TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für
Betriebswissenschaften und Montagetechnik (*iwb*)

A WORK-PIECE BASED APPROACH FOR PROGRAMMING COOPERATING INDUSTRIAL ROBOTS

Sherif Zaidan

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Michael Zäh

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Gunther Reinhart
2. Univ.-Prof. Dr.-Ing., Dr.-Ing. habil Heinz Ulbrich

Die Dissertation wurde am 13.10.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 10.02.2012 angenommen.

Sherif Zaidan

A Work-Piece Based Approach for Programming Cooperating Industrial Robots

Herbert Utz Verlag · München

Forschungsberichte IWB

Band 259

Zugl.: Diss., München, Techn. Univ., 2012

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.

ISBN 978-3-8316-4175-8

Printed in Germany
Herbert Utz Verlag GmbH, München
089-277791-00 · www.utzverlag.de

Geleitwort der Herausgeber (Foreword)

Die Produktionstechnik ist für die Weiterentwicklung unserer Industriegesellschaft von zentraler Bedeutung, denn die Leistungsfähigkeit eines Industriebetriebes hängt entscheidend von den eingesetzten Produktionsmitteln, den angewandten Produktionsverfahren und der eingeführten Produktionsorganisation ab. Erst das optimale Zusammenspiel von Mensch, Organisation und Technik erlaubt es, alle Potentiale für den Unternehmenserfolg auszuschöpfen.

Um in dem Spannungsfeld Komplexität, Kosten, Zeit und Qualität bestehen zu können, müssen Produktionsstrukturen ständig neu überdacht und weiterentwickelt werden. Dabei ist es notwendig, die Komplexität von Produkten, Produktionsabläufen und -systemen einerseits zu verringern und andererseits besser zu beherrschen.

Ziel der Forschungsarbeiten des iwb ist die ständige Verbesserung von Produktentwicklungs- und Planungssystemen, von Herstellverfahren sowie von Produktionsanlagen. Betriebsorganisation, Produktions- und Arbeitsstrukturen sowie Systeme zur Auftragsabwicklung werden unter besonderer Berücksichtigung mitarbeiterorientierter Anforderungen entwickelt. Die dabei notwendige Steigerung des Automatisierungsgrades darf jedoch nicht zu einer Verfestigung arbeitsteiliger Strukturen führen. Fragen der optimalen Einbindung des Menschen in den Produktentstehungsprozess spielen deshalb eine sehr wichtige Rolle.

Die im Rahmen dieser Buchreihe erscheinenden Bände stammen thematisch aus den Forschungsbereichen des iwb. Diese reichen von der Entwicklung von Produktionssystemen über deren Planung bis hin zu den eingesetzten Technologien in den Bereichen Fertigung und Montage. Steuerung und Betrieb von Produktionssystemen, Qualitätssicherung, Verfügbarkeit und Autonomie sind Querschnittsthemen hierfür. In den iwb Forschungsberichten werden neue Ergebnisse und Erkenntnisse aus der praxisnahen Forschung des iwb veröffentlicht. Diese Buchreihe soll dazu beitragen, den Wissenstransfer zwischen dem Hochschulbereich und dem Anwender in der Praxis zu verbessern.

Gunther Reinhart *Michael Zäh*

Acknowledgment

This dissertation is the result of my work at the Institute for Machine Tools and Industrial Management (*iwb*) at the Technical University of Munich (TUM) during the period from December 2006 till November 2011.

First and foremost I would like to thank my supervisor Prof. Dr.-Ing. Gunther Reinhart for giving me the chance to realize a lifelong dream of pursuing a Doctor in engineering while simultaneously taking part in state-of-the-art research projects. His continuous guidance and the level of trust he granted me made the journey all the more exciting and joyful. I am also grateful to Prof. Dr.-Ing. Michael Zäh for his support throughout my work at the institute and for heading the examination committee. Hereby I would also like to thank Prof. Dr.-Ing. Heinz Ulbrich for his willingness to be part of the examination committee.

The research conducted during my five years at *iwb* wouldn't have been possible without the enormous support of the following colleagues; Andrea Acker, Marwan Radi, William Tekouo, Jens Hatwig, Wolfgang Rösel and Stefan Krug. Furthermore, the following students contributed immensely to my research work through their Bachelor's and Diplom thesis which I supervised; Jörg Hubele, Wolfgang Wittman, Robert Schamlholz, Oliver Krieg, Alexandre Girard, Thomas Menauer, Joachim Michniewicz and Maximilian Völkl. I thank you all wholeheartedly and sincerely wish you future success in your private and professional life.

I am also bound to my childhood friend Omar Hassan for his continuous encouragement, relentless support and his faith in me even when I doubted myself. In spite of the geographical factor, my father, my brother and my three lovely sisters Amira, Sherihan and Hind have always been there for me with their care, their prayers and their wholehearted love. Words fail to express the gratitude I owe to my mother, her unwavering support and her belief that I can reach my goals whatever they may be, instilling in me the confidence I needed to see this endeavor through. Thank you for all you have done for me, I am forever grateful.

Sherif Zaidan
March 2012

Abstract

One of the main characteristics of cooperating industrial robots is their capacity to share a common work-space as well as a common work-piece. Due to their cooperative nature and simultaneous contact through the work-piece, programming represents a challenging and complicated task. The objective of this research is to simplify the programming, by imparting full position and force control over the work-piece to the user. In order to achieve this, force observance and control had to be tightly integrated into the programming. As a result, a multi-layered control architecture and a flexible software environment were developed and deployed on an industrial test-rig. The approach was successfully validated by three different application scenarios covering off-line, on-line and intelligent programming of industrial robots.

Eines der wichtigsten Merkmale von kooperierenden Industrierobotern ist deren Fähigkeit, sich einen gemeinsamen Arbeitsraum sowie ein gemeinsames Werkstück zu teilen. Die Programmierung der Roboter stellt auf Grund des kooperativen Charakter und dem gleichzeitigem Kontakt mit dem Werkstück eine komplexe Aufgabe dar. Ziel dieser Forschungsarbeit ist es, die Programmierung zu vereinfachen, indem der Nutzer die Kontrolle über Positions- und Kraftregelung des Werkstücks erhält. Um dies zu erreichen, müssen Kraftbeobachtung und -regelung fest in die Programmierung integriert werden. Es wurden eine mehrschichtige Regelungsarchitektur und eine flexible Software-Umgebung entwickelt und in einem industriellen Prüfstand implementiert. Der Ansatz wurde erfolgreich durch drei unterschiedliche Anwendungsszenarien validiert. Diese sind die off-line, die on-line und die intelligente Programmierung.

to my mother

Nomenclature

Variables (coordinate systems)

$\mathcal{W}$	World/Global frame
$\mathcal{R}$	Robot/Base frame
$\mathcal{T}$	Tool/TCP frame
$\mathcal{O}$	Work-piece/Object frame

Variables (control)

$x_{\mathcal{K}}$	Vector of posture in $\mathcal{K}$ cartesian coordinates
θ	Vector of joint angles in a manipulator
$F_{\mathcal{K}}$	Vector of forces and torques in $\mathcal{K}$ cartesian coordinates
τ	Vector of joint torques i.e. motor torques in a manipulator
$\mathbf{0}$	Null matrix
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{M}$	Mass matrix (inertial components)
$\boldsymbol{B}$	Damping matrix (inertial components)
$\boldsymbol{K}$	Stiffness matrix (inertial components)
$\boldsymbol{C}$	Corilios and friction matrix
$\boldsymbol{G}$	Gravity matrix (gravitational components)
Λ	Workspace (degrees of freedom of a device)

Abbreviations

2D	2-dimensional
3D	3-dimensional
API	Application programming interface(s)
CAD	Computer aided design
CIR	Cooperating industrial robots
CNC	Computer numerical control(led)
DIN	Deutsches Institut für Normung (German Institute for Standardization)
DOF	Degree(s) of freedom
e.g.	For example

etc.	And so forth (Latin: et cetera)
FTS	Force torque sensor(s)
GUI	Graphical user interface(s)
HMI	Human machine interface(s)
HTN	Homogeneous transformation notation
I/O	Input/Output
ISO	International Organization for Standardization
i.e.	That is (Latin: id est)
PC	Personal computer
RCC	Remote center compliance
RTOS	Real-time operating system
RTP	Real-time platform
SME	Small and medium enterprise(s)
TCP	Tool center point
VQN	Vector quaternion notation
w.r.t.	with respect to
WPBA	Work-piece based approach

Contents

The first verses revealed in the Holy Quran (surat el-alaq 1-5)

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ

خَلَقَ الإِنْسَانَ مِنْ عَلَقٍ

اقْرَأْ وَرَبُّكَ الأَكْرَمُ

الَّذِي عَلَّمَ بِالْقَلَمِ

عَلَّمَ الإِنْسَانَ مَا لَمْ يَعْلَمْ

Read: In the name of thy Lord Who createth

Createth man from a clot

Read: And thy Lord is the Most Bounteous

Who teacheth by the pen

Teacheth man that which he knew not

(Marmaduke Pickthall translation)

1 Introduction

Coming together is a beginning. Keeping together is progress. Working together is success.

HENRY FORD

1.1 Cooperative manipulation

It is commonly known that losing one human limb may severely affect the ability to tackle tasks requiring physical effort. This is clearly evident in the difficulties facing humans with impairments or physical disabilities (WORLD HEALTH ORGANIZATION (WHO) 2007)[1]. Losing even a finger may disturb the balance of gripping and manipulating an object (LAURIG & VEDDER 1998, P. 29.76). From a mechanical point of view, the human body could be seen as composed of several cooperative manipulator systems working in tandem to facilitate the execution of everyday-life tasks (Figure 1.1). On the smallest scale of manipulation, fingers represent the first cooperative system in the human's arsenal. Roughly thirty degrees of freedom (DOF) exhibited by the human hand offer an unprecedented level of fine manipulation and numerous techniques for flexible gripping (LIN ET AL. 2000). On a larger scale the human arms represent the second system, whether attached to the same body or not. Their ability to augment fingers with load sharing and manipulation in larger workspaces, render the combined hand-arm system a more so sophisticated system. Although the human legs are seldom applied together with the arms[2], it nevertheless constitutes an important ingredient of the larger manipulation capability manifested in the human body. Legs provide stable locomotion which physically expand the manipulation workspace and render it virtually infinite. Remarkably, the latter three systems rarely act alone. Instead they work together in different arrangements in accordance with the task given. For instance, moving an object from one location to another entails the simultaneous and coordinated application of all three systems. Not only are human limbs capable of working alone, but they could be cascaded with others to collaborate on larger tasks.

[1] The WHO's International Classification of Functioning, Disability and Health (ICF)

[2] This statement is only true in the sense that hand-leg dependency is not biologically equivalent to hand-arm dependency. Otherwise it is clear that leg movement (locomotion), especially running, requires a specific synchronized arm movement

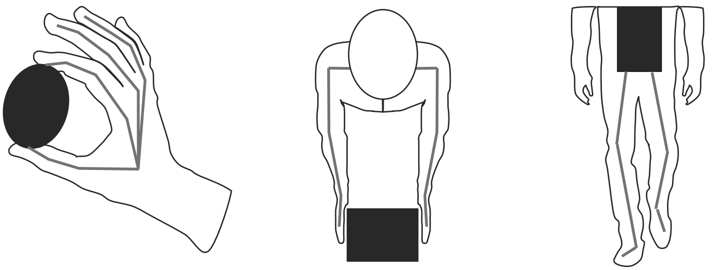


Figure 1.1: Cooperating manipulators in human beings

1.2 Cooperation in industrial manipulators

In the quest to automate human tasks, engineers have sought to build manipulators which could mimic the manipulation capabilities of humans[3]. Although far from replicating their performance, industrial robots helped propel production forward by rendering automation of monotonous tasks cost-effective, hereby immensely contributed to increasing manufacturing productivity (Craig 2005). As processes became more complicated and sophisticated, teams of robots cooperatively working together in unison were envisioned (Christensen et al. 2009, P. 10). From sharing a common workspace to processing a shared work-piece in tandem to working with humans in their vicinity; cooperating industrial robots (CIR) were regarded as a natural extension to the development of industrial robotics and a prerequisite for future flexible manufacturing paradigms (Rückel 2006)(Gausemeier et al. 2011). As the next logical step in a booming industry, robot manufacturers started marketing their respective cooperating robot systems in the mid 2000's. Given their distributive and dexterous nature, CIR are ideal as flexible jigs and fixtures (Figure 1.2). By allowing several robots to hold a work-piece and to move it in tandem, the work-piece could be flexibly positioned to be processed (Stoddard et al. 2004)(Inaba & Sakakibara 2009, P. 357). Furthermore, this permits access to areas on the work-piece which are otherwise deemed inaccessible with static fixturing (Kurth 2005). The dexterity of CIR also makes them very useful in handling very large or even flexible work-piece which require synchronized movements to retain a given (Sun et al. 1996)(Tzeranis et al. 2005).

In addition to utilizing the robots as dynamic fixtures, they also simplify the material flow through the work-cell by undertaking a secondary role as part handlers. Analogous

[3] According to the Merriam-Webster dictionary a robot is: *a machine that looks like a human being and performs various complex acts (as walking or talking) of a human being*

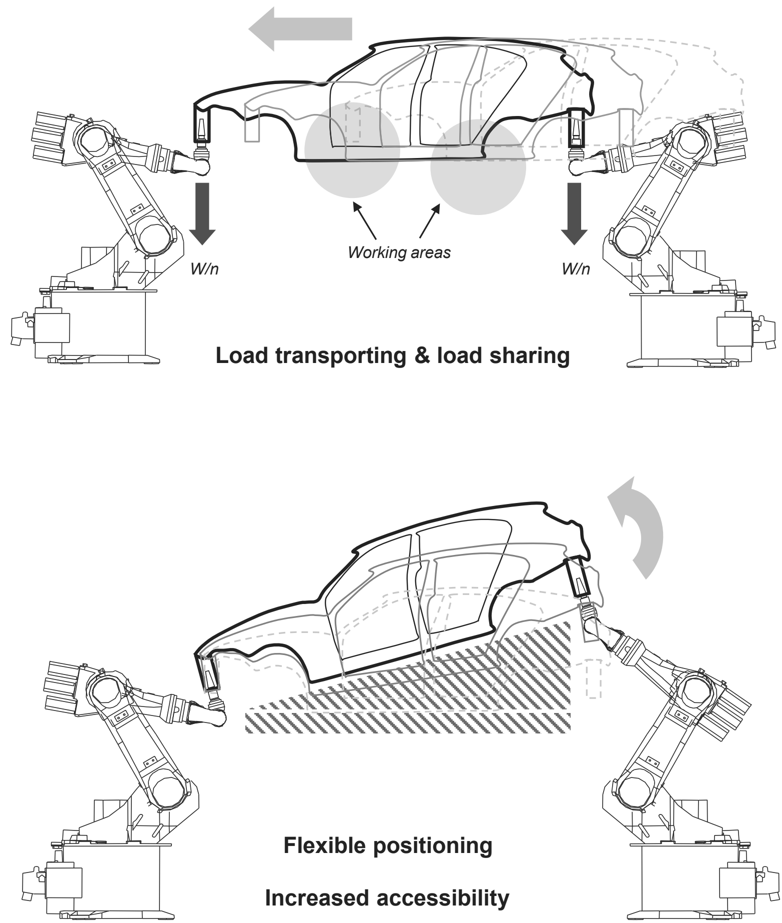


Figure 1.2: Capabilities of cooperating industrial robots

to cooperation in human bodies, the advantages of CIR are increased dexterity and load sharing capability. Indeed these are readily translated into manufacturing-wide gains such as enhanced process productivity, achieving higher quality and additional flexibility (Arai et al. 1995)(Rückel & Feldmann 2005). Attaining higher process productivity ensues through reducing the investments by building more compact systems and hence requiring less components. Additionally saving redundant intermediate steps by executing processes during transport and avoiding changing fixtures which in turn reduces process cycle time and thereby increasing productivity (Reinhart et al. 2008b). On the other hand, process quality has a direct impact on product quality. Given that industrial robots are capable of accurate positioning with high repeatability this significantly increases the product quality (Reinhart & Zaidan 2009). During the execution of a process, maintaining a constant relative speed would result in a lower cycle time while simultaneously guaranteeing the process quality stays within the permissible ranges. Regarding flexibility, industrial robots are without a doubt considerably easier to reconfigure by re-programming, compared to conventional jigs and fixtures which are constructed of static structures (Lewis et al. 2003). Since industrial robots are distinguished by their capability to be adopted to variable process needs, they automatically increase the flexibility of a given process (Venkatesan & Karnan 2009).

1.3 Scope of work

Regarding tightly and loosely coupled systems
Cooperative operation with industrial robots covers two basic areas. The first represents the act of active engagement in executing a task together. This dictates that the manipulators are dynamically connected, commonly termed tightly coupled (Tzafestas et al. 1998). On the other hand if the manipulators don't physically interact i.e. executing tasks simultaneously, they are commonly termed loosely coupled (Cardarelli et al. 1995). Since in both cases the manipulators share a common workspace, the emphasis on the former is the dynamic interaction between the manipulators and on the latter is how to avoid collision between the manipulators during movement (Terada 1998). The thesis addresses the first case and to some degree, the second.

Regarding robot type or class
Although the manipulators under consideration in this work are typical commercial robots similar to those used in production facilities worldwide, it is not the author's intent to limit the scope of the ideas developed here to this particular class of robots. Hence, the arguments and methods will be treated in a manner rendering them applicable to other classes of robots, for instance dual-arm or mobile robots (Park & Park 2008)(Kume et al. 2002b). These are particularly interesting due to their small weight/payload ratio and possible mobility. Furthermore, their limited perimeter requirements make them an ideal candidate for two-handed tasks, entailing a high level of dexterity and fine manipulation.

1.4 Thesis outline

This thesis is divided as shown in Figure 1.3: Chapter 2 explores topics addressing CIR in both research and industrial field. Interdisciplinary topics such as programming and interaction control are not exclusively handled from a CIR point of view in order to maintain their generic scope. Based on apparent drawbacks of available technology and research efforts in this field, an approach to simplify programming of CIR is proposed in chapter 3. To translate this approach to reality, task and system prerequisites are investigated in chapter 4. Consequently a conceptual framework is developed to implement the proposed approach is developed. This framework serves as the backbone of the three subsequent chapters. In chapter 5 the control ideas behind the proposed concept are analyzed and a control architecture representing a realization of the *control module* in the framework will be developed. In chapter 6 a software environment representing the technical realization of the *software module* will be developed. Chapter 7 discusses the development of an industrial test-rig which covers the rest of the framework, namely *device components* and *communication architecture*. Chapter 8 demonstrates the feasibility of the proposed approach and its technical implementation by validating it in three different programming scenarios. Subsequently, the approach will be assessed from a technical and economical point of view. Finally, chapter 9 concludes the thesis and presents an additional outlook for further research in this field.

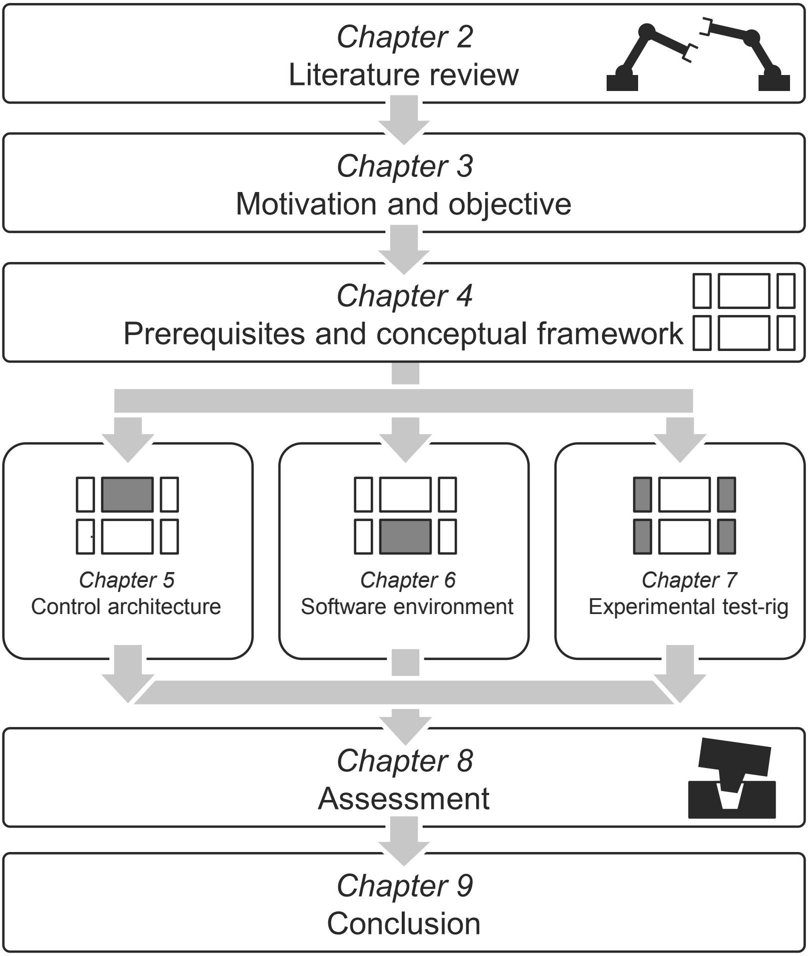


Figure 1.3: Thesis outline

2 Literature Review

To know, is to know that you know nothing. That is the meaning of true knowledge

SOCRATES

2.1 Introduction

CIR denote a group of industrial manipulators sharing a common workspace (LUH & ZHENG 1987). This definition avoids specifying the type of interaction between the manipulators and hence implicitly includes all interaction types in a common or shared workspace. Accordingly, the field of CIR could be roughly divided into two main branches (TZAFESTAS ET AL. 1998). The first branch primarily addresses the case of CIR sharing a common workspace, but nevertheless performing independent tasks namely contact-free (CARDARELLI ET AL. 1995). The main issue encountered here is to generate optimal trajectories for the concurrent movement of the manipulators while simultaneously avoiding inter-robot collisions, satisfying task requirements and keeping within the manipulators' joint constraints (CHOI ET AL. 1999). On the other hand, the second branch deals with robots which not only share a common workspace but also share the same work-piece (KOSUGE ET AL. 1997). In this case the robots are expected to physically interact with each other through this work-piece which represents the focus of the process. As mentioned in section 1.3 this work covers the latter class. Owing to the diverse nature of the thesis, the literature review is not limited to scientific research in this area but also to state-of-the-art commercial systems and modern industrial practices. The multi-disciplinary nature of the work requires dividing this chapter into three main sections:

1. Industrial robot programming techniques: classical and modern methods for programming industrial robots are introduced.

2. Interaction control of industrial robots: definition and classification of interaction control for robotic manipulators is discussed in both research and industrial capacities.

3. Cooperating manipulators: the first part thereof will discuss the work done in cooperative manipulation in the research community and the second will investigate state-of-the-art commercial systems and their common features.

At the end of the chapter a discussion will highlight the limitations of current CIR systems. As seen in Figure 2.1. The case for the proposed approach in this thesis will be constructed from the overlap between the latter three topics.

