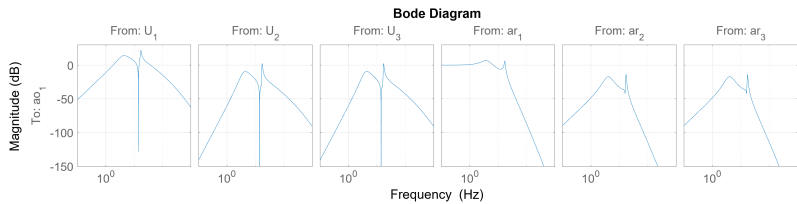


Figure 5.14: Bode diagram of isolation platform - configuration 1

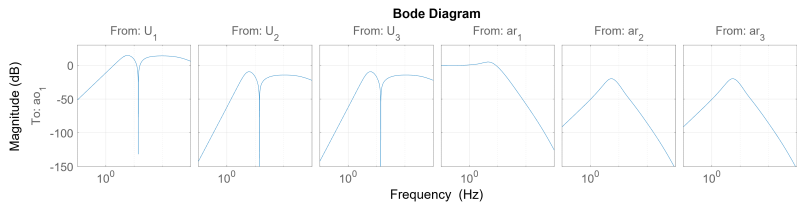


Figure 5.15: Bode diagram of isolation platform - configuration 2

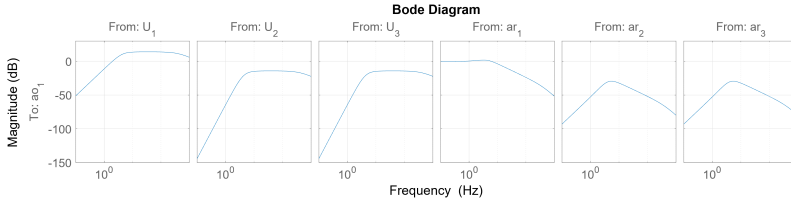


Figure 5.16: Bode diagram of isolation platform - configuration 3

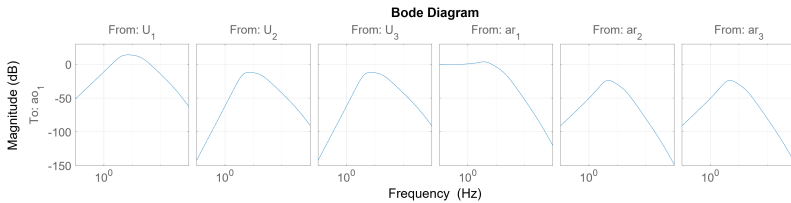


Figure 5.17: Bode diagram of isolation platform - configuration 4

5.2 Simulation design

Banks [8] defines simulation as the imitation of a real-world process or system over time. The aim of the simulation is to replicate the physical world in computer as good as possible in order to do some investigations on the system without using the physical system. The main advantage is that, simulation can take any input without damaging the physical system. So, we can test whatever we can think of in the simulation and predict the behavior of the physical system without the risk of damaging the physical system. However, it is difficult and even impossible to build a simulation that is identical to the physical system. Of course, the precision can be increased, but the both systems will not be identical.

To build the simulation Matlab/Simulink is used. There are at least three ways to build simulations with this software. The first alternative is to use Matlab to write a code which modeled the system dynamic. The second one is to use Simulink and to model the system dynamic using different Simulink blocks like integrator, differentiator, sum etc. These first two alternatives require the dynamic equations to be known before designing the simulation, and this implicitly required a very good understanding of physical systems and processes, and of the methods to derived their dynamic. So if the system dynamic cannot be well derived, it will be

to model the voice-coil motor:

For the isolation platform, the Simscape multibody library is used to connect each suspension to the platform. The simulation is build according to the model in Figure 5.13. The Simscape model is displayed in Figure 5.19.

The simulation models built in Simscape are just approximations of real models of the test bench for single suspension and the test bench of the isolation platform. These models will be used to test control algorithms that will be designed before testing them on the test benches. The next section focuses on controller design.

5.3 Summary

In this Chapter, the first steps of the development process of control system. The mathematical models of each suspension configuration were derived and analyzed. The analysis has shown that the configuration 3 has the best active behavior. Configuration 1 has the best passive behavior, but also the worse active behavior. Then, an analysis of the isolation platform with each suspension configuration was done. The analysis shows a similar behavior as for single suspensions. However, it exists a coupled dynamic between the different actuators.

After the model analysis, simulation environments have been set using the Simulink environment of Matlab. The Simscape Toolbox has been used for this purpose. Simulation environments have been set for single suspension systems, and for the whole isolation platform.

Now that the analysis of the plant is done, and the simulation environments are set, the next step is the controller design and analysis.

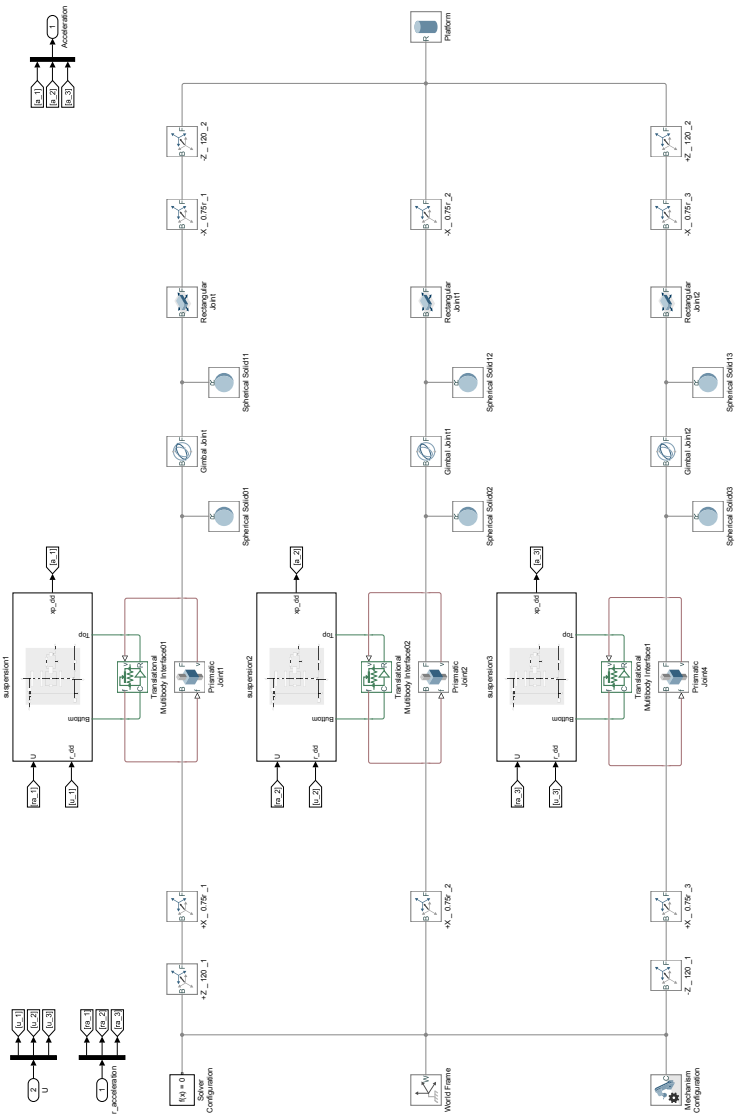


Figure 5.19: Simscape model of isolation platform

6 Controller design

In this section, controllers will be designed for the isolation platform. The main goal is to reduce the vibration of the platform mounted on the drone using suspension systems. The controllers to be designed will be deployed in a rough environment. The environment is considered as rough because there are many disturbance sources which cannot be clearly identified. As an example, the disturbance resulting from wind is not constant and is not repeatable. It depends on the weather that cannot be controlled by the user. Additionally, the effect of the disturbances coming from the motor-propeller subsystems on the actuator ground depends on the propagation path from the motor to the actuator. This path depends on the physical structure of the drone. Despite all these disturbances, the controllers should achieve their tasks and should be able to handle those disturbances. Therefore, we should choose controller designs that can handle changes in systems or that are robust enough to achieve the tasks despite these changes. Therefore, we have chosen to use adaptive control design and robust control design as strategies.

6.1 Adaptive Control

Due to its effectiveness the ANC technique is one of the most popular adaptive control strategies. As explained in Chapter 2, the aim of ANC is to generate a signal that will be transferred through the secondary path to destructively interfere with the primary noise. While fixed ANC is adequate for constant systems, adaptive ANC is suitable for time-variant systems or systems operating in rough environments. Adaptive ANC is usually designed using adaptive filters. Therefore, either adaptive FIR filters or adaptive IIR filters can be used. ANC with adaptive FIR filters has fast algorithms and is stable due to the filter structure that does not allow poles. However FIR filters are not adequate for resonant systems or systems with internal feedback because they have no poles and they need a lot of coefficients to approximate the behavior of poles without having inherent poles. This will increase the computational load and slow down the algorithm. For those kinds of systems, adaptive IIR filters are more appropriate. IIR filters have an inherent pole-zero structure which allows them to approximate resonant systems with less coefficients. But the algorithms for adaptive IIR filters are more complex and the stability of the IIR filters during the adaptation is not guaranteed. To fix the problem of stability some adaptive algorithms using

inherent stable IIR structures like Lattice structure and Laguerre structure have been developed [82] [49] [18]. But this is done on cost of increasing computational load and decreasing convergence rate.

In this thesis, the focus is on made the ANC strategy with adaptive IIR filters. The first step of the ANC is to find a good identification algorithm. This is handled in the next subsection.

6.1.1 Identification algorithm

The problematic of identification is illustrated in Figure 6.1. The goal is to find

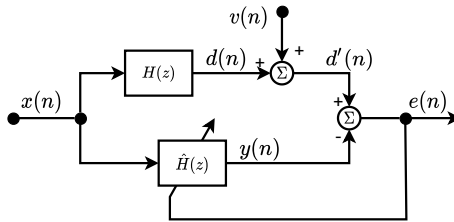


Figure 6.1: Problem of system identification

optimal parameters for $\hat{H}(z)$ to approximate the system $H(z)$ to identify. In case of identification with an IIR filter it can be formulated as

$$H(z) = \frac{Y(z)}{X(z)} \approx \hat{H}(z) = \frac{A(z)}{1 - B(z)} = \frac{\sum_{k=0}^M a_k z^{-k}}{1 - \sum_{l=1}^N b_l z^{-l}}$$

$$\Rightarrow y(n) = \sum_{k=0}^M a_k x(n-k) + \sum_{l=1}^N b_l y(n-l).$$

The objective is the minimization of the error

$$e(n) = d(n) - y(n),$$

where $y(n)$ is the output of the adaptive IIR filter. The IIR filter is composed of two paths: the feedforward path, and the feedback path. Depending on which input is used for the feedback path, we distinguish two types of error formulations: the equation-error formulation, and the output-error formulation.

Output-error formulation

The structure of the output-error formulation is shown in Figure 6.2. The adaptive IIR

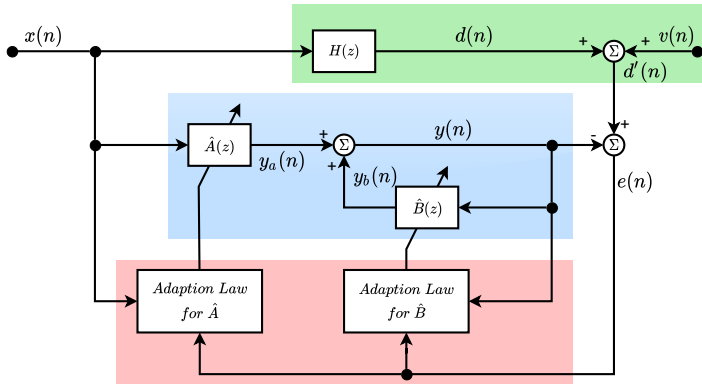


Figure 6.2: Output-error formulation

filter has the classic tapped delay structure with the output of the filter as feedback input. This leads to the formulation

$$y(n) = \sum_{k=0}^M a_k x(n-k) + \sum_{l=1}^N b_l y(n-l).$$

As explained in Chapter 2, using y as input in the feedback path will lead to a nonlinear optimization problem with a multimodal cost function. So, solving the optimization does not necessarily lead to the optimal solution. However, if the optimal solution is reached, it will not be biased. To avoid the problem of local minima, there is the equation-error formulation.

Equation-error formulation

Figure 6.3 shows the structure of the equation-error formulation. In contrast to the output-error formulation, the equation-error formulation uses the output of the real plant instead of the filter output as input for the feedback path. This leads to the

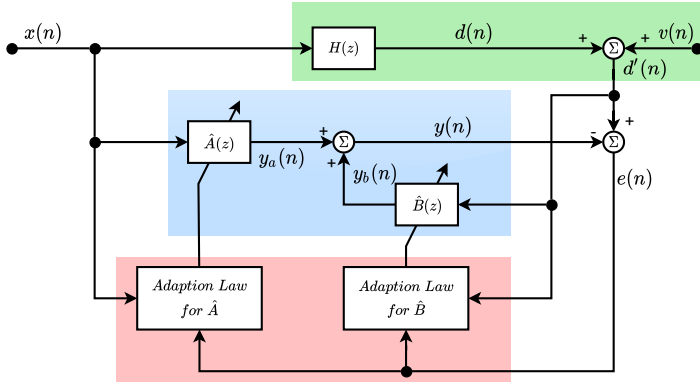


Figure 6.3: Equation-error formulation

formulation

$$\begin{aligned}
 y(n) &= \sum_{k=0}^M a_k x(n-k) + \sum_{l=1}^N b_l d'(n-l) \\
 &= \sum_{k=0}^M a_k x(n-k) + \sum_{l=1}^N b_l d(n-l) + \sum_{l=1}^N b_l v(n-l).
 \end{aligned}$$

This structure leads to a linear quadratic optimization problem which has only one optimum. However, the signal fed back is corrupted by $v(n)$. So, if $v(n)$ is neither zero nor white noise, the optimum will be biased.

Algorithms for adaptive IIR filtering

The purpose of the algorithms here is to find the optimal parameters a_k and b_l minimizing the cost function

$$\xi(n) = \mathbb{E} \{ e^2(n) \}.$$

As mentioned in Chapter 2, the gradient descent algorithm formulated as

$$\begin{aligned}
 \boldsymbol{\theta}(n+1) &= \boldsymbol{\theta}(n) - \mu \frac{\partial \xi(n)}{\partial \boldsymbol{\theta}} \quad \text{with} \\
 \boldsymbol{\theta}(n) &= [a_0(n) \quad \dots \quad a_M(n) \quad b_1(n) \quad \dots \quad b_N(n)]^T
 \end{aligned}$$

is used to solve this problem. After some simplifications shown in Chapter 2, the Feintuch's algorithm results as

$$\begin{aligned} a_k(n+1) &= a_k(n) + 2\mu e(n)x(n-k) \\ b_l(n+1) &= b_l(n) + 2\mu e(n)\eta(n-l) \\ \eta(n-l) &= \begin{cases} y(n-l) & \text{for output-error formulation} \\ d'(n-l) & \text{for equation-error formulation} \end{cases} \end{aligned}$$

[71]. To increase the performance many other algorithms are developed based on the Feintuch's algorithm. Most of them deal with the step size μ that is not defined as a constant but as a variable. Akhtar, Abe, and Kawamata [4] proposed to adapt the step size depending on the error. In fact, at the adaption beginning the error is high, and according to Akhtar, Abe, and Kawamata [4] a small step size should be used until the error signal decrease, and the step size can be increased when the error signal is low. This is well resumed in [5] as

$$\begin{aligned} \mu(n) &= \rho(n)\mu_{\min} + [1 - \rho(n)]\mu_{\max}, \quad \text{with} \\ \rho(0) &= 1 \quad \text{and} \quad \lim_{n \rightarrow \infty} \rho(n) = 0. \end{aligned} \quad (6.1)$$

According to Ahmed, Akhtar, and Tufail [3], the method proposed by Akhtar, Abe, and Kawamata [4] allows a fast initial convergence, but compromises the steady state performance due to the high step size near the eventual optimum. They proposed the new formulation

$$\mu_s(n) = [1 - \rho(n)]\mu(n),$$

with $\rho(n)$ having the same formulation as in 6.1. $\mu(n)$ is given as

$$\mu(n) = \beta_{vss},$$

and is bounded by μ_{\min} and μ_{\max} to guarantee respectively the tracking performance and the stability, and $\beta_{vss}(n)$ is recursively computed as

$$\beta_{vss}(n+1) = \beta_{vss}(n) + [1 - \alpha]f^2(n),$$

where α is a control parameter and $f(n)$ is a kind of error signal [3].

Narasimhan, Veena, and Loksha [58] proposed an algorithm called variable step-size Griffiths' algorithm which is a combination between the Griffiths'

algorithm and the Okello's algorithm. While the Griffiths' algorithm uses the cross-correlation between desired signal and the input signal to update the parameters, the Okello's algorithm uses the cross-correlation between the input signal and the error signal to update the step size [58]. Narasimhan and Veena [57] updated this algorithm by proposing a bias-free adaptive structure for IIR filter (Figure 6.4). This leads to the update equations defined as

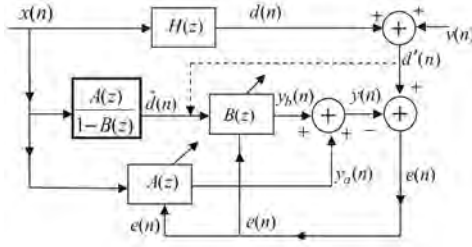


Figure 6.4: Bias-free IIR adaptive filter structure proposed by Narasimhan and Veena [57]

$$a_k(n+1) = a_k(n) + \frac{\mu}{MP_x(n)} [G_{ak}(n) - y(n)x(n-k)], \quad k = 0, \dots, M$$

$$b_l(n+1) = b_l(n) + \frac{\mu}{NP_d(n)} [G_{bl}(n) - y(n)\hat{d}(n-k)], \quad l = 1, \dots, N$$

where $P_x(n)$ and $P_d(n)$ are the powers of the input signals, and are computed recursively as

$$P_x(n) = \gamma P_x(n-1) + (1-\gamma)x^2(n),$$

$$P_d(n) = \gamma P_d(n-1) + (1-\gamma)\hat{d}^2(n), \quad 0 < \gamma < 1,$$

$G_{ak}(n)$ and $G_{bl}(n)$ are the cross-correlation between the desired signal and the input signal defined as

$$G_{ak}(n) = \beta G_{ak}(n-1) + [1-\beta]d(n)x(n-k),$$

$$G_{bl}(n) = \beta G_{bl}(n-1) + [1-\beta]d(n)\hat{d}(n-k), \quad 0 < \beta < 1.$$

[57]. In case of variable step size, the step sizes are updated using Griffiths' variable step-size algorithm

$$\begin{aligned}\mu_a(n) &= \alpha_a \mu_a(n-1) + \sigma_a \sum_{k=0}^M [G_{ak}(n) - y(n)x(n-k)]^2, & 0 < \alpha_a, \sigma_a < 1 \\ \mu_b(n) &= \alpha_b \mu_b(n-1) + \sigma_b \sum_{l=1}^N [G_{bl}(n) - y(n)\hat{d}(n-k)]^2, & 0 < \alpha_b, \sigma_b < 1\end{aligned}$$

[58].

6.1.2 Proposed algorithm

No matter which formulation is used for the IIR adaptive filtering, there are always problems occurring during the adaption. The output-error formulation has the risk of leading the algorithm to a local optimum, and the equation-error formulation leads to a unique solution, which may be biased if there are noises in the system. To fix this problem, Narasimhan, Veena, and Lokeshia [58] used an updated structure of the equation-error formulation to solve the problem of biased optimum (Figure 6.4). However, taking a closer look at the mathematical formulation of this structure shows that this structure is equivalent to the output-error formulation, at least in a pure mathematical point of view. This indeed may solve the problem of biased estimator, but it brings the problem of multimodal cost function back.

Our proposition is a combination of the two error formulations for the IIR adaptive filtering to build a better algorithm. The idea is to avoid local minima appearing in the output-error formulation and simultaneously avoid the biased estimator appearing in the equation-error formulation. We literally want to reach the global optimum of the cost function of the output error formulation, respectively to decrease the chance of getting stuck in a local optimum. Therefore, block diagrams in Figures 6.5 and 6.6 are developed. The main difference between the two error formulations is the input of the feedback path. Using an appropriate algorithm, we estimate \hat{B} for the equation-error formulation \hat{B}_{EQE} and for the output-error formulation \hat{B}_{OE} . Therefore, the following formulation is used to fuse both estimators

$$\hat{B}(n) = [1 - \tau^n] \hat{B}_{OE}(n) + \tau^n \hat{B}_{EQE}(n), \quad 0 < \tau < 1,$$

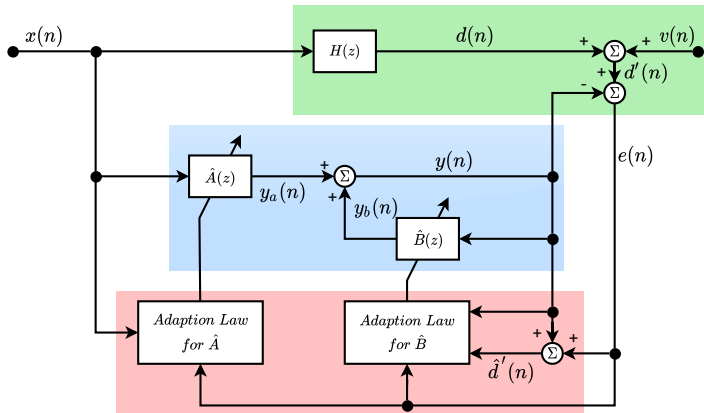


Figure 6.5: Combination of output-error and equation-error

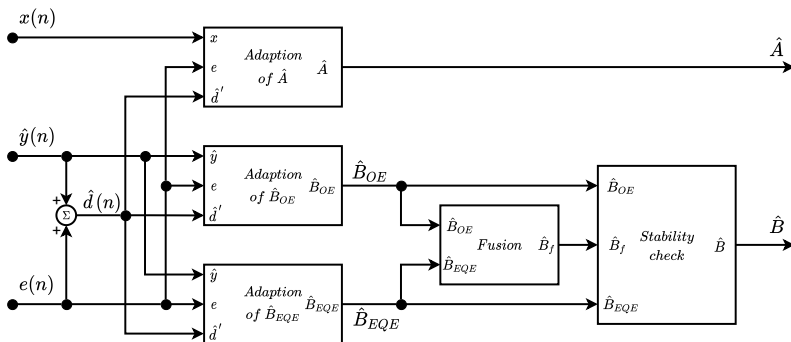


Figure 6.6: Adaptive IIR filter with OE and EQE fusion block

where $f(n) = \tau^n$ is the interpolation between both estimators. One of the main challenges of the output the output-error formulation is to find a good starting point in order to avoid local minima as shown in Figure 6.7. The equation-error formulation always converges to a unique point, since its optimization problem is linear quadratic. This solution may in fact be biased depending on the nature of the noise, but if the biased term is not too high, then the biased estimator will not be far from the unbiased optimum which is actually the global minimum of the multimodal cost function of the output-error formulation. In order to avoid local minima, the estimator should begin with the equation-error formulation and gradually goes over to the output-error formulation with increasing n . This is what

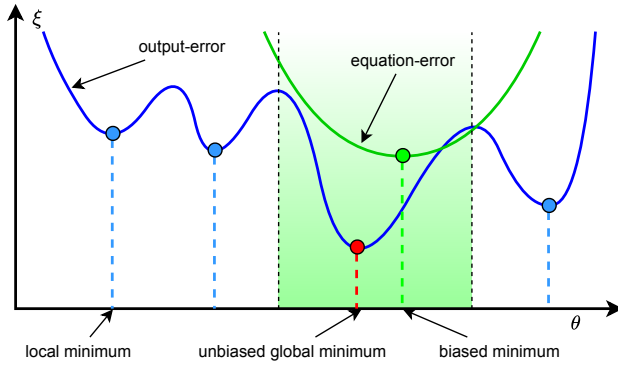


Figure 6.7: Possible optima for different formulations

is implemented through the interpolation function $f(n) = \tau^n$:

$$\hat{B}(0) = \hat{B}_{EQE}(n) \quad \text{and} \quad \lim_{n \rightarrow \infty} \hat{B}(n) = \hat{B}_{OE}(n).$$

If after the first iterations the estimator reaches the green area from Figure 6.7, then it will reach the unbiased global minimum while going over to output-error formulation.

