In: The 5th IEEE International Advanced Motion Control Workshop (AMC'98)

A distributed planning and control system for industrial robots

Christian WURLL, Dominik HENRICH, and Heinz WÖRN
Institute for Process Control and Robotics (IPR), University of Karlsruhe
[wurll, dHenrich, woern]
@ira.uka.de
http://wwwipr.ira.uka.de/

Jan SCHLOEN and Martin DAMM
Fa. IRA-TEC, Hardtstraße 35, D-76185 Karlsruhe,
[mdamm, schloen]
@ira-tec.de,
http://www.ira-tec.de/

Wolfgang MEIER
Fa. AMS, Mannheimer Str. 72, D-68753 Waghäusel
meier@ams-online.de,
http://www.ams-online.de/

Abstract

A practical distributed planning and control system for industrial robots is presented. The hierarchical concept consists of three independent levels. Each level is modularly implemented and supplies an application interface (API) to the next higher level. At the top level, we propose an automatic motion planner. The motion planner is based on a best-first search algorithm and needs no essential off-line computations. At the middle level, we propose a PC-based robot control architecture, which can easily be adapted to any industrial kinematics and application. Based on a client/server-principle, the control unit establishes an open user interface for including application specific programs. At the bottom level, we propose a flexible and modular concept for the integration of the distributed motion control units based on the CAN bus. The concept allows an online adaptation of the control parameters according to the robot's configuration. This implies high accuracy for the path execution and improves the overall system performance

Keywords: robot motion planning, search algorithm, distributed control system, client/server-architecture, PC-based robot control

1 Introduction

Today, industrial robots are often used in complex and expensive installations in big industries (i.e. car industry) and their supplier companies. For economic rentability, powerful control and planning systems are necessary [Wörn98]. The high number of pieces to be produced and their wide individualistic product diversity mean that programs and workcells are steadily changing. Additionally, the high quality standard can often only be achieved by a sensor controlled assembly method. Therefore, the up-to-date control and planning systems for industrial robots must be easy to program and modularly designed, thus enabling a simple extension or integration of new components.

Due to the widespread achievability, its good price-cost-ratio and its easy extensibility, the application of standard PCs is ever increasing. Even in the field of robotics, low-cost PCs equipped with standard operation systems like Windows are used more and more.

In this paper, we present a new distributed planning and control system for industrial robots, which fulfills all requirements mentioned above.

The paper is organized as follows: In Section 2, the new approach for the planning and controlling of industrial robots is introduced. Section 3 describes an experimental environment which has been presented at the industrial trade fair in Nürnberg (Germany). The paper ends with the conclusion and an outlook of the future investigation in Section 4.

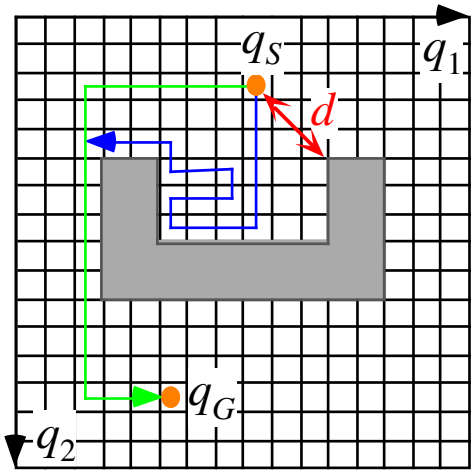

Figure 1: A 2D illustration of the path search in the implicit C-space from the start $\mathbf{q}_S$ to the goal $\mathbf{q}_G$ using the Cartesian obstacle distance $\mathbf{d}$ for collision detection

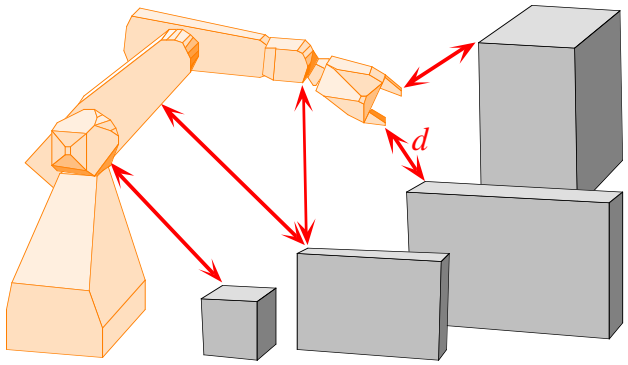

Figure 2: Collision detection in the explicit workspace by computing the minimum distance $\mathbf{d}$ between robot and obstacles