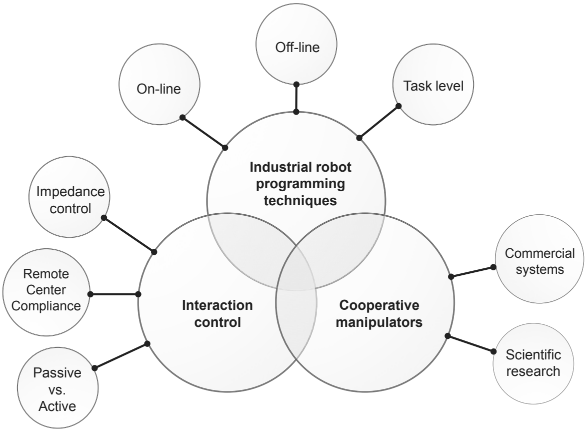


Figure 2.1: Literature classification

2.2 Industrial robot programming techniques

As specified in (DIN EN ISO 8373 1996), programming an industrial robot entails creating a user application for a production task. Research efforts tend to classify different programming techniques according to the location of programming (Humburger 1997)(Weck & Brecher 2006, P. 378). In this classification scheme, if programming occurs either in the vicinity of the robot or directly on the factory floor, it is termed ’on-line’ (DIN EN ISO 8373 1996, section 5.2.3). While creating a program in a location separate from the robot is termed ’off-line’ (DIN EN ISO 8373 1996, section 5.2.4). Other researchers classified programming techniques according to the level of task abstraction (Biggs & Macdonald 2003)(Haun 2007)(Dillmann 2010). In the following sections classical on-line/off-line programming techniques will be introduced and subsequently augmented with task-oriented methods thereby covering goal-oriented programming techniques as described in (DIN EN ISO 8373 1996, section 5.2.5).

2.2.1 On-line programming

On-line programming is characterized by leading the robot through a sequence of desired movements by an operator in the vicinity of the robot (DIN EN ISO 8373 1996, section 5.2.3). Although these methods are easy to learn and produce highly accurate robot programs executed on the robot (see Figure 2.2), they cause a significant loss in production time especially if the task is complicated (Hesse & Malisa 2010, P. 74). This is attributed to the fact that programming takes place in a production cell on the factory floor and hence contributes to production downtime (Haegele et al. 2008, P. 977). According to (DIN EN ISO 8373 1996, section 5.2.3) the two major on-line programming techniques are programming by Teach-In and Lead-through methods.

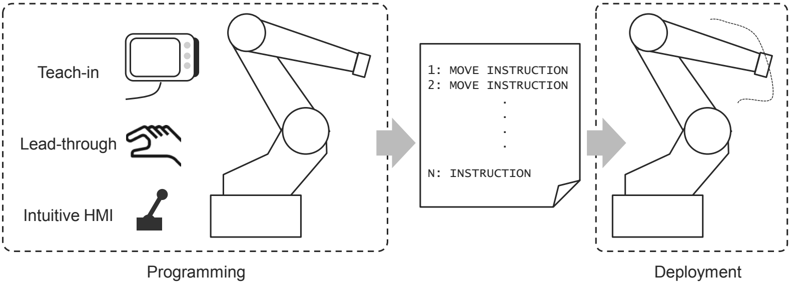


Figure 2.2: On-line programming process

2.2.1.1 Teach-in

This method is based on an ergonomically designed teach-pendant which allows the user to precisely control the robot's movement whilst being in its vicinity (Hesse & Malisa 2010, P. 74). Such pendants usually contain programming specific features such as buttons for saving and editing the program, a joystick[1] for easy manipulation of the robot or even a 3D Mouse[2]. Additionally, process specific features such as emergency stop and safety indicators are usually included (Kreuzer et al. 1994, P. 247)(Holzbock 1996, P. 344). In addition to programming those pendants serve as the human machine interface (HMI) with the robot. Hence they are coupled to the robot controller from the respective manufacturer (Deisenroth & Krishnan 1999, P. 341). The teach-in process commences by moving the robot in a reduced speed mode[3] along the desired path while constantly checking whether the points on the path correspond to the actual process. The programmer is allowed to move the robot in either tool center point (TCP) coordinates or joint coordinates i.e. separately moving each motor, during which key-points are saved in the program. Finally, the user can add branching and logic instructions to the movement instructions and test the program to check whether it corresponds to the desired functionality (Holzbock 1996,

[1] e.g. FlexPendant from ABB Group
[2] e.g. KR CX Teach Pendants from KUKA Robot Group
[3] In most commercial robots the speed during teach mode is automatically reduced

P. 348). Although teach-in represents one of the most simple programming techniques, it requires considerable knowledge of robot kinematics and coordinate systems in order to fully understand the programming operation.

2.2.1.2 Lead-through

Lead-Through or playback methods are characterized by enabling the programmer to hold the robot either from its TCP or from similar kinematic structure and move it along the required paths (Schraft & Meyer 2006)(Deisenroth & Krishnan 1999, P. 347). During movement, the robot controller saves data sets containing either the actual position and orientation of the TCP or the readings of the joint angles at equal time intervals (Hesse & Malisa 2010, P. 74). By replaying these data sets the robot duplicates the motion executed by the programmer. The most basic form of this method started in the 80's where back-drivable motors with mechanisms to counterbalance the weights were utilized (Haegele et al. 2008, P. 977). This soon evolved to applying compliance control in order to counter the size restrictions. By measuring the forces using a force torque sensor (FTS) at the TCP and utilizing a simple compliance control scheme, the robot could be rendered movable from the its tool by sheer human effort (Winkler 2006, P. 25-67). A similar scheme utilizes torque sensors on the joint motors and hence making each joint independently movable by applying force to it (Grunwald et al. 2003). Modern light weight robots take advantage of their light structures and offer this teaching method either by back-drivable motors or joint torque measurement[4]. The apparent advantage of this method is its intuitiveness and the ability to execute complicated motions within a task by mimicking the human's motions. However, the recorded points can not easily be edited later making changing the whole trajectory module inevitable each time a small deviation occurs.

2.2.1.3 Modern approaches

On-line teaching has garnered a lot of attention from researchers world-wide due to its simplicity and intuitiveness. Many in the research community have geared their work to overcome the drawbacks of the aforementioned methods. Whether it be by sensor-assisted programming or by re-inventing the discipline all together through incorporating new devices or techniques. A fresh take on the classical lead-through programming method was presented in (Maeda et al. 2008). Where the operator moves the robot through a volume hence defining a collision free space for the given task. Consequently this information is used to automatically generate a TCP based trajectory conforming to the swept volume. Reinhart et al. (2008c) for example, attempted to augment the trajectory taught using an input device with a force sensor. This was located at the TCP to overcome

[4] An example thereof is the KUKA LWR (http://www.kuka-robotics.com/en/products/addons/lwr/) and Universal Robot's UR6 robot (http://www.universal-robots.com/)

positional discrepancies during contact with stiff environments. A similar approach was investigated in (Neto et al. 2009), where force control assisted tasks were mandated by verbal commands and translated to robot code with speech recognition. Furthermore, the Nintendo wireless remote device (WiiMot) was utilized as an intuitive and wireless teach pendant, where human gestures were extracted and interpreted in robot motion instructions using artificial neural networks. Other popular research directions were based on utilizing camera based tracking for tracking the operator's hand movements holding a stylus or guiding device. This can be extended to include assistance functions in a model based on-line simulation environment (Hein et al. 2008). Additionally, the user can even directly interact with spatial trajectories through intuitive metaphors projected on a work-piece (Reinhart et al. 2007).

2.2.2 Off-line programming

Off-line programming denotes the act of programming the robot using a PC-based simulation environment or other means without actually needing an actual industrial robot (DIN EN ISO 8373 1996, section 5.2.4). Therefore allowing the robot's program to be developed independently from the production cell and subsequently deployed on it (Inaba & Sakakibara 2009, P. 357). The need for off-line programming stems from the deficiencies inherent in on-line programming as discussed in the latter section. The major disadvantage being unable to create large, complex programs. By separating the programming process from the production cell, robot downtime is avoided hence increasing productivity of the manufacturing line. Furthermore, using standard interfaces with commercial robots make it possible to create programs applicable to robots from different vendors (Yong & Bonney 1999, P. 354). Thus making it feasible to change the hardware without having to re-program the whole task. Standardization efforts have culminated in the RSS (Realistic Robot Simulation); a de facto standard for defining robot models (Bernhard et al. 2002). Despite its numerous advantages compared to other programming methods, off-line suffers from an inevitable and fatal issue. Differences and discrepancies between the actual real world and the idealized virtual world render the code generated in an off-line manner useless without further on-line adaptation or sensor-based calibration (Weck & Brecher 2006, P. 380) as illustrated in Figure 2.3. These differences and deviations could be attributed to (Yong & Bonney 1999, P. 357-358):

1. The exact spatial locations of objects in a robotic work-cell are never precisely conveyed to the virtual world.

2. Tight tolerances in the robot components (linkages, motor etc.) cause errors, which are eventually compounded to produce large errors at the TCP.

3. Difference between the numerical accuracy of the simulation environment and the controller's resolution.

4. Environmental factors, such as temperature can affect the robot's operation in the most unpredictable ways.

To overcome the latter factors, researchers have been investigating methods and techniques to simplify calibration (Sunnanbo 2003, p. 17). Moreover, some research works advocated using real world data to update off-line simulation (Denkena et al. 2004)(Brecher et al. 2010). Robot manufacturers have also been concentrating on the accuracy of off-line programming which has been dramatically enhanced (Hesse & Malisa 2010, p. 199). In spite of all the latter efforts, off-line programming and its subsequent factory floor adaptation still represents a major monetary and technical hindrance against the proliferation of robots in many production facilities, specifically in SME (Schraft & Meyer 2006)(Pires 2008a).

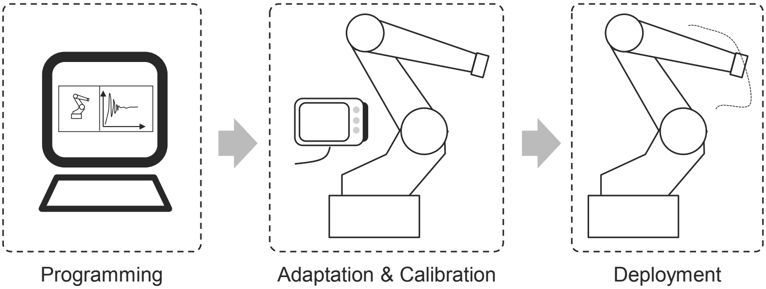


Figure 2.3: Off-line programming process

2.2.2.1 Text-based

The most basic form of off-line programming is the text based method. Each robot manufacturer offers a native language specific only to its robot. Notable examples thereof include KUKA KRL, ABB Rapid, Adept V+. Although the syntax of each language differs according to respective implementation details and design preferences, basic requirements have to be met. For instance, motion instructions have to include both linear motions and also joint motions. Additionally, loops (*FOR*-loop, *WHILE*-loop) and branching statements (*IF-THEN-ELSE*) have to be included (Weck & Brecher 2006, p. 398). Analogous to writing code for a PC-task, an editor is needed to create and edit programs, which may be included with the robot software or independently acquired. Robot languages are increasingly becoming more complicated to cater for the needs of modern robotic tasks. They have also been burrowing structural features from PC languages like the ability to organize functions in modules and executing several tasks in parallel (*multi-tasking*). Given that the robot is usually one component of several devices in a work-cell, most languages include external communication capabilities. Whether bus-based e.g. CAN-bus, PROFIBUS and Ethernet or signal-based e.g. digital and analogue inputs/outputs.

2.2.2.2 Simulation-based

The wide spread nature of graphic simulation environments make them synonymous with off-line programming. This is made possible by the multitude of robot simulation software available nowadays from robot manufacturers to independent software houses. Figure 2.4 shows several commercial off-line simulation environments which are widely employed in production facilities (Siemens AG 2011)(Dassault Systems 2011)(KUKA GmbH 2011c) (ABB-Group 2011). These software tools sport several features in common (Yong & Bonney 1999, P. 355)(Haegele et al. 2008, P. 977), such as:

3D Modeller Although not necessarily a fully featured CAD package, the modeller contains the basic shapes (sphere, box, planes etc.) and related 3D editing functions (union, merge etc.). This however is meant to complement the ability to import 3D models from dedicated CAD-packages. Supported 3D data usually include the most widely used standard formats in computer aided design (CAD) tools such as IGES (International Graphics Exchange Standard) and STEP (Standard for the Exchange of Product Model Data).

Tree View Mechanical components (fixtures, robots etc.) and virtual components (coordinate systems, trajectories etc.) in a project are listed in a heirarchial manner. Using this visualization technique, the structure of the system is not only intuitive to understand but also to find and edit the relationships between components which is easier to accomplish.

Kinematic Simulation
The main aim of an off-line simulation is the ability to simulate the movement and behavior of the robot in the confines and safety of a virtual world. Most environments contain kinematic engines capable of a near-realistic simulation. Some even offer dynamic simulation which is much more complicated and offers no significant advantage w.r.t. motion programming. Extra capabilities may also include collision warning and detection between object groups in the virtual world.

Robot Jogging
Jogging a robot is the ability to move a robot either by text input or through direct graphical interaction with the virtual model. Jogging modes can be divided into *joint* mode, where each motor is independently operated or *cartesian* mode, where the robot's TCP is moved w.r.t. a given coordinate system.

Program Editor
The program editor facilitates a convenient method to create and edit the robot's program in its native language (see latter section). Assist functions, such as copy, paste, search and replace are usually the norm

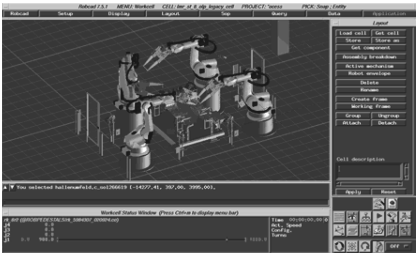
for such editors. Additionally, the programmer can execute the program and visualize the robot's movement in a virtual mode allowing him to debug it before downloading it on to the real controller.

Automatic Code Generation

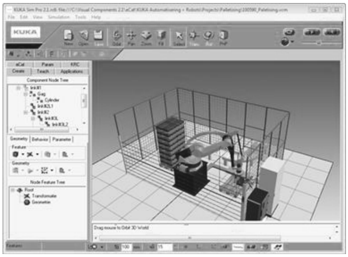
The user could automatically generate robot code after manually jogging the robot through the task space and saving key positions. Moreover some environments extend this capability to synchronize the code directly with the real controller.

Assistance Functions

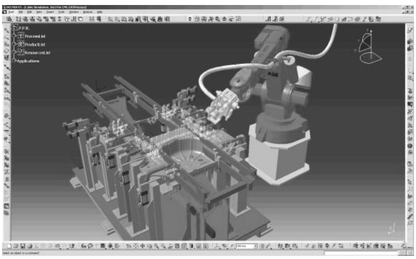
Many environments offer functionalities specific only to their respective platforms to simplify the programming process. Two examples of task-oriented-functionalities are path optimization w.r.t. cycle time and palletizing functions. Moreover some offer development-oriented functionalities in the form of an application programming interface (API) or a remote control interface.



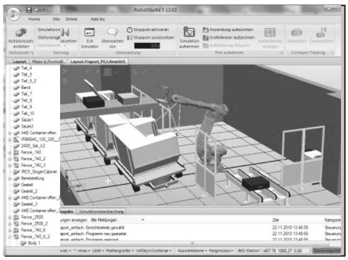
[a] Tecnomatix Robcad



[b] KUKA.Sim



[c] DS Delmia



[d] ABB RobotStudio

Figure 2.4: Commonly used commercial off-line simulation software

2.2.2.3 Visual programming

On the contrary to text-based programming, visual programming tries to simplify the interaction with the programming environment by allowing the user to visually model the operation sequence (Weck & Brecher 2006, P. 403). Most of them use cues from flow-chart, graph or diagram representations (Weck & Dammertz 1995)(Biggs & MacDonald 2003). Hence the required program behavior is described in an easy-to-decipher manner with graphical icons and connections between them. Furthermore, understanding the relationships between the different program parts and subsequent debugging is easier to achieve. Although the user experience and usability of the software is greatly enhanced, the versatility and capability of the software is correspondingly jeopardized making it less flexible than its text-based counterparts (MacDonald et al. 2003)(Biggs & Macdonald 2003). A successful example is the `Lego Mindstorm NXT` (Lego Group 2011) graphical programming environment introduced in 2006, namely `NXT-G`. its intuitiveness and simplicity allows children to program sensor-based robots with remarkable ease. Another example from Microsoft is the `Robotics Developer Studio` (Microsoft Corporation 2011) which is not limited to industrial robots and contains a multitude of functions and simulation capabilities. Within the European research initiative AMIRA Consortium (Advanced Man Machine Interfaces for Robot System Applications) a general style guide for developing graphical user interfaces for industrial robots was developed (Schreck 1998). The follow-up project MORPHA focused on the programming user interface area, uses touch and speech to manipulate an icon-based programming language (Bischoff et al. 2002). Given the need to standardize such interaction techniques, the findings of these projects culminated in an ISO standard for designing graphical user interfaces for robots (ISO 15187 2002).

2.2.3 Task-level methods

A departure from the classical programming techniques which depend on the location of programming(on-line vs. off.line), task-level or task-oriented programming methods emphasize the description of the goals to be achieved (MacDonald et al. 2003). They advocate a higher level of instruction abstraction, thus making the task of programming accessible to anyone with task knowledge rather then programming knowledge. This however entails three essential steps. The first is creating or obtaining a highly detailed model of the robot and its surrounding environment including all geometrical and physical properties. The second is the encapsulation of low level motion instructions into high level task descriptions. And finally, the system should be capable of translating the task description into robot motion (Al-Qasimi et al. 1995). Moreover, autonomous or intelligent functions to interpret the task specifications according to the sensor-based perception could be added to increase the work-cell's adaptability and flexibility (Inaba & Sakakibara 2009, P. 352). Task-oriented programming systems could be considered as to cover the spectrum between what Hägele (Haegele et al. 2008, P. 978) terms a product-centric and a process-centric programming system. Closer inspection reveals that this class of methods and their manifestation in programming systems are but a step in the direction of full automation (Yong & Bonney 1999, p. 360). Although different implementations of task-level programming are abundant in literature they are yet to find wide spread application in practice. This could be attributed to the modeling inaccuracy and the limitation of defining manufacturing process in abstract forms without sacrificing the solution's generality (Haegele et al. 2008, p. 979). Such systems are usually tailored for one application thereby addressing specific domains. For example, in the welding domain, a programming system for remote-laser welding was developed and tested on real parts (Reinhart et al. 2008a). While in the assembly domain (Castuera et al. 2004) clues for assembly were specified to an artificial neural network controller which implements different assembly strategies according to both task specification and forces on the TCP. Researchers have also investigated hybrid systems which are based on the classical on-line and off-line programming paradigms coupled with task-level strategies (Chen 2005).

2.3 Interaction control

2.3.1 Types of tasks

This section addresses the interaction between a robot and its immediate environment. From a task point of view, interaction is defined in terms of the constraints acting on the robot (Vukobratović et al. 2008, p. 1-3), and hence could be readily divided into tasks imposing no constraints and those imposing specific constraints on the robot.

Unconstrained tasks
Unconstrained tasks are defined as those that require no physical interaction between the robot and the environment (Sciavicco & Siciliano 2005, p. 5). It is therefore the objective of the robot to position itself as accurately as possible in a predefined pose w.r.t to the environment, or move along a trajectory with a predefined speed. Depending on the number of DOF of the manipulator, controlling the position also involves controlling the orientation of the TCP. Motion control was one of the earliest problems extensively studied in robotics (Craig 2005, p. 299). Consequently, numerous successful implementations of position control schemes were deployed in commercial robots leading to their near ubiquitous presence in certain manufacturing processes nowadays. Typical applications for this class are for example spray painting, pick and place, arc welding and remote laser welding. Despite their proliferation, these tasks represent only a small fraction of the whole range of manufacturing tasks that could be automated by robots (Vukobratović et al. 2008, p. 2).

Constrained tasks
Constrained tasks on the other hand involve direct interaction with the robot's environment. Interaction in this regard means creating (at the moment of contact) and maintaining (during contact) a mechanical coupling with static or dynamic objects in the robot's immediate environment (Sciavicco & Siciliano 2005, p. 271). Such tasks can be broadly

divided into two classes. The first class represents tasks requiring precise control of the forces and moments arising at the TCP, usually necessary in machining tasks e.g. cutting and deburring (Pires et al. 2002)(Stelzer et al. 2008). In this case forces directly affect parameters vital to the quality of the process for instance metal removal rate and plastic deformation (Zhang 2005). While the second class represents tasks which require precise positioning; but due to deviations between the programmed path and the actual environment they can not be classified as unconstrained tasks. On that account, contact here develops as a byproduct of positional deviations commonly experienced during the assembly of mating parts in manufacturing processes (Newman et al. 1999)(Park et al. 2008).

2.3.2 Interaction control classification

Despite being a very active topic in robotic research for the last three decades, no standard classification scheme for interaction control actually exists. Furthermore, terminology and naming notations of similar control schemes vary from one researcher to another. Surdilovic & Vukobratović (2002, P. 23.3) published a comprehensive classification, which starts by dividing interaction control into passive and active control schemes as shown in Figure 2.5.

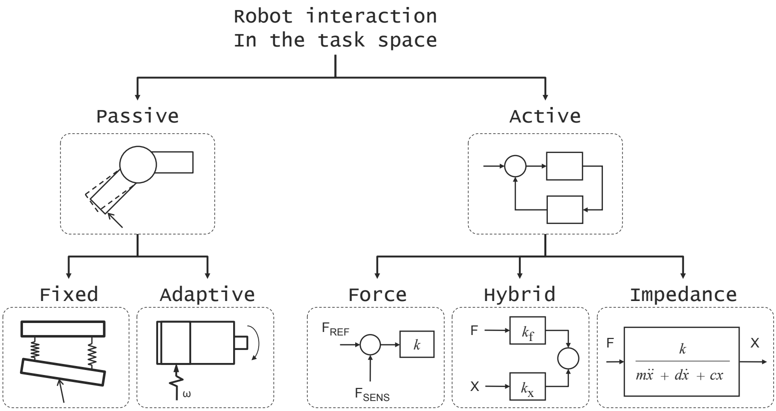


Figure 2.5: Classification of interaction control schemes in the robot task space according to Surdilovic & Vukobratović *(2002)*

2.3.2.1 Passive control

Passive control schemes allow the robot to react to forces due to inherent compliance whether structural or controller based (Khalil & Dombre 2002, P. 378-380). Duly the stiffness of the robot is intentionally lowered by a multitude of methods to achieve this. One of the more popular methods widely deployed in industrial practice is the remote center compliance (RCC) device (Whitney 2004, p. 260). Another method is building sufficient compliance in the robot's mechanical structure i.e. increasing the joint and arm elasticity (Pfeiffer 1992)(Ang Jr & Andeen 1995) to accommodate forces on the TCP. The latter are termed 'fixed methods' due to their constant behavior, since they depend on components that are not interchangeable or adaptable to the needs of the task. If however, RCC devices were adjustable (Chun 1992) or structural compliance was tunable through motor servo-gains they, would then fall under the adaptive category.