The proposed method increases the probability of finding the unbiased global optimum. However, it is not guaranteed that the global optimum will be found. Since the equation-error estimator and the output-error formulation run in parallel, the computing effort for \hat{B} is doubled. To notice is that if the number of coefficients chosen for the IIR filter does not exactly correspond to the number of coefficients required to model the real plant, it may happen that the solution considered as global optimum is not unique. Let us for example assume that

$$\hat{H}^* = \frac{A^*(z)}{1 - B^*(z)}$$

is the optimal approximation of the plant to identify, then all estimations that can lead to the form

$$\hat{H}(z) = \frac{A^*(z) \cdot \prod_{i=1}^p [1 + q_i z^{-1}]}{[1 - B^*(z)] \cdot \prod_{i=1}^p [1 + q_i z^{-1}]}$$

are also candidates for the optimal solution [71].

Like with all algorithms using the tapped-delay structure, this algorithm does not guarantee the stability of the IIR filter. To ensure the stability, Algorithm 2 is proposed. This algorithm always ensures the stability of system by switching

Algorithm 2 Stability monitoring

if \hat{B}_{OE} stable and \hat{B}_{EQE} stable **then**
 $\hat{B}(n) = [1 - \tau^n] \hat{B}_{OE}(n) + \tau^n \hat{B}_{EQE}(n)$
else if \hat{B}_{OE} stable and \hat{B}_{EQE} not stable **then**
 $\hat{B}(n) = \hat{B}_{OE}(n)$
else if \hat{B}_{OE} not stable and \hat{B}_{EQE} stable **then**
 $\hat{B}(n) = \hat{B}_{EQE}(n)$
else
 $\hat{B}(n) = \hat{B}(n - 1)$
end if

between different estimators, if one of them is not stable. But it may also happen that the stability monitoring gets stuck when \hat{B} does not upgrade over many time samples. This may happen, when \hat{B}_{OE} and \hat{B}_{EQE} remain unstable for many steps. This situation leads to

$$\hat{B}(n) = \hat{B}(n - 1) = \hat{B}(n - 2) = \dots = \tilde{B}, \quad (6.2)$$

where \tilde{B} corresponds to the last estimated stable \hat{B} , but does not necessarily correspond to a good approximation of the real plant. In this case, since the adaption of \hat{A} is still proceeding, the solution now only depends on the adaption law of \hat{A} .

The stability monitoring as presented in Algorithm 2 ensures the stability of the system, but not the effectiveness of the adaptive filter. In order to increase the performance of the stability algorithm, some modifications can be applied to the interpolation function $f(n) = \tau^n$. Some propositions are:

1. $f(n) = \tau^n$.
2. $f(n) = \tau^{n-n_0}$, with n_0 defined as the last step where \hat{B}_{EQE} or \hat{B}_{OE} was unstable.
3. $f(n) = \tau^{n-n_0}$, with n_0 defined as the last step where \hat{B}_{EQE} or \hat{B}_{OE} was unstable and f was lower than a predefined value γ .
4. $f(n) = \tau^{n-n_0+\kappa} = \tau^{n-n_0} \cdot \tau^\kappa$, with n_0 defined as the last step where \hat{B}_{EQE} or \hat{B}_{OE} was unstable and f was lower than a predefined value γ , and τ^κ represents

the new start value for f in case of instability of \hat{B}_{EQE} or \hat{B}_{OE} . κ is a predefined constant value which is considered as 0 until one unstable estimator occurs.

Proposition (1) is the interpolation function proposed at the beginning. The aim of this interpolation is to start the adaptive filtering algorithm with equation-error formulation which has only one but eventually biased minimum, and to go gradually over to the output-error formulation which has the unbiased minimum but also local minima. It is important to understand The meaning behind this in order to follow the other propositions. Let us consider the case where one of the two formulations or both return unstable \hat{B} . The estimator is somewhere it should not be, and should not stay longer there. In this case, a new start point should be given to the estimator. But it may be better, if the estimator is not moved around randomly. At the current step, the estimator is somewhere between the equation-error estimator and the output-error estimator. Setting the interpolation function to 1 will give the estimator a new start point, which is not randomly chosen and correspond to one of the actual estimator. This is what is proposed in the Proposition (2).

However, if one of the estimators stay permanently unstable, the Proposition (2) may only use the equation-error formulation since the interpolation function will remain 1. To avoid this case, a reset of the interpolation function can be made if the conditions of Proposition (2) are verified, and additionally if the interpolation function has already reached a defined value γ . This protects the algorithm from some bad start effects after resetting the interpolation function. After the interpolation function is reset, the algorithm is constrained to perform some steps. A reset will then just occur only if after these steps unstable estimators still appear. This corresponds to Proposition (3).

Each time the interpolation function is reset, the estimator jumps to the actual equation-error formulation, which may be biased. This may eventually lead to an offset in the error between the plant and the output of the adaptive filter. In order to reduce this effect, a factor $\nu = \tau^\kappa$ can be set as new start value for the reset interpolation function. This is the Proposition (4). ν should be considered as a balance factor between the two estimators for the start value after the reset. The new start value is in this case a mixture of both estimators.

The different Propositions made until now allow the interpolation function reset in case of unstable equation-error formulation or unstable output-error formulation. However the interpolation function rapidly goes over to output-error estimator. This means that the instability of the equation-error estimator does not heavily affect stability of the whole estimator after a certain number of steps, because the current estimator from the fusion block is almost influenced by the output-error

formulation alone. Therefore, it would not be wrong to reset the interpolation function just in case instability of the output-error estimator. So, other candidates of interpolation function to improve the effectiveness of the estimator are Propositions (2), (3) and (4) with an update of "the last step where \hat{B}_{EQE} or \hat{B}_{OE} was unstable" to "the last step where \hat{B}_{OE} was unstable".

6.1.3 Application on ANC algorithms

Adaptive ANC is very similar to adaptive filtering or system identification. In fact, the main difference is that in ANC, there exists a secondary path $S(z)$ also known as cancellation path. In case of system identification, the secondary path transfer function is either 1 or -1 depending on how the adaption algorithm is formulated. While the goal of the system identification is to find an \hat{H} such that

$$H(z)x(n) - \hat{H}(z)x(n) \approx 0,$$

the ANC has the same goal, but it should take the secondary path in consideration

$$P(z)x(n) + S(z)C(z)x(n) \approx 0. \quad (6.3)$$

$P(z)$ is the primary path, $S(z)$ is the secondary path, and the both form the plant. $C(z)$ is the controller to design. Let us assume that the primary path model can be decomposed in the secondary path $S(z)$ and another transfer function $Q(z)$. Then, Equation 6.3 can be reformulated as

$$\begin{aligned} P(z)x(n) + S(z)C(z)x(n) &= S(z)Q(z)x(n) + S(z)C(z)x(n) \\ &= S(z)[Q(z) + C(z)]x(n) \approx 0, \end{aligned}$$

which in case of SISO can be expressed as

$$\begin{aligned} P(z)x(n) + S(z)C(z)x(n) &= [Q(z) + C(z)]S(z)x(n) \\ &= [Q(z) + C(z)]x_f(n) \approx 0. \end{aligned}$$

This formulation shows that ANC can be considered as the identification of $Q(z)$ respectively $-Q(z)$ with $x_f(n) = S(z)x(n)$ as input signal. Hansen et al. [34] presented this for ANC with adaptive FIR filters as SISO filtered- x LMS algorithm with the corresponding block diagram in Figure 6.8. It should be noticed that the notations in Figure 6.8 are different from the notations used here. To compute

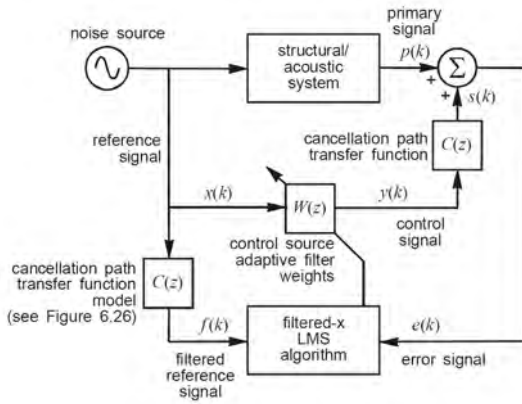
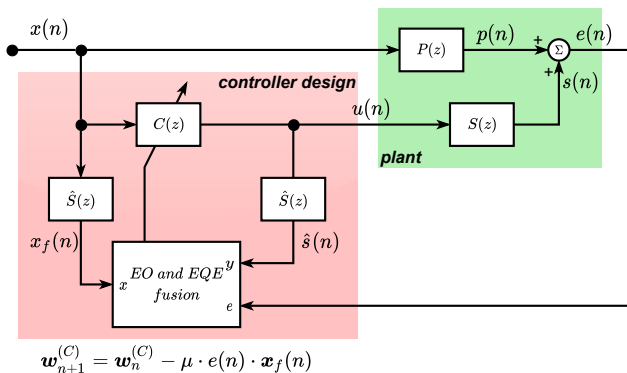


Figure 6.8: ANC block diagram with adaptive FIR filter [34]

$x_f(n)$ for the adaptive algorithm, the model of the secondary path is needed. Applying the structure of Figure 6.8 on ANC with adaptive IIR filters leads to the block diagram in Figure 6.9. In contrast to FIR adaptive filters, IIR adaptive filters need the filter output to compute the filter parameters. Therefore, the output of the real plant is modeled as

$$\hat{d}(n) = -\hat{S}(z)C(z)x(n) + e(n) \approx -s(n) + e(n) = p(n).$$

As shown in Figure 6.9, it is necessary to have the model of the secondary path $S(z)$



$$\mathbf{w}_{n+1}^{(C)} = \mathbf{w}_n^{(C)} - \mu \cdot e(n) \cdot \mathbf{x}_f(n)$$

Figure 6.9: ANC block diagram

or at least an estimation $\hat{S}(z)$ to successfully cancel the primary noise. Up on the system, the secondary path may be known respectively easy to estimate, or may be unknown respectively time-variant. Therefore, there are two types of secondary path modeling. The offline secondary path modeling is for systems where the secondary path is easy to model or easy to estimate. This can be done by deriving the differential equation describing the dynamic model of the secondary path. Another alternative called random noise injection (Figure 6.10) consists in injecting some identification noises through the secondary path without primary disturbance, then the secondary path modeling is equivalent to an estimation task respectively to an adaptive filter problem.

The secondary path may also vary, but only within a very small range. Else, the stability and the performance of the system will be badly affected. For systems with time-variant secondary path or deployed on a rough environment, the online secondary path estimation is more adequate. The most popular method is the secondary path modeling through the injection of an additional uncorrelated noise into the secondary path as perturbation signal [34]. This method is somehow

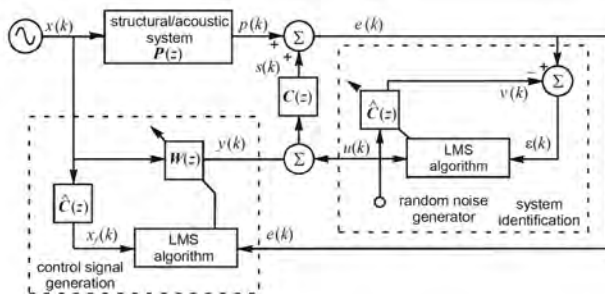


Figure 6.10: ANC with random noise injection for secondary path modeling [34]

counter-productive, since additional noises are injected to the system. Therefore, the level of the noises injected should be as low as possible. However, it should not be too low. Else, the effect of the injected noises will not be strong enough to allow an effective identification of the secondary path. Hansen et al. [34] reported that 30dB below the unwanted primary noise (around 3.2%) is low enough and can still provide a model with a suitable accuracy over a relatively long time interval.

The other method is the extended online secondary path modeling method which uses the control signal as modeling signal (Figure 6.11). The idea is to avoid adding additional noises to the system for identification purposes, and instead use the

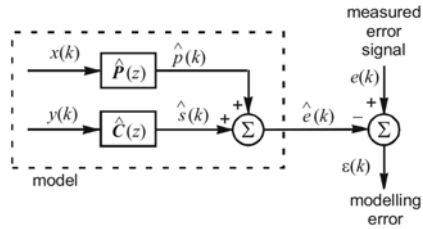


Figure 6.11: Structure used for extended least-square method [34]

primary noise to achieve the task. In [34], this is done by using an extended least-squares approach.

Each of both approaches has advantages and drawbacks. The method with injection of random noises indeed introduces additional disturbances to the system, but the model obtained for the secondary path contains information over the full frequency bandwidth of the injected noises. In contrast to this, the method of extended least-squares does not require an extra disturbance and uses the primary noise for secondary path estimation. So, the model from the secondary path obtained with this method has only information about frequencies appearing in the primary noises. So, the control system may not react well if some changes in primary disturbances frequencies occur while deploying the controller. Furthermore, the extended least-squares method needs to be updated more often than the random noise injection, because an update of the random noise injection after it has converged is necessary only in case of significant changes in the secondary path, meanwhile even a change in the primary path necessitates an update of the extended least-squares method. According to Hansen et al. [34], the random noise injection method is better in many aspects like the convergence speed of the estimator and of the controller, the reaction to changes in primary or secondary path etc. However, its implementation requires more memory than the extended least-squares method.

For the reasons mentioned above, we will use online secondary path modeling with random noise injection (Figure 6.12). For control and estimation, the Griffiths' algorithm with variable step size as presented in [58] is applied. We should pay attention, that an adaptive filter here is the equivalent of three FIR adaptive filter: one for the feedforward path \hat{A} of the filter, and two for the two formulations of the feedback path \hat{B}_{EQE} and \hat{B}_{OE} . To determine the controller using the Griffiths'

$$\begin{aligned}\mu_a^{(C)}(n) &= \alpha_a^{(C)} \mu_a^{(C)}(n-1) + \sigma_a^{(C)} \sum_{k=0}^{M_C} \left[G_{ak}^{(C)}(n) - y(n)x(n-k) \right]^2 \\ \mu_{eqeb}^{(C)}(n) &= \alpha_b^{(C)} \mu_{eqeb}^{(C)}(n-1) + \sigma_b^{(C)} \sum_{k=0}^{N_C} \left[G_{eqeb}^{(C)}(n) - y(n)\hat{d}(n-k) \right]^2 \\ \mu_{oeb}^{(C)}(n) &= \alpha_b^{(C)} \mu_{oeb}^{(C)}(n-1) + \sigma_b^{(C)} \sum_{k=0}^{N_C} \left[G_{oeb}^{(C)}(n) - y(n)\hat{y}(n-k) \right]^2\end{aligned}$$

and for the secondary path estimation:

$$\begin{aligned}a_k^{(S)}(n+1) &= a_k^{(S)}(n) + \frac{\mu_a^{(S)}}{M_S P_x^{(S)}(n)} \left[G_{ak}^{(S)}(n) - y(n)x(n-k) \right], \quad k = 0, \dots, M_S \\ b_{eqel}^{(S)}(n+1) &= b_{eqel}^{(S)}(n) + \frac{\mu_b^{(S)}}{N_S P_{\hat{d}}^{(S)}(n)} \left[G_{eqel}^{(S)}(n) - y(n)\hat{d}(n-k) \right], \quad l = 1, \dots, N_S \\ b_{oel}^{(S)}(n+1) &= b_{oel}^{(S)}(n) + \frac{\mu_b^{(S)}}{N_S P_{\hat{y}}^{(S)}(n)} \left[G_{oel}^{(S)}(n) - y(n)\hat{y}(n-k) \right], \quad l = 1, \dots, N_S\end{aligned}$$

$$\begin{aligned}G_{ak}^{(S)}(n) &= \beta_a^{(S)} G_{ak}^{(S)}(n-1) + [1 - \beta_a^{(S)}] d'(n)x(n-k) \\ G_{eqel}^{(S)}(n) &= \beta_b^{(S)} G_{eqel}^{(S)}(n-1) + [1 - \beta_b^{(S)}] d'(n)\hat{d}(n-k) \\ G_{oel}^{(S)}(n) &= \beta_b^{(S)} G_{oel}^{(S)}(n-1) + [1 - \beta_b^{(S)}] d'(n)\hat{y}(n-k)\end{aligned}$$

$$\begin{aligned}P_x^{(S)}(n) &= \gamma_a^{(S)} P_x^{(S)}(n-1) + [1 - \gamma_a^{(S)}] x^2(n) \\ P_{\hat{d}}^{(S)}(n) &= \gamma_b^{(S)} P_{\hat{d}}^{(S)}(n-1) + [1 - \gamma_b^{(S)}] \hat{d}^2(n) \\ P_{\hat{y}}^{(S)}(n) &= \gamma_b^{(S)} P_{\hat{y}}^{(S)}(n-1) + [1 - \gamma_b^{(S)}] \hat{y}^2(n)\end{aligned}$$

$$\begin{aligned}\mu_a^{(S)}(n) &= \alpha_a^{(S)} \mu_a^{(S)}(n-1) + \sigma_a^{(S)} \sum_{k=0}^{M_S} \left[G_{ak}^{(S)}(n) - y(n)x(n-k) \right]^2 \\ \mu_{eqeb}^{(S)}(n) &= \alpha_b^{(S)} \mu_{eqeb}^{(S)}(n-1) + \sigma_b^{(S)} \sum_{k=0}^{N_S} \left[G_{eqeb}^{(S)}(n) - y(n)\hat{d}(n-k) \right]^2 \\ \mu_{oeb}^{(S)}(n) &= \alpha_b^{(S)} \mu_{oeb}^{(S)}(n-1) + \sigma_b^{(S)} \sum_{k=0}^{N_S} \left[G_{oeb}^{(S)}(n) - y(n)\hat{y}(n-k) \right]^2\end{aligned}$$

In the above relations,

$$\begin{aligned}\beta_a^{(C)}, \gamma_a^{(C)}, \alpha_a^{(C)}, \sigma_a^{(C)}, \beta_b^{(C)}, \gamma_b^{(C)}, \alpha_b^{(C)}, \sigma_b^{(C)} \\ \beta_a^{(S)}, \gamma_a^{(S)}, \alpha_a^{(S)}, \sigma_a^{(S)}, \beta_b^{(S)}, \gamma_b^{(S)}, \alpha_b^{(S)}, \sigma_b^{(S)}\end{aligned}$$

are parameter that should be chosen in the interval $(0, 1)$. The choice of these parameters is system-dependent and directly affects the performance of the algorithm. There is a total of 16 parameters whose optimal values should be chosen in the interval $(0, 1)$ and depend on the system considered. Now, the question is how to choose the optimal parameters.

6.1.4 Determination of optimal parameters for ANC algorithms

As mentioned in the last subsection, the application of ANC required different parameters to be chosen in defined intervals. However, to achieve a good performance of ANC, optimal values for these parameters should be chosen, and they are system-dependent. Testing the chosen values without damaging the physical system requires a simulation. The optimal parameters should be first found for the simulation, and they will be later adjusted for the physical system. Searching the optimal parameters may be a very exhausting and time-consuming task, especially when there are a large number of parameters, and in our case, there are 16 parameters.

To find the optimal parameters for the ANC in simulations, we will proceed in a methodical way. This problem will be reformulated in a kind of optimization problem. Therefore, an adequate cost function is needed. The goal of ANC is to minimize the amplitude of the error signal. So, a kind of power formulation for the error signal can be considered as a cost function:

$$J = \sum_{n=0}^{end} e^2(n).$$

The error signal $e(n)$ depend on the parameter vector \mathbf{v} defined as

$$\mathbf{v} = \left[\beta_a^{(C)}, \gamma_a^{(C)}, \alpha_a^{(C)}, \sigma_a^{(C)}, \beta_b^{(C)}, \gamma_b^{(C)}, \alpha_b^{(C)}, \sigma_b^{(C)} \dots \right. \\ \left. \beta_a^{(S)}, \gamma_a^{(S)}, \alpha_a^{(S)}, \sigma_a^{(S)}, \beta_b^{(S)}, \gamma_b^{(S)}, \alpha_b^{(S)}, \sigma_b^{(S)} \right]^T,$$

this means that the cost function also depends on \mathbf{v} , $J = J(\mathbf{v})$. The optimization problem can be reformulated as

$$\min_{\mathbf{v}} J(\mathbf{v}), \quad J : (0, 1)^{16} \longrightarrow \mathbb{R}_+.$$

The simulation contains some random signals that are chosen as non-repeatable. This is for example the case for the random noise injected to identify the secondary path model. This means that the cost function is non-deterministic with respect to \mathbf{v} , because it often happens that

$$\mathbf{v}_1 = \mathbf{v}_2 \quad \text{but} \quad J(\mathbf{v}_1) \neq J(\mathbf{v}_2).$$

The cost function is not only dependent from the parameter vector \mathbf{v} but also from a random parameter ε :

$$J = J(\mathbf{v}, \varepsilon).$$

ε is a random parameter that changes each simulation run and characterizes the non-repeatable aspect of random signals in the simulation. It is therefore responsible for the non-deterministic characteristic of the cost function. Due to the non-deterministic aspect of the cost function, some changes are made in it. For each parameter vector \mathbf{v} , the simulation is run Q -times and for each simulation, a cost function $J_q = J(\mathbf{v}, \varepsilon_q)$ is calculated. The new cost function for \mathbf{v} is defined as

$$J(\mathbf{v}, \varepsilon) = \max\{J_1(\mathbf{v}, \varepsilon_1), \dots, J_q(\mathbf{v}, \varepsilon_q), \dots, J_Q(\mathbf{v}, \varepsilon_Q)\}$$

The idea is to choose the worst simulated case, and to find the best worst case of all parameter vectors tested. It should be noticed that the gradient of the simulation cannot be determined.

In summary, the optimal argument for a non-deterministic function whose gradient cannot be calculated should be found. For such a problem, metaheuristic procedures are a convenient option. For this purpose, a Matlab App named "Simulation Parameters - Global Optimization Toolbox" was developed with the Matlab AppDesigner Toolbox.