2 Distributed approach

The proposed distributed approach consists of three independent levels. Each level is modularly implemented and supplies an application interface (API) to the next higher level. The communication between these levels is mainly established based on the TCP/IP protocol. At the top level, we propose an automatic motion planner (see Section 2.1). Although the middle and bottom levels, which are responsible for the robot trajectory control and the robot's motor drive control, are independently developed, they are closely coupled and will be explained together (see Section 2.2).

2.1 Motion planning

The issue of robot motion planning has been studied for a couple of decades and many important contributions to the problem have been made. Motion planning algorithms are of great theoretical interest, but are rarely used in practice because of their computational complexity. But new applications, like robot assistant surgery [Burghart98], automatic off-line programming [Wörn98] or virtual engineering, enforces efficient motion planners which are able to find a valid solution either on-line or off-line. Concerning the application of off-line programming for example, even now, industrial robots are programmed manually by the "Teach-in" method. During this programming time, the capital intensive installations cannot be used. Afterwards, a time consuming optimization of the programmed trajectory follows (for example, by an iterative adaptation of the velocity profiles). The result is an optimized trajectory which strongly depends on the programmer's experience. But with the help of a fast motion planner the quality of the robot programs can be increased and the idle times can be reduced.

In comparison, the application of robot assistant surgery, here a fast on-line motion planner, is needed to control the robot automatically. This is necessary to relieve the surgeon from the robot control. Thus, he is able to concentrate completely on his surgical intervention.

In order to satisfy both the on-line as well as the off-line planning capabilities, we have developed a new motion planning approach.

Most of the off-line path planners are based on an explicit representation of the *free configuration-space (C-space)*. The free C-space computation consists of the obstacle transformation into the C-space and the construction of a free-space representation. Both tasks are very time and memory consuming, and their calculation effort increases with the robot's DOF. In order to avoid these time consuming obstacle transforma-

tions, one can search in an implicitly represented C-space and detect collisions in the Cartesian workspace. This strategy basically enables the planner to cope with on-line provided environments and moving obstacles. See Figure 1 and Figure 2.

For searching the implicit C-space, any best-first search mechanism can be applied. We choose a variation of the well known A*-search algorithm [Hart68]. Robot configurations (nodes) still to be processed are stored in a priority list OPEN, while already processed nodes are stored in a hashing table CLOSED. Contrasting with the original A*, here, no reopening of nodes in CLOSED is performed.

An evaluation function $f(n) = (1–w)\ g(n) + w h(n)$ is used, where $g(n)$ is the number of nodes of the path from the start node to node n, and $h(n)$ is the Airplane distance in C-space between node n to the goal node. Increasing the weight $w \in [0, 1]$ beyond 0.5 generally decreases the number of investigated nodes, while increasing the cost of the solutions generated. To improve the on-line capabilities of the path planner, our search is strongly directed to the goal by setting $w = 0.99$ [Wurll97b].

Collisions are detected by a fast, hierarchical distance computation in the 3D workspace, based on the polyhedral model of the environment and the robot provided by common CAD systems [Henrich92, Henrich97]. With the help of the "MaxMove Tables", introduced in [Katz96], the Cartesian distances are then transformed into joint angles in order to determine whether the current configuration collides or not. For obtaining similar joint intervals, thus implicating an efficient distance exploitation, the optimal joint discretization is automatically computed based on the method of [Qin96].