2.3.2.2 Active control

Active control differs from passive control in that it requires the measurement and feedback of forces on the robot's TCP to achieve a specific robot behavior during interaction. In the force control scheme, the forces and position are simultaneously controlled to track the nominal positions and forces (Eppinger & Seering 1987)(Bigras et al. 2007a). While in the hybrid control scheme both forces and positions are controlled in orthogonal subspaces. Therefore the directions in which the robot interacts with the environment (constrained) are force controlled while the directions in which the robot is allowed to move freely (unconstrained) are position controlled (Raibert & Craig 1981)(Craig 2005, p. 373-378). On the other hand impedance control advocates controlling the relationship between the robot and the environment thereby acquiring a target impedance (Hogan 1985)(Bigras et al. 2007b). Implementing latter control schemes is usually executed in either the joint space or in the task space. This depends on two major factors; the first is the availability of a near-realistic dynamic model of the robot and the second is the interfaces available to control the robot i.e. whether access to motor control is permitted (Zhang et al. 2004). The latter factor also plays a role in re-structuring the control schemes around the position controller of the robot where the controller's output is defined in terms of a correctional position signal (Pelletier & Doyon 1994)(Lange & Hirzinger 2005)(Winkler & Suchy 2007b).

2.3.3 Remote center compliance

Given the wide spread application of these types of devices in manufacturing processes (involving interaction), they will later be separately discussed. RCC devices were first developed in The Charles Stark Draper Laboratory (Drake & Simunovic 1979). Initial investigations resulted in mathematical models to facilitate the fitting of parts with non-compliant mechanisms i.e. robots (Holzbock 1996, p. 91). Based on these models, the first devices for mechanical compliance were developed and patented (Drake & Simunovic 1979)(De Fazio & Whitney 1984)(Whitney 1985). Although their development continues to present day, the principles upon which the devices are built are unaltered (Holzbock 1996, p. 92). As seen in Figure 2.6 an RCC consists of two parts; a translational part allowing movement in the lateral direction and a rotational part allowing movement in the angular direction. The 'remote center' represents the optimum location at which the insertion process may take place. In order to avoid jamming or wedging, specific conditions regarding clearances and friction coefficients have to be met. Additionally, at least one of the parts in the assembly process has to possess chamfered edges. The mechanical embodiment of this principle consists of two plates connected by several columns of elastic pads interspersed with metal plates (Hoshizaki & Bopp 1990, P. 238). This geometrical formation offers much higher stiffness in compression than in shear, subsequently causing the lower plate to move in a lateral and/or torsional direction w.r.t. the upper plate (Whitney 2004, P. 260). By changing the type of elastic material and the geometrical stacking shape, the compliance of the device could be altered to correspond to specific application requirements. Furthermore, commercial devices are built in a manner allowing them to be easily mounted between the manipulator and the gripper using standard adapters. In practice, RCC devices are selected and deployed according to the following criteria (Hoshizaki & Bopp 1990, P. 238-240):

1. Nominal values of misalignment correction could be achieved at the *remote center point*. Any change in the distance between the mounting face of the RCC and the remote center point adversely affects the insertion process.

2. The repeatability of the RCC must be considered in the context of the whole robotic system. Hence it is imperative to select the RCC conforming to the tolerances expected from a system.

3. Given the inherent compliance of an RCC, oscillation of the work-piece during movement of the robot is inevitable. Hence the natural frequency of the device is selected so as to reduce the damping time of the oscillation based on the given weight class.

Although the vast majority of RCC devices exhibit constant behavior, tunable devices have also been developed (Chun 1992). Additionally, researchers investigated the classical RCC principle to achieve more accurate mathematical models (Ciblak & Lipkin 1996)(Ciblak & Lipkin 2003).

2.3.4 Impedance control

On the contrary to direct force control, impedance represents a general framework for controlling the dynamic interaction of robots with environments (Vukobratović et al. 2003, P. 4) which was introduced by Hogan in his seminal work in 1985 (Hogan 1985)(Hogan 1989). The naming convention owes itself to the electrical engineering discipline from which the original idea stemmed. Instead of defining and controlling a certain force at the interaction's interface i.e. TCP, it defines the *shape* of the dynamic interaction. Hence, the controlled variable is neither the positions nor the forces but rather the relationship of interaction (Nagchaudhuri & Garg 2001). Such control techniques are beneficial when the magnitude and direction of the forces are not vital to accomplishing the task at hand.

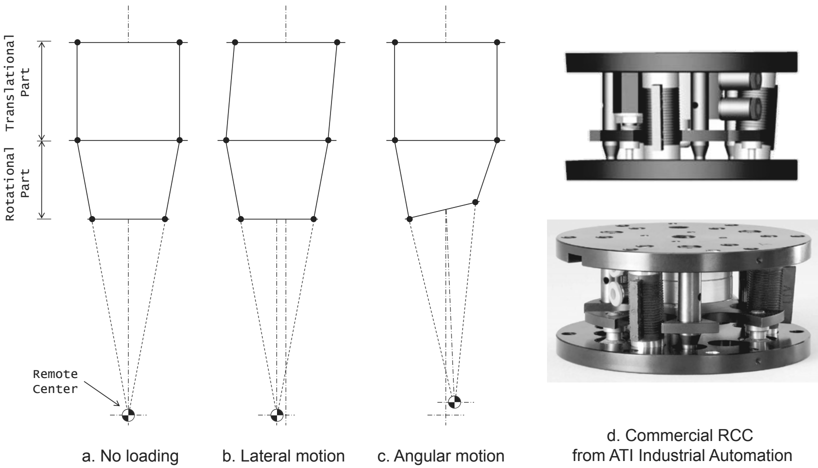


Figure 2.6: Design principle of an RCC device

In other words, it is adequate to position tasks with forces as a byproduct e.g. assembly rather than tasks that require specific forces e.g. machining (Bogue 2009). This does not imply the fact that forces arising at the impedance interface are uncontrollable. By augmenting control loops with force monitoring, forces can be conveniently kept within specific limits. Additionally impedance control can be implemented without the explicit knowledge of the robot's model, which is particularly beneficial in industrial systems since they only offer a direct interface to their position control and not to the motor current control (Pelletier & Doyon 1994)(Tan et al. 1997)(Bigras et al. 2007a)(Bigras et al. 2007b). Ongoing research in impedance since the mid-eighties resulted in different implementations which in turn reflected a large variety of situations suitable for this control strategy. Several researchers experimented with higher order impedance by introducing force derivatives in the relation between positions and forces (Lee & Lee 1991). Others applied adaptive control techniques such as varying the target parameters during runtime to accommodate changes in the system state (Matko et al. 1999) or on-line adaptation of impedance parameters using artificial neural networks (Tanaka & Tsuji 2004) or other learning mechanisms (Qiao & Zhu 1999).

2.3.5 Applications

Historically, commercial robot manufacturers focused on application domains which represent unconstrained tasks for instance material handling and simple pick and place tasks.

So to say, positional capability[5] instead of the interaction capability of the robot was the focus of development. Albeit industrial deployment staggered, research in unconstrained tasks didn't lag behind (Caccavale et al. 2005)(International Federation of Robotics - IFR Statistical Department 2009, P. 329). The two application domains where research in constrained task flourished were robotic assembly and robotic machining.

2.3.5.1 Industrial assembly

Since industrial assembly is considered one of the most labor intensive and respectively the most cost intensive process, it was at the forefront of research in robot interaction control (Rampersad 1995). But due to the delicacy and complicated nature of assembly processes, the promise of robotic automation in practice -at least not to its fullest potential- never materialized. Nevertheless several research efforts culminated in control frameworks, guidelines and approaches to the problem. One of the major contributions to this field was W.S. Newman and his team. They developed controllers suitable for a wide range of assembly tasks (Glosser & Newman 1994) and showed that by combining it with autonomous search and parameter optimization (Newman & Wei 2002), they could be utilized for intelligent assembly (Newman et al. 2001) in practice at the FORD assembly plant (Newman et al. 1999)(Gravel & Newman 2001). In several other initiatives, interaction control was coupled with image processing to facilitate a combined visual and force servoing (Morrow et al. 1995)(von Collani et al. 1999). This combination entrusts the image processing module with the task of finding the target location and hence planning the gross transport motion Schmitt et al. (2008). Once the assembly phase starts and given the tight-tolerance nature of most assembly tasks, the force control module is applied to ensure the successful completion of the task (Jorg et al. 2000)(Chen et al. 2007). Furthermore, given the 3D geometric models of the parts, off-line optimization techniques were investigated to generate optimal force control parameters(Arai et al. 2006).

2.3.5.2 Machining processes

Although machining and finishing processes were traditionally dominated by computer numerical controlled machine tools (CNC) and custom built machines, robots are finding increased deployment in this field. Their key advantage compared to traditional CNC lies in their flexibility regarding DOF and the comparatively larger workspace (Abele et al. 2008a)(Han & Sun 2008). Hence, making them ideal for machining complicated and/or large work-pieces (Abele et al. 2005). Additionally, integrated material handling and easier programming compared to CNC machine tools results in quicker changeover times for small batch sizes (Bogue 2009). In recent years, solutions appeared on the market

[5] This includes both absolute accuracy and repeatability.

regarding operations ranging from heavy-duty tasks such as cutting and deburring to lighter tasks such as polishing (Pires et al. 2002)(Bilbao et al. 2005)(Abele et al. 2008b)(Han & Sun 2008). However, not all machining operations are suited for robots given the relatively low structural stiffness of the robots; resulting in unacceptable process quality adversely affecting the process productivity (Zhang 2005). By making the technology more affordable for customers, ongoing research in robotic machining has been trying to transfer the wealth of knowledge in this type of interaction control to practice (Elbestawi et al. 1992)(Robertsson et al. 2006). Furthermore, advances in robotic machining made it possible to apply robots in different domains, for instance robot-assisted bone milling during surgery (Meister et al. 1998) or for sculpting for in artistic front (Owen et al. 2008).

2.4 Cooperative manipulators: scientific research

Researchers took huge interest in cooperation to achieve the dexterity and load sharing capability of human arms. The introduction of industrial robots on a wide scale made it more attractive for researchers to implement their theories and hence encouraged more work in this field. Issues pertaining to cooperative manipulation inherently overlap with interaction control, nevertheless it is beneficial to consider them in a light of their own. According to Zivanovic & Vukobratović (2006, P. 19-26) the issues concerning cooperative manipulators are classified into: kinematic uncertainty, force uncertainty and control problems. Although the latter issues appear non-related they are in reality mutually exclusive. The possibility of encountering one or more thereof depends to a great extent on several factors: structure and DOF of manipulators, force measurement techniques and precision, control bandwidth and available hardware/software interfaces etc. In the first two issues, uncertainty arises due to redundant manipulators and assumptions of rigid contact points respectively rendering the models indeterministic (Yun et al. 1989)(Kumar & Gardner 1990). Accordingly, the problem is more of a mathematical nature, and hence introducing certain constraints facilitates in attaining a unique solution. Additionally, the extent of the solution depends on the kinematic and dynamic models available and whether the forces could be measured at the contact points or not. Thus the problem of control remains the focus of most researchers in this field justifying the huge body of work. Kosuge & Hirata (2005, P. 20.5) formulated the problems of cooperative manipulator control into several questions

- How to control the spatial pose (position/orientation) and its time derivatives (velocity/acceleration) of the manipulated object? (Tzafestas et al. 1998)(Sun & Mills 2002b)
- How to control the behavior of the manipulated object relative to an external force acting on that object? (Khatib et al. 1996b)(Kosuge et al. 1997)
- How to control the internal forces arising at the contact points between the object and the manipulators? (Adli et al. 1991)(Osumi & Arai 1994)

- How to share the load of the object among the manipulators in an equal or a predefined manner? (Kim & Zheng 1991)(Zhang et al. 2004)(Al-Jarrah & Zheng 1997)

In the next subsections the answers to the latter questions will be implicitly discussed within the scope of the most common methods of controlling cooperative manipulators investigated in the research community during the last decades.

2.4.1 Master/slave control

One of the earliest approaches to coordination and control of cooperative manipulators is the master/slave which was sometimes referred to as the leader/follower technique. Whereby one robot acts as the *master* and the other follows it and acts as the *slave*. However, this technique suffers from the fact that the slave robot considerably lags the master robot (Uchiyama & Nakamura 1988). The first implementations utilized an FTS at the TCP of the slave robot to measure the forces (Ishida 1977). While the master moved along a trajectory the slave was controlled by the forces on its TCP. This was extended upon to solve the problem of load distribution between cooperating manipulators in (Kim & Zheng 1991). Whereby the slave robot is triggered to start carrying its share of the load after the master robot reaches its limit. Other researchers implemented the idea w.r.t. coordinated motion according to the constraints between the robots and the manipulated object (Luh & Zheng 1987). Alford & Belyeu (1984) investigated the real-time aspect of master/slave coordination by designing and testing a hierarchical computer structure to plan, generate and control the coordinated movement of two robots. This however was specifically designed for industrial robots which are mainly position controlled. Some researchers tried to reduce the lag between the slave and master robot. This was achieved by defining the kinematic coordination problem in terms of kinematic redundancy and flexible grasping which consequently increased the maneuvering capabilities of the slave robot (Suh & Shin 1988).

2.4.2 Hybrid position/force control

Although rarely under research nowadays, the hybrid position/force represented one of the early successes of cooperative manipulation. The scheme was based on the work done by Raibert & Craig (1981) (refer to section 2.3.2.2). One of the first implementations for cooperative control was done in (Hayati 1986). The controller also included features facilitating tension/compression in the position subspace while minimizing the tracking error w.r.t. the desired forces in the force subspace. Hayati (1988) also utilized the operational space formulation (Khatib 1987) to design a decoupled hybrid position/force controller, which was validated with simulation. Further works by Uchiyama et al. investigated the decoupled spaces in a central controller, wherefore making the information from each manipulator available for all other manipulators (Uchiyama & Dauchez 1988). This was successfully implemented on an experimental test-rig with two 4-DOF cartesian robots

(Uchiyama & Nakamura 1988). Yoshikawa (1991) extended this work to include not only the dynamics of the manipulators but also of the object. He used a method, developed in earlier work, to derive the dynamic equations of cooperative manipulators environment constraints and decouple them w.r.t. the object motion, constraint forces and internal forces (Yoshikawa et al. 1988). This method was validated with two 2-DOF cartesian manipulators. Koivo & Unseren (1991) attempted to reduce the order of the derived models in order to simplify the solution of both forward and inverse dynamics. Consequently simplifying the decoupling of the position and force subspaces for the hybrid position/force control scheme. Additionally, comparisons to the master/slave method proved the superior performance of the hybrid position/force (Duelen et al. 1991). This was mainly attributed to the fact that pulling and pushing forces could be successfully controlled without disturbances from the object movement (Uchiyama & Nakamura 1988). Recently the approach was revived by combining it with a robust controller and designing a computationally inexpensive algorithm for practical implementation (Gueaieb et al. 2007a). Generally speaking, the hybrid position/force control scheme suffers a great disadvantage. It requires monitoring of the task at hand and continuous switching between different position and force modes. Additionally, the set of environment constraints is usually difficult to determine given the irregularities in the real world, making it less attractive for practical implementations (Schneider & Cannon 1992).

2.4.3 Adaptive control

According to Sastry & Bodson (1994, P. 2), adaptive control is a technique used to identify system parameters, and accordingly, use this knowledge to design and influence the plant controller in order to compensate for uncertainty during design. The challenge here lies in the ability to prove the asymptotic convergence in tracking the control variables, which in this case range from the desired position/velocity to the internal forces at the contact points. One of the common approaches to the problem was assuming that a perfect model of the manipulator kinematics and dynamics existed (Walker et al. 1989)(Zribi & Ahmad 1992). Other researchers proposed a more realistic approach and investigated adaptive coordination schemes in which the parameters of the dynamic model is estimated, and the adaptive control compensates for the discrepancies. The advantage here lies in the fact that explicit force control is not required given the implementation of the control on the joint level (Kawasaki et al. 2006)(Sun & Mills 2002b). Not only was adaptive control separately investigated, but researchers combined other control techniques e.g. hybrid position/force and impedance, to achieve superior performance. Seraji (1988) filed a patent outlining the design of adaptive control strategies for dual arm manipulators in the cartesian space i.e. without requiring complex dynamical models of the manipulators. While the basic control law was based on the hybrid position/force algorithm, the adapting parameters compensated for the cross-coupling effects between the manipulators. Adaptive impedance was investigated by Huang et al. (2004), where the controller continuously tuned the desired impedance and consequently the tracking error. Adaptive control was also used for dual-arm assembly (Zhu 2005). Where for example a relative jacobian which represents the inverse kinematics of both manipulators was used to execute a peg in a hole task (Choi et al. 1999).

2.4.4 Impedance control

As already discussed in section 2.3.4 impedance control aims at controlling the behavior or the interaction relationship between the manipulator and the environment at the contact point (Hogan 1985). After thoroughly examining hybrid position/force for cooperative manipulation during the 80's and early 90's, researchers started taking note of impedance and its potential. Schneider & Cannon (1992) analyzed the cooperative manipulation problem from a system's perspective. Consequently a control policy that enforced object impedance was developed. Experiments with two 2-DOF robots exhibited how the controller fully compensated for the system dynamics and allowed for a powerful specification of object behavior. The work was extended upon by Pfeffer & Cannon (1993) to include flexible drivetrains where a hierarchical control scheme (joint, arm, object, task) was developed which enforced object impedance on the object level built upon the arm level. Similarly, Surdilovic et al. (2001) proposed an impedance control framework for assembly tasks defined at higher control levels. This was augmented with an approach for designing robust impedance controllers ensuring both coupled and transition contact stability. Kosuge et al. (1994) proposed a unified coordinated motion control algorithm based on arm impedance, which was defined relative to the manipulated object and was designed for both manipulation of a common object and part mating. Consequently, it was deployed in a decentralized scheme on two 6-DOF industrial manipulators (Kosuge et al. 1997). The algorithm outlined was able to specify the apparent impedance in a decoupled fashion after satisfying certain geometrical constraints. This actually amounted to a centralized scheme, since the constraints represent the object's equations of motion. Recently Caccavale et al. (2008) proposed a two level impedance scheme to overcome both internal forces acting on the robots' TCP and the external forces acting on the manipulated objects. They also proved the controller's performance on an industrial test-rig. Furthermore, non-model based impedance with reduced computational overhead for limited hardware resources was investigated (Moosavian & Ashtiani 2008). Researchers also combined different approaches together. For instance an adaptive impedance controller to compensate for uncertainty in object dynamics was investigated by Nagchaudhuri & Garg (2001). Bekalarek et al. (2001) proposed a decentralized hybrid position/impedance control and tested it on industrial robots. Where the robots were motion controlled until excessive forces above a given threshold triggered the impedance controller.

2.4.5 Intelligent control

The term intelligent control encompasses a wide variety of techniques and mathematical tools aiming at imparting intelligent behavior to a controller. It is commonly used to denote methods such as artificial neural networks, fuzzy logic and genetic algorithms

(Russell & Norvig 2009, P. 727,550,126). These techniques are either used alone or combined with other basic interaction controllers e.g. impedance or hybrid position/force. Early works regarding the utilization of intelligent controllers for cooperative manipulation included using a learning algorithm to control the cartesian movement of cooperating robots handling an unknown object (Kuc et al. 1995). The algorithm compensates for uncertainty in robot parameters, viscous friction and manipulated object parameters. In (Lin & Huang 1997) an fuzzy logic based controller was applied to a master/slave control scheme. Whereby the master is position controlled while the slave has a FTS and moves to keep the forces constant in a decentralized fashion. Kumar & Garg (2005) used a combination of a neuro-fuzzy controller on an experimental test-rig. The parameters of fuzzy logic engine were tuned with the help of an artificial neural network. In (Gueaieb et al. 2007b) a similar setup was designed, whereby a fuzzy logic engine was combined with a systematic on-line adaptation mechanism to achieve control objectives without an accurate dynamic model of the system.

2.4.6 Telepresence

In telepresence/teleoperation a human user -termed the operator- interacts with a remote environment through a manipulator -termed the teleoperator. The human not only remotely controls this manipulator, but also receives feedback signal e.g. visual and haptic to increase the feeling of actually being immersed in a remote environment (Hokayem & Spong 2006). Issues in multi-user or cooperative telepresence consists of problems pertaining to telepresence such as communication time delay, passivity and stability (Arcara & Melchiorri 2002)(Tanner 2005) compounded with problems encountered in classical cooperative manipulation. These sets of problems are not mutually exclusive, therefore the analysis, design and implementation of controllers for such systems entail more complexity compared to their respective parts. Sirouspour (2005b) tackled the latter issues by increasing the modeling level through incorporating the dynamics of all static and moving constituents in a scenario. Based on this model, a control framework to ensure information flow between all operators and teleoperators was introduced (Sirouspour 2005a). The framework guaranteed a robust and stable interaction and enhanced task coordination by optimizing performance objectives. Furthermore, an adaptive controller based on the later framework was designed proving its robustness and stability with respect to time delay (Sirouspour & Setoodeh 2005a)(Sirouspour & Setoodeh 2005b). An additional LQG (Linear-quadratic-Gaussian) controller for stable and robust operation under constant time delay was implemented and tested under different environmental modes (soft vs. hard) (Setoodeh et al. 2006). Another control framework was developed by Lee & Spong (2005)(Lee et al. 2005) using passive decomposition while ensuring energetic passivity. To overcome instability due to time delay, scattering based communication was utilized. The principle idea was to allow one OP to control several teleoperators and execute fixtureless manipulation of an object (i.e. without grasping) while controlling and ensuring the passivity of the communication channel and the mutual interactions between the operators, the teleoperators and the environment. However, most of the latter research was only validated under controlled conditions with 1-DOF experimental manipulators.

2.4.7 Miscellaneous control approaches

Owing to the complexity of the controlling cooperative manipulators, variable approaches attempting to tackle the issue appeared in literature. A classical approach to the problem in the early research phase was to model the system and try to linearize it and consequently design an adequate controller for it (Yun et al. 1989)(Tarn et al. 1986). Investigations utilizing this method ranged from decoupling the equations of motion for easier control (Yun 1991) to examining the effect of flexible joints (Jankowski et al. 1993). A similar method embodying the concept of the RCC (refer to section 2.3.3) was investigated. The method advocated introducing controllable flexibility at the interaction points between the object and the manipulators (Osumi & Arai 1994)(Osumi et al. 1995)(Sun & Mills 2002a). While others combined traditional methods with robust control techniques to enforce bounds on the position and force tracking error in the face of uncertain model parameters (Gao et al. 1992)(Kim et al. 1997). One of the more exotic methods was the utilization of a fractional-order position and force controller (Fonseca Ferreira & Tenreiro Machado 2007). Supported by simulations, the authors claimed the superiority of its performance compared to traditional methods.

2.5 Cooperative manipulators: commercial systems

Starting in 2004 several robot manufacturers introduced commercial multi-robot systems (Reinhart & Zaidan 2009). Their introduction could be divided into two mutually intersecting phases. The first phase was the introduction of control units capable of addressing more than one drivable mechanism at a time. Thus fulfilling the market's need for an integrated operation involving external drives and coordination between those drives and the robot controller. This clearly led to the second phase where several robots were envisioned to work together. This, however entailed not only a controller to drive the motors but also to guarantee cooperative functionalities for instance cooperative workspace definition and collision detection and avoidance. An overview of the state-of-the-art in commercial CIR will begin with an extensive review of related patents and subsequently some of the common features in commercial CIR is presented. This serves to better understand the development of the mentioned commercial systems.