The App developed is a kind of optimization toolbox for simulations. It can handle non-deterministic simulations and can also use the parallel tools of Matlab to run the simulations in parallel. In the App, three algorithms have been implemented so far, namely genetic algorithm (GA), particle swarm optimization (PSO), and general pattern search optimization (GPSO). Three versions of the App were design. The first version is for optimizing a one-dimensional signal in a simulation. The second version named "Simulation Parameters - Global Optimization Toolbox - Weighted Version" is for optimizing two one-dimensional signals under consideration of given weights for each signal. Figure 6.13 shows

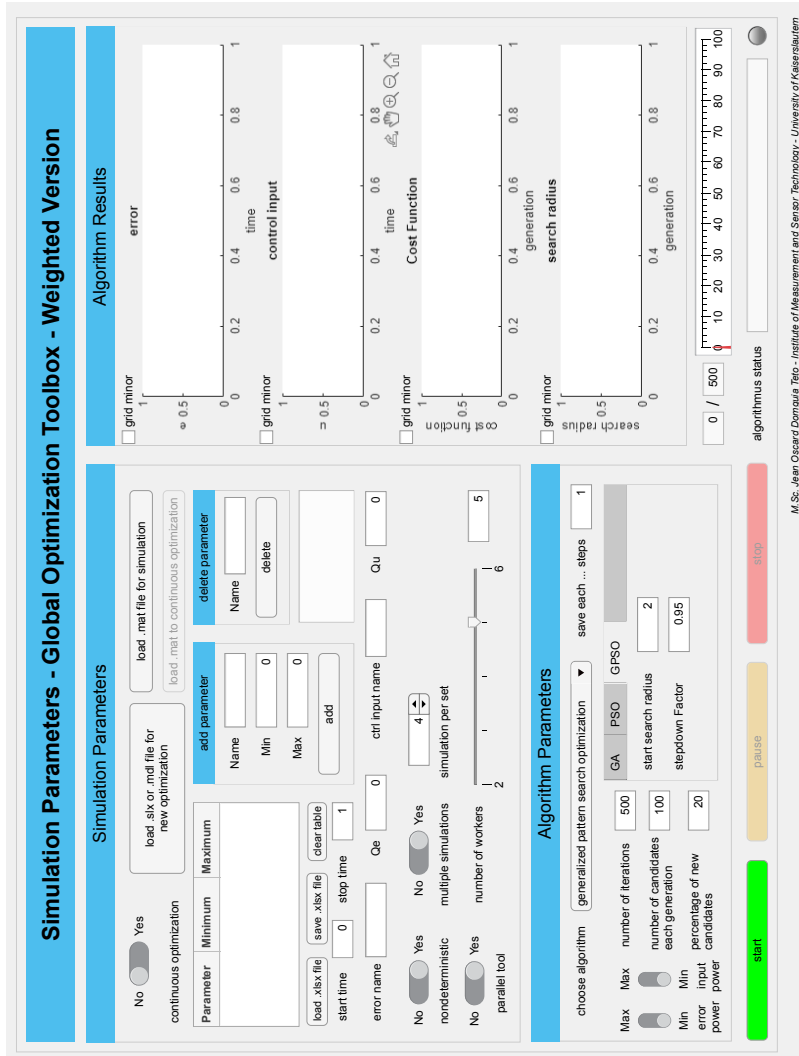


Figure 6.13: Simulation optimization app - SISO Weighted version

a screenshot of this version of the App. The third version named "Simulation Parameters - Global Optimization Toolbox - Weighted Version - MIMO" has the same function as the second version, but can also handle multidimensional signals. More

details to the App and the different versions are given in the Appendix A.1.

6.1.5 SISO-system

In this section, the result of the simulations for single suspensions corresponding to each configuration will be presented. Here, we used feedforward ANC. For that reason a feedforward signal (primary noise) and a feedback signal (error signal) should be provided to the control algorithm. This means that not only the error signal should be sensed, but also the primary noise. To minimize vibrations, three physical quantities can be considered: the position, the velocity and the acceleration. Which physical quantity is to choose depends on the application. In our case for example, the acceleration is the physical quantity that can be easily measured with low effort compared to the velocity and the position. Therefore, the acceleration is used as quantity to minimize. So, the primary noise is in this case the ground acceleration, and the acceleration measured on the payload is considered as error signal. The secondary path is the path from the control input to the error signal. In this case, the control input is the input voltage of the voice-coil motor.

The block diagram of the simulation is shown in Figure 6.14. For all configurations, a signal containing the frequencies of $5Hz$, $10Hz$, $15Hz$ and $20Hz$ with an amplitude around $21.5m/s^2$ is input as primary noise. A band-limited white noise is injected in the secondary path for identification purposes. According to Hansen et al. [34], the amplitude of the secondary noise should be chosen such that the output caused by it is only around 4% of the output caused by the primary noise. The simulation will run for 100 seconds. From the beginning, the secondary noise is injected to the system and after 30 seconds the primary noise is activated.

It should be mentioned, that the simulations designed here are only for test purposes on some arbitrarily chosen frequencies. To complete the simulation design, some tests should be done on test benches to identify which frequencies really appear in the physical system. Then, tests should be made to find if the found optimal parameters are also effective for those frequencies, or if the parameters should also be optimized for those frequencies.

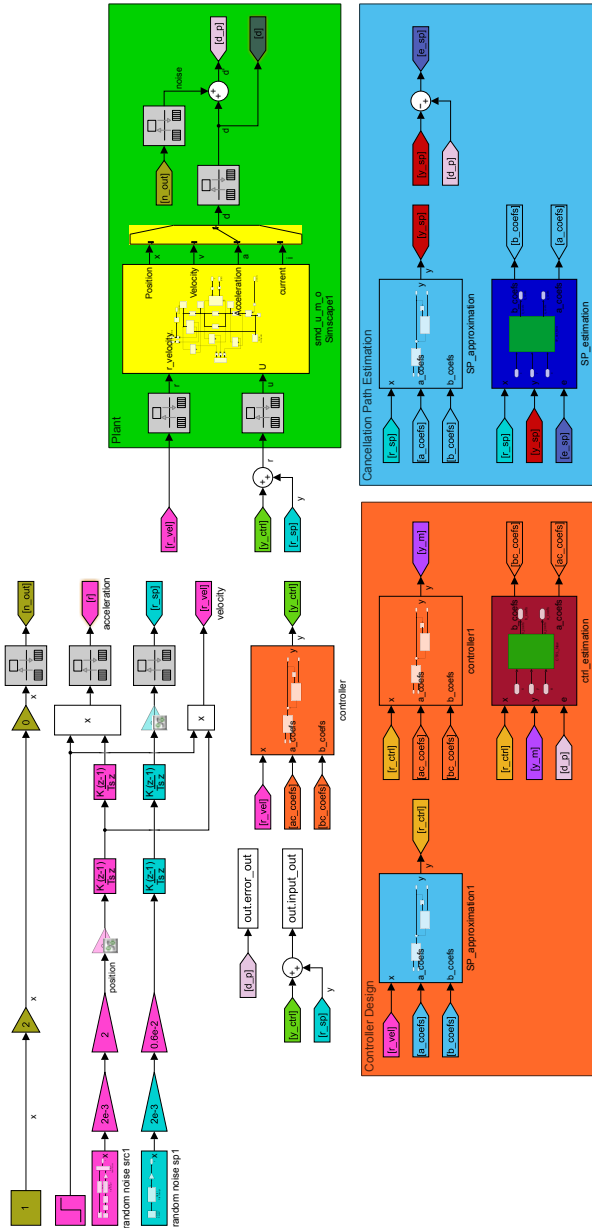


Figure 6.14: Simulink block diagram of ANC simulation

Configuration 1

Figure 6.15 shows the result of the simulation for the suspension configuration 1. In the first column, there are the plots shown in the optimization app.

For all configurations, the general pattern search algorithm is used as optimization algorithm. From the App, we get the graphs of the error and the generated input for the best candidate of the last iteration. We also get the progression of the cost function and of the search radius over the generations. On the right column, the during the simulation generated primary and secondary noises, and the resulting error with inactive ANC controller are plotted. To make the comparison easier, we have then plotted the error generated from the noises without any control strategy (black curve), and the error generated when the ANC controller is applied to the system (red curve) are plotted in the same graph.

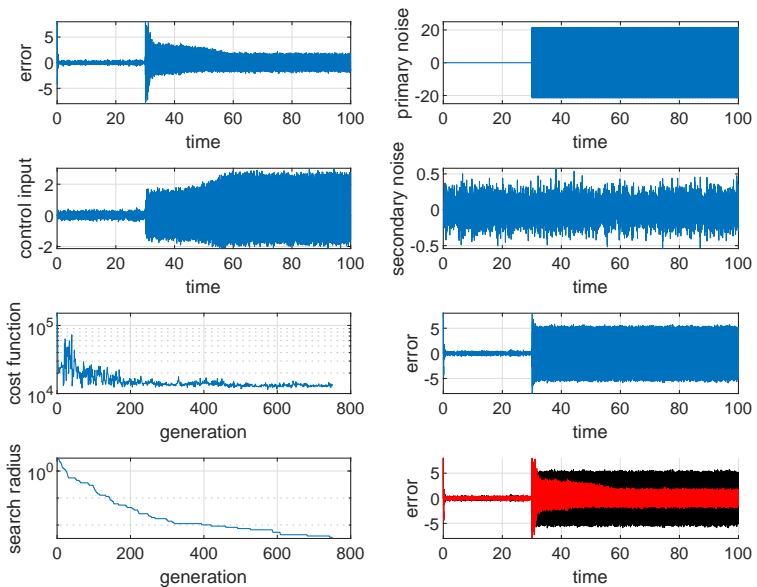


Figure 6.15: Result of the Simulation Optimization App - Configuration 1

Figure 6.15 shows that the passive behavior of the system reduces the acceleration from an amplitude around $21.5m/s^2$ to an amplitude around $5.5m/s^2$, which

corresponds to an attenuation of $11.84dB$. The application of the ANC reduces this noise to $1.90m/s^2$, which is an additional reduction of $9.23dB$. So, the total attenuation is $21.07dB$. The control signal generated by the ANC controller has an amplitude around $2.25V$ with an offset of $0.45V$.

Configuration 2

Figure 6.16 shows the result for the suspension configuration 2. The suspension passively attenuates the noise to an amplitude of $5m/s^2$ with an offset of $1m/s^2$. So, for this configuration a passive attenuation of $12.67dB$ is reached. The ANC controller leads to a reduced error signal with an amplitude around $1.2m/s^2$ with no offset. This corresponds an additional attenuation of $12.40dB$ and a total reduction of $25.07dB$. The control signal generated has an amplitude around $14.4V$ with an offset of $-0.6V$.

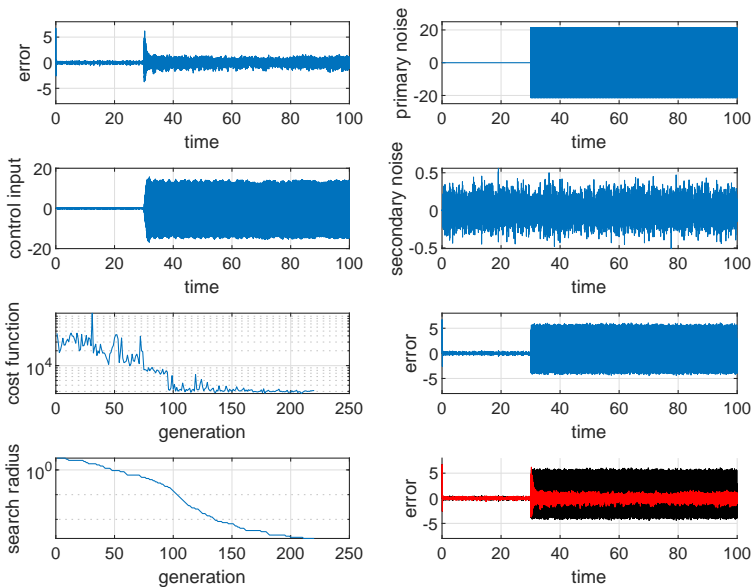


Figure 6.16: Result of the Simulation Optimization App - Configuration 2

Configuration 3

The result of the simulation of suspension system with the configuration 3 is in Figure 6.17. Without controller, the suspension was able to damp the noise to an error with an amplitude of $6.4m/s^2$ and an offset of $1m/s^2$. This corresponds to a reduction of $10.53dB$. The ANC controller reduces the error signal to an amplitude of $0.75m/s^2$, so an additional attenuation of $18.62dB$, which gives a total reduction of $29.15dB$. The control signal has an amplitude around $2.13V$ with an offset of $0.1V$.

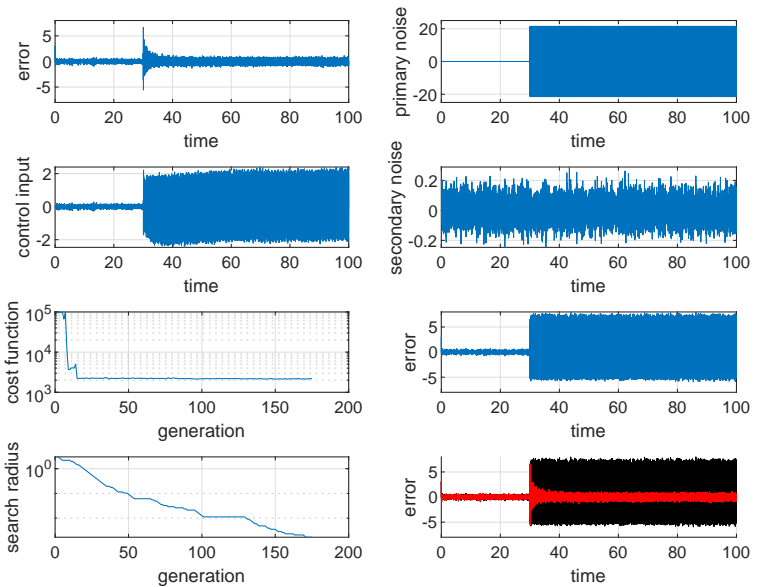


Figure 6.17: Result of the Simulation Optimization App - Configuration 3

Configuration 4

Figure 6.18 presents the result of the simulation of suspension system with the configuration 4. This configuration passively reduces the noise to an error signal with an amplitude of $5.72m/s^2$ and an offset of $1.38m/s^2$. This is an attenuation of $11.5dB$. The integration of the ANC controller to the system leads to an error signal

with an amplitude of $1m/s^2$. This an additional attenuation of $15.15dB$ and a total attenuation of $26.65dB$. The control signal generated by the ANC controller has an amplitude around $2V$ with an offset of $-0.3V$.

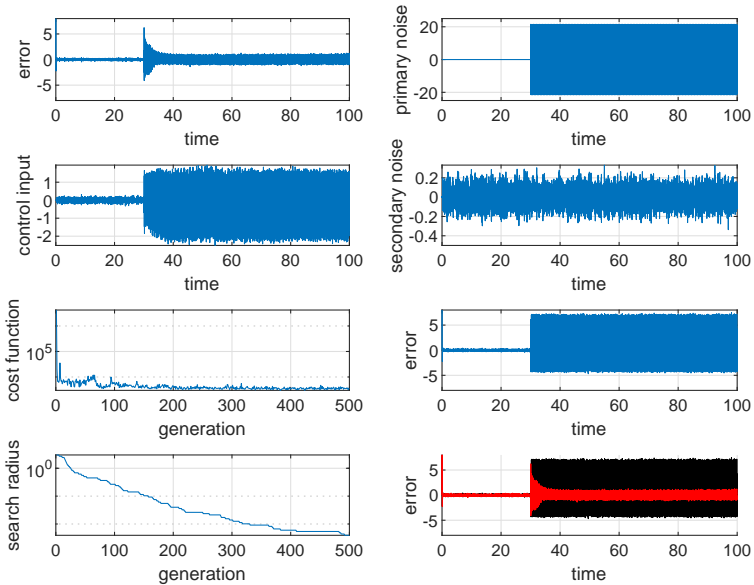


Figure 6.18: Result of the Simulation Optimization App - Configuration 4

The simulation results of the four suspension configurations are resumed in Table 6.1. We can noticed that the configurations that do not have a lower spring

Table 6.1: Simulation results of ANC on single suspensions

Configu- ration	Passive damping		Active damping		Actuation		Total attenua- tion
	Ampli- tude in m/s^2	Attenua- tion in dB	Ampli- tude in m/s^2	Attenua- tion in dB	Ampli- tude in V	Offset in V	
1	5.50	11.84	1.90	9.23	2.25	0.45	21.07
2	5.00	12.67	1.20	12.40	14.40	-0.60	25.07
3	6.40	10.53	0.75	18.62	2.13	0.10	29.15
4	5.72	11.50	1.00	15.15	2.00	-0.30	26.25

(configurations 3 and 4) have a lower passive attenuation, but a higher active attenuation with lower control effort. Configuration 2 has also a good overall attenuation, however, the control effort required is too high compared to others configurations. The found parameters were also used to run the simulations with other frequencies. They still worked, and the result was similar to the ones presented here. However, there are some frequencies for which they does not work.

6.1.6 MIMO-system

There are two possibilities to use ANC on the isolation platform (MIMO-system). The first one is to use the proposed ANC algorithm for SISO-system and to apply it on each suspension. This is a kind of decentral control, since the ANC algorithm does not consider the influences of the suspensions on each other. The second possibility is to adjust the algorithm proposed for SISO-systems to apply it on MIMO-systems. This would be the central control with ANC.

Decentral Control

For decentral control, the same ANC algorithm used for the single suspension is used. The Simulation Optimization App is used to determine the optimal parameters for the simulation for each configuration. Since simulations for single suspensions have already been optimized, the optimal values found are used as start value for the new optimization problem. Therefore, an additional option has been added to the Simulation Optimization App. This is the start value, which is optional and should be stored in the mat-file required for the simulation.

Figures 6.19, 6.20, 6.21, and 6.22 show the result of the App for decentral control of the isolation platform respectively with the suspension configurations 1, 2, 3, and 4. The first remark here is that the solutions obtained for the parameters for single suspension problem are not optimal for the decentral control problem.

Another optimization problem should be solved using the MIMO version of the Optimization App to find better parameters. The results are also shown in Table 6.2.

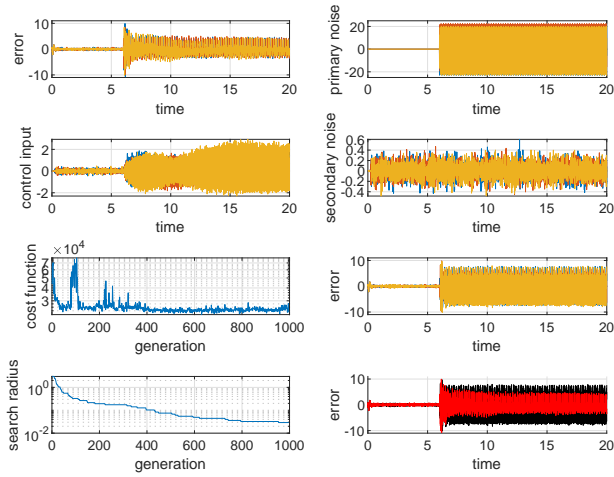


Figure 6.19: Result of Simulation Optimization App - Decentral - Configuration 1

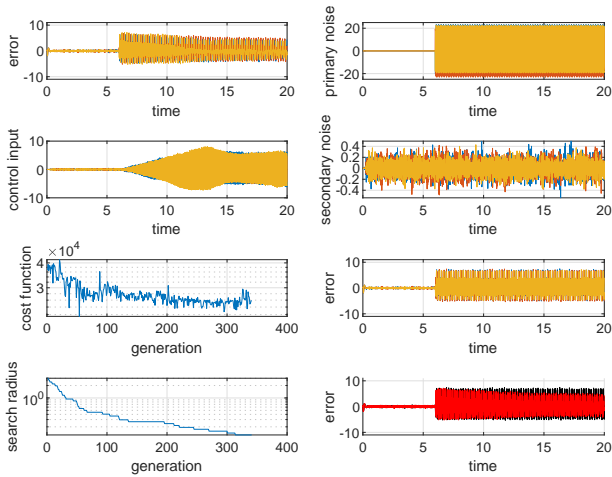


Figure 6.20: Result of Simulation Optimization App - Decentral - Configuration 2

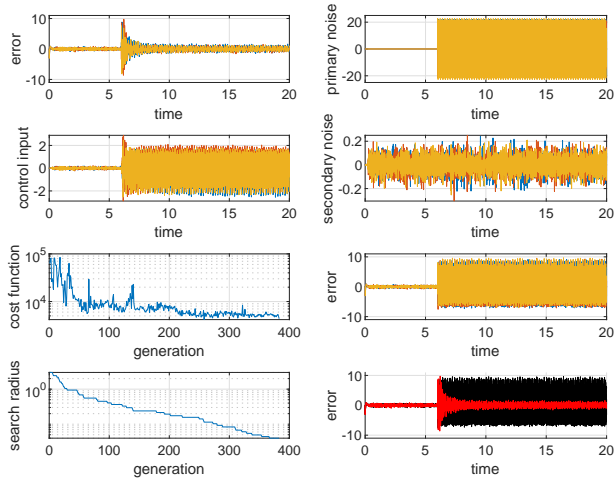


Figure 6.21: Result of Simulation Optimization App - Decentral - Configuration 3

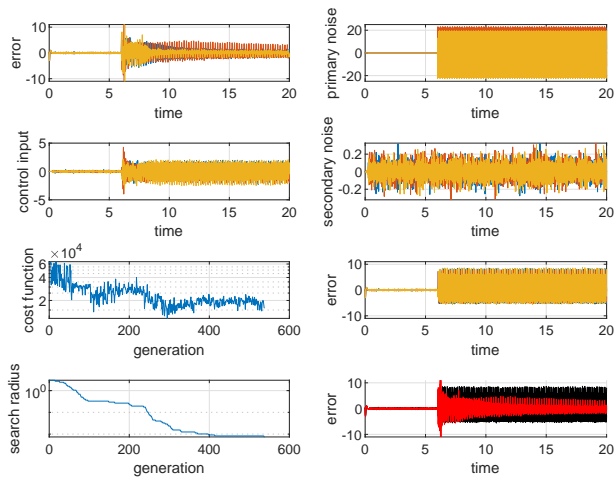


Figure 6.22: Result of Simulation Optimization App - Decentral - Configuration 4

Table 6.2: Simulation results of decentral ANC for MIMO-system

Configu- ration	Passive damping		Active damping		Actuation		Total attenua- tion
	Ampli- tude in m/s^2	Attenua- tion in dB	Ampli- tude in m/s^2	Attenua- tion in dB	Ampli- tude in V	Offset in V	
1	7.40	9.26	4.02	5.30	1.85	0.10	14.56
2	5.50	11.84	4.55	1.65	5.75	-0.25	13.49
3	7.50	9.15	1.05	17.08	2.00	0.10	26.23
4	6.70	10.13	3.14	6.58	2.00	-0.10	16.71

According to the results, the configuration 3 has the best attenuation behavior, followed by the configuration 4. In the configurations 1 and 2, the in the App used algorithm could not find reasonable parameters to increase the primary noise reduction.