For speeding up the motion planner, we have enhanced the basic approach in different ways. It was greatly accelerated by parallelizing the A*-algorithm. By decomposing the 6D C-space into hypercubes and mapping them on multiple processing units in a cyclic manner, a good load distribution can be achieved. This new approach usually shows linear and sometimes even super-linear speedups. In on-line environments with static obstacles, the parallel planning times are only a few seconds [Wurll98, Henrich98a]. To use the properties of the inverse kinematics, the search algorithm was extended to support multiple goals in the C-space, thus selecting the suitable goal configuration by itself [Henrich98b]. For an additional acceleration of the motion planning, we discretisize the C-space hierarchically, thus reducing the search space [Osterroht98, Henrich98c].

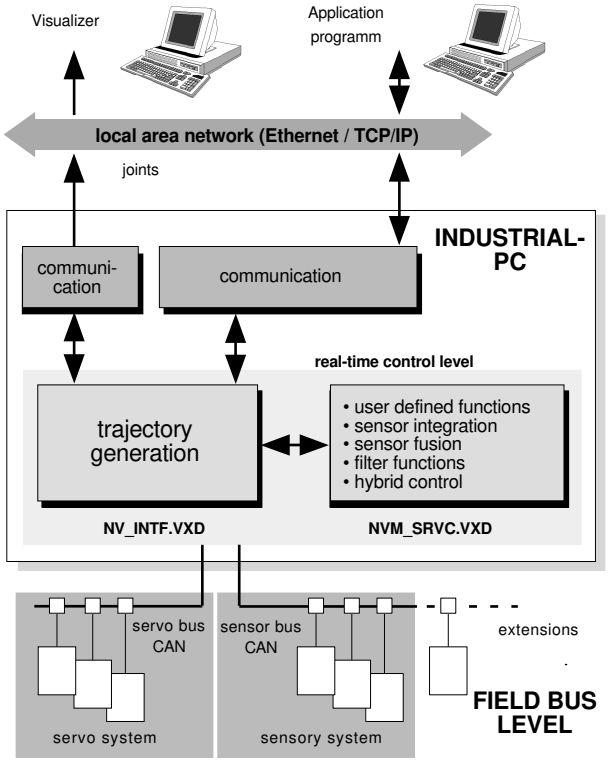

Figure 3: Structure of the PC-based robot control system

2.2 Robot control system

The robot control system being used for the experiments is the system NoVAL+, which is a modular and flexible motion control system for industrial robots based on an industrial PC (IPC) and the operating system Windows. The system has been developed within a large research project [Damm93] and is now commercially avail-

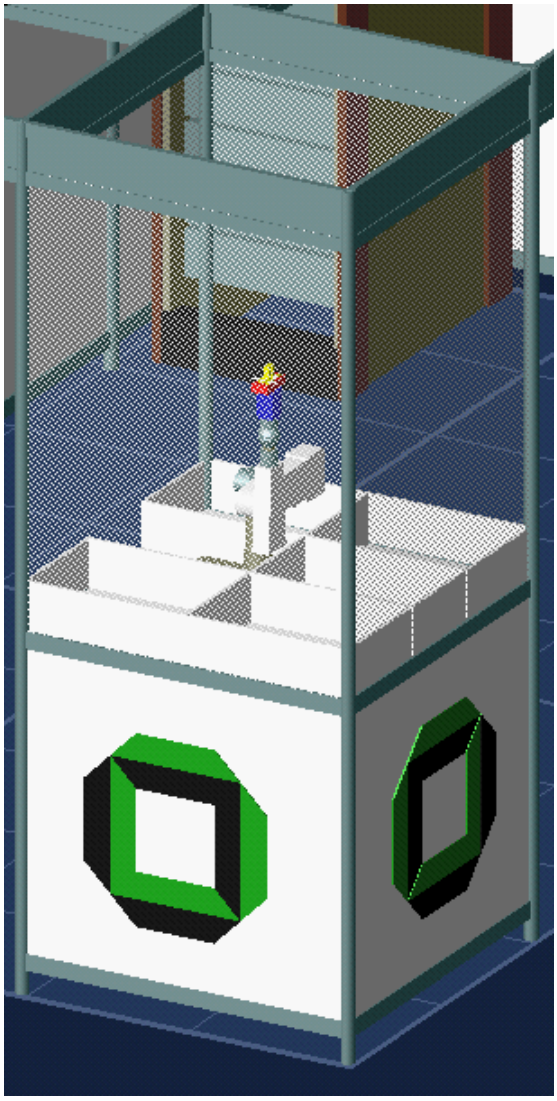

Figure 4: Screenshot of the CAD-model in a robot simulation system (ROBCAD)

able. Because of its modular concept, it can easily be adapted to different kinematics and applications. The main structure of NoVAL+ is presented in Figure 3.