2.5.1 Patent Review

On control systems
Two of the earliest patents in this field were filed by Japanese corporations in the early 90's. Otera et al. (1993) were one of the first to propose a central control system with a single program to interpret the motion instructions and transfer them to its respective robots. The program thereby guaranteed a sequential coordination of not only the motion instructions

Figure 2.7: A cooperating robot test-rig from KUKA GmbH

but also central error detection and general program modification for process adaptation. On the other hand Kanitani (1993) proposed a similar architecture but emphasized the overall system operation by introducing the robot-based master notion which discussed how the motion and velocity could be synchronized in a shared program. The more recent wave of inventions started in the early 2000's. FANUC filed a patent in 2001 (Takahashi & Kosaka 2001) proposing a control system and its accompanying operation scheme where a plurality of robots working in a cooperative mode store 3 different operation programs. The first one makes the robot a master, the second would make it a slave and the third would operate the robot independent of the other robots. According to the patent, the arrangement guaranteed operation of all the robots in either a cooperative mode (irregardless of being a master or slave) or an independent mode at all times. To extend the applicability of the latter patent for an unlimited number of robots, KUKA filed a patent in 2004 (Stoddard et al. 2004) proposing outfitting the robots' controllers with multiple motion instruction sources. Consequently, the robots were operated either in a synchronous mode (when they receive their instructions from an external source through a network) or operated in an independent fashion (when they use their internal motion instruction source). KUKA also proposed a detailed description of the workings of such a coordination system in a subsequent patent (William Kneifel II & Stoddard 2004). This was achieved by devising a mutual exclusion mechanism to prevent several instruction sources from simultaneous access to the same robot. Lapham (2005) proposed a control system based on a general purpose computer which interfaces with the real-time subsystems of a mechanism or robot. An execution module on this computer generates the move command and synchronizes it with the interpreters or move modules on the real-time subsystems. ABB filed a patent in 2009 (Fortell et al. 2009) proposing a two level path planner scheme for cooperating robots. While each robot possesses its own path planner, a global path planner generates the robots' respective paths during a cooperative task. During any one time either the separate path planner or the coordination path planner is activated using a switching technique with a mutual exclusion mechanism similar to that proposed from William Kneifel II & Stoddard (2004). Hence, during a cooperative task one or more robots could break free of the pack and execute an independent task and after its completion return to where it/they left off in the cooperative task. While the latter patent avoided specific hardware configurations, ABB's subsequent patent exclusively addressed that issue. Not limiting itself to robots but extending to arbitrary mechanical units, ABB's patent (Kettu et al. 2009) describes how a single server controller side consisting of independent control units communicates over a network with the mechanical units facilitating synchronized movement. Control programs are stored on a memory on the server and subsequently interpreted for execution instructions. Additionally, the server is connected to a shared HMI and employs software to distribute available real and virtual resources to the connected control units.

On coordination and time synchronization

To avoid operation disturbances due to variable time latency through the communication channel, FANUC proposed a coordination method based on the communication time delay (Hashimoto & Ohya 2008). Synchronizing the robot controllers with the master robot controller depends on the time interval between the generation of a basic operation signal and its echo from the respective slave controllers. This time interval is used to synchronize the basic operation signal of the respective slave controllers with the corresponding basic operation signal from the master controller. ABB touched upon the issue of coordination by overcoming latency incurred for synchronizing a master/slave arrangement since in a cooperative state the cooperative path planner is switched on, performing a parallel interpolation of the robots' movements (Fortell et al. 2009). Hence motion instructions are directly sent to each robot avoiding the time delay caused by communication of non-interpreted motion instructions. Although barely discussed, KUKA tackled the problem of synchronization from the program execution point of view (Graf et al. 2006b). It proposed special data structures containing control data which are exchanged between control units. By adding certain trigger points in the program, those data structures are accessed during cooperation which facilitate the synchronization of movement between the robots according to prescribed data stored in them.

On Calibration

Despite its importance, calibration was a topic largely absent from the manufacturers focus especially w.r.t. cooperating manipulators. However, KUKA proposed an extended calibration to overcome the deficiency of depending on an academic approach in calibrating multiple robots w.r.t. each other (Graf et al. 2006a). Taking into consideration the fact that deflections in the real-world occur during the movement of the robots, the method calls for calibrating each robot w.r.t. all other robots using an independent coordinate transformation which is saved in each robot's respective memory storage. Basically, the method calls for moving from a master-slave based calibration scheme to a distributed one.

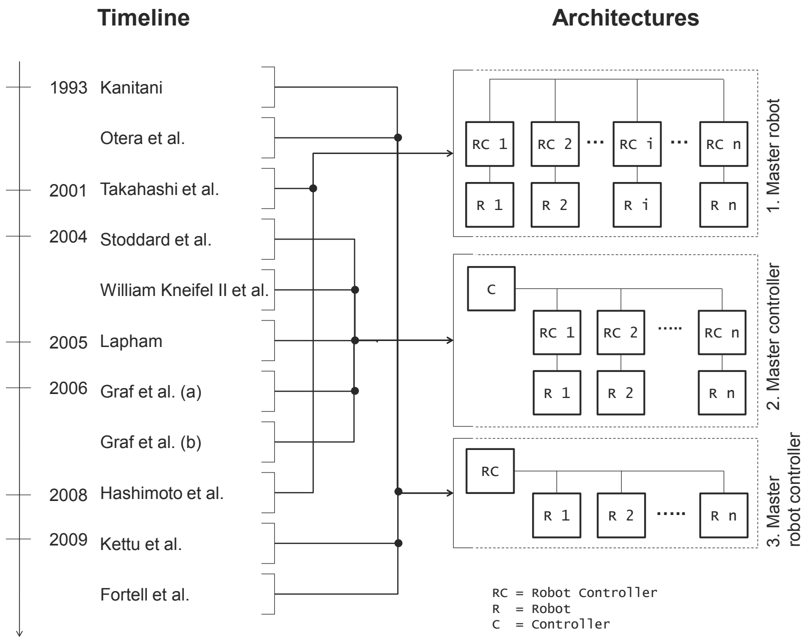


Figure 2.8: Timeline of CIR architectures according to patents filed by commercial manufacturers

2.5.2 System Features

A noteworthy observation, is that the patents reflect not a method to solve a generic problem, but a method to solve a problem specific to the architecture and abstraction level of the respective robot manufacturer. Moreover, some ideas are directly derived from issues arising during the execution of specific tasks. Closer investigation of the patents reveal stark similarities but nevertheless subtle differences in implementation. These differences allude to the variable terminology and notations underlying a specific system. Furthermore, the ideas discussed mostly pertain to different architecture schemes for operations thereby neglecting the extent of how the architecture influences the programming capabilities of those systems. All in all, most commercial CIR systems employ the master/slave principle albeit in varying forms. Three basic CIR architectures could be deduced from the patents (as illustrated in Figure 2.8).

- **Master robot**: The master is one of the robots but could exchange his master role any time with one of the other slaves.
- **Master controller**: A dedicated master controller controls the robots through their robot controllers.
- **Master robot controller**: A dedicated master robot controller directly controls the robots. In this sense, the mechanical manipulators don't need dedicated robot controllers.

Based on the patent investigation in the latter section and the available information on commercial CIR systems (as shown in Table 2.1), the predominantly common features in most systems can be summarized as follows:

Off-line programming
Due to the complexity of a cooperative operation -whether it is loose cooperation or tight cooperation- off-line programming with a simulation environment remains the method of choice for engineers. Software features such as emphasizing collision areas greatly simplify the planning of such operations (ABB-Group 2011)(KUKA GmbH 2011c)(YASKAWA Motoman Robotics 2011)(FANUC Robotics America Corporation 2011). Subsequently after the task has been defined and measures have been taken to ensure collision-free and synchronized operation, code is automatically generated for each respective robot. However, as already discussed in section 2.2.2, the discrepancies between the virtual world and the real world render the generated code useless without post-adaptation which is performed in an on-line manner directly on the factory floor. This becomes more complicated when increasing the number of robots involved in a given operation.

Data Coupling
The nature of the information exchanged between the robots and their respective controllers defines the data coupling type. The most common is a *geometrical coupling* (Fortell et al. 2009) where all robots and their controllers receive instantaneous information about the positions and velocities of the other robots in the wok-cell. In that event, deflections occurring due to heavy loads on the TCP could be conveyed to the others in order to adjust their trajectories accordingly (Stoddard et al. 2004). Another type of coupling is the *dynamic coupling*. In this type the position data -including velocities- is extended to include forces and torques on the TCP of each respective robot. Hence deflections and irregularities can be conveyed to the other robots and eventually avoided in a controlled manner. Unfortunately this type of coupling is rarely found in commercial CIR and is usually implemented for research purposes.

CIR system	Control panel	Software	Robot/Motor limit
KUKA RoboTeam	Shared Pendant	KUKA-ArcTech	15 (R+M)
ABB MultiMove	FlexPendant	RobotStudio	4/36 (R/M)
Motoman Multi-Robot	Teachbox	MotoSimEG	4/36 (R/M)
FANUC i Series	iPendant	ROBOGUIDE	16 (R+M)

Table 2.1: A review of the commercial CIR systems available on the market

Synchronization Mechanisms
To ensure the timely exchange of information between the robots, a synchronization mechanism has to be implemented. Synchronization on the implementation level mainly requires quantifying time delay (Fortell et al. 2009), generating operation master signals (Hashimoto & Ohya 2008) and designing data structures for triggering signals (Graf et al. 2006b). For instance, KUKA applied the IEEE 1588 standard (Eidson 2006) to synchronize multiple robot controllers (Gerstenberger et al. 2005). However the implementation side of the synchronization is only valid in light of the signal model behind it. The most widely used is the master/slave model, where a master robot controller or a master controller generates a signal to synchronize all other controllers. It essentially offers a hierarchal control scheme, where the slaves' movements are coupled to the master. This corresponds to the second and third architecture type in Figure 2.8.

2.6 Discussion

2.6.1 The position/force dilemma

Unique among measured signals in a robotic system, forces and positions are mutually exclusive. Regardless whether they are defined at the robot's end-effector or at the joints. Other correction signals, for instance video camera or laser sensors, are eventually translated to a position offset that exhibits no implicit effect on the source signal, making the relationship of a unilateral nature. In case of force signals, the derived position signal loops back to influence that force. This is only logical in light of Newton's second law of motion (Newton 1846, P. 83) which dictates this relationship. Consequently, under constrained conditions where interaction between manipulators and the environment occurs, it is impossible to define the shape of one signal without implying the shape of the other. This is analogous to the voltage/current interdependency in electrical engineering (Nagchaudhuri & Garg 2001). Hence, the overall objective here is to achieve a predefined tracking performance simultaneously for both positions and forces of the object being manipulated. This latter force/position dependency defines the context in which most research in robotic interaction takes place. Researchers have argued that issues in cooperative manipulation are limited to control issues, if the forces at the interaction surfaces can be determined (refer to section 2.4).

Despite that, commercial robot manufacturers introduced geometrically coupled systems capable of executing synchronized movements in real-time through exchanging positional data (position, velocity and acceleration) as detailed in section 2.5. This however addresses only the first part of the aforementioned dilemma, namely the position. In an ideal situation and after exact calibration, motion generated from an off-line simulation would be identical to that performed by the manipulators. This is unfortunately not the case. Far from perfect, actual environmental conditions deviate from those in the virtual environment, which is eventually translated into positional discrepancies between them (refer to section 2.2.2). Additionally, dynamic effects due to gravitational or inertial forces which ultimately cause deviations from the generated trajectories are rarely taken into consideration during programming. Since industrial robots were mainly utilized for purely positioning tasks, the mechanical structure was kept as stiff as possible to (re)produce maximum positioning accuracy (Bilbao et al. 2005, P. 7) thus implicitly reducing their adequacy for constrained tasks. Given the highly stiff nature of environment, those deviations are readily translated into large forces on both robots and environment. Both types of interaction; (a) object-manipulators and (b) object-environment, occurring simultaneously in a tightly coupled cooperative operation develop into a complex set of problems which ultimately results in a case where disregarding the forces is practically impossible.

2.6.2 Robot architectures

Robot architectures arise from the interplay between hardware, software and communication components in a robotic work-cell. The specific order in which the components are arranged relative to each other and the information exchange between them dictate the functionality of the system. Hence, architectures define the boundaries and possibilities expected from a robotic system based on the underlying design of the respective components and their relation to each other w.r.t. signal flow. Although a body of work exists on robot architectures, no unifying or coherent definition of a robot architecture is available (MacDonald et al. 2003). Since their conception and worldwide proliferation in production facilities, industrial robots have retained a relatively rigid architecture. This can be mainly attributed to the repetitive and constant nature of the task types assigned to industrial robots i.e. handling and welding (International Federation of Robotics - IFR Statistical Department 2007, P. 17). The two main aspects that characterize classical architectures are a weak information flow and lack of sensor-based programming and operation. Both aspects significantly affect the programming and operational capability of any robotic work-cell.

Information flow or exchange in a robot architecture could be labeled weak if information flows predominantly in one direction i.e. unidirectional. In a classical architecture, the simulation environment allows the user to define the task in an off-line manner and automatically generate code for the robot controller. Given that on-line adaptation is inevitable due to the differences between the virtual world and the real world, the robot behavior has to be correspondingly modified. During operation the user interface allows the user to send supervisory signals or operations setting to the work-cell. It also displays the actual status of the robot during operation. However, data pertaining to the behavior of the robot generated from on-line adaptation is scarcely fed back to the simulation environment. Consequently, this prevents the simulation environment from updating the virtual world with information gathered in the real world. Therefore, in spite of the research efforts aimed in this direction (as mentioned in section 2.2.2), off-line and on-line programming methods rarely overlap.

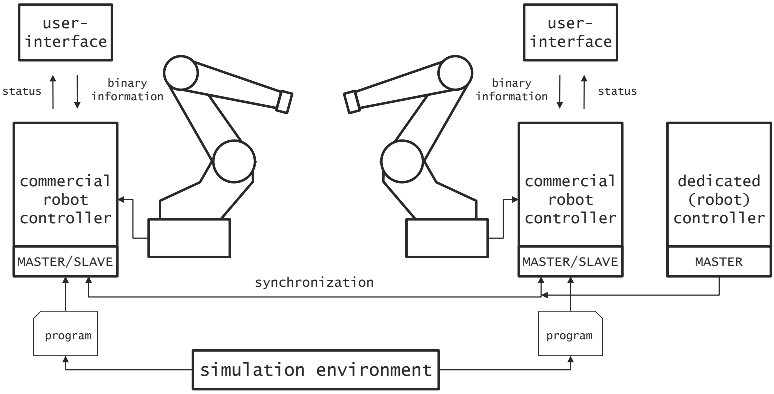


Figure 2.9: A typical CIR architecture

The second aspect is sensor-based programming and operation in industrial robots. Despite the fact that sensor-based robotics has been a staple of robotics research in the last 20 years (Christensen et al. 2009, p. 4), it has yet to gain wide-spread acceptance in production facilities. Reasons ranging from doubtful sensor readings and the robustness of control algorithms to the lack of RT interfaces on commercial robots are among the most commonly encountered (Christensen et al. 2009, p. 14)(Georgia Institute of Technology et al. 2009, p. 17). Not only are they lacking, but their implementation is also troublesome considering the closed nature of much of today's industrial manipulators. They lack standardized real-time interfaces and open access to internal data in the controllers. In a technical sense the latter reasons are enough to hinder sensor-based architectures from becoming more popular. However a holistic view of the issue discloses an overarching reason, namely that tight integration of sensors in robot architectures in practice is virtually nonexistent. For instance the utilization of sensors in different capacities during programming and during operation is rarely investigated. Furthermore, simulating the sensors off-line is still limited to research platforms such as the Player/Stage/Gazebo framework (Gerkey et al. 2003) (Collett et al. 2005) or Microsoft Robotics Developer Studio (Microsoft 2011). The latter discussion also applies for CIR which aren't much radically different than industrial robots (see Figure 2.9). As already summarized in Figure 2.8 such architectures additionally possess a communication channel through which all the robot controllers can synchronize their movement according to the master/slave principle referred to in section 2.5.2. Where a dedicated controller/robot controller or simply one robot controller generates a clock-cycle to be followed by all the others.

2.6.3 Programming paradigms

Although robot programming has been in the forefront of research and development for robot manufacturers, it has yet to experience radical changes regarding the applied programming paradigms. This could be readily traced back to the relation between programming, architectures and additionally the human machine interface (HMI). In order to understand the extent of this work the intrinsic and sometimes ambiguous relation between robot programming and architectures will be elaborated upon. As already discussed in the latter section the types of components in a robotic work-cell and how they are arranged and interconnected directly influences not only the operation capability but also the programming capability. MacDonald et al. (2003) emphasized this acute connection by considering the robot infrastructure and its interconnections as a major component of a robot programming system. They stated that one of the conceptual components necessary for a programming system is:

> *"the underlying infrastructure including designs for architectures that support and execute robot behavior descriptions, especially in distributed environments"*

The rigidity of robot architectures and their significant influence on programming systems led to the emergence of two distinct phases that occur during robot operation:

1. First phase: Robot programming is regarded as the process during which a human operator/programmer define instructions for the robot in order to execute a certain task.
2. Second phase: Robot execution is regarded as the process during which those instructions are executed at runtime without human intervention.

The classical programming paradigm handles the two phases distinctively in a manner so that the phases do not overlap with each other, i.e. the robot program (instruction list) is invariable throughout the runtime until further human intervention. Any disturbance arising during task execution can not be accounted for. Similar to a new task, any disturbance has to be defined and incorporated in the final instruction list. This classical definition assumes that the robot cell is incapable of gathering information about its surrounding environment and thus incapable of taking decisions regarding any dynamic change in runtime. This is, in turn, bound to the capability of the work-cell to integrate and utilize sensory feedback to enhance and adapt its behavior which is directly linked to the work-cell's architecture. Moreover, other than a few efforts from manufacturers such as integrating a space mouse or a joystick, no simple HMI have found their way in commercial CIR systems. Furthermore, graphical user interfaces utilized for programming both single industrial robots and CIR systems are usually cluttered with functionalities that render them complicated and hence require extra training.

3 Motivation and Objective

A straight path never leads anywhere except to the objective

ANDRE GIDE

3.1 Motivation

Despite the fact that most commercial robot manufacturers offer cooperating robotic systems in some form, either as an add-on technology or through a dedicated central controller (ref to section 2.5), their wide spread acceptance in production facilities have yet to materialize. This is attributed to a multitude of problems concerning their programming and operation. The discussion in the previous chapter highlights the shortcomings of both existing research and industrial practices in the field of CIR. Accordingly, these issues can be summarized as follows:

- The conventional method of programming tightly coupled CIR by off-line programming with on-line adaptation represents a technically complex process and hence incurs high costs. Furthermore, the complexity increases disproportionately with the number of robots involved in the planned operation (REINHART & ZAIDAN 2009).
- Synchronization schemes in CIR are limited to kinematic signals (position and its derivatives) (GERSTENBERGER ET AL. 2005). Hence no provision for dynamic signals (force and torques) and consequently no force control is possible.
- Although stresses build up in the work-piece during operation due to inaccurate positioning and/or deflection through loading, force measurement and subsequent control in CIR in an industrial context is rarely investigated (SPILLER & VERL 2010)(REINHART ET AL. 2010b).
- Even if the dynamic loading on the work-piece is determined, integrating this information in the architecture comes as an afterthought. With the exception of a few research works (SCHNEIDER & CANNON 1993)(SURDILOVIC ET AL. 2001), tight integration of dynamic loading in programming and throughout task execution is virtually non-existent in industrial systems.
- Research activities were fixated on proving the validity of control schemes for cooperating manipulators but hardly investigated the integration of force control in programming and subsequent operation.

- Providing simple programming techniques through integration of intuitive HMI is seldom discussed in the context of CIR.
- Most experimental test-rigs use research components instead of commercial off-the-shelf components and hence make it more difficult to replicate the performance on an industrial test-rig.

3.2 Objective

Most CIR programming approaches focus on the robotic manipulators, irregardless of the type of cooperative task (tightly or loosely coupled). Given that for tightly coupled CIR only one work-piece is manipulated, it would be beneficial to focus on the work-piece instead of the manipulators themselves. Hereby the objective of this research can be derived as follows:

> ***Simplifying the process of programming tightly coupled cooperating industrial robots by investigating an approach for programming based on defining the required task of the work-piece and not of the robots. The approach requires imparting full control of the work-piece to the user.***

The idea builds upon and extends the work done by several other researchers albeit with focus on the *programming* aspect and not the *control* aspect solely. Particular inspirations for this research work are the operational space concept developed by Khatib (1987), the object impedance for cooperative manipulation developed by Schneider & Cannon (1992) and the cooperative bilateral teleoperation scheme investigated by Lee & Spong (2005). As commonly known for robotic manipulators, defining equations of motions is done either in the generalized coordinates also known as the joint space or in the operational space also termed the task space (Khatib 1987). In the latter formulation, the manipulator's motion is allowed to be defined according to the task at hand and not the manipulator's structural characteristics as in the former case. Although the operational space formulation contains an implicit definition of the manipulator's dynamics, it specifically moves the focus of the corresponding control analysis to the forces acting on the end-effector of the manipulator as seen in Figure 3.1. Similarly, the work-piece based programming approach (WPBA) moves the focus of programming from the robots to the work-piece. Thereby building on and extending control concepts pertaining to cooperating manipulators to cover the programming aspect.

On the rationale for a work-piece based approach
The WPBA encompasses a methodology to define and satisfy practical task objectives formulated in the work-piece space. This is done by imparting to the user full control over the work-piece in spatial coordinates irregardless of the manipulator's configuration. By extension, since the object lies in the operational space of the task i.e. the work-piece space, and the cooperative task mainly entails manipulation of the object, the work-piece

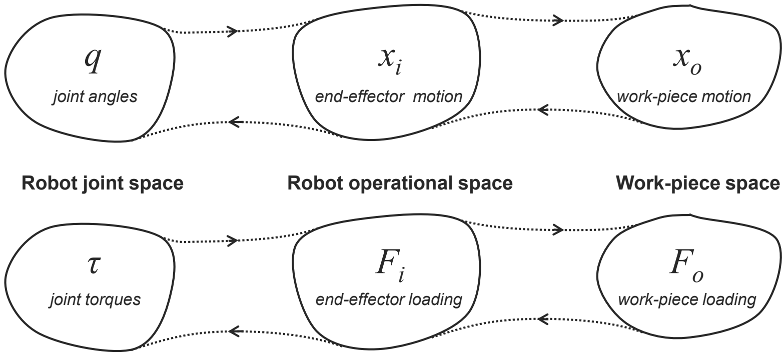


Figure 3.1: Overview of control spaces

based approach becomes particularly appealing for application on industrial robots due to the following reasons:

Modeling-wise
: If the object being manipulated possesses no DOF i.e. no actuated or under-actuated joints, it is considerably less complicated to model its dynamics than modeling the manipulator's dynamics. An extremely important consequence thereof is virtually decoupling the dynamic models of the robots from those of the whole system.

Control-wise
: Regarding industrial manipulators, it is easier to define force control strategies w.r.t. forces on the TCP in a cascade arrangement on top of the existing position control loops. Despite decreasing the bandwidth for the possible force control attainable, it guarantees a robust inner control loop to build on.