Central Control

For central control with ANC, the algorithm is adapted for MIMO-systems. This leads to following reformulations for the control part:

$$\mathbf{a}_k^{(C)}(n+1) = \mathbf{a}_k^{(C)}(n) + \frac{1}{M_C P_x^{(C)}(n)} \boldsymbol{\mu}_a^C \left[\mathbf{G}_{ak}^{(C)}(n) - \mathbf{y}(n) \mathbf{x}^T(n-k) \right],$$

$$k = 0, \dots, M_C$$

$$\mathbf{b}_{eqel}^{(C)}(n+1) = \mathbf{b}_{eqel}^{(C)}(n) + \frac{1}{N_C P_d^{(C)}(n)} \boldsymbol{\mu}_b^C \left[\mathbf{G}_{eqel}^{(C)}(n) - \mathbf{y}(n) \hat{\mathbf{d}}^T(n-k) \right],$$

$$l = 1, \dots, N_C$$

$$\mathbf{b}_{oel}^{(C)}(n+1) = \mathbf{b}_{oel}^{(C)}(n) + \frac{1}{N_C P_y^{(C)}(n)} \boldsymbol{\mu}_b^{(C)} \left[\mathbf{G}_{oel}^{(C)}(n) - \mathbf{y}(n) \hat{\mathbf{y}}^T(n-k) \right],$$

$$l = 1, \dots, N_C$$

$$\mathbf{G}_{ak}^{(C)}(n) = \beta_a^{(C)} \mathbf{G}_{ak}^{(C)}(n-1) + [1 - \beta_a^{(C)}] \mathbf{d}'(n) \mathbf{x}^T(n-k)$$

$$\mathbf{G}_{eqel}^{(C)}(n) = \beta_b^{(C)} \mathbf{G}_{eqel}^{(C)}(n-1) + [1 - \beta_b^{(C)}] \mathbf{d}'(n) \hat{\mathbf{d}}^T(n-k)$$

$$\mathbf{G}_{oel}^{(C)}(n) = \beta_b^{(C)} \mathbf{G}_{oel}^{(C)}(n-1) + [1 - \beta_b^{(C)}] \mathbf{d}'(n) \hat{\mathbf{y}}^T(n-k)$$

$$P_x^{(C)}(n) = \gamma_a^{(C)} P_x^{(C)}(n-1) + [1 - \gamma_a^{(C)}] \mathbf{x}^T(n) \mathbf{x}(n)$$

$$\begin{aligned}
P_{\hat{\mathbf{d}}}^{(C)}(n) &= \gamma_b^{(C)} P_{\hat{\mathbf{d}}}^{(C)}(n-1) + \left[1 - \gamma_b^{(C)}\right] \hat{\mathbf{d}}^T(n) \hat{\mathbf{d}}(n) \\
P_{\hat{\mathbf{y}}}^{(C)}(n) &= \gamma_b^{(C)} P_{\hat{\mathbf{y}}}^{(C)}(n-1) + \left[1 - \gamma_b^{(C)}\right] \hat{\mathbf{y}}^T(n) \hat{\mathbf{y}}(n) \\
\boldsymbol{\mu}_a^{(C)}(n) &= \alpha_a^{(C)} \boldsymbol{\mu}_a^{(C)}(n-1) + \sigma_a^{(C)} \sum_{k=0}^{M_C} \left\{ \left[\mathbf{G}_{ak}^{(C)}(n) - \mathbf{y}(n) \mathbf{x}^T(n-k) \right]^T \dots \right. \\
&\quad \left. \left[\mathbf{G}_{ak}^{(C)}(n) - \mathbf{y}(n) \mathbf{x}^T(n-k) \right] \right\} \\
\boldsymbol{\mu}_{eqeb}^{(C)}(n) &= \alpha_b^{(C)} \boldsymbol{\mu}_{eqeb}^{(C)}(n-1) + \sigma_b^{(C)} \sum_{k=0}^{N_C} \left\{ \left[\mathbf{G}_{eqeb}^{(C)}(n) - \mathbf{y}(n) \hat{\mathbf{d}}^T(n-k) \right]^T \dots \right. \\
&\quad \left. \left[\mathbf{G}_{eqeb}^{(C)}(n) - \mathbf{y}(n) \hat{\mathbf{d}}^T(n-k) \right] \right\} \\
\boldsymbol{\mu}_{oeb}^{(C)}(n) &= \alpha_b^{(C)} \boldsymbol{\mu}_{oeb}^{(C)}(n-1) + \sigma_b^{(C)} \sum_{k=0}^{N_C} \left\{ \left[\mathbf{G}_{oeb}^{(C)}(n) - \mathbf{y}(n) \hat{\mathbf{y}}^T(n-k) \right]^T \dots \right. \\
&\quad \left. \left[\mathbf{G}_{oeb}^{(C)}(n) - \mathbf{y}(n) \hat{\mathbf{y}}^T(n-k) \right] \right\}
\end{aligned}$$

and to the following reformulations for the secondary path identification:

$$\begin{aligned}
\mathbf{a}_k^{(S)}(n+1) &= \mathbf{a}_k^{(S)}(n) + \frac{1}{M_S P_x^{(S)}(n)} \boldsymbol{\mu}_a^S \left[\mathbf{G}_{ak}^{(S)}(n) - \mathbf{y}(n) \mathbf{x}^T(n-k) \right], \\
k &= 0, \dots, M_S \\
\mathbf{b}_{eqel}^{(S)}(n+1) &= \mathbf{b}_{eqel}^{(S)}(n) + \frac{1}{N_S P_{\hat{\mathbf{d}}}^{(S)}(n)} \boldsymbol{\mu}_b^S \left[\mathbf{G}_{eqel}^{(S)}(n) - \mathbf{y}(n) \hat{\mathbf{d}}^T(n-k) \right], \\
l &= 1, \dots, N_S \\
\mathbf{b}_{oel}^{(S)}(n+1) &= \mathbf{b}_{oel}^{(S)}(n) + \frac{1}{N_S P_{\hat{\mathbf{y}}}^{(S)}(n)} \boldsymbol{\mu}_b^S \left[\mathbf{G}_{oel}^{(S)}(n) - \mathbf{y}(n) \hat{\mathbf{y}}^T(n-k) \right], \\
l &= 1, \dots, N_S \\
\mathbf{G}_{ak}^{(S)}(n) &= \beta_a^{(S)} \mathbf{G}_{ak}^{(S)}(n-1) + \left[1 - \beta_a^{(S)}\right] \mathbf{d}^T(n) \mathbf{x}^T(n-k) \\
\mathbf{G}_{eqel}^{(S)}(n) &= \beta_b^{(S)} \mathbf{G}_{eqel}^{(S)}(n-1) + \left[1 - \beta_b^{(S)}\right] \mathbf{d}^T(n) \hat{\mathbf{d}}^T(n-k) \\
\mathbf{G}_{oel}^{(S)}(n) &= \beta_b^{(S)} \mathbf{G}_{oel}^{(S)}(n-1) + \left[1 - \beta_b^{(S)}\right] \mathbf{d}^T(n) \hat{\mathbf{y}}^T(n-k) \\
P_x^{(S)}(n) &= \gamma_a^{(S)} P_x^{(S)}(n-1) + \left[1 - \gamma_a^{(S)}\right] \mathbf{x}^T(n) \mathbf{x}(n) \\
P_{\hat{\mathbf{d}}}^{(S)}(n) &= \gamma_b^{(S)} P_{\hat{\mathbf{d}}}^{(S)}(n-1) + \left[1 - \gamma_b^{(S)}\right] \hat{\mathbf{d}}^T(n) \hat{\mathbf{d}}(n) \\
P_{\hat{\mathbf{y}}}^{(S)}(n) &= \gamma_b^{(S)} P_{\hat{\mathbf{y}}}^{(S)}(n-1) + \left[1 - \gamma_b^{(S)}\right] \hat{\mathbf{y}}^T(n) \hat{\mathbf{y}}(n) \\
\boldsymbol{\mu}_a^{(S)}(n) &= \alpha_a^{(S)} \boldsymbol{\mu}_a^{(S)}(n-1) + \sigma_a^{(S)} \sum_{k=0}^{M_S} \left\{ \left[\mathbf{G}_{ak}^{(S)}(n) - \mathbf{y}(n) \mathbf{x}^T(n-k) \right]^T \dots \right.
\end{aligned}$$

$$\left[\mathbf{G}_{ak}^{(S)}(n) - \mathbf{y}(n)\mathbf{x}^T(n-k) \right] \Big\} \\ \boldsymbol{\mu}_{eqeb}^{(S)}(n) = \alpha_b^{(S)} \boldsymbol{\mu}_{eqeb}^{(S)}(n-1) + \sigma_b^{(S)} \sum_{k=0}^{N_S} \left\{ \left[\mathbf{G}_{eqeb}^{(S)}(n) - \mathbf{y}(n)\hat{\mathbf{d}}^T(n-k) \right]^T \dots \right. \\ \left. \left[\mathbf{G}_{eqeb}^{(S)}(n) - \mathbf{y}(n)\hat{\mathbf{d}}^T(n-k) \right] \right\} \\ \boldsymbol{\mu}_{oeb}^{(S)}(n) = \alpha_b^{(S)} \boldsymbol{\mu}_{oeb}^{(S)}(n-1) + \sigma_b^{(S)} \sum_{k=0}^{N_S} \left\{ \left[\mathbf{G}_{oeb}^{(S)}(n) - \mathbf{y}(n)\hat{\mathbf{y}}^T(n-k) \right]^T \dots \right. \\ \left. \left[\mathbf{G}_{oeb}^{(S)}(n) - \mathbf{y}(n)\hat{\mathbf{y}}^T(n-k) \right] \right\}$$

Like in SISO-case, the parameters

$$\beta_a^{(C)}, \gamma_a^{(C)}, \alpha_a^{(C)}, \sigma_a^{(C)}, \beta_b^{(C)}, \gamma_b^{(C)}, \alpha_b^{(C)}, \sigma_b^{(C)} \\ \beta_a^{(S)}, \gamma_a^{(S)}, \alpha_a^{(S)}, \sigma_a^{(S)}, \beta_b^{(S)}, \gamma_b^{(S)}, \alpha_b^{(S)}, \sigma_b^{(S)}$$

should be chosen in the interval $(0, 1)$. Here, we should notice, that the formulations above were written intuitively based on SISO ANC, and a formal derivation is not available.

The parameter optimization for each configuration is done using the MIMO version of the Simulation Optimization App. The results of the Optimization App for each configuration are shown in Figures 6.23, 6.24, 6.25, and 6.26. Table 6.3 gives an overview of the results in a quantitative form.

Table 6.3: Simulation results of central ANC for MIMO-system

Configu- ration	Passive damping		Active damping		Actuation		Total attenua- tion
	Ampli- tude in m/s^2	Attenua- tion in dB	Ampli- tude in m/s^2	Attenua- tion in dB	Ampli- tude in V	Offset in V	
1	7.40	9.26	4.50	4.32	1.50	0.03	13.58
2	5.50	11.84	6.10	-0.90	15.00	-0.02	10.94
3	7.50	9.15	4.90	3.69	1.90	0.04	12.84
4	6.70	10.13	5.00	2.54	1.75	-0.05	12.67

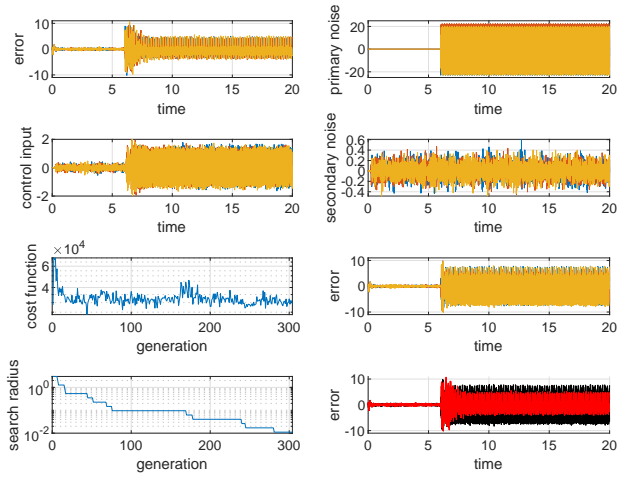


Figure 6.23: Result of Simulation Optimization App - Central - Configuration 1

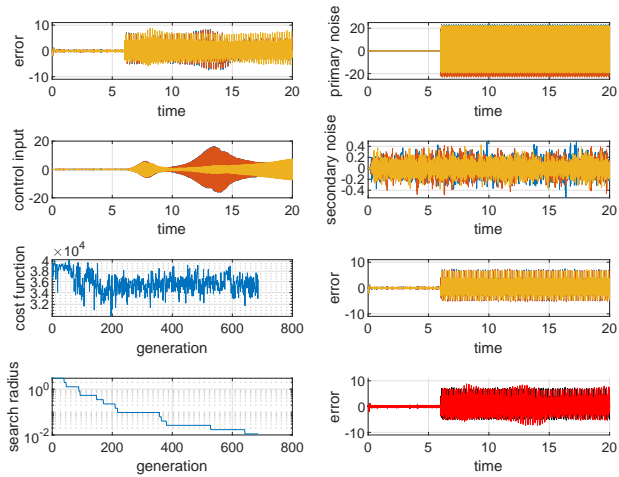


Figure 6.24: Result of Simulation Optimization App - Central - Configuration 2

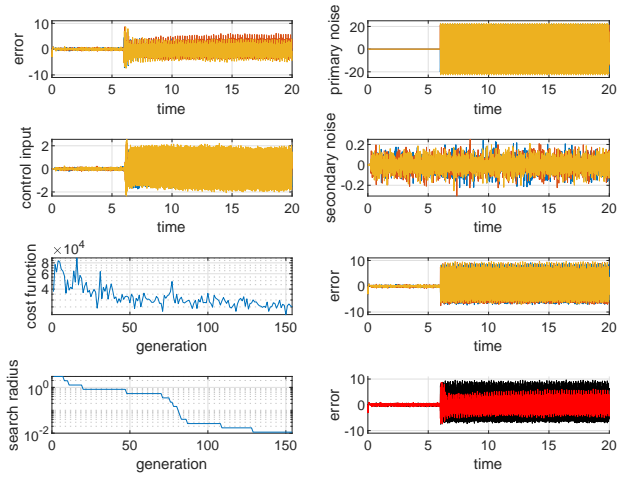


Figure 6.25: Result of Simulation Optimization App - Central - Configuration 3

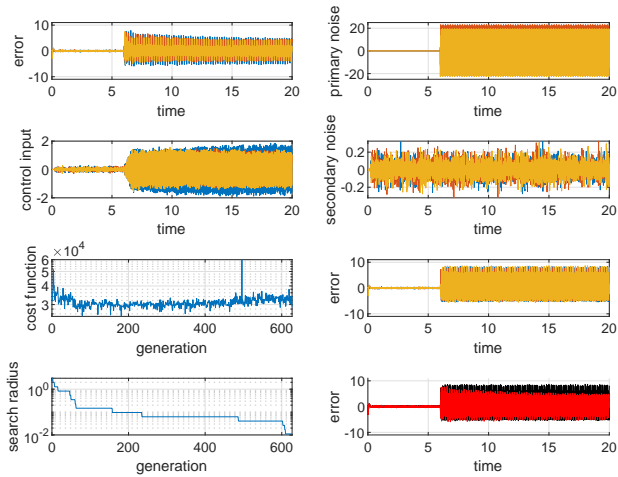


Figure 6.26: Result of Simulation Optimization App - Central - Configuration 4

The results show that the configuration 1 has the best noise reduction. For the configuration 2, the App could not find parameter to achieve noise reduction with the active part of the system. What can be observed here is that central ANC was not able to achieve better noise reduction than decentral ANC.

6.2 Robust control

Robust control is the second alternative we have chosen for suspension control. In contrast to feedforward ANC, robust controllers based on \mathcal{H}_∞ optimization do not need the primary noise signal to be measured, and the controller is not adaptive. Therefore, it does not need as much computational resources as feedforward ANC to be implemented. The main goal is to design a controller that will ensure the robust stability and the robust performance of the system. To test the functionality of robust control on our system, we will first design robust controllers for each suspension configuration considered as SISO-system. Then, they will be applied on the whole platform considered as MIMO-system.

6.2.1 SISO-system

In order to design a robust controller for each suspension, the suspension dynamic is reformulated in form of a generalized plant as shown in Figure 6.27.

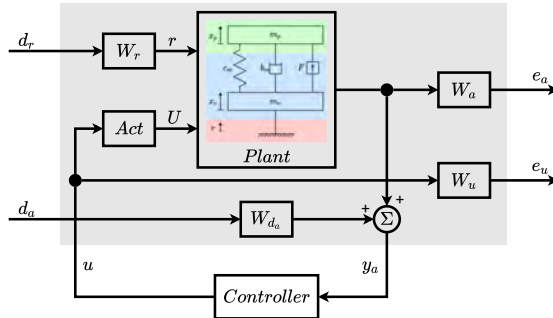


Figure 6.27: Generalized plant for single suspension

As weighted exogenous outputs, we consider the signals e_a and e_u , which are respectively the weighted payload acceleration and the weighted control input. The

weights W_a and W_u are kind of performance weighting functions which implicitly specify some design requirements. At the very beginning, the idea were to consider the ground acceleration as system input. However, the generalized plant derived using ground acceleration as system input did not fulfill all necessary requirements for a controller design with this method (e.g. controllability, observability). For that reason, the generalized plant is considered with the ground displacement r as suspension input.

Robust controllers should be designed for the four suspension configurations. However, the generalized plants of the configurations 3 and for 4 have implicit dynamics in form of

$$\begin{cases} E\dot{x} = Ax + Bu \\ y = Cx + Du \end{cases},$$

where E is a singular matrix. Therefore, their dynamic cannot be transformed in an explicit dynamic, and this is an obstacle to the design of an \mathcal{H}_∞ controller. So, for the robust controller design using \mathcal{H}_∞ optimization based methods, only configurations 1 and 2 are concerned.

The first step to describe the generalized plant is the definition of the uncertainties and the different weight functions. As plant uncertainties, we considered the payload with an uncertainty of $\pm 20\%$, and the motor damping with an uncertainty of $\pm 50\%$. The weight function W_a of the acceleration output is a kind of performance weight function defined in this case as

$$W_a(s) = 1.67 \cdot \frac{s + 173.21}{s + 0.8661}.$$

The weight W_u of the control effort is defined a high pass with

$$W_u(s) = 0.1 \cdot \frac{s + 5.44}{s + 1088}.$$

The weight functions of the input disturbance and sensor noises are chosen as constants:

$$W_r = 0.005, \quad W_{da} = 0.5.$$

In the block *Act*, the uncertain dynamic of the actuator is considered. The dynamic of the voice-coil actuator itself has already been considered in the plant dynamic. As dynamic uncertainty, we consider a multiplicative uncertainty coupled with a delay

of 1 *m.s.* The nominal dynamic of this block is

$$Act(s) = \overline{W}_{unc}(s) = \frac{2000}{s + 2000}.$$

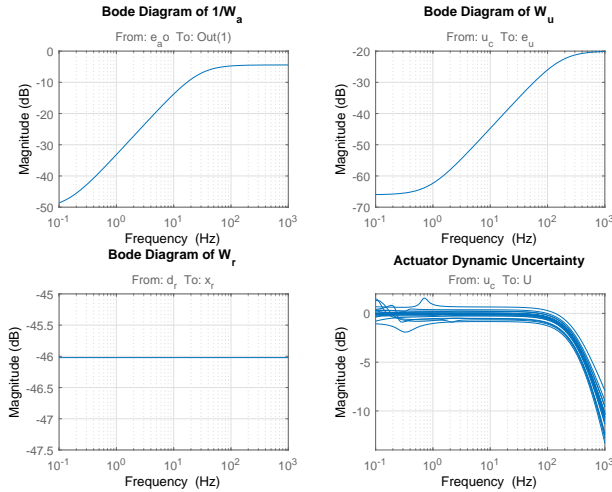


Figure 6.28: Bode diagram of weight functions

Figure 6.28 shows the bode diagrams of the different weight functions and the multiplicative uncertainty of the actuator dynamic.

To design the controller, the Matlab functions for \mathcal{H}_∞ -synthesis, and for the μ -synthesis with μ - K iteration and D - K iteration are used.

Configuration 1

As mentioned above, three \mathcal{H}_∞ -based controllers were designed: K_{H_∞} , $K_{\mu K}$ and K_{DK} . They are all robust stable but the robust performance is not guaranteed.

Figure 6.29 shows the different singular values for each controller. The left row is for the robust stability and the right row for the robust performance.

The designed controllers have the orders

$$N_{K-H_\infty} = 10, \quad N_{K-\mu K} = 19, \quad \text{and} \quad N_{K-DK} = 25.$$

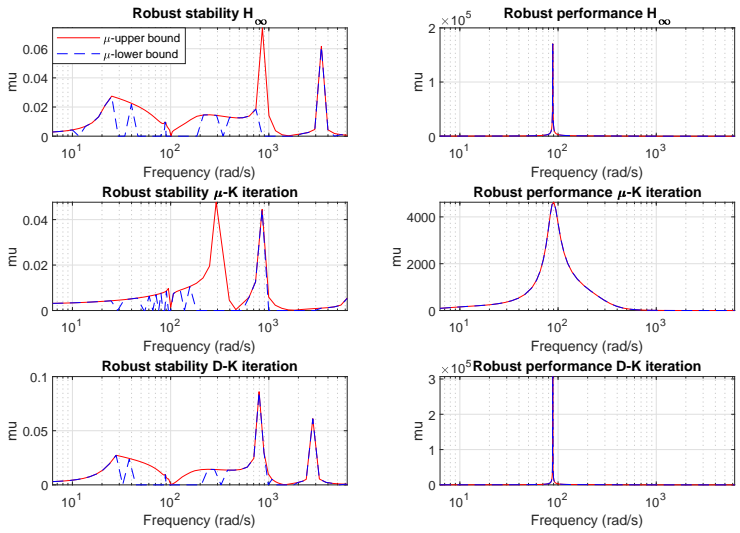


Figure 6.29: Singular value for robust stability (left) and performance (right) - Configuration 1

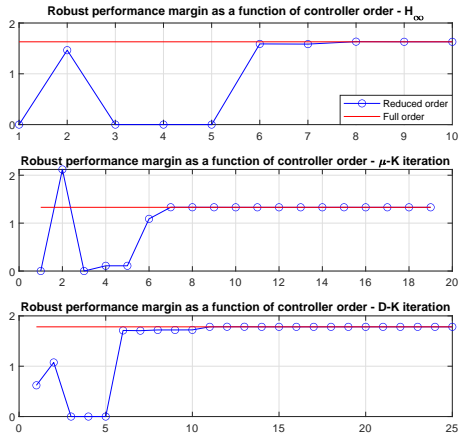


Figure 6.30: Controller order reduction - Configuration 1

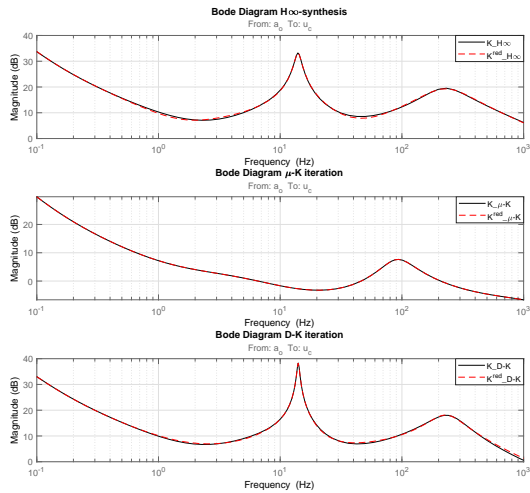


Figure 6.31: Bode diagram of designed controllers - Configuration 1

For implementation, it is desirable to have controllers with low orders. Therefore, an order reduction is applied on the controllers (Figure 6.30). The aim of the order reduction is to reduce the order of a transfer function while preserving its main characteristics. After the order reduction, the order of the new controllers are

$$N_{K-H\infty}^{red} = 7, \quad N_{K-\mu K}^{red} = 8, \quad \text{and} \quad N_{K-DK}^{red} = 7.$$

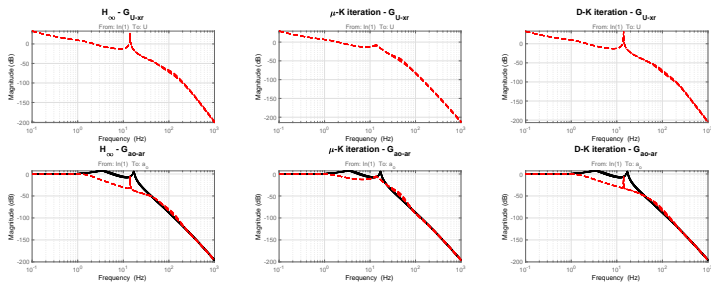


Figure 6.32: Bode diagram of closed-loop - Configuration 1

The bode diagrams of the designed controllers are shown in Figure 6.31. The bode diagrams of the closed-loop system with each controller are shown in Figure 6.32.