The trajectory generation module is realized as a device driver module within the operating system using the NMI[1] timer of the IPC due to the strict real-time requirements. The integration of the distributed axis control modules is realized using the CAN bus. An on-line adaptation of the control parameters, based on the robot's configuration and the robot task being executed, is implemented to improve the accuracy for the path execution and the overall system performance, especially for high speed applications. All external peripheral components, like grippers and sensors, are connected to the IPC via a second CAN bus.

The control unit establishes an open user interface for including application specific modules on the real-time control level to easily realize user defined control functions.

A main feature of NoVAL+, in comparison to conventional robot control systems, is the client-server architecture, which enables the control system to be easily integrated into TCP/IP-based local control networks. Therefore, a robot application program can be implemented on the robot control IPC itself or on any other PC in a network. To realize powerful robot applications, a C/C++-based application programmer interface is available. Additionally, a TCP/IP-based on-line visualization is provided for monitoring the robot's movements.

3 Experimental environment

For testing the different components, we have designed an experimental setup. Basically, it consists of a Silicon Graphics (SGI) Workstation on which the robot simulation program ROBCAD runs to define the CAD model of the workcell and to specify the motion planning task. The CAD model of the experimental environment in robot simulation system is shown in Figure 4. The corresponding real robot work cell is depict in Figure 5. The workcell includes an industrial robot Puma260, fixed on a platform with five boxes and surrounded by Plexiglas windows. Supported by a graphical user interface based on the ROBCAD Open System Environment (ROSE), the parallel motion planner can be started and controlled. One simple task consists, for example, of planing a collision-free path from

[1] NMI stands for "non maskable interrupt".

the upper left box to the lower right box (see Figure 4).

The motion planner itself runs on the IPR-ParaStation, a PC-based workstation cluster [Wurll97a]. It only needs the CAD model and the start and goal configurations and an address to which it sends the solution path. In our scenario, a solution is successfully found in less than one second. The path is then sent to the robot simulation tool for visualization and is sent to the robot control unit for execution. For an optimal path execution the trajectory has been smoothed by a spline interpolation in the C-space, which is a special motion skill on the robot control unit.

Figure 5: The real robot work cell presented at the trade fair SPS/IPC/Drives in Nürnberg, 1997

To our knowledge, this is the first published[2] experimental setup, where a 6 DOF motion planner is efficiently used for an on-line path planning with a life path execution on an industrial robot.

4 Conclusions and future work

In this paper, we have introduced a new approach to a distributed planning and control system for industrial robots. It consists of three independently developed components. At the top level of the hierarchical concept, there is an efficient parallel motion planner for 6 DOF industrial robots which does not need any time consuming off-line computations. In the middle level, we have presented a PC-based robot control unit which can easily be adapted to new kinematics and which offers an open interface for the peripheral integration. At the bottom level, we have proposed a bus-based coupling of distributed motion control units. The concept allows an on-line adaptation of the control parameters according to the robot's configuration. By setting up an experimental result, we have proven the performance of the overall concept. This is the first scenario known to the authors, in which a motion planner has computed a suitable path for execution on a real workcell.

Future investigation will focus on the improvement of the presented components. Concerning the motion planner, we concentrate on extending this path planner to be able to cope with moving obstacles, such as other robots. Regarding the robot control unit, new applications, like medical surgery, enforces high safety requirements and short reaction times, which will be included in the next software revision.

5 Acknowledgment

This research work has been performed by the Institute for Process Control and Robotics (IPR) at the University of Karlsruhe and by the German companies IRA-TEC and AMS. Parts of the work is funded by the German Research Foundation in the project "Scalable algorithms for parallel motion planning in dynamic environments" and part of the special research program *SFB314*. Further information can be found in the Web at

- http://wwwipr.ira.uka.de/~paro/,
- http://wwwipr.ira.uka.de/~megi/
- http://www.ira-tec.de/, and
- http://www.ams-online.de/.