Mobility-wise
: By moving the control from joint space to operational space, the absolute translational movements of the manipulators' end-effectors could be commutatively divided between relative movements of the end-effectors w.r.t. the manipulators' base and the absolute velocity of the base. Due to the restricted nature of the cooperative workspace, this could be beneficial in increasing its overall volume.

Intuitive point of view
: Common automation tasks being executed in a normal or cooperative manner are more intuitively expressed in terms of the task not in terms of the task enablers.

4 Prerequisites and Conceptual Framework

Out of intense complexities intense simplicities emerge

WINSTON CHURCHILL

4.1 Overview

The work-piece based approach calls for imparting full control of the work-piece to the programmer during a cooperative task. As described in the latter chapter, full control includes posture of the work-piece and force loading on it in each phase of the given task. In order to achieve this, tightly coupled cooperative tasks have to be synthesized and understood. Combining this knowledge with the requirements imposed by the proposed approach on the system architecture, a conceptual framework for implementing the work-piece based programming approach is presented. This framework represents a conceptual blueprint for implementing the proposed approach according to generic design principles.

4.2 Task prerequisites

A comprehensive synthesis of cooperative tasks is introduced in this section. The aim is to gain a deeper insight into the separate phases and consequently derive the motion and loading i.e. position and force requirements for each phase. According to SURDILOVIC ET AL. (2001)(ZIVANOVIC & VUKOBRATOVIĆ 2006, P. 2-3) nine phases could be identified during a cooperative task. By omitting the 'planning of the approach' phase and combining the 'grasping' and 'gripping' phases together, seven phases are left. To cover the phases in a more task-oriented form, an extra phase 'machining' will be added to them, while extending the 'lowering' phase to include 'assembling'. Thereby, a total of eight phases will be presented here as shown in Figure 4.1. The synthesis however excludes the case of manipulators cooperatively assembling a work-piece, where each manipulator is holding one part of the final work-piece commonly known as dual-arm cooperative assembly (CHOI ET AL. 1999).

Phase 1: Approaching

The manipulators approach their respective grasping locations on the work-piece. This phase is characterized by a pure positional movement of the manipulator without interaction with the work-piece hence the need for force control is not necessary. Due to the shared workspace of the manipulators, mechanisms for collision detection and avoidance must be taken into account. After reaching the grasping location, the manipulators' velocity should be successively reduced. In this regard, grasping zones could be defined, that place constraints on the manipulator's movement once it enters.

Phase 2: Engaging/Grasping
After entering the grasping zone in the latter phase the manipulators start moving to their predefined grasping positions where they come for the first time in contact with the work-piece. During this phase, two aspects have to be taken into consideration. The first is that only one manipulator at a time should attempt interacting with the work-piece otherwise unexpected movement of the work-piece can happen. The second aspect concerns the force/position constraints during grasping. If the manipulators possess a relatively good positional accuracy and the work-piece is exactly positioned at the predefined position, no forces are bound to arise. Otherwise forces will immediately appear right after grasping, which represents the usual case given positional discrepancies on the factory floor. Additionally, force control here should be activated only after the grasping phase is terminated and deactivated once another manipulator starts its respective grasping phase. In order to consider all the latter aspects a grasping strategy should be implemented. This has to include the order in which the manipulators start and stop their grasping phase which implicitly includes the force and position constraints on each manipulator.

Phase 3: Lifting
The main aim of this phase is to allow the manipulators to detect and distribute the load among themselves. It begins when all manipulators are tightly grasping the work-piece i.e. no slip or friction is assumed. For synthesis purposes, friction between the gripper and the work-piece is always assumed non-existent. The center of gravity of the work-piece could be deduced from the force/torque distribution and the absolute positions of the manipulators. Before this phase ends, load distribution on the manipulators could be carried out according to a predefined distribution ratio (Zheng & Luh 1988)(Hsu 1989). However, it has to be taken into consideration that the position correction involuntarily leads to different force loading on the work-piece.

Phase 4: Transporting/Positioning
This phase represents the main body of the cooperative task. Whether transporting heavy loads or precisely positioning the work-piece, the objective is to change the absolute spatial position and orientation of the work-piece or to track a given predefined path. The main requirement here is of absolute position/orientation accuracy in terms of either positioning (point) or tracking (path).

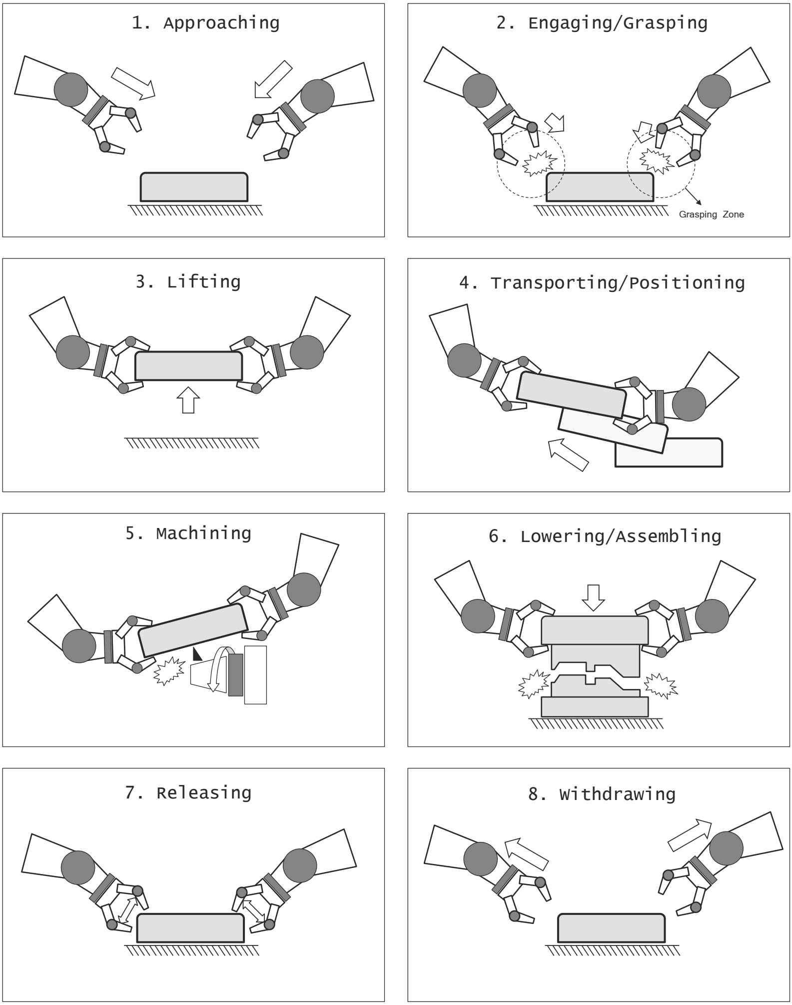


Figure 4.1: Different phases in a cooperative task

Any reduction in forces appearing during this phase directly affects the aforementioned accuracy. In order to deal accordingly with those forces, one must understand and categorize those forces. An adequate classification would split the types of forces into two. The first class represents forces arising due to misalignment between the manipulators which could be quantified according to the deviation between the actual end effector spatial position w.r.t a global coordinate system and the theoretical position. This misalignment arises however accurate the work-cell was calibrated. Millimeter-wise differences in geometrical features are also an extension of this class of forces. Apparently, this type calls for a local force reduction strategy. Thereby forces can be minimized or controlled at each manipulator within predefined constraints. The second type of loads arise directly due to interaction of the work-piece with an unexpected obstacle which was not taken into consideration during task planing. Such a case is common in dynamic or unstructured environments. Contrary to the first class, a global force reduction strategy must be employed here, in order to execute a synchronized motion aiming at adapting the path of the work-piece to the new state of the environment. Generally speaking, forces in the latter class would be more menacing to the system than those from the former class. Thus, it is safe to assume that if no mechanism to differentiate between the two classes is available and that the task takes place in an unstructured environment, the first class can be ignored if strategies for the second are activated.

Phase 5: Machining

This phase is directly related to the latter phase, since it represents a task being executed on the work-piece while it is positioned w.r.t a static or moving machine tool. Given that machining tasks are very complicated processes especially regarding the interaction between forces and positions, any definition of such an interaction is strongly coupled with task requirements especially accuracy and process quality. It is also safe to assume that the forces arising during machining should be handled by a global control strategy which takes the task specification into consideration.

Phase 6: Lowering/Assembling

This phase represents one of the more challenging phases of a cooperative task. Assembling or mounting could well be the main aim of the cooperative task instead of just changing the absolute spatial location and orientation of the work-piece as in the transport/positioning phase. In this case, the loading on the work-piece has to be controlled according to the process requirements e.g. tolerances, quality control etc. Hence a global control strategy is necessary to fulfill the given requirements without resorting to local strategies. This is based on the assumption that movement and thus local loading between the manipulators and the work-piece will be negligible considering the loading between the environment and the manipulator.

Phase 7: Releasing

During this phase the manipulators disengage from the work-piece thus ceasing any connection to it. Although the phase may seem simple, it requires a sophisticated and timed releasing strategy similar to that of the engaging/grasping phase to avoid a build-up of stresses. Such a strategy minimizes the risk of any destruction by guaranteeing either a simultaneous release by all manipulators or by ensuring compliant behavior from the manipulators while they disengage one at a time. It should also be noted that gravitational forces should be compensated on a local level to counter the sudden loss of load.

Phase 8: Withdrawing

The final phase in a cooperative action is withdrawing, whereby the manipulators withdraw from the vicinity of the work-piece. Akin to the approach phase, withdrawing requires pure position control and a free trajectory to prevent any collisions with the immediate environment or any other manipulators.

It is sometimes beneficial to cluster specific phases together. For instance phases 4, 5 and 6 could be clustered in a so called *process phase*, since the main process within the cooperative activity takes place during those phases. According to the latter analysis of the different phases, motion and force requirements associated with each respective phase in a cooperative task are listed in Table 4.1.

Phase	Motion/Force/Strategy	Features
Approach	Y/N/Local	Collision detection
Engage/Grasp	N/Y/Local	Grasping zones and strategy
Lift	Y/Y/Global	Weight distribution
Transport/Position	Y/Y/Local	Accurate positioning min. force
Machining	Y/Y/Global	WP control according to task
Lower/Mount	Y/Y/Global	WP control according to task
Release	Y/N/Local	Collision detection
Withdraw	Y/N/Local	Strategy similar to grasping

Table 4.1: Motion and force requirements for each phase in a cooperative task

4.3 System prerequisites

The discussion in section 2.6 points out the drawbacks of current robot work-cells w.r.t. their architecture and their adverse effect on the robot's programming capability. To overcome this, it is imperative to develop a flexible robot architecture that facilitate a simpler and more intuitive (re-)programming and operation of robotic work-cells. A flexible architecture should not dictate specific programming technique to the user, but allow him to create and mix different paradigms together.

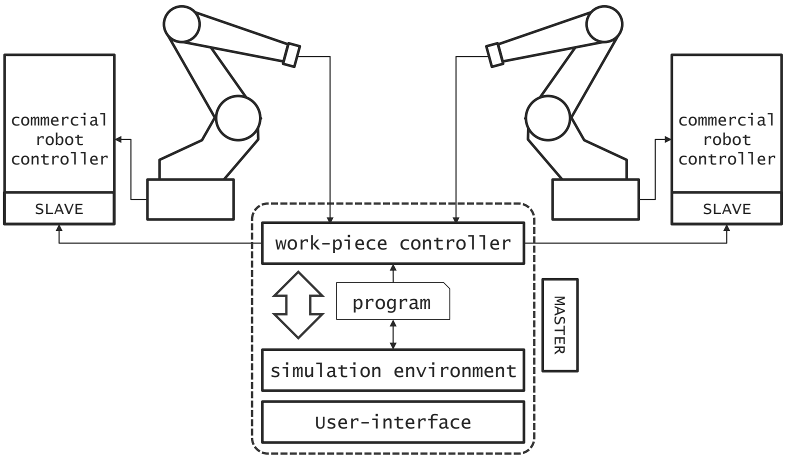


Figure 4.2: The envisioned architecture for implementing the WPBA on industrial cooperating robots

Consequently, rethinking of components and interactions in a robot architecture is deemed necessary to achieve this. To implement the WPBA a robotic architecture (as seen in Figure 4.2 compared to the classical architecture in Figure 2.9) is envisioned. Despite the similarity between both architectures, the proposed architecture exhibits nevertheless a significant departure from the classical controller. For instance, the work-piece controller and the master controller in classical arrangements represent the same centralized control approach but the former is tightly integrated with the simulation environment and the user interface. Additionally, force signals are exchanged between the robots and the controller, making dynamic information available for different control schemes irregardless of their location (centralized vs. decentralized schemes). Hence both kinematic and dynamic synchronization is guaranteed. Moreover, the integrated nature of the architecture allows the signals to be available for all other system components. On the whole, the architecture reduces the robots to mechanical units while control on all levels is unloaded to an external controller which communicates with the robots through appropriate interfaces. Consequently, it renders the work-cell more flexible to reconfigure according to the task at hand and the selected programming paradigm. Thus, the latter architecture leads to certain assumptions about what the CIR system should be capable of. In this section the prerequisites for such a system -which would fulfill the WPBA- are discussed.

4.3.1 Bandwidth-oriented control

In the classical sense, bandwidth defines the ability of the controller to respond to system inputs within a given time period (Mastascusa 2005) (System Dynamics - Response Time). This is mainly affected by actual implementation of the control loops i.e. the type and speed of hardware and communication means available to preform the control task in the real world. In this work, bandwidth oriented control is a redefinition of the classical layered control theory, where the layers are stacked upon each other according to decreasing bandwidth. Since decreasing bandwidth also means variable sample rate, it is convenient to define each layer with its own sample rate. Although the term multi-rate control could be theoretically used to define such a scheme, it is crucial to point out the significant difference between the two concepts. Multi-rate control denotes the existence of several sample rates in the control loop without alluding to any functionality inherent to these different rates. On the other hand bandwidth oriented control contains layers with different sample rates that are prescribed certain functionalities. Thus, not only is the control system limited to low-level control (i.e. high BW) but also to intelligent control and decision making. This technique is famous among robotic researchers which investigate both high-level and low-level control (Gueaieb et al. 2007b).

4.3.2 Sensor-based architecture

A sensor-based control architecture would allow the manipulators to continuously update their knowledge base and thus be more adaptive to the immediate environment they are in contact with (Schmitt et al. 2010). With this capability, adopting the needs of the task to those of the process would be greatly simplified by augmenting the control capabilities with sensoric measurements. Sensoric values augment an architecture in three different capacities. In a programming capacity, it could be used for correcting automatically generated code from an off-line simulation tool. Additionally sensors could also be directly utilized for programming the work-cell and hence facilitating simpler programming techniques as mentioned in section 2.2.1.3. In a control capacity, sensors are used in a closed loop control manner for disturbance rejection and achieving control objectives associated with the operation; whether they are motion or force requirements. Another common use is for safety measures during operation, where critical values are monitored and consequently safety procedures are triggered if they exceed defined limits. And finally in a high-level intelligence capacity, it is utilized for decision making and its accompanying real-time control during operation.

4.3.3 Flexible information flow

Guaranteeing that signals between different components and devices are freely exchanged in a robotic system is necessary to ensure maximum flexibility of information flow and signal transparency. Consequently, this provides a solid base for a flexible architecture as earlier proposed. Signal transparency here means that all signals flowing in the subsystems must be accessible to all decision-making entities whether they are human operators, intelligent control units or even remote entities. However, at one point in the control stack a line has to be drawn between low-level control and decision making. Although this line is up to the human system designer, it is imperative to reach a compromise to avoid over-flooding the signal interface with a huge number of signals. Safety is also an issue to be considered, if critical signals were allowed to be freely manipulated. On the other hand restricting the number of signals would lead to a classical situation where the interface is limited and would thereby restrain the capability of the whole architecture regarding information flow. Nevertheless, during actual implementation the designer should take performance issues into consideration. Details such as signal origins, signal data types, communication data rate and data types have a direct impact on inter-communication in an architecture and can't be overlooked.

4.3.4 Intuitive human machine interfaces

Since programming represents the focus of this, simple and intuitive methods to interact with the system are of paramount importance. Interaction here could be classified according to the activity of the human operator. Initially, during programming the operator should be able to dictate the motion of the robots and/or work-piece using intuitive input devices. Additionally, this has to be augmented with an easy-to-understand graphical user interface (GUI) to enable quick editing of programs. These devices and interaction methods should also be extensible to be utilized during testing and subsequent execution. Hence, mechanisms for uncomplicated configuration between programming and execution have to be designed.

4.4 Framework

Based on the latter task and system requirements a generic framework was derived to simplify programming of CIR by implementing the WPBA. It is imperative to understand that the framework is not a detailed design for a literal solution to the problem of programming CIR. It rather represents a conceptual blueprint which could be utilized in a generic manner. This also renders it flexible enough to be scaled and adopted according to the required functionality and specifications. Naturally, this makes it open for interpretation in light of several factors, chiefly being the designers' engineering backgrounds, technical preferences and prior experience. Furthermore, available hardware and budget constraints have a profound influence on the interpretation of the framework and subsequent implementation of the solution. As seen in Figure 4.3, the components of the framework could be aggregated into three main classes:

`Class 1:` `Functional Components`

This class contains two modules

1. **Control Module**: Encompasses algorithms and functions that possess real-time constraints i.e. that require high bandwidth. Any break down in the deterministic operation of this module leads to a failure of major functionalities in the system, in other words reliability has to be guaranteed at all times.
2. **Software Module**: Encompasses high-level control functions which are not critical to the safe operation of the system. Additionally it should provide the user with an intuitive GUI and simplify access to different HMI.

`Class 2:` `Device Components`

This class contains four modules

1. **Robot Module**: Is composed of the actual commercial manipulator and any extra actuators or linear motors attached to it. Any connections between the manipulators' commercial controllers and the rest of the components are defined according to the type and capabilities of the accompanying interface, whether it is real-time or not.
2. **Sensor Module**: Consists of all types of sensors such as FTS or infrared sensors.
3. **HMI Module**: This module consists of all HMI devices available to the user. It is interfaced through the software module. Visual monitors and haptic devices are examples of such devices.
4. **Task-level Peripheries**: These are devices that are imperative to task execution but are usually independent from low level control, such as grippers.

Class 3: Communication Architecture
Encompasses all communication buses connecting all the components together on an inter- and intra- module level.

The framework represents the backbone of this research work and accordingly the next three chapters will discuss the technical realization thereof. In chapter 5, the **control module** is realized as a multi-layered control architecture. Subsequently, in chapter 6, the **software module** in the form of a flexible software environment is developed. Since the remaining two classes namely [device components] and [communication architecture] rely heavily on the available hardware, they will be discussed in chapter 7 within the development of an industrial test-rig.

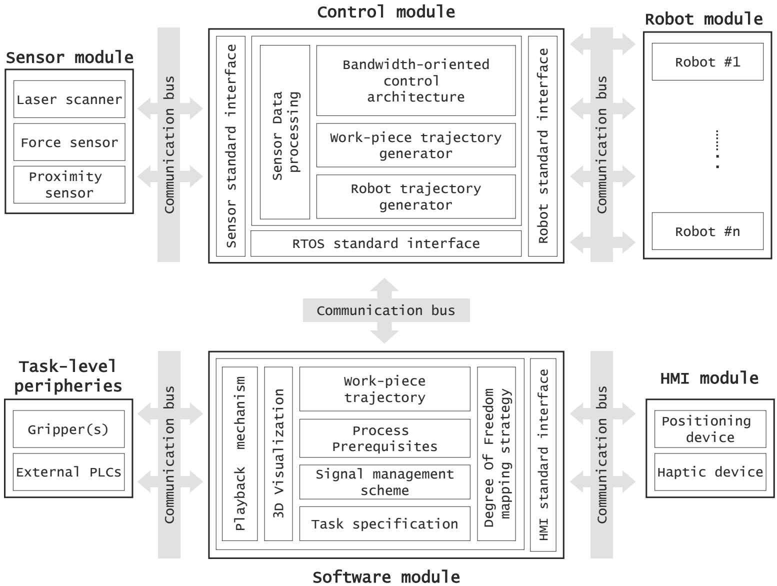


Figure 4.3: Conceptual framework for implementing the work-piece based approach

5 Control Architecture

The alteration of motion is ever proportional to the motive force impressed; and is made in the direction of the right line in which that force is impressed.

Mathematical Principles of Natural Philosophy
Isaac Newton

5.1 Overview

In this chapter the **control module** of the framework will be realized in the form of an encompassing control architecture. This architecture spans not only the control module but also its interaction with the software module to enforce the WPBA. Based on previous research in the field of cooperative manipulation and in order to gain valuable insight into the control problem, it is imperative to synthesize it into two separate, but nevertheless mutually exclusive sets of problems. The first addresses the interaction of a robot with its environment i.e. transition from the state of free movement to the state of contact and how this contact could be controlled (discussed in section 5.3). While the second problem addresses the control of cooperating manipulators working in tandem and how their interaction with each other and with the environment is controlled (discussed in section 5.5).

5.1.1 Basic definitions

Although controlling manipulators was one of the very first problems that caught researchers' attention with the advent of modern robots, research in this field continues uninterrupted to this day. The complex nature of that robotic behavior aspired to with state-of-the-art manipulators makes the underlying control architecture all the more so important. Consequently, high-level automation and artificial intelligence techniques hoping to mimic complex human behavior are rendered useless unless accompanied by adequate control algorithms. Those algorithms make sure the required behavior is translated into motion instructions, which the robot dynamics are capable of executing (Georgia Institute of Technology et al. 2009, P. 44). This led to different interpretations of control architectures and their constituent parts. Thereby, to avoid confusion relating to the terms in this chapter, the following definitions for all control-related entities and activities are introduced:

Control Laws

Control laws describe the unique relationship between an input and an output signal defined in the frequency domain or the time domain as a transfer function[1]. This definition also includes multi-variable control relationships described by the state space representation (Dorf & Bishop 2007, P. 632).

Control Structures

Control structures are built with different components to achieve a certain control functionality through a dedicated routing of signals in a system. Consequently, the structures usually overlap irrespective of their content, thereby several blocks could build more than one control structure according to the routing.

Control Loops

Control loops describe the arrangement of control structures/components w.r.t. each other and according to the location of execution. Hence a loop could be characterized as centralized or decentralized if an operation takes place in a central or a distributed fashion respectively (Khatib et al. 1996a).

Control Modes

Control modes are digital[2] (ON/OFF) signals which are mainly used to enable/trigger specific control structures or any other assistance algorithms. They could be in a sense considered as enablers of discrete automation i.e. programmable logic controllers (Weck & Brecher 2006, P. 81).

Control Layers

Control layers are differentiated by their varying bandwidth. Each layer is essentially executed with a unique sample rate or within a specific bandwidth[3]. The layers could contain any number of control entities sampled at the same rate. It allows to organize those entities not in a functional manner but from an aspect of implementation (Yin et al. 1997).

Control Architecture

The sum of all components that involve decision making, control activities and control entities within a system built in a hierarchical manner. One may consider the hierarchies as organizational building blocks orthogonal to the aforementioned control entities. The topmost hierarchy describes the control layers. The second is the operation modes which trigger other control structures. The third and most basic block is the control law. An architecture can also be regarded as composed of several layers, which may contain overlapping control

[1] Whether it is a continuous-time signal (s) or a discrete-time signal (z)
[2] Also termed binary or boolean signals.
[3] A bandwidth is characterized by an upper and lower sample rate limit.

structures triggered by operation modes. It is mainly characterized by a multi-rate nature as seen in Figure 5.1, allowing it to be mapped to varying bandwidth (Gueaieb et al. 2007b).