Configuration 2

The same procedure is applied for the suspension configuration 2. Like for configuration 1, the designed controllers guarantee the robust stability but not the robust performance (Figure 6.33). The orders of the initial controllers are

$$N_{K-H\infty} = 8, \quad N_{K-\mu K} = 37, \quad \text{and} \quad N_{K-DK} = 35.$$

After applying the order reduction on them (Figure 6.34), their orders become

$$N_{K-H\infty}^{\text{red}} = 7, \quad N_{K-\mu K}^{\text{red}} = 7, \quad \text{and} \quad N_{K-DK}^{\text{red}} = 10.$$

The bode diagrams of the closed-loop system with each controller are shown in Figure 6.36.

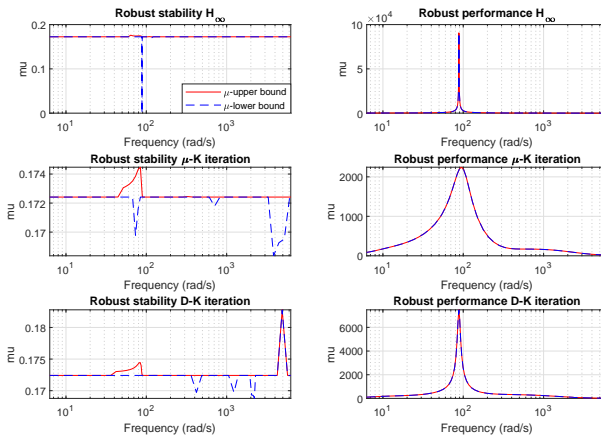


Figure 6.33: Singular value for robust stability (left) and performance (right) - Configuration 2

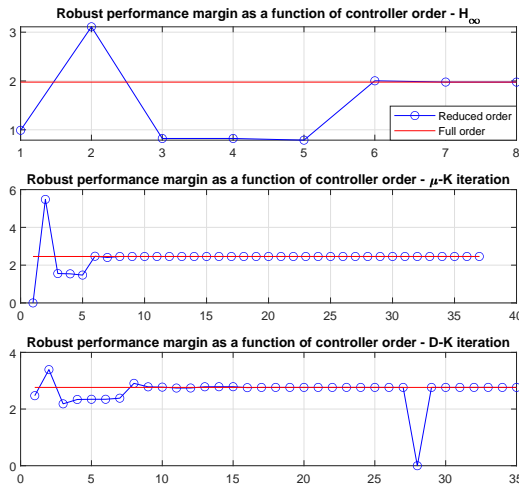


Figure 6.34: Controller order reduction - Configuration 2

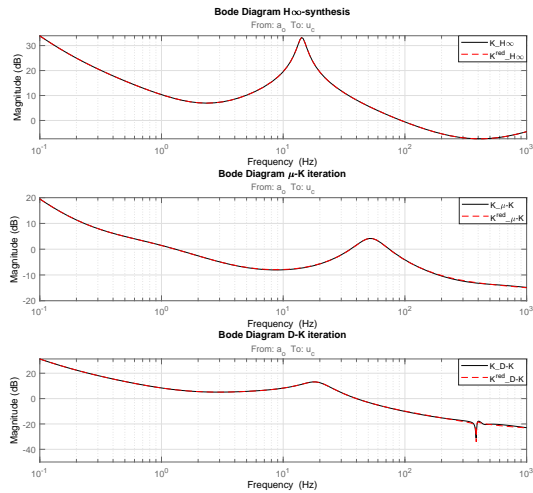


Figure 6.35: Bode diagram of designed controllers - Configuration 2

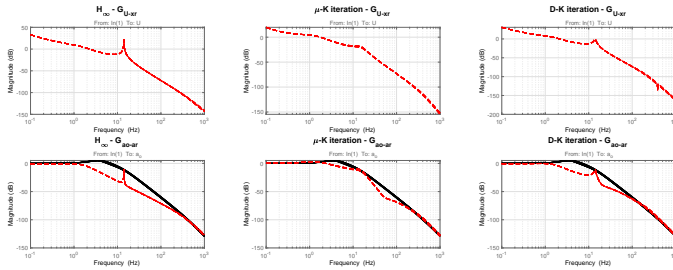


Figure 6.36: Bode diagram of closed-loop - Configuration 2

In summary, three types of \mathcal{H}_∞ -based controllers were design for the robust control of single suspensions. The bode diagrams of the closed-loops (Figures 6.31 and 6.35) show that the controllers designed with \mathcal{H}_∞ -synthesis (K_{H_∞}) have the best damping behavior and the ones designed with the μ - K iteration ($K_{\mu K}$) the worst. However, the singular values (Figures 6.29 and 6.33) show that the K_{H_∞} have the worse performance behavior, and $K_{\mu K}$ the best. The controllers designed with the D - K iteration K_{DK} seem to be a good compromise. Considering Figures 6.32 and 6.36, we can say that the damping behavior of the closed-loop is better for the configuration 2 than for the configuration 1.

6.2.2 MIMO-system

For the three dimensional platform, two control structures are applied: decentral structure and central structure.

Decentral control

For the decentral control, no new controllers are designed. Instead, the controllers designed for single suspensions are applied on each suspension of the isolation platform. Since the controllers were designed with position excitation as input, we also use the dynamic which has the position excitation as disturbance input, and we apply the reduced order controllers. Figures 6.37, 6.38 and 6.39 show the bode diagrams of the closed-loop systems with the suspension configuration 1, with respectively \mathcal{H}_∞ controller, μ - K controller and D - K controller. Figures 6.40, 6.41 and 6.42 show the same diagrams for the suspension configuration 2. Although the

positions x_T is written as input in the bode diagrams, the accelerations a_T are considered as input to plot these diagrams.

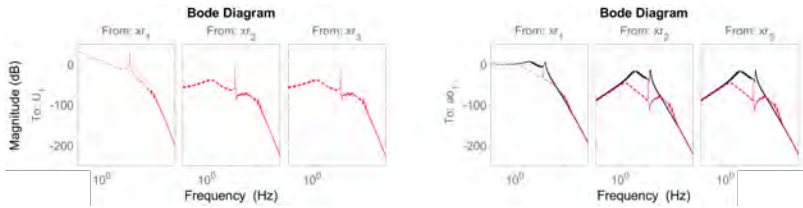


Figure 6.37: Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Decentral control - Configuration 1

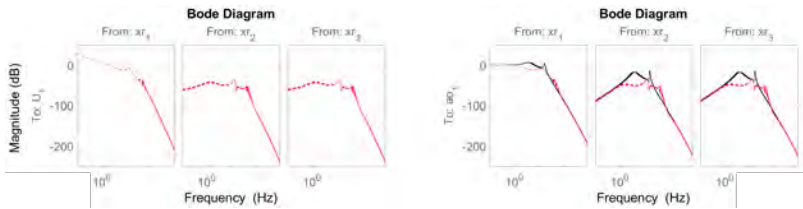


Figure 6.38: Bode diagram of closed-loop of MIMO system - μ - K iteration - Decentral control - Configuration 1

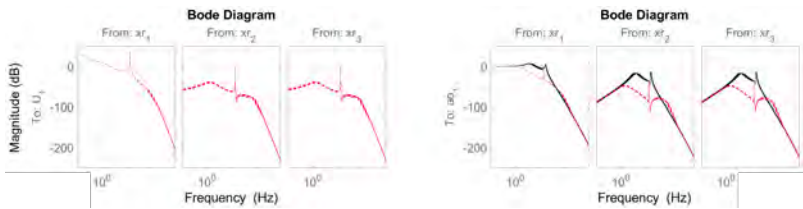


Figure 6.39: Bode diagram of closed-loop of MIMO system - D - K iteration - Decentral control - Configuration 1

The results of the bode diagrams show that the decentral controllers stabilize the isolation platform and have a good damping behavior. Like for single suspension, the robust stability is guaranteed, but not the robust performance, and the best tradeoff remains the controllers designed with the D - K iteration.

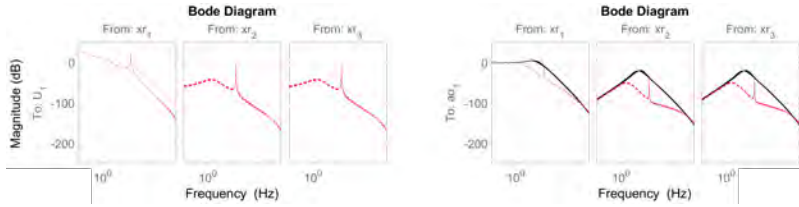


Figure 6.40: Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Decentral control - Configuration 2

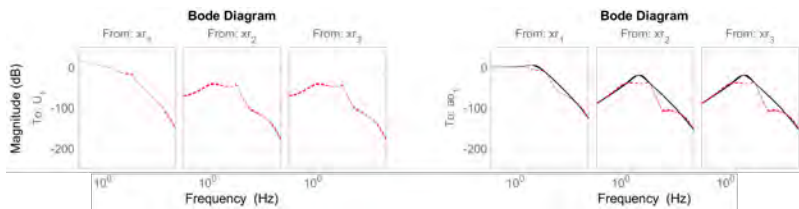


Figure 6.41: Bode diagram of closed-loop of MIMO system - μ - K iteration - Decentral control - Configuration 2

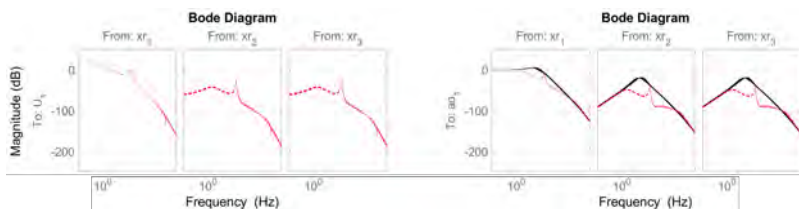


Figure 6.42: Bode diagram of closed-loop of MIMO system - D - K iteration - Decentral control - Configuration 2

Central control

In the central control structure, not only single suspensions are considered for controller design, but the whole system dynamic. The same weight functions as for single suspensions are used here. The three types of \mathcal{H}_∞ -based controllers used for single suspensions are designed for the MIMO-system. They are all robust stable but do not achieve the robust performance as their singular values show in Figures

6.44 and 6.48. After controller order reduction, we get controllers with orders

$$N_{K-H\infty}^{red} = 19, \quad N_{K-\mu K}^{red} = 24, \quad \text{and} \quad N_{K-DK}^{red} = 19$$

for configuration 1, and

$$N_{K-H\infty}^{red} = 16, \quad N_{K-\mu K}^{red} = 22, \quad \text{and} \quad N_{K-DK}^{red} = 24$$

for configuration 2. Figures 6.45, 6.46 and 6.47 show the bode diagrams of closed-loop systems for each controller with configuration 1, and Figures 6.50, 6.51 and 6.52 show the same for configuration 2. Like in other control structures, controllers designed with D - K iteration give the best tradeoff between the robust stability and the robust performance.

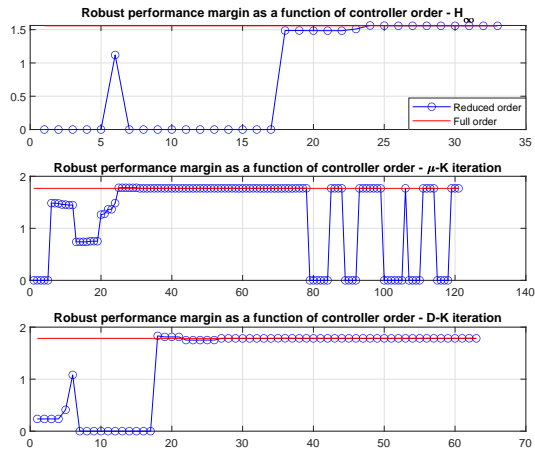


Figure 6.43: Controller order reduction - Central control - Configuration 1

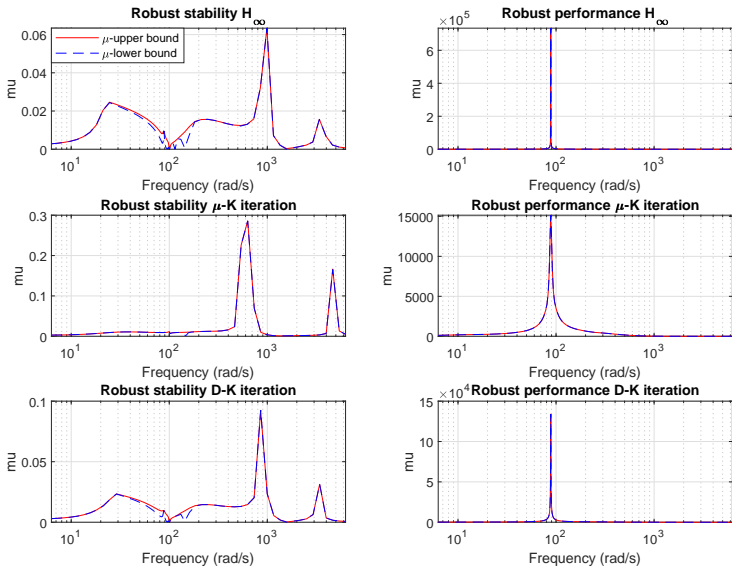


Figure 6.44: Singular value for robust stability (left) and performance (right) - Central control - Configuration 1

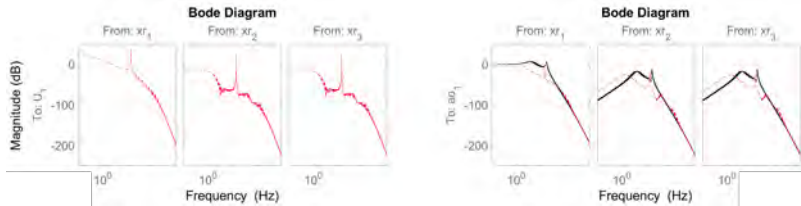


Figure 6.45: Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Central control - Configuration 1

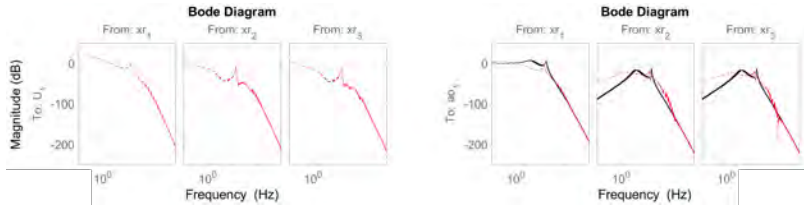


Figure 6.46: Bode diagram of closed-loop of MIMO system - μ -K iteration - Central control - Configuration 1

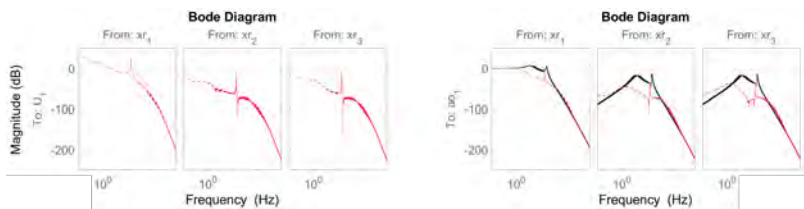


Figure 6.47: Bode diagram of closed-loop of MIMO system - D-K iteration - Central control - Configuration 1

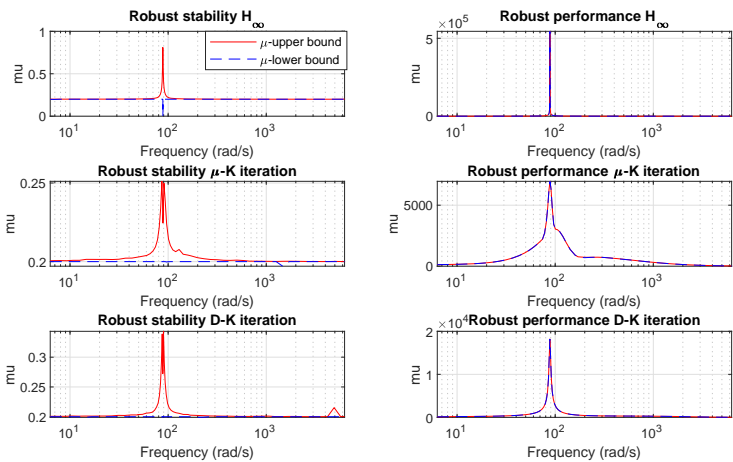


Figure 6.48: Singular value for robust stability (left) and performance (right) - Central control - Configuration 2

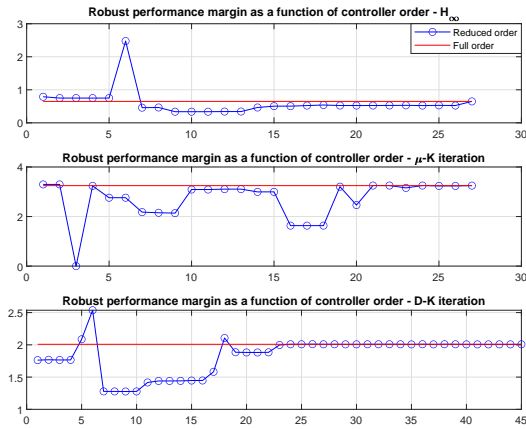


Figure 6.49: Controller order reduction - Central control - Configuration 2

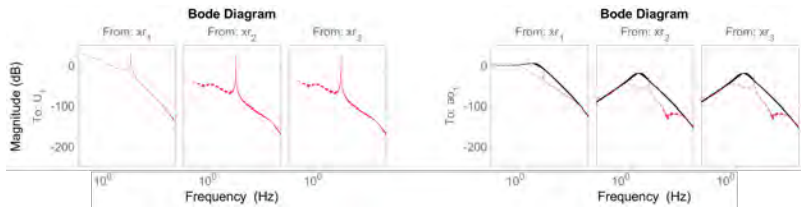


Figure 6.50: Bode diagram of closed-loop of MIMO system - H_∞ - Central control - Configuration 2

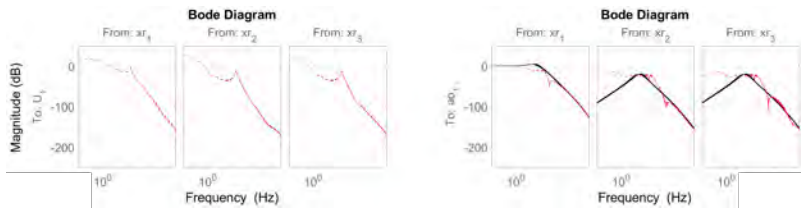


Figure 6.51: Bode diagram of closed-loop of MIMO system - μ -K iteration - Central control - Configuration 2

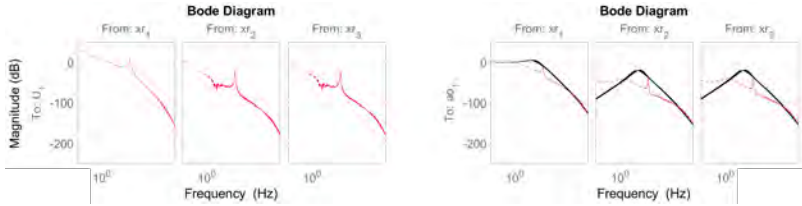


Figure 6.52: Bode diagram of closed-loop of MIMO system - D - K iteration - Central control - Configuration 2

Out of this section about robust control, we can say that the controllers designed guaranteed the robust stability but not the robust performance. A crucial element of robust control with \mathcal{H}_∞ optimization is the choice of the weight functions. However, there are no formal procedures describing how to choose these functions, just some indications on how they should look like. The choice of good weight functions required a good knowledge of the physical system and a good experience of the control engineer. And even if these criteria are met, they do not guarantee the choice of the optimal weight functions for the controller design.

From the three types of \mathcal{H}_∞ -based controllers designed here, the μ -synthesis with D - K iteration had the best results. The closed-loop behavior was better for decentral control than for central control. This is a bit surprising because the central control is designed based on the whole system dynamic including the influence of the suspensions on each other, which is not taken into account while designing decentral controllers. This may be due to the inappropriate choice of the weighting functions.

6.3 Summary

The aim of this Chapter was the design and simulation of controllers for the vibration control on the isolation platform. Two types of control strategies have been used. The first one is the ANC. For ANC, an algorithm based on adaptive IIR filters has been proposed. With this algorithm, the chances of landing on a local optimum or having a biased optimum while solving the optimization problem of adaptive IIR filtering should be reduced. Since the stability of adaptive IIR filters is not guaranteed, propositions have been made to monitor the stability during the adaptation. The adaptation algorithm contains many parameters whose values are not clearly specified, and are case-dependent. Only the intervals in which the values

should be chosen are given. For an application with many parameters, find the optimal parameters without any tool is very time consuming, and it is not guaranteed that good parameters will be found. This issue was solved by reformulating the problem as an optimization problem w.r.t the parameters. Therefore, a Matlab-App has been developed. This App is designed to find the optimal parameters in simulation (Simulink) optimizing specific signals. There are three versions of the App depending on the application.

The Optimization App was used for parameter optimization for the simulation of ANC controllers. The application of ANC for single suspensions gave good results, especially for the configuration 3. The configuration 2 also achieved a good attenuation, but it needs much more energy compared to the others configurations. This results were summarized in Table 6.1.