[2] There is an other experimental setup at the University of Munich. But this has not been published yet.

6 References

[Burghart98] Burghart Ch., Wurll Ch., Henrich D., Raczkowsky J., Rembold U., Wörn H.: "On-line motion planning for medical applications", In: IECON'98: The 24th Annual Conference of the IEEE Industrial Electronics Society, Aachen, Germany, August 31 - Sept. 4, 1998.

[Damm93] M. Damm, D. Kappey, J. Schloen: "A Multi-sensor and Adaptive Real-time Control Architecture for an Autonomous Robot", Intelligent Autonomous Systems IAS-3, International Conference, Pittsburgh, Pennsylvania, Februar 1993, pp. 439-448.

[Hart68] Hart P.E., Nilsson N.J., Raphael B.: "A formal basis for the heuristic determination of minmum cost paths", IEEE Trans. Syst. Sci. Cybern, pp. 100-107, 1968.

[Henrich92] Henrich D., Cheng X., "Fast distance computation for on-line collision detection with multi-arm robots", IEEE Int. Conf. on Robotics and Automation, Nice, France, May 10-15, pp. 2514-2519, 1992.

[Henrich97] Henrich D., Gontermann S., Wörn H.: "Kollisionserkennung durch parallele Abstandsberechnung". In: 13. Fachgespräch Autonome Mobile Systeme (AMS'97), Stuttgart, October 6-7, 1997, Springer-Verlag, Reihe "Informatik Aktuell".

[Henrich98a] Henrich H., Wurll Ch., Woern H.: " Parallel motion planning for industrial robots ". In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA'98), Leuven, Belgium, May 16.-21., 1998

[Henrich98b] Henrich D., Wurll Ch., Wörn H.: "Multi-directional search with goal switching for robot path planning ". In: The 11th Int. Conf. on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Castellan, Spain, July 1-4, 1998.

[Henrich98c] Henrich D, Wurll Ch., Wörn H.: " Heuristic Hierarchical Motion Planning in Dynamic Environments", In: IECON'98: The 24th Annual Conference of the IEEE Industrial Electronics Society, Aachen, Germany, August 31 - Sept. 4, 1998.

[Katz96] Katz G.: "Konzeption einer Entwicklungsumgebung unter ROBCAD für die parallele Bewegungsplanung", Master's Thesis, Computer Science Department, University of Karlsruhe, 1996.

[Osterroht98] Osterroht T.: "Hierarchische parallele Bewegungsplanung für dynamische Umgebungen", Master's Thesis, Computer Science Department, University of Karlsruhe, 1998.

[Qin96] Qin C., Henrich D.: "Path planning for industrial robot arms - A parallel randomized approach", In Proc. of the Int. Symp. on Intelligent Robotic Systems (SIRS´96), Lissabon, Portugal, pp. 65-72, July 22-26, 1996.

[Wörn98] Wörn H., Wurll Ch., Henrich D.: "Automatic off-line programming and motion planning for industrial robots". In: The 29th Int. Symp. on Robotics, Birmingham, United Kingdom, April 27-30, 1998.

[Wurll97a] Wurll C., Henrich D.: "Ein Workstation-Cluster für paralleles Rechnen in Robotik-Anwendungen". In: Proceedings der 4. ITG/GI-Fachtagung Arbeitsplatz-Rechensysteme (APS'97), Universität Koblenz-Landau, May 21-22, 1997, pp 187-196.

[Wurll97b] Wurll Ch., Henrich D., Wörn H.: "Parallele Bewegungsplanung in dynamischen Umgebungen", Technical Report 20/97, Computer Science Department, University of Karlsruhe, 1997

[Wurll98] Wurll Ch., Henrich D., Woern H., 1998, "Parallel on-line motion planning for industrial robots", Proceedings of Robotics 98: The Third ASCE Specialty Conference on Robotics for Challenging Environments, Albuquerque, New Mexico, April 26-30, 1998