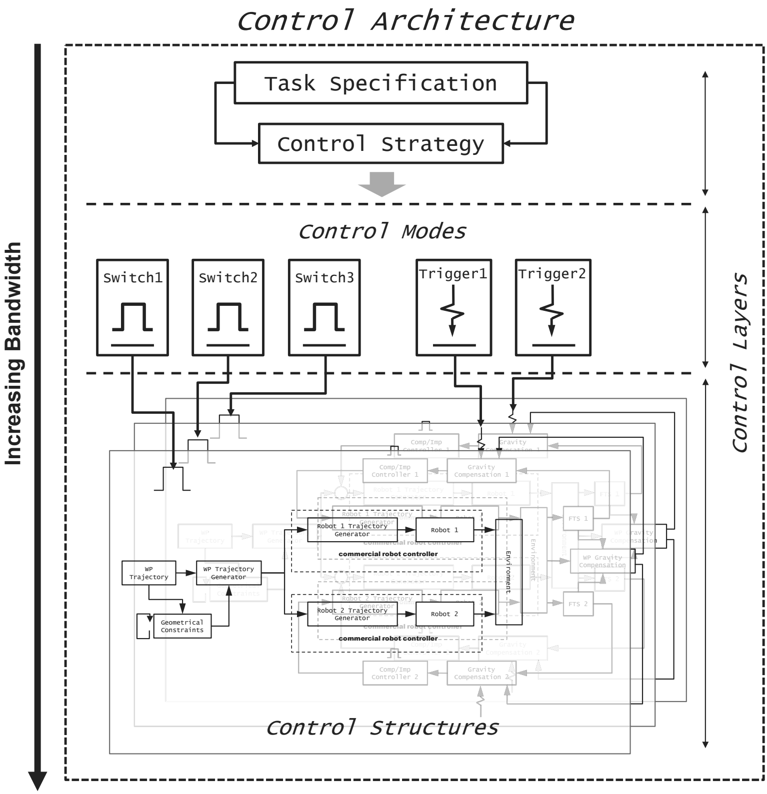


Figure 5.1: The different definitions in a control architecture

It is imperative to note that this chapter mainly discusses the undermost level i.e. the control structures. Nevertheless the information with the other layers is also outlined and will be thoroughly discussed in the next chapter.

5.2 Modeling

Modeling serves to understanding relations between the different components of the system in light of kinematic and dynamic analysis. This assists in designing the required control architecture. Furthermore it facilitates decreasing the design complexity of the controllers by incorporating assumptions to simplify the analysis.

5.2.1 Coordinate systems

In this section the different coordinate systems encountered in the analysis of cooperative manipulation will be introduced (Figure 5.2).

World frame ($\mathcal{W}$)
The world frame, also termed the 'global frame' is a fixed coordinate system which is usually set in an arbitrary location somewhere in the vicinity of the manipulators. To aid the mathematical description of the system, the frame is usually positioned so as to simplify the derivation of equations. A common practice is to adjust it to coincide with the initial position of the work-piece frame. All or most of the other coordinate systems are at some point defined w.r.t the world frame.

Robot frame ($\mathcal{R}$)
The robot frame is a coordinate system attached to the base of a given robot. It is usually considered fixed w.r.t the world frame unless the robot is capable of moving i.e. mobile or installed on a linear motor. In industrial robots, it is usual to have the z-axis of the frame coincide with the axis of the first joint, thus simplifying kinematic analysis.

Tool frame ($\mathcal{T}$)
This frame is attached to the TCP of a given robot. In industrial practice and sometimes in the research literature it is commonly/interchangeably termed the TCP frame. It is imperative to point out that this frame *does not* lie at the robot's flange, but is directly attached in the middle of the gripper. In most cases, the gripper is not flexible in a manner that the mathematical transformation between both the tool frame and the flange frame is constant, hence the interchangeable nature of the term. Additionally, the posture of this frame in practice is defined w.r.t the robot frame.

Work-piece frame ($\mathcal{O}$)
This frame is usually attached to the work-piece's center of gravity. The initial orientation thereof is usually aligned to that of the World Frame.In this work only one work-piece is considered at any given time. Therefore, only one work-piece frame exists through this ongoing analysis.

Generalized coordinates
In dealing with rigid bodies, sometimes it is more advantageous to describe the system in a set of generalized coordinates which are independent from a fixed frame or coordinate system. This eliminates the variables needed to express constraints between the variables.

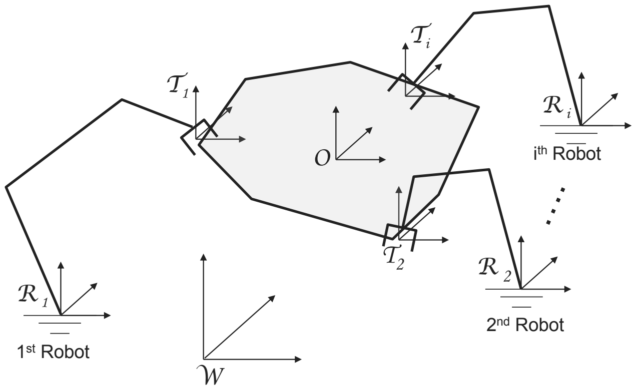


Figure 5.2: Different coordinate systems in a cooperative task

In robotics, it is common to use the joint angles and their derivatives as generalized coordinates. When those sets of coordinates are utilized, the analysis is made in the *joint space*. Thus differentiating it from an analysis in one of the latter frames, which could be collectively termed the *task space*[4].

5.2.2 Posture representation

A rigid body is completely described in space by its position and orientation w.r.t a given reference frame or coordinate system. Since the total DOF of a body in space are equal to six, therefore the posture $x_{\mathcal{W}}$ of any rigid body w.r.t to the world frame could be uniquely described with six mutually exclusive parameters:

$$x_{\mathcal{W}} = [\ \vec{r}\ ,\ \vec{\theta}\ ]^T \in \mathbb{R}^6 \tag{5.1}$$

where:

$\vec{r} \in \mathbb{R}^3$ describes the three translational DOF *along* the x, y and z in the world frame

$\vec{\theta} \in \mathbb{R}^3$ describes the three rotational DOF *around* the x, y and z axis in the world frame

[4] Using the term *task space* to describe world, robot, tool or work-piece frame depends on the specific phase of the task

Postures could be described in various mathematical forms. The choice thereof depends on how convenient this form could derived, manipulated and eventually implemented in a controller. In this section two forms of posture representation and their notation will be introduced. Their mathematical properties makes them ideal candidates for the implementation in the next sections.

Homogeneous transformation notation (HTN)
This type of notation is defined as a matrix that is artificially composed of the translational vector $\vec{r}$ and the rotational vector $\vec{\theta}$ shaped in a rotation matrix $\boldsymbol{R}(\theta)$. To convert from one coordinate system to another or to move a vector around several coordinate systems, one must simply multiply the 4×4 matrices together in the right order. The homogeneous transformation matrix is mathematically defined as (Spong et al. 2006, P. 61)

$$\boldsymbol{\Gamma}_x = \begin{bmatrix} & \boldsymbol{R}(\theta) & & \vec{r} \\ 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{5.2}$$

where:

$\boldsymbol{R}(\theta) \in \mathbb{R}^{3\times3}$ is the rotation matrix representing the rotation of $\vec{x}$ around $\vec{\theta}$

and is defined as

$$\boldsymbol{R}(\theta) = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & u_z \end{bmatrix} \tag{5.3}$$

Vector quaternion notation (VQN)
This notation is applied here for its simple structure, although it entails more complicated programming practices for implementation. Its simplicity lies in the fact that it maintains the posture represented in a vector format albeit with a different length. The VQN is mathematically defined as (Khalil & Dombre 2002, P. 53)

$$x_W = [\ \vec{r}\ ,\ \breve{\theta}\ ]^T \in \mathbb{R}^7 \tag{5.4}$$

where:

$\breve{\theta} \in \mathbb{R}^4$ is the quaternion representation of the rotation vector $\vec{\theta}$

5.2.3 Kinematics

5.2.3.1 Analysis for single manipulator

In order to deploy a robot for a given task, it is imperative to either know or to dictate the path of the robot end-effector or TCP. Thus the goal of the kinematic equations is to establish the relation between the joint movement and the end-effector movement (Spong et al. 2006, P. 76). Since the number of joints of a robot defines its kinematic capability, any definition starts by listing the joint variables. Essentially, two types of joints exist in robotics: revolute and prismatic (Sciavicco & Siciliano 2005, P. 39). Revolute joints are those allowing rotational motion around their axis through geared drives, while prismatic allow translational motion along their principle axis through linear drives. For an n-drive robot, the vector of joint variables is written as

$$q = [\ q_1\ ,\ q_2\ \ \dots\ \ q_n\ ]^T \in \mathbb{R}^n \tag{5.5}$$

If the number of robot drives is equal to or exceeds six drives, then the end-effector is capable of a six DOF motion in spatial space. At any moment in time, the posture of the end-effector w.r.t the robot's frame can be expressed as a function of the joint variables

$$x_{\mathcal{R}} = f(q) \tag{5.6}$$

To obtain expressions for the velocity and acceleration of the TCP, equation (5.6) is differentiated twice w.r.t time, yielding the following expressions

$$\dot{x}_{\mathcal{R}} = \dot{f}(q) = J(q)\dot{q} \tag{5.7}$$

$$\ddot{x}_{\mathcal{R}} = \dot{J}(q)\dot{q} + J(q)\ddot{q} \tag{5.8}$$

where $J(q) \in \mathbb{R}^{6\times n}$ is termed the *jacobian matrix* and is defined as

$$J(q) = \frac{\partial x}{\partial q} \tag{5.9}$$

Depending on the manipulator configuration, the *jacobian matrix* represents mapping from the joint space to the task space of the TCP. It constitutes one of the most useful mathematical tools in robotics. It is utilized to find singular configurations of a manipulator, to analyze redundancy and even to map the joint torques to the forces/torques arising on the TCP (Sciavicco & Siciliano 2005).

5.2.3.2 Analysis for cooperative manipulators

In order to extend the latter definitions to a group of cooperative manipulators interacting with the same work-piece, the following is assumed:

- **Assumption 1**: The system exhibits flexibility only through the control system and not due to any flexibility in the work-piece, hence the manipulators and the object are considered rigid and non-deformable (Kosuge et al. 1997)(Reinhart & Zaidan 2009).
- **Assumption 2**: The grasp between the manipulators and the object at the gripping point is tight i.e. no slipping or sliding occurs (Caccavale et al. 2000).
- **Assumption 3**: The *jacobian matrices* either in an analytical or in a geometric form are known and hence the kinematics of the manipulators are known (Gueaieb et al. 2007a).
- **Assumption 4**: The geometrical and dynamical properties of the object are known (Gudiño Lau & Arteaga 2005).

The starting point of this analysis are the 1^{st} and 2^{nd} assumptions. According to which the distance between the object frame and any tool frame is constant throughout the movement of both the manipulators and the object. Consequently, the rotation of the TCP frame of any manipulator i is equal to the rotation of the object frame o

$$\vec{\theta}_i = \vec{\theta}_o \tag{5.10}$$

or in rotation matrix form:

$$\boldsymbol{R}(\theta)_i = \boldsymbol{R}(\theta)_o \tag{5.11}$$

according to Figure 5.3, vector $\vec{r}_{(O-\mathcal{T}_i)}$ defines the distance between the the two frames O and $\mathcal{T}_i$, which is constant throughout the motion. Hence, the position of the tool frame $\mathcal{T}_i$ w.r.t. the world frame is equal to

$$\vec{r}_i = \vec{r}_o + \boldsymbol{R}(\theta)_o \vec{r}_{(O-\mathcal{T}_i)} = \vec{r}_o + \vec{r}_{\theta_o} \tag{5.12}$$

where

$$\vec{r}_{\theta_o} = \boldsymbol{R}(\theta)_o \vec{r}_{(O-\mathcal{T}_i)} \tag{5.13}$$

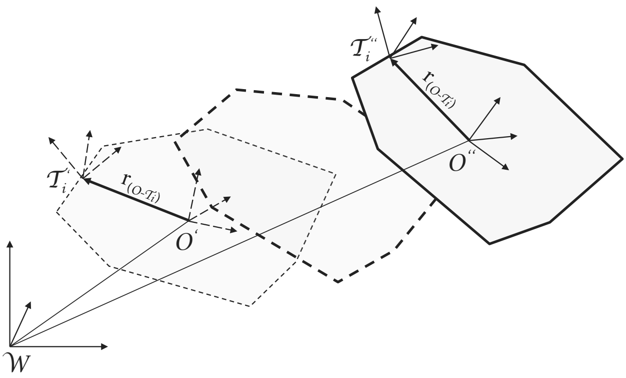


Figure 5.3: Movement of the tool frames relative to movement of the object frame

Equations (5.10), (5.11) and (5.12) represent the geometrical constraints on the body and consequently on the movement of the manipulators. By differentiating twice w.r.t time, the expressions for the kinematic constraints (Caccavale & Villani 2000)

$$\dot{\theta}_i = \dot{\theta}_o \tag{5.14}$$

$$\dot{r}_i = \dot{r}_o - \boldsymbol{S}(\vec{r}_{\theta_o})\dot{\theta}_i \tag{5.15}$$

and

$$\ddot{\theta}_i = \ddot{\theta}_o \tag{5.16}$$

$$\ddot{r}_i = \ddot{r}_o - \boldsymbol{S}(\vec{r}_{\theta_o})\ddot{\theta}_i - \boldsymbol{S}(\vec{r}_{\theta_o})\boldsymbol{S}(\dot{\theta}_i)\dot{\theta}_i \tag{5.17}$$

where $\boldsymbol{S}(\cdot)$ is the skew symmetric matrix performing the cross vector multiplication on its arguments.

5.2.4 Dynamics

In the upcoming section all discussions and mathematical derivations will be conducted based on non-redundant manipulators with six DOF.

5.2.4.1 Analysis for single manipulator

The dynamic equation of a manipulator in joint space with $q \in \mathbb{R}^6$ can be written as follows (Kurfess 2005)

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D\dot{q} + G(q) = \tau - J^T(q)F \tag{5.18}$$

where:

$M(q) \in \mathbb{R}^{6\times6}$ is the symmetric positive definite inertia matrix, which also represents the kinetic energy of the dynamic system in terms of a *lagrangian* analysis

$C(q, \dot{q}) \in \mathbb{R}^{6\times6}$ represents the Coriolis and centrifugal forces/torques

$D(q, \dot{q}) \in \mathbb{R}^{6\times6}$ is a diagonal matrix representing the viscous friction coefficients

$G(q) \in \mathbb{R}^6$ represents the vector of gravitational forces acting on the manipulator's structure

$\tau \in \mathbb{R}^6$ represents the joint force/torque applied by the actuators on the manipulator

$F \in \mathbb{R}^6$ represents the forces/torques arising on the TCP

The operational space formulation allows robotic researchers to formulate manipulator problems in the task space of the robot (Khatib 1987). By using the acceleration at robot's TCP as derived in Equation (5.8) and by neglecting the friction components in Equation (5.18), the following expression can be easily obtained[5]

$$\boldsymbol{M}_a(x)\ddot{x} + \boldsymbol{C}_a(x, \dot{x})\dot{x} + \boldsymbol{G}_a(x) = \tau_a - F \tag{5.19}$$

where:

$\boldsymbol{M}_a = f(M, J) \in \mathbb{R}^{6\times6}$ is representative of the *kinetic pseudo-energy* of the dynamic system in the operational space (Sciavicco & Siciliano 2005)

$\boldsymbol{C}_a = f(M, C, J, \dot{q}) \in \mathbb{R}^{6\times6}$ represents the equivalent of the Coriolis and centrifugal torques/forces in the operational space

$\boldsymbol{G}_a = f(G, J) \in \mathbb{R}^6$ represents the equivalent of the gravitational forces in the operational space

τ_a represents the contribution of the TCP forces due to joint actuation

[5] The question of redundant and non-redundant manipulators will not be handled here due to its irrelevancy. Further information regarding this topic w.r.t. the operational space formulation could be found in (Khatib 1987)

Equation (5.19) not only transfers the analysis and consequently control of a manipulator from the joint space to the task space but also generates a unique mathematical relationship between the joint forces and the forces arising on the TCP during contact with the environment. This relationship could also be obtained using the principles of virtual work in static analysis (Sciavicco & Siciliano 2005)

$$\tau = J(q)^T F \tag{5.20}$$

5.2.4.2 Analysis for cooperative manipulators

The dynamic equations for a set of m cooperative manipulators in the operational space for $i = 1, ..., m$ could be written as follows

$$\begin{aligned}
&\boldsymbol{M}_{a1}(x_1)\ddot{x}_1 + \boldsymbol{C}_{a1}(x_1, \dot{x}_1)\dot{x}_1 + \boldsymbol{G}_{a1}(x_1) = \tau_{a1} - F_1 \\
&\boldsymbol{M}_{a2}(x_2)\ddot{x}_2 + \boldsymbol{C}_{a2}(x_2, \dot{x}_2)\dot{x}_2 + \boldsymbol{G}_{a2}(x_2) = \tau_{a2} - F_2 \\
&\qquad\vdots \\
&\boldsymbol{M}_{ai}(x_i)\ddot{x}_i + \boldsymbol{C}_{ai}(x_i, \dot{x}_i)\dot{x}_i + \boldsymbol{G}_{ai}(x_i) = \tau_{ai} - F_i \\
&\qquad\vdots \\
&\boldsymbol{M}_{am}(x_m)\ddot{x}_m + \boldsymbol{C}_{am}(x_m, \dot{x}_m)\dot{x}_m + \boldsymbol{G}_{am}(x_m) = \tau_{am} - F_m
\end{aligned} \tag{5.21}$$

In a matrix form, the latter equations could be compounded to

$$\boldsymbol{M}_A\ddot{x}_A + \boldsymbol{C}_A\dot{x}_A + \boldsymbol{G}_A = \tau_A - F_A \tag{5.22}$$

where:

$$x_A = [\ x_1, \ldots, x_m\ ]^T \in \mathbb{R}^{6m}$$

$$\boldsymbol{M}_A = blockdiag(\boldsymbol{M}_{a1} \ldots \boldsymbol{M}_{am}) \in \mathbb{R}^{6m\times 6m}$$

$$\boldsymbol{C}_A = blockdiag(\boldsymbol{C}_{a1} \ldots \boldsymbol{C}_{am}) \in \mathbb{R}^{6m\times 6m}$$

$$\boldsymbol{G}_A = blockdiag(\boldsymbol{G}_{a1} \ldots \boldsymbol{G}_{am}) \in \mathbb{R}^{6m\times 6m}$$

$$\tau_A = [\ \tau_1, \ldots, \tau_m\ ]^T \in \mathbb{R}^{6m}$$

$$F_A = [\ F_1, \ldots, F_m\ ]^T \in \mathbb{R}^{6m}$$

Equation (5.22) summarizes the collective dynamics of the manipulators but stops short of describing the cooperative action. However, this could be completed by considering the dynamics of the work-piece. A quick dynamic analysis thereof reveals the following equation

$$\boldsymbol{M}_o(x_o)\ddot{x}_o + \boldsymbol{G}_o(x_o) = F_o \tag{5.23}$$

where:

$\boldsymbol{M}_o(x_o) \in \mathbb{R}^{6\times6}$ is the symmetric positive definite inertia matrix of the work-piece

$\boldsymbol{G}_o(x_o) \in \mathbb{R}^6$ represents the vector of gravitational forces acting on the work-piece

$F_o \in \mathbb{R}^6$ represents the sum of all forces/torques acting on the work-piece's center of gravity

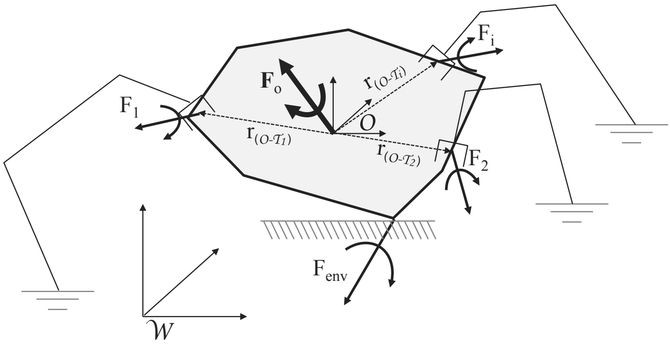


Figure 5.4: Force analysis on the work-piece

When more than one manipulator interact through a common object under the assumptions mentioned in section 5.2.3.2, the external force vector on the work-piece from the i^{th} manipulator are

$$F_{o-i} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{S}(\vec{r}_{(O-\mathcal{T}_i)}) & \boldsymbol{I} \end{bmatrix} F_i \tag{5.24}$$

and hence the resultant forces/torques acting on the center of gravity of the work-piece from the manipulators are equal to

$$F_o = \sum_{i=1}^{m} F_{o-i} \tag{5.25}$$

which connects the dynamics of the manipulator (Equation (5.21)) to the dynamics of the object (Equation (5.23)). Thereby transferring the focus of the analysis from the operational space of the robots to the work-piece space as envisioned in Figure 3.1. It is important to remember that all forces arising on the work-piece can not be solely attributed to the interaction between the work-piece and the manipulators. To obtain a comprehensive expression of the forces on the work-piece during interaction with the environment, Equation (5.25) has to be augmented with the extra term $F_{env} \in \mathbb{R}^6$, which represents the component of the total forces on the center of gravity of the work-piece generated by the interaction of the work-piece with the environment as illustrated in Figure 5.4.

$$F_o = \sum_{i=1}^{m} F_{o-i} + F_{env} \tag{5.26}$$

5.3 Interaction control

As already discussed in section 2.3 and section 2.4 different control algorithms, laws and structures have been heavily investigated in the literature. In this section, the overarching interaction method or control law implemented in this work will be introduced. This method is based on *force augmented impedance control* similar to what was investigated by Lopes & Almeida (2006). Initially, a fully fledged impedance controller is developed. The most popular form for such a controller though remains the mass-damper-spring system (MDSS) which is well studied and understood (Surdilovic & Vukobratović 2002, P. 23.14). During runtime the controller tries to achieve the target impedance by enforcing the relationship between the manipulator and its immediate environment. This target impedance is defined as (Kamnik et al. 1998)

$$f = \boldsymbol{M}_t(\ddot{x} - \ddot{x}_0) + \boldsymbol{B}_t(\dot{x} - \dot{x}_0) + \boldsymbol{K}_t(x - x_0) \tag{5.27}$$

where:

$x_0 \in \mathbb{R}^6$ is the reference posture signal

$\boldsymbol{M}_t \in \mathbb{R}^{6\times 6}$ is the symmetric positive definite matrix defining the target inertia

$\boldsymbol{B}_t \in \mathbb{R}^{6\times6}$ is the symmetric positive definite matrix defining the target damping

$\boldsymbol{K}_t \in \mathbb{R}^{6\times6}$ is the symmetric positive definite matrix defining the target stiffness

In the latter description a 6 DOF manipulator movement is assumed. By substituting the difference $x - x_0$ by an error signal X_e and expressing Equation 5.27 in the Laplace (s)-domain, one obtains:

$$F(s) = (\boldsymbol{M}_t s^2 + \boldsymbol{B}_t s + \boldsymbol{K}_t)X_e = Z_t(s)X_e \tag{5.28}$$

where:

$Z_t(s)$ is the target impedance in the s-domain and is equal to $(\boldsymbol{M}_t s^2 + \boldsymbol{B}_t s + \boldsymbol{K}_t)$

Although implementations on industrial robots generally refer to impedance, they deploy in reality another form referred to as *admittance* control. The difference arises from the nature of industrial robots since most of them offer no or limited access to the joint torques but rather to a task space position interface (Winkler & Suchy 2007b). Additionally the forces are usually not deduced from the loading on the motors but are measured directly at the TCP using sensors. Hence, the resultant control loop inverses the mathematical definition of impedance, but nevertheless maintains the concept intact

$$\frac{F(s)}{X_e(s)} = Z_t(s) \quad \textit{and} \quad \frac{X_e(s)}{F(s)} = A_t(s) \tag{5.29}$$

where:

$A(s)$ is the target admittance in the s-domain and is equal to $Z(s)^{-1}$

The latter control law is not only used for manipulator/work-piece interaction but also work-piece/environment interaction in what could be termed cooperative impedance (Kosuge & Hirata 2005, P. 20.8). Eventually the aim is to enforce this impedance across all interacting bodies and hence enforce controlled relationships between them, as shown in Figure 5.5. This will be apparent once the corresponding control structures are discussed. This approach is similar to the approaches investigated by Caccavale et al. (2008) and Moosavian & Ashtiani (2008), where *internal impedance/manipulator level impedance* denotes manipulator/work-piece interaction while *external impedance/object level impedance* denotes work-piece/environment respectively. The second step is to augment the latter control law with an active force controller in all directions (Szewczyk et al. 1996). This is readily achieved by subtracting a reference force from the measured forces

$$\frac{Xe_(s)}{F(s) - F_{ref}(s)} = A_t(s) \tag{5.30}$$

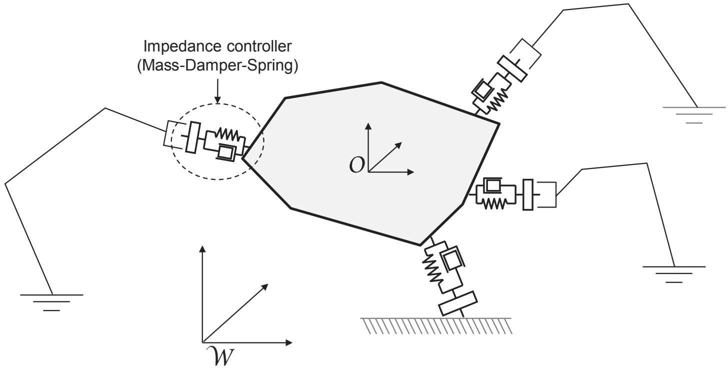


Figure 5.5: Cooperative impedance enforces impedance across manipulators, work-piece and environment

where:

$F_{ref}(s) \in \mathbb{R}^6$ is the reference force signal

Equation (5.30) represents the general form of the controller used throughout this work. This controller will be referred to in the upcoming structures as the interaction control law (ICL). According to the parameters of the controller different functionalities could be enforced. To parametrize this controller a total of 24 variables is required; the target inertia in all DOF, the target damping in all DOF, the target stiffness in all DOF and the reference force in all DOF. It is important to note that this controller can also be parameterized in run-time according to the task phase, predefined limits and triggers.