ANC has also been applied on the whole platform. Since the platform is a MIMO system, there were two possibility. The first one is the decentral control. The parameters found for the single suspension did not worked for decentral control. Another parameter optimization problem had to be solved with the optimization App to find better parameter values. The results are shown in Table 6.2. All configurations could achieved an active attenuation except the configuration 2. The best attenuation was achieved by the configuration 3.

For central ANC, an algorithm for MIMO system were proposed based on SISO ANC algorithm. Similar to decentral ANC, all configurations could achieved an active attenuation except the configuration 2. The best configuration was achieved by the configuration 1. Overall, using ANC, the decentral control has achieved better results than the central control.

The second control strategy used was robust control. This was only possible for the configurations 1 and 2. Three types of \mathcal{H}_∞ -based controllers were used: \mathcal{H}_∞ -synthesis, μ -synthesis with μ - K iteration, and with D - K iteration. The analysis of controllers for single suspensions showed that the \mathcal{H}_∞ -synthesis controllers had the best nominal damping, but the worst performance. In contrast, the μ - K iteration controllers had the best performance and the worst damping behavior. The D - K iteration controllers appeared like a compromise between \mathcal{H}_∞ -synthesis and μ - K iteration. The robust control design for the whole platform gave similar results for decentral control as well as central control. The results of the decentral control were even better than the results of the central control.

After controller design, the next step of the development process for control system is the controller implementation. This is theme of the next Chapter.

7 Hardware implementation

The three first steps of the development process for control systems have been worked through in Chapters 5 and 6. In this chapter, we will deal with a part of the controller implementation, which is next stage of the control system development procedure. Our focus here is on the On-Target Rapid Control Prototyping (OTRCP). The objective of the OTRCP is the design and the analysis of the controller under real environment conditions and with real target hardware. Therefore, the controller should be implemented by code generation on a Target system, and the real plant is considered. OTRCP allows data logging and visualization blocks so that further analysis of the system is possible. However, it is not optimal from memory usage and stability point of view [32]. Figure 7.1 shows the signal flow of OTRCP for our control system. The control algorithm designed in Matlab/Simulink is flashed

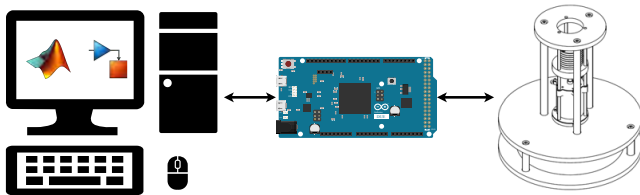


Figure 7.1: On-Target Rapid Control Prototyping

on a processing unit. The processing unit receives information from sensors. Based on the algorithm flashed on it, it should process the received data and generate a control signal for the actuators. For analysis purposes, the connection and the signal flow between the development computer and the processing unit should remain active. However, this causes an additional latency, which can affect the performance of the system.

7.1 Components

For the implementation of a control system, three main components are needed: the processing unit, the sensors, and the actuators.

Processing unit

The processing unit is the central component of a control system. It must manage and regulate all communications between the different components and perform the operations and calculations implemented in the control algorithm. One of the requirements here is the compatibility with Matlab/Simulink and the possibility to run Matlab/Simulink in external mode on the processing unit. Regarding these requirements, Arduino boards are a good choice. There are already a Matlab extension and Simulink extension for Arduino boards, which facilitate the implementation of Simulink programs on Arduino boards and the real-time analysis of data coming from Arduino in Simulink. There are many variants of Arduino boards. For our control system, Arduino Due were chosen (Figure 7.2). Arduino Due is the first Arduino board with an 32-bit ARM architecture. It has a clock speed

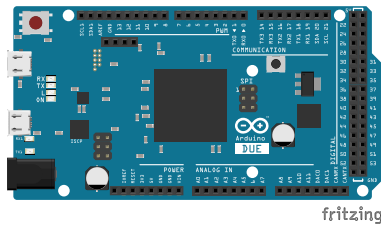


Figure 7.2: Arduino Due board

of $84MHz$, a flash memory of $512KB$ for storing code, and a SRAM of $96KB$. In comparison, the well known Arduino Uno board has a clock speed of $16MHz$, a flash memory of $32KB$ and a SRAM of $2KB$. The operating voltage of Arduino Due is $3.3V$. Arduino Due provides many interfaces. The most relevant for our application are the USB interface for the connection with the computer; the analog, digital, PWM, SPI, I2C interfaces for reading sensor data and sending commands to actuators.

Accelerometer

Accelerometers are used to sense the vibration on the platform, and also on the ground for ANC. As accelerometer, ADXL345 was chosen. It is a low power 3-axis accelerometer that can measure up to $\pm 16g$ and has an I2C as well as a SPI digital interface. Its maximum SPI clock speed is $5MHz$, and with I2C it can reach a

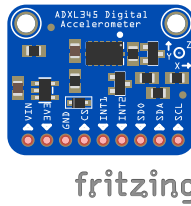


Figure 7.3: ADXL345 accelerometer

maximum clock speed of $400kHz$. It has an output data rate up to $3200Hz$. However, this is only possible with an SPI communication rate greater than $2MHz$. The maximum output data rate with I2C communication is $800Hz$ and can only be reached if the $400kHz$ I2C are used.

Voice-coil motor and motor driver

The voice-coil motor chosen for our application has already been described in Chapter 5. To interpret the actuation command given from Arduino, a motor driver is needed. The motor driver is generally connected to the control board, to the voltage source, and to the actuator. Its task is to convert the motor commands coming from control board into input voltage for the motor. MD10C was chosen as motor driver (Figure 7.4). MD10C is a bidirectional motor driver for brushed DC



Figure 7.4: Voice-coil motor and motor driver

motors. It is designed for high current applications (up to $13A$ continuous and $30A$ peak) and does not required a heat sink. It supports motor voltages between $5V$ and $25V$, and has a logic voltage level of $3.3V$ and $5V$. It can support PWM frequency up to $20kHz$ and both locked-antiphase and sign-magnitude PWM operations. For the test benches, the voltage for the motor drivers is supplied with a laboratory power supply unit.

Disturbance generation

To test the control algorithms, ground vibrations should be generated in the test benches. In order to build a noise profile similar to the real plant disturbances, a drone motor and a corresponding propeller are used to generate the vibrations. In



Figure 7.5: Excitation system components

the single suspension test bench, the drone motor is attached on the bottom of the actuator, so that the vibrations generated by the motor-propeller subsystem are directly transmitted to the actuator. To control the motor velocity, an Electronic Speed Controller (ESC) is used, which is controlled with an Arduino Nano and a potentiometer. The voltage is provided from a Lithium Polymer (LiPo) battery. These different components are shown in Figure 7.5.

7.2 SISO-system

Figure 7.6 shows the OTRCP system for the single suspension test bench. The disturbance generation system is an independent system as described above. The Arduino Due board used here as processing unit is connected to the computer via a USB cable. Then, it is connected to other components as shown in Figure 7.8. We have two accelerometers to measure the ground vibrations and the vibrations of the payload. The accelerometers communicate with the Arduino Due using SPI protocol. SPI communication has the advantage of being faster than I2C communication. Furthermore, the I2C communication with ADXL345 accelerometers has only two addresses. This means that more than two accelerometers cannot be connected to the Arduino board using the I2C communication protocol. The motor driver is control with two PWM signals. One signal modulates the amplitude of the motor voltage, and the other signal modulates the sign of the voltage.

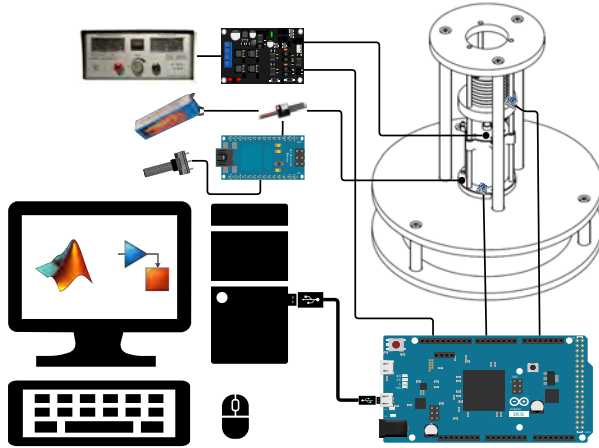


Figure 7.6: Test bench for single suspension - OTRCP system

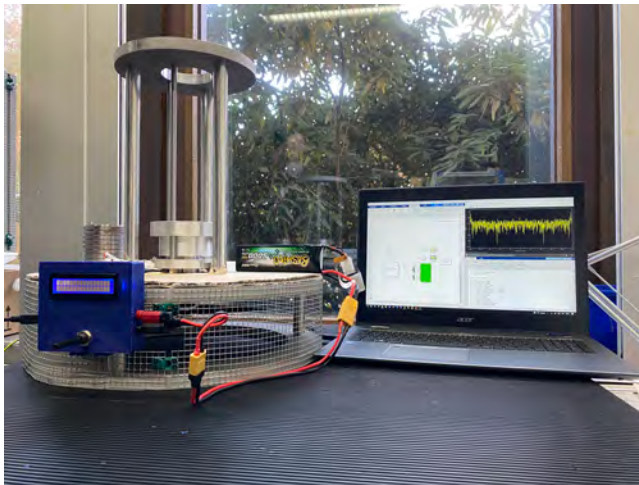


Figure 7.7: Test bench for single suspension - Physical system

Figure 7.7 shows the single suspension test bench in the laboratory.

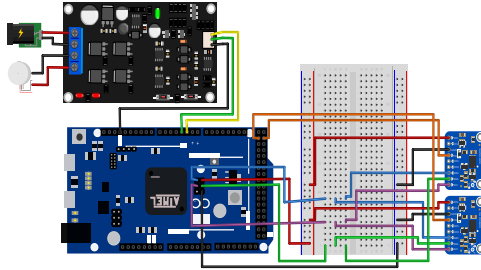


Figure 7.8: Sensor and actuator system for single suspension test bench

7.3 MIMO-system

In the isolation platform test bench, six accelerometers are used for three suspensions mounted. Each suspension has an accelerometer at the bottom and another on the top. The SPI communication is used. For each suspension, there is

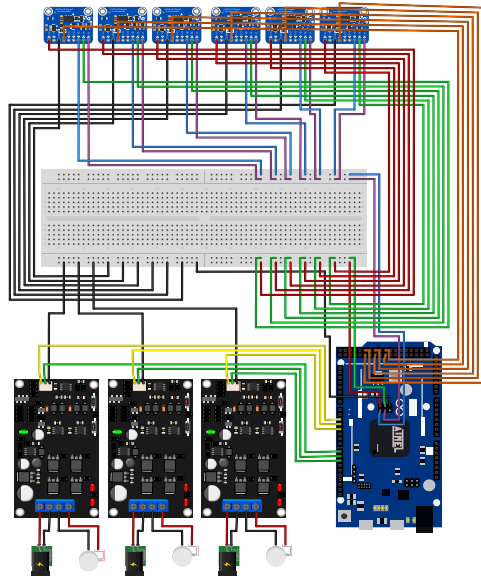


Figure 7.9: Sensor and actuator system for isolation platform test bench

one motor driver, which is connected to the Arduino board through PWM pins

(Figure 7.9). The excitation system is similar to the excitation system of the single suspension. But here, all four motors of the drone form an independent system which is controlled with an Arduino board. The vibrations generated by the motors are transmitted through the drone frame to the suspension systems. Figure 7.10

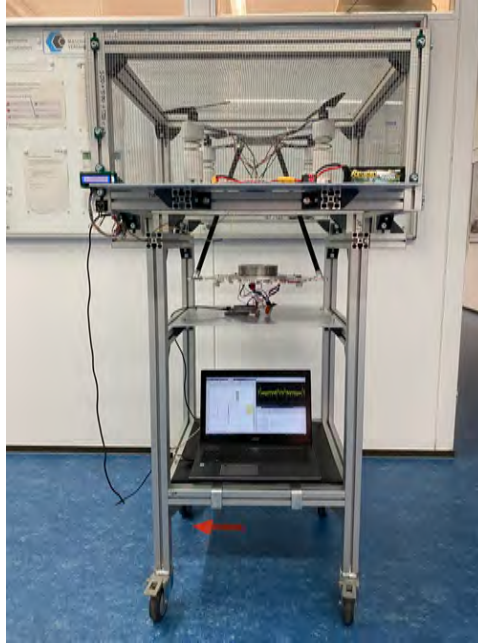


Figure 7.10: Test bench for isolation platform - Physical system

shows the isolation platform test bench in the laboratory.

7.4 Summary

The goal of this Chapter were the OTRCP. The different components of the OTRCP have been presented. The test bench for single suspension and the test bench for the whole platform in the laboratory were shown in Figures 7.7 and 7.10.

Due to delays in the production, actuator components could not be manufactured on time. Hence, it was not possible to go to the end of OTRCP.

8 Conclusion and outlook

Owing to their low cost and wide perspectives they offer, UAVs are becoming more and more popular for several applications. However, there are some limitations on their applications due to the vibrations occurring. There are several concepts developed to reduce the effect of vibrations on the payload. The concepts that can be classified as mechanical vibration damping have the capacity of damping a wide range of vibration frequency and amplitudes.

Within the scope of this work, a concept for the isolation of measurement equipments on an UAV should be developed. Therefore, the design of an isolation platform for measurement equipments for UAV applications was the main objective of this thesis. The platform should be usable for several applications.

Based on the application of the platform, different requirements were established. Then a concept were derived following the technical product design procedure according to VDI 2221. From this procedure, a mechanical model for the isolation platform and the actuator could be derived. Additionally, two test benches were developed for test purposes.

Also for test purposes, simulation environments were designed. Therefore, four structural configurations of the actuators have been considered. The dynamic of each configuration was derived and analyzed. The configuration 1 offered the best passive property while the configuration 3 offered the best active property. The two other configurations were somewhere between the both, but the configuration 4 showed a better compromise. The analysis was also done on the dynamic of the whole isolation platform, and it showed similar behavior as in single suspensions. But it also showed that the system dynamic is coupled. The simulation environment was designed using the Simscape Toolbox of Matlab/Simulink.

After the dynamic analysis, the next step of the development process of the control system was controller design and analysis. Since the platform should operate in a rough environment, adaptive control and robust control were chosen as control strategies. Both strategies are known to be capable of handling uncertain systems. As adaptive control strategy, ANC was chosen. An algorithm for IIR ANC was proposed. The aim of this algorithm is to decrease the probability of finding a local optimum or biased optimum while solving the adaptive IIR filtering problem. Propositions are also made for stability monitoring. The proposed algorithm and one of the propositions for stability monitoring were implemented in the simulation environment. There was also the problem of finding the best algorithm

parameters fitting the application. For that, a Matlab-App were develop with three versions depending on the problem to solve. The combination of the self-developed App with the proposed algorithm and stability monitoring gave satisfactory simulation results for single suspensions. The best isolation were achieved by the configuration 3. The configuration 2 also achieved a good isolation as all other configurations, but it required to high control input.

After using ANC on single suspensions, it was also used on the isolation platform in form of decentral and central control. For both strategies, the best isolation was achieved by the configuration 3, and the configuration 2 could not achieve satisfactory attenuation in the system. In general, decentral control provided better results than central control. The best isolation of the platform was achieved by the configuration 3 in the decentral control strategy.

The second control strategy considered was the robust control. This strategy could only be used for two configurations. For controller design, a payload uncertainty of $\pm 20\%$, and a motor damping uncertainty of $\pm 50\%$ were considered. The \mathcal{H}_∞ -based controllers were design: \mathcal{H}_∞ -synthesis, μ -synthesis with μ - K iteration, and μ -synthesis with D - K iteration. For single suspensions, the best nominal damping was achieved by the \mathcal{H}_∞ -synthesis controllers, and the best performance was achieved by the controllers designed with the μ - K iteration. The controllers designed with D - K iteration are somewhere between both other methods in term of attenuation and performance. The design of controller for the whole platform showed similar results for both central and decentral control, whereby the decentral control achieved a better result.

After controller design, the first steps of the implementation in terms of On-Target Rapid Control Prototyping (OTRCP) of the isolation platform were done. The OTRCP is one of the first steps to the implementation of the designed controllers in the physical system. The different components used for OTRCP were presented for both test benches. Due to delays in the production, actuator components could not be manufactured on time, and a further analysis using OTRCP data was not possible.

In summary, during this thesis, we achieved to design a mechanical model for an isolation platform for several UAV applications. Different actuator designs were also proposed for the active isolation of the platform. A simulation environment were designed in Matlab/Simulink using Simscape Toolbox. To control the simulation, two control strategies were proposed. The first proposition was ANC. A new algorithm for ANC with propositions for stability monitoring for IIR filters was developed. Combined with a self-developed Optimization Matlab-App, the new algorithm achieved to attenuate vibrations in the system. Robust control strategy

was the second proposition. Attenuation could also be achieved with this strategy, and the closed-loop behavior were represented in Bode-diagrams. After controller design, the first steps in the implementation of the controller were made in term of OTRCP.

To find the parameters for ANC using the self-developed Optimization App, only the global pattern search algorithm was used. Other algorithms can be tried out for optimizing the ANC parameters. In the Optimization App, there are already two other algorithms that can be tried: genetic algorithm, and particle swarm optimization. There are also other biologically-inspired algorithms and heuristic algorithms that may generates better results, especially for MIMO-systems. After finding better optimization algorithms, the proposed algorithm for IIR adaptive filtering problem can be compared to the already existent bias-free algorithms. The effect of the different stability monitoring propositions on ANC with adaptive IIR filter can also be examined.

Regarding the robust control design, there was a problem of finding optimal weighting functions. They usually depend on the application, but there are no formal method how to find them. In the medium term, a task may be to examine and to characterize the system in order to derived suitable weighting functions corresponding to the problem. Another task in the medium term is to continue the OTRCP and the commissioning of the test benches.

A task in the long term could be the integration of the isolation platform on a real drone with measurement equipments, the analysis of its behavior, and the comparison with existing isolation systems. Furthermore, the effect of the stability monitoring algorithm on ANC could be analyzed and an extension of the proposed ANC algorithm to MIMO systems can be derived.

List of Figures

1.1	UAV for civil applications [23]	1
1.2	Measurement platform for 3D reconstruction attached to an UAV [68]	3
2.1	Source of vibrations in aerial robotic [79]	7
2.2	Stabilization method [79]	9
2.3	Optical image stabilization compensation [43]	9
2.4	Digital image stabilization compensation [47]	10
2.5	Gimbal system architecture [39]	11
2.6	Wire-rope isolator [51]	13
2.7	Configuration of HILS system for vibration damping in UAV [59]	14
3.1	Lifecycle of technical product [36]	17
3.2	Product Development Process (PDP)	18
3.3	Example of passive vibration control configurations [37]	22
3.4	Example of semi-active vibration control configurations	23
3.5	Feedback and feedforward control for active vibration control [11]	23
3.6	Pneumatic actuator [17]	24
3.7	Electrodynamical actuator [77]	24
3.8	Piezo actuator [29]	25
3.9	Classical feedback control system	26
3.10	Requirements on $T(s)$ and $S(s)$	27
3.11	Bode plot of the inverse performance weight	28
3.12	Shape of L based on the requirements on S and T [33]	30
3.13	Control structure for mixed-sensitivity \mathcal{H}_∞ [69]	31
3.14	Robust stability (left) and robust performance (right) [50]	32
3.15	M/Δ -structure [69]	32
3.16	M/Δ -structure with D [69]	33
3.17	N/Δ -structure for robust performance analysis [69]	33
3.18	Destructive interference [2]	36
3.19	ANC feedforward (above) and feedback (below) [14]	37
3.20	ANC with adaptive filtering [34]	38
3.21	Tapped-delay structure of FIR filter (above) and adaptive FIR-filter (below) [34]	39
3.22	Cost function [34]	39
3.23	Adaptive IIR-filter	41
3.24	Multimodal function	42
3.25	Central control for Gimbal	44
3.26	Decentral control for Gimbal	44
3.27	Principle of evolutionary algorithms	47
3.28	Particle swarm optimization: position update	50
3.29	Global pattern search optimization	51
4.1	Data acquisition platform on DJI Matrice 600 Pro [68]	53
4.2	Image of data acquisition [68]	54

4.3	Main vibration sources in UAV	54
4.4	Black box model of the isolation platform	57
4.5	Functional structure of isolation platform	58
4.6	Mechanical designs	61
4.7	Actuator design	66
4.8	Free-body diagram of the actuator	67
4.9	Free-body diagram of the isolation platform	68
4.10	CAD model of isolation platform on quadcopter	69
4.11	Test bench for single suspension	69
4.12	Test bench for isolation platform	70
5.1	Voice-coil motor	71
5.2	electical circuit of voice-coil motor	71
5.3	Car suspension (left) and single mass damper (right)	72
5.4	Different suspension configurations	73
5.5	Free-body diagram of suspension configuration 1	74
5.6	Bode Plot of transfer functions - Configuration 1	77
5.7	Free-body diagram of suspension configuration 2	78
5.8	Bode Plot of transfer functions - Configuration 2	79
5.9	Free-body diagram of suspension configuration 3	79
5.10	Bode Plot of transfer functions - Configuration 3	81
5.11	Free-body diagram of suspension configuration 4	82
5.12	Bode Plot of transfer functions - Configuration 4	83
5.13	Free-body diagram of the isolation platform	85
5.14	Bode diagram of isolation platform - configuration 1	85
5.15	Bode diagram of isolation platform - configuration 2	85
5.16	Bode diagram of isolation platform - configuration 3	86
5.17	Bode diagram of isolation platform - configuration 4	86
5.18	Simscape model of single suspension Configuration 1	87
5.19	Simscape model of isolation platform	89
6.1	Problem of system identification	92
6.2	Output-error formulation	93
6.3	Equation-error formulation	94
6.4	Bias-free IIR adaptive filter structure proposed by Narasimhan and Veena [57]	96
6.5	Combination of output-error and equation-error	98
6.6	Adaptive IIR filter with OE and EQE fusion block	98
6.7	Possible optima for different formulations	99
6.8	ANC block diagram with adaptive FIR filter [34]	103
6.9	ANC block diagram	103
6.10	ANC with random noise injection for secondary path modeling [34]	104
6.11	Structure used for extended least-square method [34]	105
6.12	ANC block diagram with secondary path estimation	106
6.13	Simulation optimization app - SISO Weighted version	110
6.14	Simulink block diagram of ANC simulation	112
6.15	Result of the Simulation Optimization App - Configuration 1	113

6.16	Result of the Simulation Optimization App - Configuration 2	114
6.17	Result of the Simulation Optimization App - Configuration 3	115
6.18	Result of the Simulation Optimization App - Configuration 4	116
6.19	Result of Simulation Optimization App - Decentral - Configuration 1	118
6.20	Result of Simulation Optimization App - Decentral - Configuration 2	118
6.21	Result of Simulation Optimization App - Decentral - Configuration 3	119
6.22	Result of Simulation Optimization App - Decentral - Configuration 4	119
6.23	Result of Simulation Optimization App - Central - Configuration 1	123
6.24	Result of Simulation Optimization App - Central - Configuration 2	123
6.25	Result of Simulation Optimization App - Central - Configuration 3	124
6.26	Result of Simulation Optimization App - Central - Configuration 4	124
6.27	Generalized plant for single suspension	125
6.28	Bode diagram of weight functions	127
6.29	Singular value for robust stability (left) and performance (right) - Configuration 1	128
6.30	Controller order reduction - Configuration 1	128
6.31	Bode diagram of designed controllers - Configuration 1	129
6.32	Bode diagram of closed-loop - Configuration 1	129
6.33	Singular value for robust stability (left) and performance (right) - Configuration 2	130
6.34	Controller order reduction - Configuration 2	131
6.35	Bode diagram of designed controllers - Configuration 2	131
6.36	Bode diagram of closed-loop - Configuration 2	132
6.37	Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Decentral control - Configuration 1	133
6.38	Bode diagram of closed-loop of MIMO system - μ - K iteration - Decentral control - Configuration 1	133
6.39	Bode diagram of closed-loop of MIMO system - D - K iteration - Decentral control - Configuration 1	133
6.40	Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Decentral control - Configuration 2	134
6.41	Bode diagram of closed-loop of MIMO system - μ - K iteration - Decentral control - Configuration 2	134
6.42	Bode diagram of closed-loop of MIMO system - D - K iteration - Decentral control - Configuration 2	134
6.43	Controller order reduction - Central control - Configuration 1	135
6.44	Singular value for robust stability (left) and performance (right) - Central control - Configuration 1	136
6.45	Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Central control - Configuration 1	136
6.46	Bode diagram of closed-loop of MIMO system - μ - K iteration - Central control - Configuration 1	137
6.47	Bode diagram of closed-loop of MIMO system - D - K iteration - Central control - Configuration 1	137
6.48	Singular value for robust stability (left) and performance (right) - Central control - Configuration 2	137

6.49	Controller order reduction - Central control - Configuration 2	138
6.50	Bode diagram of closed-loop of MIMO system - \mathcal{H}_∞ - Central control - Configuration 2	138
6.51	Bode diagram of closed-loop of MIMO system - μ - K iteration - Central control - Configuration 2	138
6.52	Bode diagram of closed-loop of MIMO system - D - K iteration - Central control - Configuration 2	139
7.1	On-Target Rapid Control Prototyping	141
7.2	Arduino Due board	142
7.3	ADXL345 accelerometer	143
7.4	Voice-coil motor and motor driver	143
7.5	Excitation system components	144
7.6	Test bench for single suspension - OTRCP system	145
7.7	Test bench for single suspension - Physical system	145
7.8	Sensor and actuator system for single suspension test bench	146
7.9	Sensor and actuator system for isolation platform test bench	146
7.10	Test bench for isolation platform - Physical system	147
A.1	Simulation optimization app - SISO version	169
A.2	Simulation optimization app - SISO Weighted version	170
A.3	Simulation optimization app - MIMO Weighted version	171

List of Tables

4.1	Compact requirements	56
4.2	Morphological box of isolation platform	60
4.3	Ranking procedure for mechanical construction	62
4.4	Guideline VDI 2225 - Rating System [13]	63
4.5	Ranking procedure for actuator	63
4.6	Evaluation of mechanical design	64
4.7	Evaluation of type of actuator	65
5.1	Extract from the voice-coil motor data sheet	76
6.1	Simulation results of ANC on single suspensions	116
6.2	Simulation results of decentral ANC for MIMO-system	120
6.3	Simulation results of central ANC for MIMO-system	122

Bibliography

- [1] Mohammad Abdulrahman Al-Mashhadani. "Random vibrations in unmanned aerial vehicles, mathematical analysis and control methodology based on expectation and probability". In: *Journal of Low Frequency Noise, Vibration and Active Control* 38.1 (2019), pp. 143–153.
- [2] *Active noise control*. 2021. URL: https://en.wikipedia.org/wiki/Active_noise_control#/media/File:Active_Noise_Reduction.svg.
- [3] Shakeel Ahmed, Muhammad Tahir Akhtar, and Muhammad Tufail. "A variable step size based method for Online secondary path modeling in active noise control System". In: *Proc. of the Second APSIPA Annual Summit and Conference*. 2010, pp. 549–554.
- [4] M Tahir Akhtar, Masahide Abe, and Masayuki Kawamata. "Modified-filtered-x LMS algorithm based active noise control systems with improved online secondary-path modeling". In: *The 2004 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS'04*. Vol. 1. IEEE. 2004, pp. 1–13.
- [5] Muhammad Tahir Akhtar, Masahide Abe, and Masayuki Kawamata. "A method for online secondary path modeling in active noise control systems". In: *2005 IEEE International Symposium on Circuits and Systems*. IEEE. 2005, pp. 264–267.
- [6] Francisco J Aragón et al. *Nonlinear optimization*. Springer, 2019.
- [7] Charles Audet. "Convergence results for generalized pattern search algorithms are tight". In: *Optimization and Engineering* 5.2 (2004), pp. 101–122.
- [8] Jerry Banks. *Discrete event system simulation*. Pearson Education India, 2005.
- [9] Mehmet Baskin and MEHMET KEMAL LEBLEBİCİOĞLU. "Robust control for line-of-sight stabilization of a two-axis gimbal system". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 25.5 (2017), pp. 3839–3853.
- [10] Andrei Battistel and Tiago Roux Oliveira. "Unit vector control of an unbalanced three-axis gimbal for application to inertially stabilized platforms". In: *Asian Journal of Control* (2021).
- [11] Francisco Beltran-Carbajal. *Vibration Analysis and Control in Mechanical Structures and Wind Energy Conversion Systems*. BoD—Books on Demand, 2018.
- [12] Beate Bender and Kilian Gericke. *Pahl/Beitz Konstruktionslehre*. Springer, Berlin, 2020.
- [13] Bernd Bender et al. *Erfolgreiche individuelle Vorgehensstrategien in frühen Phasen der Produktentwicklung*. VDI-Verlag Berlin, 2004.
- [14] Sten Böhme. "System zur aktiven Reduktion der Schallabstrahlung von Lärmquellen". In: (2010).

- [15] Daniel Bratton and James Kennedy. "Defining a standard for particle swarm optimization". In: *2007 IEEE swarm intelligence symposium*. IEEE. 2007, pp. 120–127.
- [16] Guowei Cai et al. "Systematic design methodology and construction of UAV helicopters". In: *Elsevier Mechatronics* 18.4 (2008), pp. 554–558.
- [17] Ping-Chang Chen and Ming-Chang Shih. "Modeling and robust active control of a pneumatic vibration isolator". In: *Journal of Vibration and Control* 13.11 (2007), pp. 1553–1571.
- [18] S-J Chen and James Steven Gibson. "Feedforward adaptive noise control with multivariable gradient lattice filters". In: *IEEE Transactions on Signal Processing* 49.3 (2001), pp. 511–520.
- [19] Chi-Wei Chiu et al. "Optimal design and experimental verification of a magnetically actuated optical image stabilization system for cameras in mobile phones". In: *Journal of Applied Physics* 103.7 (2008), 07F136.
- [20] Nguyen Cong Danh. "The Stability of a Two-Axis Gimbal System for the Camera". In: *The Scientific World Journal* 2021 (2021).
- [21] Reiner Creutzburg. "European activities in civil applications of drones: an overview of remotely piloted aircraft systems (RPAS)". In: *Mobile Multimedia/Image Processing, Security, and Applications 2015*. Vol. 9497. International Society for Optics and Photonics. 2015, p. 949707.
- [22] Christopher Dahlin Rodin et al. "A survey of practical design considerations of optical imaging stabilization systems for small unmanned aerial systems". In: *Sensors* 19.21 (2019), p. 4800.
- [23] Pasquale Daponte et al. "Metrology for drone and drone for metrology: Measurement systems on small civilian drones". In: *2015 IEEE Metrology for Aerospace (MetroAeroSpace)*. IEEE. 2015, pp. 306–311.
- [24] James C DeBruin et al. "Feedforward stabilization test bed". In: *Acquisition, Tracking, and Pointing X*. Vol. 2739. International Society for Optics and Photonics. 1996, pp. 204–214.
- [25] Pakize Erdogmus. "Introductory Chapter: Swarm Intelligence and Particle Swarm Optimization". In: *Particle Swarm Optimization with Applications* (2018), p. 1.
- [26] SARP Erturk. "Digital image stabilization with sub-image phase correlation based global motion estimation". In: *IEEE transactions on consumer electronics* 49.4 (2003), pp. 1320–1325.
- [27] *Forschungsprojekte am Institut für mess- und regelungstechnik*. URL: <https://www.imr.uni-hannover.de/de/forschung/forschungsprojekte/>.
- [28] K Gjika and R Dufour. "Rigid body and nonlinear mount identification: application to onboard equipment with hysteretic suspension". In: *Journal of Vibration and Control* 5.1 (1999), pp. 75–94.

- [29] Physik Instrumente (PI) GmbH. *P-882 – P-888 PICMA® Stack Multilayer Piezo Actuators*. URL: <https://www.physikinstrumente.com/en/products/piezoelectric-transducers-actuators/p-882-p-888-picma-stack-multilayer-piezo-actuators-100810/>.
- [30] Pedro Gomes, Pedro Santana, and José Barata. "A vision-based approach to fire detection". In: *International Journal of Advanced Robotic Systems* 11.9 (2014), p. 149.
- [31] GJ Grenzdörffer, A Engel, and B Teichert. "The photogrammetric potential of low-cost UAVs in forestry and agriculture". In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 31.B3 (2008), pp. 1207–1214.
- [32] Robert Grepl. "Real-Time Control Prototyping in MATLAB/Simulink: Review of tools for research and education in mechatronics". In: *2011 IEEE International Conference on Mechatronics*. IEEE. 2011, pp. 881–886.
- [33] Da-Wei Gu, Petko Petkov, and Mihail M Konstantinov. *Robust control design with MATLAB®*. Springer Science & Business Media, 2005.
- [34] Colin Hansen et al. *Active control of noise and vibration*. CRC press, 2012.
- [35] Jie-Liang Huang and Wen-Yu Cai. "UAV low altitude marine monitoring system". In: *2014 International Conference on Wireless Communication and Sensor Network*. IEEE. 2014, pp. 61–64.
- [36] VEREIN DEUTSCHER INGENIEURE. *VDI 2221 - Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte*. 1993.
- [37] Roman Karsten. "Integrierte dielektrische Elastomerwandler für aktive Schwingungsdämpfung". PhD thesis. Technische Universität Darmstadt, 2016.
- [38] Jaka Katrasnik, Franjo Pernus, and Bostjan Likar. "A survey of mobile robots for distribution power line inspection". In: *IEEE Transactions on power delivery* 25.1 (2009), pp. 485–493.
- [39] Min Kim, Gi-Sig Byun, and Gwan-Hyung Kim. "Gimbal System Control for Drone for 3D Image". In: *Journal of the Korea Institute of Information and Communication Engineering* 20.11 (2016), pp. 2107–2112.
- [40] Tobias K Kohoutek and Henri Eisenbeiss. "Processing of UAV based range imaging data to generate detailed elevation models of complex natural structures". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39.B1 (2012), pp. 405–410.
- [41] Michael Kontitsis, Kimon P Valavanis, and Nikos Tsourveloudis. "A UAV vision system for airborne surveillance". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*. Vol. 1. IEEE. 2004, pp. 77–83.
- [42] Olivier Küng et al. *The accuracy of automatic photogrammetric techniques on ultra-light UAV imagery*. Tech. rep. 2011.

- [43] Fabrizio La Rosa et al. "Optical Image Stabilization (OIS)". In: *STMicroelectronics*. Available online: http://www.st.com/resource/en/white_paper/ois_white_paper.pdf (accessed on 12 October 2017) (2015).
- [44] Ying-Chih Lai and Shau-Shiun Jan. "Attitude estimation based on fusion of gyroscopes and single antenna GPS for small UAVs under the influence of vibration". In: *GPS solutions* 15.1 (2011), pp. 67–77.
- [45] Robert M Lewis and Virginia Torczon. *Rank Ordering and Positive Bases in Pattern Search Algorithms*. Tech. rep. INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE and ENGINEERING HAMPTON VA, 1996.
- [46] Zhenming Li et al. "Development and Design Methodology of an Anti-Vibration System on Micro-UAVs". In: *International Micro Air Vehicle Conference and Flight Competition (IMAV 2017)*. IEEE. 2017, pp. 223–228.
- [47] Chia-Kai Liang et al. "The effect of digital image stabilization on coding performance [video coding]". In: *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004*. IEEE. 2004, pp. 402–405.
- [48] Albert Yu-Min Lin et al. "Combining GeoEye-1 satellite remote sensing, UAV aerial imaging, and geophysical surveys in anomaly detection applied to archaeology". In: *IEEE Journal of selected topics in applied earth observations and remote sensing* 4.4 (2011), pp. 870–876.
- [49] Jing Lu et al. "Lattice form adaptive infinite impulse response filtering algorithm for active noise control". In: *The Journal of the Acoustical Society of America* 113.1 (2003), pp. 327–335.
- [50] Uwe Mackenroth. *Robust control systems: theory and case studies*. Springer Science & Business Media, 2013.
- [51] Jean-Luc Maës, Stéphane Binczak, and Vincent Lhenry. "A passive stabilization solution for camera embedded onboard small planes". In: *2014 Integrated Communications, Navigation and Surveillance Conference (ICNS) Conference Proceedings*. IEEE. 2014, U3–1.
- [52] Chris Reitsma Maj. "A novel approach to vibration isolation in small, unmanned aerial vehicles". In: *2009 IEEE International Conference on Technologies for Practical Robot Applications*. IEEE. 2009, pp. 84–87.
- [53] G Nicolás Marichal et al. "Vibration reduction for vision systems on board unmanned aerial vehicles using a neuro-fuzzy controller". In: *Journal of Vibration and Control* 20.15 (2014), pp. 2243–2253.
- [54] IZ Mat Darus and MO Tokhi. "Genetic algorithms based adaptive active vibration control of a flexible plate structure". In: *Systems Science* 29.3 (2003), pp. 65–79.
- [55] Thomas J Mueller and James D DeLaurier. "Aerodynamics of small vehicles". In: *Annual review of fluid mechanics* 35.1 (2003), pp. 89–111.
- [56] Paul Naefe and Michael Kott. *Konstruktionslehre Für Einsteiger*. Springer, 2018.

- [57] SV Narasimhan and S Veena. "New unbiased adaptive IIR filter: it's robust and variable step-size versions and application to active noise control". In: *Signal, Image and Video Processing* 7.1 (2013), pp. 197–207.
- [58] SV Narasimhan, S Veena, and H Lokesha. "Variable step-size Griffiths' algorithm for improved performance of feedforward/feedback active noise control". In: *Signal, image and video processing* 4.3 (2010), pp. 309–317.
- [59] Jong-Seok Oh, Young-Min Han, and Seung-Bok Choi. "Vibration control of a camera mount system for an unmanned aerial vehicle using piezostack actuators". In: *Smart materials and structures* 20.8 (2011), p. 085020.
- [60] Yu-Chun Peng et al. "Digital image stabilization and its integration with video encoder". In: *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005*. IEEE. 2005, pp. 544–549.
- [61] Ashley E Pete et al. "The role of passive avian head stabilization in flapping flight". In: *Journal of The Royal Society Interface* 12.110 (2015), p. 20150508.
- [62] G. Nicolás Plasencia and M. Tomas-Rodriguez. "Modelling and Analysis of Vibrations in a UAV Helicopter with a Vision System". In: *International Journal of Advance Robotic Systems* 9.5 (2012), p. 220.
- [63] Paolo Bellezza Quater et al. "Light Unmanned Aerial Vehicles (UAVs) for cooperative inspection of PV plants". In: *IEEE Journal of Photovoltaics* 4.4 (2014), pp. 1107–1113.
- [64] Paresh Rawat and Jyoti Singhai. "Review of motion estimation and video stabilization techniques for hand held mobile video". In: *Signal & Image Processing: An International Journal (SIPIJ) Vol 2* (2011).
- [65] Maurizio Rossi et al. "Gas-drone: Portable gas sensing system on UAVs for gas leakage localization". In: *SENSORS, 2014 IEEE*. IEEE. 2014, pp. 1431–1434.
- [66] David Sachs, Steven Nasiri, and Daniel Goehl. "Image stabilization technology overview". In: *InvenSense Whitepaper* (2006).
- [67] *Simscape*. URL: <https://de.mathworks.com/products/simscape.html>.
- [68] *Simultan-Lokalisation und kartierung system unter Verwendung von Luftkamera*. URL: <https://www.imr.uni-hannover.de/de/forschung/forschungsprojekte/forschungsprojekte-detailansicht/projects/simultan-lokalisierung-und-kartierung-system-unter-verwendung-von-luftkamera/>.
- [69] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: analysis and design*. Vol. 2. Citeseer, 2007.
- [70] Myeong-Gyu Song et al. "Design of a voice-coil actuator for optical image stabilization based on genetic algorithm". In: *IEEE Transactions on Magnetics* 45.10 (2009), pp. 4558–4561.
- [71] Bob Stewart. "Adaptive IIR filtering". In: *5th IEE Adaptive Signal Processing Development Course* (2006).

- [72] Katie J Stuckel and William H Semke. "A piezoelectric actuated stabilization mount for payloads onboard small UAS". In: *Advanced Aerospace Applications, Volume 1*. Springer, 2011, pp. 295–305.
- [73] Katie J Stuckel et al. "A high frequency stabilization system for UAS imaging payloads". In: *Structural Dynamics, Volume 3*. Springer, 2011, pp. 1411–1419.
- [74] João Marcelo Teixeira et al. "Teleoperation using google glass and ar, drone for structural inspection". In: *2014 XVI Symposium on Virtual and Augmented Reality*. IEEE, 2014, pp. 28–36.
- [75] Mohit Verma et al. "Active stabilization of unmanned aerial vehicle imaging platform". In: *Journal of Vibration and Control* 26.19-20 (2020), pp. 1791–1803.
- [76] Adam L Webster and William H Semke. "Broad-band viscoelastic rotational vibration control for remote sensing applications". In: *Journal of Vibration and Control* 11.11 (2005), pp. 1339–1356.
- [77] *What is a Voice Coil Actuator?* URL: <https://www.machinedesign.com/mechanical-motion-systems/article/21836669/what-is-a-voice-coil-actuator>.
- [78] Darrell Whitley. "An overview of evolutionary algorithms: practical issues and common pitfalls". In: *Information and software technology* 43.14 (2001), pp. 817–831.
- [79] Jens Windau and Laurent Itti. "Multilayer real-time video image stabilization". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 2397–2402.
- [80] Zhangfan Xu et al. "A double gimbal optical image stabilizer driven by piezoelectric stacks". In: *Review of Scientific Instruments* 92.4 (2021), p. 045008.
- [81] Hsing-Cheng Yu and TS Liu. "Adaptive model-following control for slim voice coil motor type optical image stabilization actuator". In: *Journal of Applied Physics* 103.7 (2008), 07F114.
- [82] Jing Yuan. "Adaptive Laguerre filters for active noise control". In: *Applied Acoustics* 68.1 (2007), pp. 86–96.
- [83] Oliver Zirn and Sascha Weikert. *Modellbildung und Simulation hochdynamischer Fertigungssysteme: eine praxisnahe Einführung*. Springer Science & Business Media, 2005.

A Appendix

A.1 Matlab application designed with AppDesigner

To solve parameter optimization problems, a Matlab App has been designed during this thesis. The main objective of the App is to solve simulation optimization problems, where the task is to find the optimal parameters optimizing the power of specific signals.

There are three versions of this App. The first version whose interface is shown on Figure A.1 is developed to optimize the power of a one-dimensional signal in a simulation. The cost function formulated in this version is

$$J = \left(\sum_{k=1}^K e_k^2 \right)^\alpha,$$

where e denotes the signal whose power is to optimize, and e_k the discrete values of e . If the problem is a power minimization problem, then $\alpha = 1$, else $\alpha = -1$.

The second version whose interface is shown in Figure A.2 is an update of the first version with an expansion of the optimization to two one-dimensional signals. The cost function is in this case defined as

$$J = \left(\sum_{k=1}^K Q_e \cdot e_k^2 \right)^{\alpha_e} + \left(\sum_{l=1}^L Q_u \cdot u_l^2 \right)^{\alpha_u},$$

where e and u denote the signals whose powers are to optimize, and e_k and u_l the discrete values of e and u . α_e respectively α_u are defined as 1 if the power of e respectively u are to minimize and as -1 if they are to maximize.

The third version whose interface is shown on Figure A.3 is designed for two multidimensional signals. The cost function used is

$$J = \left(\sum_{k=1}^K e_k^T Q_e e_k \right)^{\alpha_e} + \left(\sum_{l=1}^L u_l^T Q_u u_l \right)^{\alpha_u}.$$

e and u are the multidimensional signals whose powers should be optimized, and α_e and α_u are defined as in the second version.

This formulation of the cost function may be problematic for maximization tasks if the case where

$$\sum_{k=1}^K e_k^T Q_e e_k = 0 \quad \text{or} \quad \sum_{l=1}^L u_l^T Q_u u_l = 0$$

occurs. This leads to an infinite cost function, independent from the power of the other signal. To avoid this problem, the cost function is reformulated as

$$J = \gamma^{1-\alpha_e} \cdot \left(1 + \sum_{k=1}^K e_k^T Q_e e_k \right)^{\alpha_e} + \gamma^{1-\alpha_u} \cdot \left(1 + \sum_{l=1}^L u_l^T Q_u u_l \right)^{\alpha_u},$$

where γ is a positive parameter to be define, especially for maximization tasks. This formulation leads to following formulations for the cost function:

$$\begin{aligned} \min e^2 \wedge \min u^2 &\Rightarrow J = 2 + \sum_{k=1}^K e_k^T Q_e e_k + \sum_{l=1}^L u_l^T Q_u u_l, \\ \max e^2 \wedge \min u^2 &\Rightarrow J = 1 + \gamma^2 \cdot \left(1 + \sum_{k=1}^K e_k^T Q_e e_k \right)^{-1} + \sum_{l=1}^L u_l^T Q_u u_l, \\ \min e^2 \wedge \max u^2 &\Rightarrow J = 1 + \sum_{k=1}^K e_k^T Q_e e_k + \gamma \cdot \left(1 + \sum_{l=1}^L u_l^T Q_u u_l \right)^{-1}, \\ \max e^2 \wedge \max u^2 &\Rightarrow J = \gamma^2 \cdot \left(\sum_{k=1}^K e_k^T Q_e e_k \right)^{-1} + \gamma^2 \cdot \left(\sum_{l=1}^L u_l^T Q_u u_l \right)^{-1}. \end{aligned}$$

These formulations are normally defined for two multidimensional signals. But it can also be used for one-dimensional signals. This will cover the functionality of the second version of the App. It can also be used for one one-dimensional or one multidimensional signal. In this case, the App considers the first signal also as second signal, but a zeros-matrix is defined as weight for the second signal. The new cost function is

$$J = \gamma^{1-\alpha_e} \cdot \left(1 + \sum_{k=1}^K e_k^T Q_e e_k \right)^{\alpha_e}.$$

So, the third version of the App can also cover the functionality of the first version. Therefore, only the third version will be used to solve simulation optimization problems. For a problem with more than two signals to optimize, the different signals can be packed in a vector an their weights in a matrix in form of

$$e = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_R \end{bmatrix}, \quad Q_e = \begin{bmatrix} Q_{s1} & 0 & \cdots & 0 \\ 0 & Q_{s2} & & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & Q_{sR} \end{bmatrix}.$$

The interface of the App is divided in three main sections. The first section titled "**Simulation Parameters**" is for the definition of the problem. It contains following components:

- **continue optimization switch**: with this switch, the user can tell the App if he

want to define a new optimization problem or continuous an already started optimization problem.