5.4 Assist functions

5.4.1 Force monitoring

In order to effectively implement active interaction control, forces/torques arising at critical points in the cooperative system have to be determined. Without adequate foreknowledge of the system dynamics i.e. structure and parameters force monitoring is deemed necessary (Liu & Abdel-Malek 2000). In this regard, force monitoring alludes not only to measuring the forces at the manipulators' TCP but also to observe the forces if not directly measurable. In practical implementations, two main methods to measure forces at the robot's TCP are commonly used. The first method utilizes the manipulators' motor currents (Böhm

1994)(Winkler 2006, P. 99-100). Assuming a constant velocity (hence zero acceleration), the following equation applies

$$F_{TCP} = J.\tau = J.K.i \tag{5.31}$$

$K \in \mathbb{R}^6$ is a constant value relating the motor currents to the joint torque

$i \in \mathbb{R}^6$ is the motor currents of n motors in the manipulator

For complex cases involving redundant motors (> 6) and variable velocity profiles, an accurate dynamic model of the manipulator is necessary. Otherwise the force/torque values computed will be not accurate enough. Although gaining access to motor currents doesn't entail extra costs, this method is highly complicated and requires real-time computation of the non-linear robot model. The second method is the direct measurement of forces using a dedicated force/torque sensor (FTS) mounted between the robot's TCP and the gripper. Modern force sensors are either based on the piezo-electric effect or on strain-gages. (Hesse & Malisa 2010, P. 65-70). Most commercial FTS utilize strain-gages mounted on different beam structures (Voyles et al. 1994). By mathematical manipulation of the gages' readings, forces and torques in the six principle directions could be determined. Regarding observing the forces/torques on the work-piece, it is quite difficult to separate between the two components indicated in Equation (5.26) (Winkler & Suchy 2007a). For practical purposes the latter equation will be handled as follows

$$F_o \approx \begin{cases} \sum_{i=1}^{m} F_{o-i}, & \text{for work-piece/manipulator interaction} \\ F_{env}, & \text{for environment/work-piece interaction.} \end{cases} \tag{5.32}$$

Equation (5.32) represents the core of the geometrical force observer used in this work. It was constructed based on the assumption that the component of the resultant force on the work-piece due to the interaction between the manipulators is negligible to F_{env} when the work-piece starts to interact with the environment[6]. The importance of such an assumption will become evident when adaptive control is discussed.

5.4.2 Gravity compensation

Due to the FTS being mounted between the gripper and the TCP of the manipulator, a gravity component arises on the FTS due to the weight of the gripper in the free state and the weight of both the gripper and the object in loaded state. Without adequately compensating for those forces before activating any controller requiring the forces, will

[6]In this case, the environment denotes not that of the manipulators i.e. the work-piece, but rather that of the work-piece itself. In interaction control literature, a robot interacting with a work-piece is considered interacting with *its* immediate environment which is not the case here.

only lead to faulty results and unnecessary drift. Gravity compensation for the work-piece compliance is more complicated than for the single robot case and requires prior knowledge of the location of the center of gravity of the work-piece w.r.t all the TCP in the system.

5.5 Control structures

In the following sections, several control structures built upon the analysis introduced in the latter sections will be presented. As stated at the beginning of this chapter, control structures exhibit different control behavior by a dedicated routing of signals through the same components in a given layer. It is imperative here to define the functions of the basic blocks encountered in the upcoming sections so as to understand the behavior of each structure:

Commercial robot controller
In this composite block, the commercial robot controller along with the robot itself is aggregated into one block. Thereby, utilizing the commercial controller instead of implementing the robot trajectory generator required by the framework. However, it must be noted here that this block is built on the assumption that the robot vendor provides a real-time interface to the robot's trajectory generator. Not only is this required, but the interface must be capable of executing commands in the robot's task space. The representation of the block in this manner allows the control designer to handle the robot as a black box and cease worrying about the inner workings of the controller, since the commercial controller guarantees robust and accurate positioning.

Work-piece trajectory This is a generic block which denotes the source of the work-piece trajectory and is formally a component of the software module in the framework. For example the trajectory could be generated in an off-line simulation environment, or on-line using a jogging device. The output is either a trajectory in a HTN or a VQN w.r.t the world frame.

Work-piece trajectory generator The work-piece trajectory generator is responsible for converting the trajectory of the work-piece into corresponding coordinated trajectories for each manipulator depending on the given geometrical constraints. These constraints are defined before the operation starts, but could also be changed during operation.

5.5.1 Coordinated motion

The most simplest and most straightforward type of control is implemented by enforcing the geometrical constraints on the robots. In principle, this type of control is what could be found in commercial CIR mentioned in section 2.5. As already discussed in section 5.2.3.2 geometrical constraints guarantee that the movement of the TCP of all robots is synchronized to the movement of the work-piece. The most important prerequisite for the operation here is the relative calibration between the robots, which is usually done through calibration w.r.t. a fixed frame. Any deviations or discrepancies would lead to internal loads and stresses on the work-piece.

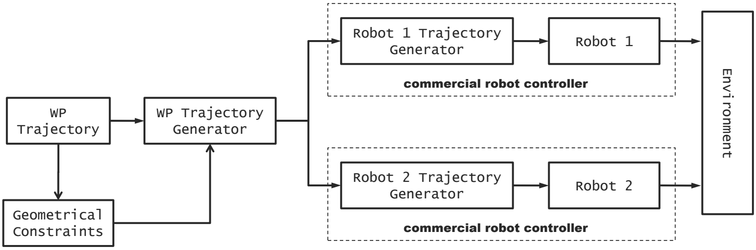


Figure 5.6: Coordinated position control in work-piece mode

The structure illustrated in Figure 5.6 with a built-in geometrical constraints block represents the basis of the upcoming structures. To set those constraints, two methods are applied here, namely the work-piece method and the offset method. In the work-piece method, the relative paths of all robots are calculated w.r.t. the work-piece path. Hence, the constraints required here are those described in Equations (5.10) to (5.12). Consequently, the implementation in the VQN takes the following form

$$\vec{r}_i = \vec{r}_o + \breve{\theta}_o \vec{r}_{(O-\mathcal{T}_i)} \breve{\theta}_o^{-1} \tag{5.33}$$

$$\breve{\theta}_i = \breve{\theta}_{(O-\mathcal{T}_i)} \breve{\theta}_o \tag{5.34}$$

where

$$\vec{r}_{(O-\mathcal{T}_i)} = \breve{\theta}_{0,o}^{-1} (\vec{r}_{0,i} - \vec{r}_{0,o}) \breve{\theta}_{0,o} = Constant \tag{5.35}$$

$$\breve{\theta}_{(O-\mathcal{T}_i)} = \breve{\theta}_{0,i} \breve{\theta}_{0,o}^{-1} = Constant \tag{5.36}$$

On the other hand, the offset method calculates the relative paths of all robots w.r.t. another robot. Hence their relative positions and orientations w.r.t the master robot stay constant as long as this mode is turned on. This could also be termed a master-slave configuration, whereby it pertains only to geometrical or kinematic synchronization. Consequently, the implementation in the VQN takes the following form

$$\vec{r}_i = \vec{r}_j + \breve{\theta}_j \vec{r}_{(\mathcal{T}_j - \mathcal{T}_i)} \breve{\theta}_j^{-1} \quad (5.37)$$

$$\breve{\theta}_i = \breve{\theta}_{(\mathcal{T}_j - \mathcal{T}_i)} \breve{\theta}_o \quad (5.38)$$

where

$$\vec{r}_{(\mathcal{T}_j - \mathcal{T}_i)} = \breve{\theta}_{0,j}^{-1} (\vec{r}_{0,i} - \vec{r}_{0,j}) \breve{\theta}_{0,j} = Constant \quad (5.39)$$

$$\breve{\theta}_{(\mathcal{T}_j - \mathcal{T}_i)} = \breve{\theta}_{0,i} \breve{\theta}_{0,j}^{-1} = Constant \quad (5.40)$$

Although both methods achieve the same behavior, the offset method suffers from the classical delay between the master and slaves which is avoided by using the work-piece method. This is attributed to the fact that the work-piece method sets the work-piece as master and the robots as slave and given that the work-piece itself is virtually controlled through the robots, the delay on all robots will be uniform.

5.5.2 Accommodation control

In this section a common type of interaction control will be implemented to facilitate one of the major aspects of the WPBA. This type of control enables the lead-through programming method already discussed in section 2.2.1.2. Different implementations refer to this type of control with several names, for instance accommodation, hand-guided and compliance. Due to the nature of cooperative tasks and its multiple phases with its respective requirements discussed in section 4.2, this type of control will be implemented in two structures. The first structure implements the control law in an independent fashion while the second implements it in a work-piece based fashion. The principle upon which the compliance[7] control law is built on, is very simple and is directly derived from the impedance concept introduced in section 5.3. The control law is a form of energy dissipation whereby a manipulator dissipates the energy imparted to it by moving in the path of least resistance and hence minimizing the forces on it. In a mechanical sense, it acts as a damper such that no energy is conserved but rather absorbed. In terms of the general controller discussed in section 5.3 takes the following form

$$F(s) = \boldsymbol{B}_t X_e s \quad (5.41)$$

[7] for simplicity's sake it will be termed *compliance* control

Owing to the simplicity of the control law and control structure, it has been widely adopted and in recent times successfully commercialized (Pires 2008b). The main application is to manually position the robots for non-interaction tasks for instance welding or for assisting elderly workers with physically demanding tasks (Neto et al. 2009)(Spillner et al. 2008).

5.5.2.1 Single manipulator mode

This structure is a local implementation of the compliance control law on each robot. No signals whatsoever connect the robots together as shown in Figure 5.7. Each manipulator has its own compliance controller and gravity compensation. The basic function of this structure, is that it allows the user to move both robots independently from each other only with the sheer force of his hand. Thus, by activating one or both compliance control blocks, the user is able to simply position the TCP of each robot manually. Another function derived from this is imparting to any manipulator a compliant nature during cooperative tasks. Before such an action is made possible, it is however imperative to change the controller gain to accommodate the difference between the dynamics of a human hand and that of a rigid object.

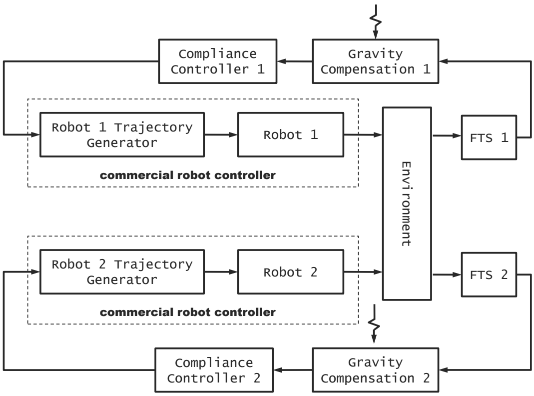


Figure 5.7: Control structure facilitating accommodation control in single manipulator mode

5.5.2.2 Work-piece mode

On the contrary to the latter implementation, this structure is of a centralized nature. All manipulators are connected to each other through the work-piece signals. The basic function of this structure is to enable the compliance control law but on the work-piece level.

To achieve the latter objective two steps are necessary. The first is to monitor or observe the forces on the work-piece i.e. the internal loading of the work-piece due to external forces. A work-piece observer based on the principle discussed in section 5.4.1 is located after local force monitoring on the manipulators' TCP. The second step entails the observation of the work-piece's posture w.r.t the world frame based on the instantaneous posture of the manipulators. By coupling this information with the geometrical constraints, the movement of the work-piece in space will be consistent and will represent the force dissipation principle discussed previously. As shown in Figure 5.8 the structure implements the same concept as that of Figure 5.7 but in a centralized way. However, all the components are work-piece oriented instead of being specific to one or the other robot. It is also important to note that the work-piece's trajectory does not exist, since it stems from interaction control i.e. no trajectory is given or predefined. The work-piece simply reacts to any force exerted on it and moves away to reduce it.

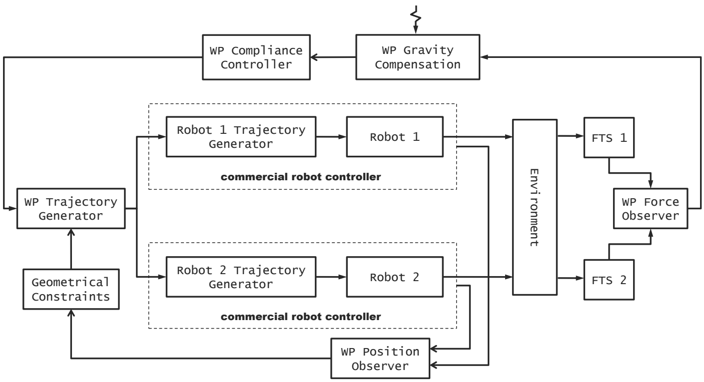


Figure 5.8: Control structure facilitating accommodation control in work-piece mode

5.5.2.3 Drift issues

Due to the nature of the control law, any forces measured or observed will be processed as a reference signal and hence will lead to a non-zero motion command. This unintended motion could arise due to a multitude of reasons. The most common is the faulty calibration of the robots. Hence the real position of the work-piece w.r.t to the TCP is not accurate resulting in an error translated as forces/torques. Another source are the residual forces on the TCP after the operator stops exerting effort on the TCP. These forces may be due to:

1. Inherent structural compliance in the FTS or in the work-piece itself.

2. Mechanical compliance between work-piece and gripper.

3. Structural and mechanical compliance of the robots.

Since the control parameters are based on stiff interactions, large compliance could render the control parameters invalid. To counter this problem, a simple technique has been employed. The movement of the manipulators and consequently the work-piece is coupled is only activated when needed. Hence the operator has to make sure an enable button is pressed as long as he intends to switch on this structure.

5.5.3 Adaptive control

The name of this type of control maybe somewhat deceptive. Adaptive here refers to the ability of the work-piece to *adapt* to its environment. As mentioned earlier in section 5.4.1, these forces arise either from positional inaccuracies during coordinated movement or from interaction with the environment. In order to differentiate between them, the interaction of the work-piece with the environment must be somehow detected. The simplest case would be a manual operator that switches between the two structures according to the given task phase. For a fully automatic operation a mechanism or an intelligent unit is required. In the next sections, two structures will be introduced to treat the two interaction cases. The two structures are based on the presumption inherent in Equation (5.32). The choice of the ICL in these structures depends on the operator and the criteria to be enforced. Consequently, two main criteria for interaction control exist. The first can be termed a *dynamic relation* criterion, which enforces a relationship between the objects in question using a fully parameterized impedance controller. While the second can be termed *minimum force* criterion which attempts to minimize the forces arising on the work-piece by dissipating energy using a compliance controller (as a special case of a full impedance controller). Given the generic interaction law discussed in section 5.3, it is possible to implement both criteria.

5.5.3.1 Manipulator/work-piece interaction

As mentioned before, Equation (5.26) reduces to the first part of Equation (5.32) when no interaction between the work-piece and the environment occurs. The idea behind this structure is simple and represents a form of a decentralized control (Kume et al. 2002a)(Khatib et al. 1996a). When all manipulators are moving in a coordinated fashion, all of them use the forces arising on their respective TCP to control or minimize the internal loading between them and the work-piece. In this case the generated trajectories are modified to accommodate the local forces, thus the reference trajectory of the work-piece is irreversibly lost. To avoid this, the same technique can implemented while retaining only one of the manipulators in position control. This would have the effect that all the manipulators (*slaves*) deviate from their reference trajectory except one (*master*) which will actually maintain its reference position and also that of the work-piece[8]. However, the trajectory of the work-piece will still be lost, but could be easily referenced to that of the master manipulator. The latter scheme can be termed a dynamically coupled master-slave approach analogous to the terminology mentioned in section 2.4.1. This is illustrated in Figure 5.9. Where the second robot is position controlled and hence represents the master manipulator while the first is the slave manipulator since the compliance/impedance controller is active. The interaction control loop of the second robot is grayed out to indicate that it is available if needed. According to preference, the roles could simply be switched to reference the work-piece trajectory to other robots.

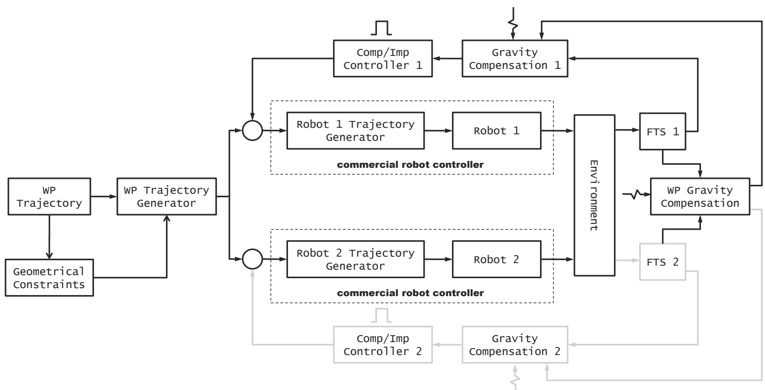


Figure 5.9: Control structure facilitating adaptive control for manipulator/work-piece interaction

5.5.3.2 Work-piece/environment interaction

This structure addresses the case when interaction between the work-piece and the environment takes place. Compared to the latter structure, this is fully centralized in nature. The forces on the respective TCP are fed into a work-piece force observer to determine the total forces on the work-piece. Consequently these forces could be controlled and a correction trajectory for the work-piece is generated, which subsequently augments the original reference trajectory specified by the operator as shown in Figure 5.10. It is apparent that the outcome here is a compliant work-piece motion, whether it be a compliance or impedance controller. This structure is of utmost importance in operations where the 6^{th} phase (refer to section 4.2) comprises the main objective of the operation

[8]Based on the assumption that the contact between them is rigid as mentioned in section 5.2.3.2

i.e. an assembly task. Not only is it possible to control the forces but by dynamically changing the controller parameters it is also possible to execute autonomous tasks with the work-piece by controlling the shape of the interaction in a variable fashion.

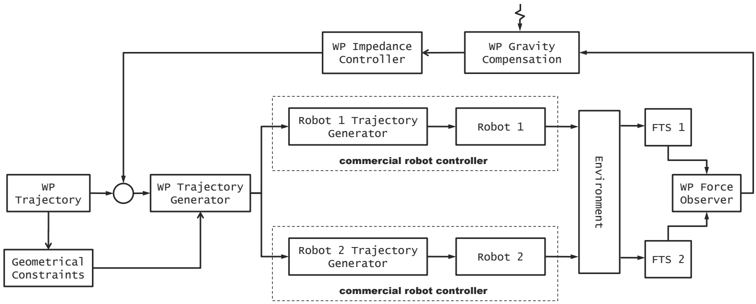


Figure 5.10: Control structure facilitating adaptive control for work-piece/environment interaction

5.6 Architecture

In the latter section the different control structures pertaining to all functionalities required from the control module were discussed. These structures are based on the interaction control principles presented in section 5.3. Along with the assistance functions, the latter components establish the core of the control architecture as a practical implementation of the control module in the framework. The architecture in its entirety is illustrated in Figure 5.11. Referring to the definitions and terminology in section 5.1.1, the architecture is based on three main layers.

The first layer, termed the **low-level control** layer, consists of what requires real-time execution and hence lies on the higher end of the bandwidth scale. It represents the integration of the components from the aforementioned control structures in a manner which facilitates an exclusive triggering of any structure at any given time without interfering with the others. A rearrangement of the blocks reveals three major loops: two *decentralized* inner loops representing the local control around each robot, a *centralized* outer loop representing the global control on the work-piece. Since the work-piece in itself is a non-drivable object, its movement is based on the robots' motion and hence the centralized loop is based on the decentralized loops. Each loop has its own fully parameterized interaction control law (ICL) making it capable of enforcing the triggered structure. The second layer, termed the **high-level control** layer, contains two main components executed in a way similar to programmable logic controllers (PLC). The operation modes trigger either the robot mode which mainly invokes the decentralized control loops or the work-piece mode which triggers the centralized control loop.