- **load .slx or .mdl file for new optimization:** with this button, the user can choose the Simulink file which will be used for the simulation. In the simulation, the signals to optimize should each go to "To Workspace" blocks. All "To Workspace" blocks should have the save format "Timeseries" and the same sample time.
- **load .mat file for simulation:** by clicking this button, the user can load a mat-file, in which all required variables for the simulation are saved.
- **load .mat file to continue optimization:** this button allows the user to load a previously from App saved mat-file of an optimization problem in order to continue the optimization. This button is only active if the continue optimization switch is on "Yes".
- **load .xlsx file, save .xlsx file and clear table:** these buttons allow the user to respectively load the parameters to optimized and their minimum values and maximum values as xlsx-file in the App, to save the defined parameters as xlsx-file, and to clear the table. The table can also be edited manually in the App with the "add parameter" and "delete parameter" subsections.
- **Text box** below "delete parameter" subsection: this text box outputs error messages if an error occurs while defining the simulation parameters.
- **error name, Qe, ctrl input name and Qu:** they are for the definition of the signals to optimize and the corresponding weights.
- **start variable:** to define a start value for the optimization problem. This is optional.
- **nondeterministic switch:** with this switch, the user can define if the problem is deterministic or not. In case of nondeterministic simulation, the "multiple simulations switch" is activated. If the "multiple simulations switch" is switched to "Yes", the user can choose how many simulations should be done for each solution candidate. In case of nondeterministic cost function, the optimization problem is formulated as

$$\min_{\mathbf{x}} J(\mathbf{x}, \varepsilon), \quad \mathbf{x} \in (\underline{\mathbf{x}}, \bar{\mathbf{x}}),$$

where ε is a random parameter. The cost function used for a nondeterministic optimization problem with Q simulation runs for each solution candidate is

$$J(\mathbf{x}, \varepsilon) = \max\{J_1(\mathbf{x}, \varepsilon_1), \dots, J_q(\mathbf{x}, \varepsilon_q), \dots, J_Q(\mathbf{x}, \varepsilon_Q)\}.$$

The idea behind this formulation is the optimization of the worst case respectively the best worst case behavior.

- **parallel tool switch:** when it is switched to "Yes", the App will use the parallel tool of Matlab for the simulation. It activates a slider to choose the number of workers that the parallel tool should use.

The second section of the interface is titled "**Algorithm Parameters**". Here, the optimization algorithm can be chosen and its parameters can be set. So far, three algorithms have been implemented in the App. They are Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Global Pattern Search Optimization (GPSO). The difference to the classical algorithms is the parameter "percentages of new candidates". With this parameter, the user can set how many solution candidates of each generation will be randomly chosen in the solution space. This helps the chosen algorithm to explore the whole solution space when the algorithm dynamic leads to a local minimum. There are two switch buttons to choose if the power of the signals defined in "Simulation Parameters" should be minimized or maximized. "**save each ...steps**" defines after how many iterations App data will be saved. The App generates a mat-file containing the solution and some App data. The generated mat-files contains three variables of type structure. The for the user most important structure variable is named "Solution" and it contains two fields: "FoundOptimum" and "FoundOptimumEachStep". The two other structure variables are "AlgoParStruct" and "SimParStruct". They are important for the App to continue an already started optimization problem. In fact, if the user want to continue an optimization problem, he should load the generated mat-file by clicking on the "**load .mat to continue optimization**" and choosing the file.

The third section of the interface is the "**Algorithm Results**". This section contains figures in which the signals and the progression of the cost function will be plotted. There is also a gauge to visualize the evolution of the iterations in percentage, and a status bar to show information about iterations.

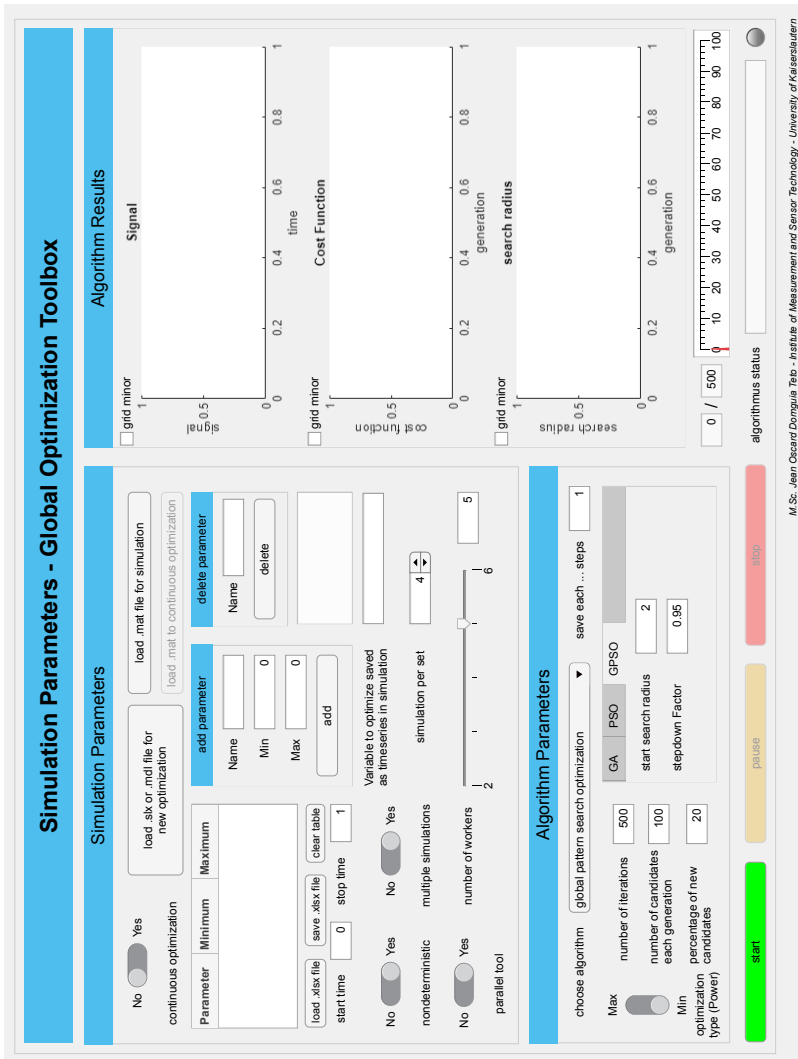


Figure A.1: Simulation optimization app - SISO version

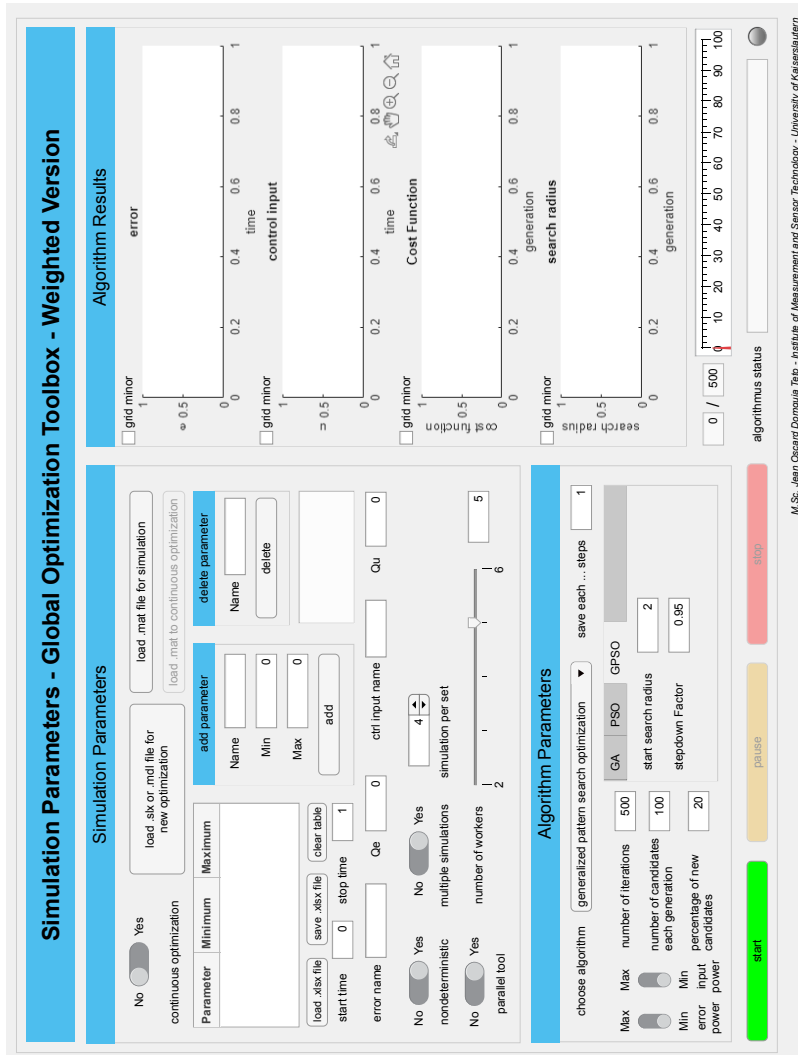


Figure A.2: Simulation optimization app - SISO Weighted version

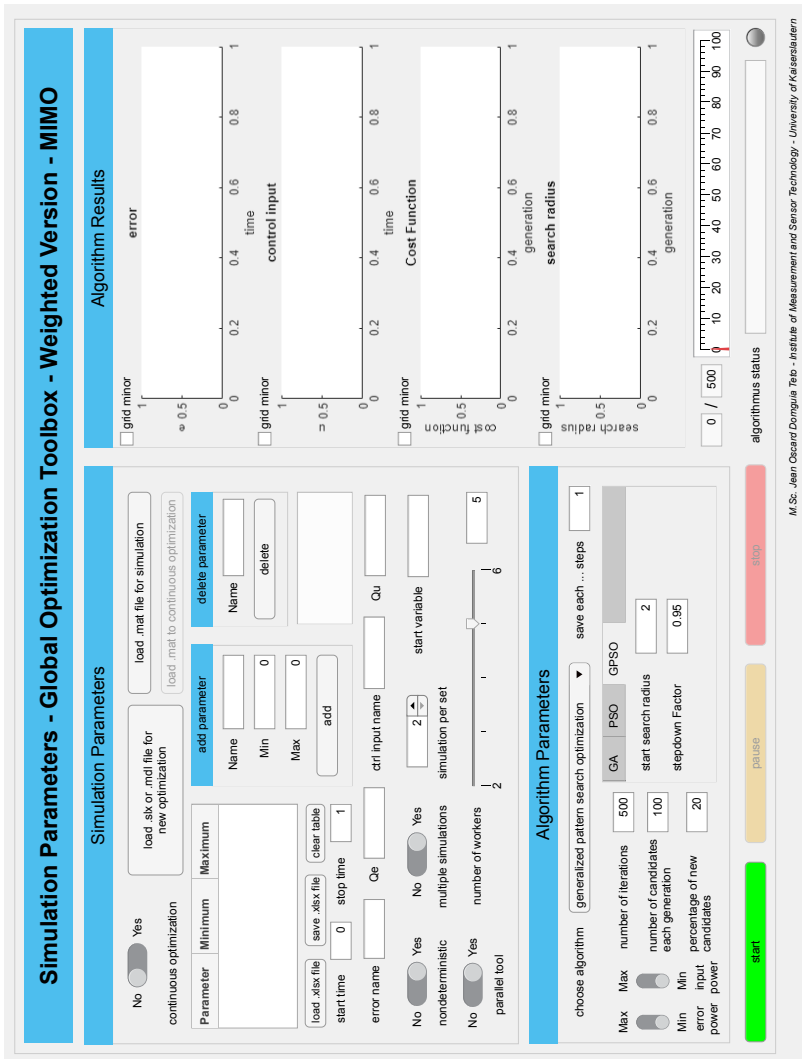


Figure A.3: Simulation optimization app - MIMO Weighted version

B Lebenslauf des Verfassers



Jean Oscard Domguia Teto

geboren in Jaunde / Kamerun

oscard.teto@yahoo.fr

Berufserfahrung:

- IAV GmbH Ingenieurgesellschaft Auto und Verkehr*
- ab 03/2022 *Entwicklungsingenieur*
- Lehrstuhl für Messtechnik & Sensorik, TU Kaiserslautern***
- 04/2019 – 02/2022 *Wissenschaftlicher Mitarbeiter und Doktorand in Gebieten der aktiven Geräuschunterdrückung (ANC) durch Gegenschall und der Vibrationsdämpfung für Anwendungen auf unbemannten Luftfahrzeugen (UAV)*
- Entwurf und Konstruktion von mechanischen Aktoren für Vibrationsdämpfung zur Anwendung auf UAVs
 - Entwurf von ANC Strategien und Regelalgorithmen zur Vibrationsdämpfung
 - Modellierung und Simulation mechanischer Systeme zur Auslegung und Bewertung von Regelstrategien
 - Konstruktion von Prüfständen zur experimentellen Validierung theoretischer Ansätze
- 06/2015 – 03/2019 *Wissenschaftliche und studentische Hilfskraft*
- Messaufgaben (optisch und taktil) im Labor
 - Programmierung Robotereinheiten
- Lehrstuhl für Technische Mechanik, TU Kaiserslautern***
- 02/2016 – 02/2019 *Wissenschaftliche und studentische Hilfskraft*
- Unterstützung der Forschungsarbeiten zur simulativen Beschreibung der Benetzung von strukturierten Bauteiloberflächen mit Tröpfchen
- Implementierung von Modellkomponenten
 - Generierung von Finite-Elemente-Netzen
 - Durchführung von Parameterstudien

KSBAG, Frankenthal

- 08/2017 – 12/2017 *Praktikum in der Abteilung „Anwendungs- und Grundlagenforschung“*
 Predictive Maintenance von Gleitringdichtungen mittels maschinellen Lernens mit Matlab und Python
- Untersuchung der Temperaturabhängigkeit der Lagerlebensdauer
 - Datenaufbereitung
 - Betriebszustandsprädiktion von Kreiselumpen mittels maschinellen Lernens
 - Predictive Maintenance von Gleitdichtungen mittels maschinellen Lernens

Akademischer & schulischer Werdegang:**TU Kaiserslautern**

- 04/2019 – 02/2022 *Promotion zum Doktor-Ingenieur (Dr.-Ing.)*
 Lehrstuhl für Messtechnik & Sensorik, Prof. Dr.-Ing. Jörg Seewig,
Dissertation: „Isolation of a measurement platform for UAV applications“
- 04/2018 – 02/2019 *Studium Fahrzeugtechnik, Master of Science (M. Sc.)*
Abschlussarbeit: „Aktive Schwingungsisolierung eines Totzeit-behafteten Systems mit Prinzipien der nichtlinearen Regelungstechnik unter Verwendung von Matlab und Simulink“
Projektarbeit: „Entwicklung eines robusten Reglers für das Einrad-Segway bezogen auf beliebige Anwender“
Zusatzleistungen: Model Predictive Control, Nichtlineare und adaptive Regelung, Robust Control, Methoden der Soft-Control, Modellbildung und Identifikation, CAE in der Regelungstechnik, Grundlagen der Automatisierung, Systemtheorie, Englisch B2 Refresher
- 10/2014 – 03/2018 *Studium Maschinenbau mit der Vertiefung Fahrzeugtechnik, Bachelor of Science (B. Sc.)*
Abschlussarbeit: „FE-Implementierung eines Phasenfeld-Elements zur Modellierung der Wechselwirkung zwischen Fluiden und Festkörpern“
Zusatzleistungen: Einführung in wissenschaftliches Programmieren, Computer Aided Styling (CAS), Systemtheorie, Getriebetechnik, Finite Elemente, Wärmeübertragung, Kontinuumsmechanik, Thermodynamik der Mischungen, Funktionstheorie, Englisch B1-2, Ingenieur und Zukunft

Institut der Erfolg

06/2013 – 07/2014 Deutschkurs: TestDaF am Goethe-Institut Kamerun

CPLP-Gymnasium – Jaunde / Kamerun

09/2007 – 06/2013 Abitur in Physik und Mathematik

Lehrtätigkeiten:**TU Kaiserslautern****Übungen**

„Angewandte Regelungstechnik“ (WS 19/20 bis WS 2021/22),
 „Technische Mechanik I“ (WS 2021/22), „Systemtheorie“ (SS 2019),
 „Mess- und Regelungstechnik“ (WS 2019/2020)

Tutor

„Technische Mechanik I, II, III, und IV, und Elemente der Technische
 Mechanik I, und II“ (WS 16/17 bis WS 18/19),
 „Höhere Mathematik I, II, und III“ (WS15/16 bis WS 17/18)

Sprach- und EDV-Kenntnisse:**Sprachen**

Französisch (Muttersprache)
 Deutsch (TestDaF 4x4)
 English (B2 Refresher)

EDV-Kenntnisse

Programmierung: Matlab/Simulink, Python, Arduino, Maple, C
 CAD: Autodesk Inventor, Siemens NX, Solid Edge, Catia
 Robotersteuerung: EV3, Microsoft Small Basic
 Latex, MS Office

Auszeichnungen:

2019 Stiftungspreis der Familie Dr. Jürgen Ziegler Stiftung

Ehrenamtliche Tätigkeiten:**Soutien Academique (Hochschulgruppe an der TU Kaiserslautern)**

seit 04/2021 Berater der Hochschulgruppe Soutien Academique

04/2018 – 03/2021 Leiter der Hochschulgruppe Soutien Academique

seit 04/2018 Gründer, Mietglied und Tutor der Hochschulgruppe Soutien Academique

Africa Development Incubator (ADI)

seit 04/2019 Projektmanager of “ADI Connect Schools-Projekt”

seit 04/2019 Mietglied von Africa Development Incubator

Publikationen und Referenzen:

ICCMA 2021

11/2021

- *Adaptive IIR filtering for online system identification with stability monitoring*
- *Stabilization of nonlinear continuous systems based on discrete integrator backstepping*

ICMCE 2020

09/2021

*Reference tracking of dynamic system using prefilter and integral extension***Referenzen****Google Scholar**<https://scholar.google.de/citations?user=B4Q91xgAAAAJ&hl=de>**UNISPECTRUMlive**<https://www.unispectrum.de/node/311><https://www.unispectrum.de/node/22>

Berichte aus dem Lehrstuhl für Messtechnik und Sensorik

bereits veröffentlicht wurden

- 1 Wendel, M.: Qualifizierung eines Streulichtsensoren und Untersuchung erster Ansätze zur dreidimensionalen Streulichterfassung
2015, ISBN 978-3-95974-006-7 € 39,-
- 2 Schäfer, P.: Modellbasierte Entwicklung pneumatischer Abstandssensoren für prozessintegrierte Messungen
2015, ISBN 978-3-95974-009-8 € 39,-
- 3 Eifler, M.: Modellbasierte Entwicklung von Kalibriernormalen zur geometrischen Produktspezifikation
2016, ISBN 978-3-95974-027-2 € 39,-
- 4 Arnecke, P.: A measurement method for characterising micro lead on ground shaft surfaces
2017, ISBN 978-3-95974-047-0 € 39,-
- 5 Kusnezowa, T.: Möglichkeiten und Herausforderungen bei der taktilen 2D Rauheitsmessung auf Konturen
2017, ISBN 978-3-95974-050-0 € 39,-
- 6 Torner, F. M.: Entwicklung virtueller, optischer Sensoren zur Charakterisierung geometrischer Oberflächen
2018, ISBN 978-3-95974-085-2 € 39,-
- 7 Alapurath George, B.: Web-Based Reference Software For Characterisation Of Surface Roughness
2018, ISBN 978-3-95974-095-1 € 39,-
- 8 Wiehr, C.: Anwenderunterstützung bei der Nutzung und Überprüfung von optischen 3D-Oberflächenmessgeräten
2019, ISBN 978-3-95974-110-1 € 39,-
- 9 Eifler, M.: Tendenzen und Herausforderungen in der geometrischen Produktspezifikation am Beispiel der Rauheitsmesstechnik
2019, ISBN 978-3-95974-117-0 € 39,-
- 10 Ströer, F.: Modellbasierte Entwicklung der Betriebselektronik für eine Rasterkraftsonde im Frequenzmodulationsverfahren zum Messen technischer Oberflächen
2020, ISBN 978-3-95974-129-3 € 39,-
- 11 Mansel, H.: Pneumatische Inline-Messung formgehoher Zylinderlaufflächen
2020, ISBN 978-3-95974-141-5 € 39,-

Berichte aus dem Lehrstuhl für Messtechnik und Sensorik

bereits veröffentlicht wurden

- 12 Sivasothy, P.: Potentialanalyse eines nichtinvasiven Sensorkonzepts zur Füllstandüberwachung bei mobilen Schüttgutsilos
2021, ISBN 978-3-95974-160-6 **€ 39,-**

- 13 Teto, JO D.: Isolation of a measurement platform for UAV applications
2022, ISBN 978-3-95974-175-0 **€ 39,-**

ISBN: 978-3-95974-175-0

ISSN: 2365-9742

Kurzfassung

Im Bereich Messtechnik kommt der Einsatz von unbemannten Fluggeräten immer häufiger in Frage. Für zahlreiche Messaufgaben bietet der Einsatz solcher Geräte viele Vorteile bezüglich der Kosten und der zu treibender Messaufwand. Jedoch sind die auftretenden Vibrationen und Störungen ein wesentlicher Nachteil für die Anwendung von diesen Geräten bei vielen Messaufgaben. Im Rahmen dieser Arbeit wird eine Plattform für Messgeräte entwickelt. Die Plattform wird speziell für den Einsatz auf Drohnen entworfen. Die Aufgabe der Plattform ist die darauf angebrachte Messeinrichtung von den Drohnenstörungen zu isolieren. Zu diesem Zweck gehen wir durch das Produktentwicklungsprozess nach VDI 2221, um ein mechanisches Modell der Plattform zu entwickeln. Dann werden Regelstrategien zur Isolierung der Plattform entworfen. Da die auf einer Drohne wirkenden Störungen nicht immer stationär sind, werden zwei Regelstrategien angewendet, die für ihren Umgang mit unsicheren Systemen bekannt sind, wobei eine von den beiden Methoden aus dem Bereich Akustik kommt.

Abstract

In the field of measurement technology, the use of unmanned aerial vehicles is becoming more and more popular. For many measurement tasks, the use of such devices offers many advantages in terms of cost and measurement effort. However, the occurring vibrations and disturbances are a significant disadvantage for the application of these devices for several measurement tasks. Within the scope of this work, a platform for measurement devices is developed. The platform is designed specifically for use on drones. The task of the platform is to isolate measurement equipments mounted on it from the drone disturbances. For this purpose we go through the product development process according to VDI 2221 to design a mechanical model of the platform. Then, control strategies are applied to isolate the platform. Since the disturbances acting on a drone are not always stationary, two control strategies known for their ability to handle uncertain systems are used. One of them comes from the field of acoustic.