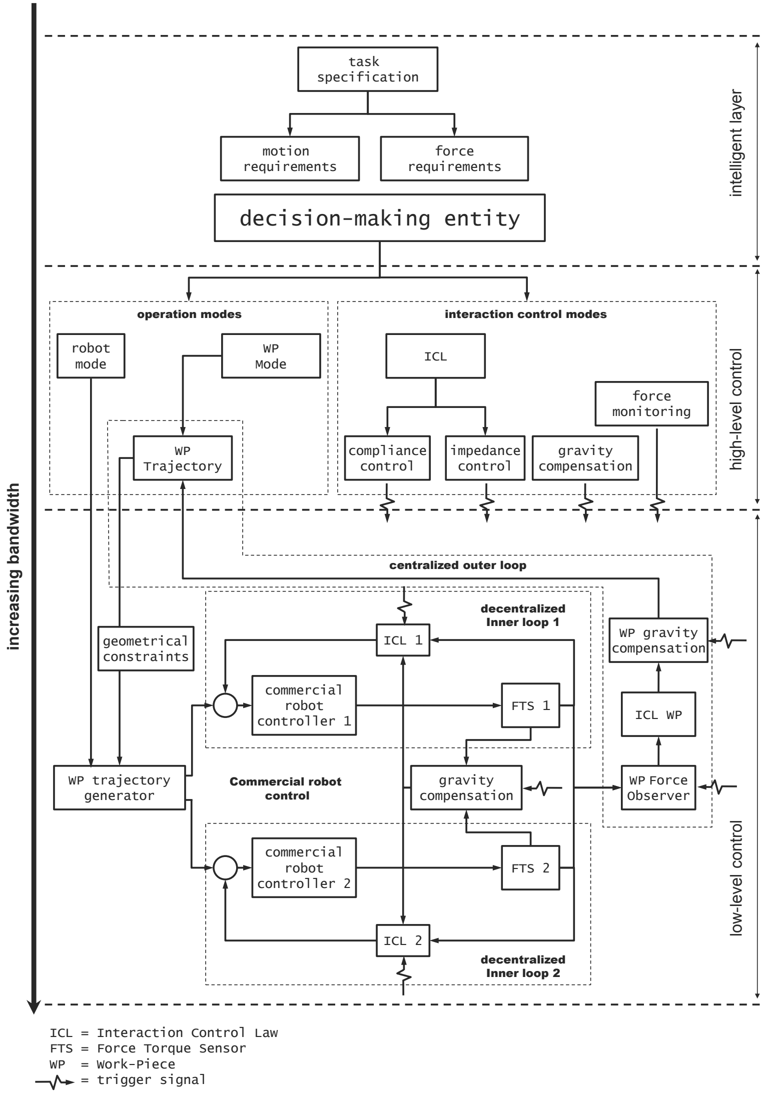


Figure 5.11: The control architecture implementing the work-piece based programming approach as a technical realization of the control module

Furthermore, this mode sets the geometrical constraints arising from the relative coordinates of the robots w.r.t. the work-piece. On the other hand, the interaction control modes are responsible for parameterizing the ICL in the low-level control layer through predefined parameters. These parameter are in turn used to enforce the required interaction functionality i.e. compliance control or impedance control. Assistance functions such as gravity compensation and force monitoring are defined and subsequently activated/deactivated utilizing similar modes. The third and uppermost layer is the **intelligent** layer. It represents the interpretation of the task into motion and force requirements by a decision-making entity. This could be a human operator or an artificial intelligent unit which is capable of deciding *move*, *generate*, *trigger* and *enable* actions to successfully complete a task.

5.7 Summary

In this chapter, the control module was technically realized as an architecture enforcing the control functionalities of the WPBA. Initially, the kinematics and dynamics of single and cooperative manipulators were investigated revealing the mathematical relationships governing their interaction. Subsequently, a generic interaction control law based on the impedance principle augmented by a reference force was developed. This law was fully parametrized to enforce different interaction behaviors either between the robots and the work-piece or between the work-piece and the environment. Given that the forces on the work-piece have to be monitored, a geometrical force monitor was developed. Moreover, gravity compensation in order to take the work-piece's weight into consideration during control activities was implemented. Subsequently, control structures to enforce certain functionalities were introduced. These structures contained overlapping blocks but differed in their respective signal routing, hence resulting in variable outcome. Additionally, the arrangements of the blocks according to their geographical locations revealed two major control loops; namely local for single robots and global for the work-piece. All in all, four structures were developed, two enforcing accomodation control and adaptive control for single robots and two enforcing the same functionalities but for the work-piece. The aforementioned components, interaction control law, assistance functions and control structures were brought together in one encompassing control architecture. The architecture features several layers each with its own bandwidth. Hereby providing the flexibility of separating control actions requiring real-time execution from others with no real-time constraints. Moreover, the architecture takes into consideration the motion and force requirements of each phase in a cooperative task, thus enabling the technical implementation of the proposed approach.

6 Software Environment

Any sufficiently advanced technology is
indistinguishable from magic

Profiles of the Future: An Inquiry Into The Limits of the Possible
Arthur C. Clarke

6.1 Overview

In this chapter a software environment is developed, representing the technical realization of the **software module** in the framework. It combines different concepts and ideas providing the user with several programming techniques to attain full control of the work-piece. Combined with the control functionalities discussed in the latter chapter it facilitates a flexible and powerful architecture allowing the user a simple interface to the robot work-cell in three different capacities (SRI International 1988):

- As a *production designer*, it allows the user to plan the work-cell and virtually simulate the process.
- As a *robot programmer*, it allows the user to combine different programming paradigms to (re)program the work-cell.
- As a *work-cell operator*, it allows the user to monitor the work-cell and make changes accordingly to boost the performance or adapt to arising disturbances.

Historically the role of software environments for industrial applications has been to simulate the feasibility of a given robot-based operation and consequently its behavior w.r.t. a specific production process. Thus robot manufacturers introduced simulation environments with their robot cells to address the customers' needs in this regard (refer to section 2.2.2). Hence, production planners were able to test the process hypothesis in a safe way, before trying to perform the operation on a real work-cell. A natural extension of this method was to automatically generate robot code from the simulation and transfer it to the robot in order to be executed. Although this constituted a very plausible and cost-effective method of robot programming, it failed to replace on-line programming methods. Owing to the fact that discrepancies between the simulation and reality abundantly exist, the generated code which executed flawlessly in the simulation is rendered useless in the real world (Yong & Bonney 1999, P. 357)(Vogl 2009, P. 20). Thus production planners were faced with two choices: either to write the program from scratch using on-line programming or manually adapt the existing code to the real world i.e. post-adaptation. This debacle lead to two parallel developments. The first was lead by robot manufacturers and software developers to develop more accurate environments, that would require little or no post-adaptation. Hence tighter calibration between the real and virtual worlds was rigorously enhanced (Brecher et al. 2010). The second was spearheaded by research institutes and partly by commercial companies to develop simpler on-line programming techniques which would render robot programming a simple and inexpensive process (Reinhart et al. 2007)(Hein et al. 2008)(Hoffmann et al. 2009). A major objective thereof is to help SME which are characterized by small production lots and hence more frequent re-programming to deploy robot-based automation in a cost-efficient manner (Schraft & Meyer 2006).

6.2 Requirements

The objective is to design a software environment that would offer the user simple and intuitive methods to plan, program, deploy and monitor a robotic work-cell. As already mentioned the software should assist the user in three different roles; production planning, robot programming and work-cell operation. The overarching aim here, is not to only develop a simulation tool but an environment which serves as the cornerstone of a flexible robot architecture in an integrated and level-oriented manner (Garlan & Shaw 1993, P. 28)(Georgia Institute of Technology et al. 2009, P. 43). Based on the latter statement and in order to avoid the limitations of commercial simulation tools available on the market, the design of this environment should fulfill the following requirements:

1. Providing a 3D-graphic simulation which could be augmented with realistic physical behavior i.e a physics simulation engine should also be included. Furthermore, compatibility with similar tools should be ensured through a simple and understandable environment specification based on standard description schemes.

2. Ensuring a transparent information flow between all components of the system. Transparency in this sense means that at any time during operation, the operator has access to all signals measured from, observed or fed into the system.

3. Interfacing intuitive HMI devices and allowing the user to (re)configure them according to both the operators and the tasks' needs. Additionally, it provide standard interfaces to external devices including sensors, and other process specific accessories like grippers. This also encompasses ethernet-based communications with real-time or non-real-time components.

The next three section are devoted for discussing how the latter requirements are technically realized in the software, which was named 'PuppetMaster4D'.

6.3 Environment

6.3.1 Structural components

To develop a software environment fulfilling the aforementioned requirements, special purpose software engines are needed. The engines should act as the core around which the environment's architecture will be developed. The growth of the PC gaming industry in the latter decade contributed immensely to the increasing number of software game engines available whether open source or proprietary. Although no unifying definition and scope for game engines exists (Anderson et al. 2008) they usually incorporate 3D graphic rendering, physics engine networking and human-interaction mechanisms among other functionalities (Lewis & Jacobson 2007). Some engines only offer graphic simulation capabilities (The OGRE Team 2011)(Nikolaus Gebhardt et al. 2011) while others offer integrated packages which contain physics-based simulation and collision detection (id Software 2011)(Fuchs 2011). In this work the open source `CHAI3D` engine was chosen as the core for the environment (Conti et al. 2003)(Conti et al. 2005)(Conti et al. 2011a). Its characteristics make it appealing for the required purpose due to the following reasons:

- The graphical rendering mechanism is based on OpenGL's `GLUT` libraries (Silicon Graphics 2011a), which are platform independent and hence easily portable to different platforms. The OpenGL standard (Silicon Graphics 2011b) itself is widely supported by graphic chips' manufacturers, making hardware acceleration possible for a large range of graphic cards.
- A simple mechanism to include all the graphical objects defined in a physics-based simulation through integration with the open source physics engine `ODE` (Open Dynamics Engine) (Smith 2006).
- The graphical and physics rendering loops are conveniently built around a haptic rendering loop. Special mechanisms in `CHAI3D` handle the interplay between the different loops allowing the user to concentrate on the task at hand without worrying about the thread handling of each loop.
- Support for a wide range of haptic and input devices. Additionally, integration of new devices is greatly simplified through template classes and simple interfaces to device drivers.
- The engine is characterized by an object-oriented structure, making it simple to build upon utilizing inheritance and polymorphism concepts central object-oriented programming (Scheibl 2003, P. 26).
- It has a simple and intuitive structure, which is augmented with many tutorials (with code) to help newcomers to quickly build applications of their own.

The QT (Nokia 2011b) framework is chosen for implementing the user interface and additionally serving as a basis for integrating other libraries. The framework itself contains diverse libraries which simplify implementing not only a dynamic GUI but also widely used interfaces. Figure 6.1 illustrates the main constituents of the software and how the different libraries are arranged w.r.t. each other. The CHAI3D libraries containing both the ODE engine and the GLUT libraries represent the core of the software, around which other libraries are connected through the QT framework. QTGui and QTCore are utilized for the GUI, which allows common mouse-based functionalities like *drag and drop* and widget rearranging. While QTNetwork and QTXML are used for external communication and manipulating XML data structures, respectively. Several device drivers and interface libraries are incorporated to provide for connectivity to diverse HMI. For instance, the free WiiUse library (Homebrew Channel 2011) is used for interfacing the Nintendo wireless remote device (WiiMot) (Nintendo 2011). The Novint Falcon and the Sensable PHANTOM haptic devices are interfaced through the hdfalcon and hdPhantom libraries respectively (Novint 2011)(Sensable Technologies Inc. 2011). Additionally, the digital input/output (I/O) card from QUANCOMM is interfaced through QLib, a vendor-specific library (QUANCOM Informationssysteme GmbH 2011b).

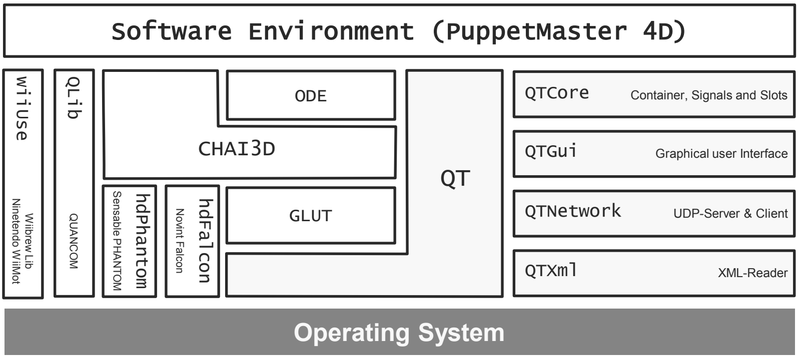


*Figure 6.1: Library components of the software environment (*Hubele *2009)*

6.3.2 Joints

The choice of CHAI3D included the choice of ODE as the physics engine. ODE is an open source physics engine with a small foot-print and efficient calculation schemes (Smith 2011). Inherent to the engine is also a collision detection capability which is connected to the physics calculation. Of particular interest in ODE are the types of joints representing several types of real life motors and actuators (Figure 6.2). They are described as follows:

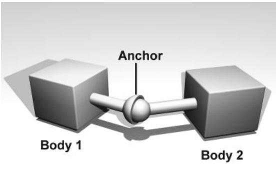
Ball possesses 3 rotational DOF and is specified by one point i.e the *anchor*.

Hinge possesses 1 rotational DOF along an *axis* at the *anchor* and is commonly used to represent rotational motors (whether geared or not).

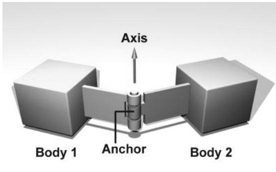
Slider possesses 1 translational DOF along an *anchor* and is mainly used to represent linear motors.

Piston possesses 1 translational and 1 rotational DOF and is in reality a combination of a slider and a piston and represents the motion in an internal combustion engine.

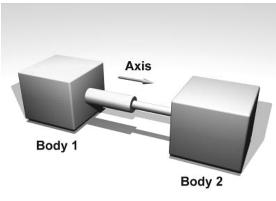
Fixed possesses no DOF, its main aim is to fix two bodies together preventing any relevant movement between.



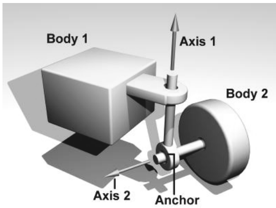
[a] Ball



[b] Hinge



[c] Slider



[d] Piston

Figure 6.2: Joint types in `ODE` *from http://www.ode.org*

6.3.3 XML-based description

XML is quickly becoming the language of choice for information saving and retrieval. Its simple and human-like structure makes it an ideal contender for cross-application information exchange (Bray et al. 2008)(World Wide Web Consortium 2011). This is reflected in the increasing number of applications which define their information in XML structures/files (Arnaud & Parisi 2007). In the computer graphic industry an effort is being undertaken to unify the environment descriptions in graphic software including modeling, animation and manipulation. The COLLADA consortium developed a simple and intuitive environment definition based on the XML language specification to describe 3D content (Sony Computer Entertainment Inc. & Khronos Group 2011). In this work, a similar data structure was developed to describe a dynamic virtual environment (Barnes et al. 2008). The data structure represents not only the graphical and dynamical description of the software but also the signals and information flowing through the software. In the following descriptions the different components of the structure will be presented:

Environment properties
A basic environment definition must contain certain information at the beginning of the description. This includes the name (`name`), dynamic simulation parameters (`erp`, `cfm`) and light properties (`ambient`,`specular`). For the dynamics to be correctly calculated the gravity vector has to be defined (`gravity`). The background color of the graphical environment could also be directly defined (`color`). According to the user's needs, different light objects could be included to give a photo-realistic graphical rendering of the environment. A combination of the latter elements in a basic environment definition would look like the following xml-listing:

```
FILE: test_environment.env
<environment name="Test_Environment" scale="1.0" erp=".8" cfm="1e-18"
     ambient="-.8" specular="1.">
  <gravity z="-9.81"/>
  <color r="1." g="1." b="1." />
  <light>
    <color r="1." g="1." b="1." ambient="0" specular="1."/>
    <translation x="50" y="100" z="80" />
  </light>
</environment>
```

Object hierarchical structure
To accommodate chained structures, the XML-description is based on a hierarchical structure. It allows the aggregation of objects that constitute one body or mechanism to be represented in a *parent-child* manner. For instance, in the following listing an articulated robot with six DOF is described accordingly from base to wrist:

```
FILE: RobotTypeX.prt
<object name="robot">
  <object name="base">
  </object>
  <object name="arm2">
  </object>
  <object name="arm3">
  </object>
  <object name="wrist">
    <object name="wrist1">
    </object>
    <object name="wrist2">
    </object>
    <object name="wrist3">
```

```
    </object>
  </object>
</object>
```

Beside being easy to understand the relation between different objects, this method is also used to build a library of commonly used components and referencing them for repeated use or saving them for use in other definitions. A prime example thereof is that of robotic mechanisms. Using the definition of a specific robot class in the latter listing, one could reference it repeatedly in the same environment but with different names and locations to create an array of similar robots. It has to be noted though, that in their definition different base locations have to be defined.

```
<environment>
  [...]

  <object name="Cooperating_Robots">
    <object group="Robot_1" url="RobotTypeX.prt">
      <translation x="0" y="0" z="0" />
    </object>
    <object group="Robot_2" url="RobotTypeX.prt">
      <translation x="100" y="100" z="80" />
    </object>
    <object group="Robot_3" url="RobotTypeX.prt">
      <translation x="200" y="100" z="80" />
    </object>
  </object>

  [...]
<environment>
```

Object I/O

Where the `<ioobject>` denotes the existence of inputs (the `<controller>` structure) and/or outputs (the `<output>` structure) for the object it is wrapped in. The latter signal structure is not only limited to simulated objects but also to graphical and external objects. The controller sets the variables for the position control of the respective component and additionally sets limits on the positional and rotational DOF. Moreover graphical lines denoting the forces from external FTS sensors could be visualized according to a predefined scale:

```
<object name="arm">
  <ioobject switchports="true">
    <controller>
      <gain position="32" rotation="40"/>
      <limit position="1" rotation="1"/>
    </controller>
    <output />
    <forces force="100" torque="10" />
    <arrow scale="1" axis=".8" />
  </ioobject>
</object>
```

6.3.4 Graphical components

The graphical components allow the user to comfortably interact with the structural components of the software environment. With the exception of the signal matrix, all of the components are dynamically defined. Hence they are only available when the user creates them in run-time. Furthermore, most of the graphical components have emitter/receiver ports (this will be detailed in in Table 6.1).

Graphical view
The graphical view represents a 3D render or a view port of the environment in run-time, through which the user can visualize the effect of interactions and changes by him or by the constraints imposed by the physics engine. The user can also create more than one view port to visualize the environment from different perspectives. Additionally he can also manipulate each view independent of the others by zooming, panning and rotating it using the computer mouse.

Signal matrix
The signal matrix represents the front-end or graphical interface to the signal management scheme that will be introduced in section 6.4. It enables the user to reconfigure the signal connections between the different components in run-time. This window is constantly updated to accommodate new signals being dynamically defined or destroyed. If the user clicks a connection between two defined objects, an attachment window opens up beside the main window exhibiting the emitter and receiver ports of those objects.

Recorder A recorder is a dynamically defined window, which contains several sub-windows. After the user creates a recorder in run-time, he is allowed to define an infinite number of signals in the shape of channels. Each channel has a type and a name and is considered both an input and output object. If the channel is connected to a signal as in input, it records the data from that signal with time and is thus a recorder. On the other hand, connecting the same channel to a signal as output makes the channel a play-back device and thus could be used to execute recorded signals.

Switch-box
A switch-box is a dynamically defined window, which is derived from a connection between two objects. It enables the user a simple and direct access to important boolean signals in the signal matrix, without having to resort to the complicated and error-prone process of clicking through and managing the signal matrix.

UDP connection
This window represents a connection to an external device via Ethernet. It can be configured using a special configuration file containing the name

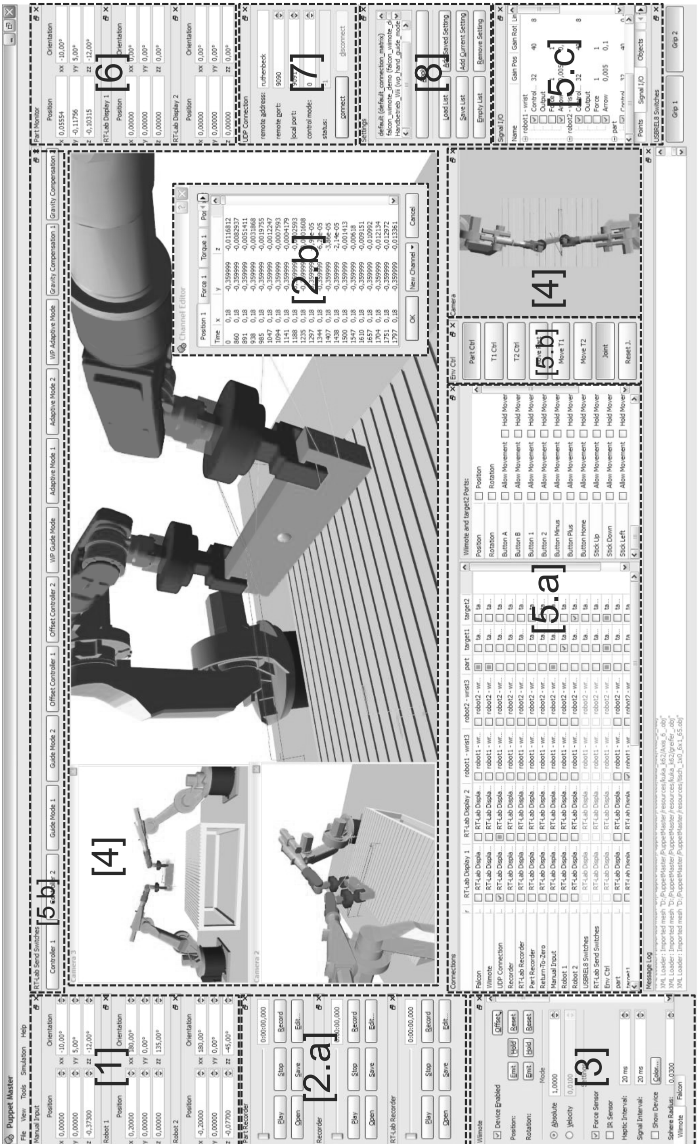


Figure 6.3: Screenshot of the PuppetMaster4D software environment

and type of signals exchanged between external devices and the software environment. An update of this file in run-time will automatically refresh the Signal Matrix to exhibit the changes in signals belonging to those devices i.e. I/O ports.

Manual input
The manual input is a dynamically defined window, with which the user can manually type in required posture data in the global coordinate system. It is considered a source, thus has no inputs.

Monitor The monitor is a dynamically defined window, with which the user can monitor the posture of an object in the global coordinate system. Hence it functions as a scope to display values in run-time. It is considered a sink, thus has no outputs.

Object I/O
This is a static window displaying all the inputs/outputs pertaining to all graphical objects defined in the environment. The definition thereof comes directly from the XML environment file loaded at the beginning. The user is also allowed to change the parameters of the joints (maximum and minimum values) and/or enable and disable position controllers pertaining to specific objects.

Settings window
The settings window is used for saving and activating a group of signal configurations. The settings could represent a group of connections for related actions, for instance the third phase in an assembly operation.

WiiMot settings
This is the graphical interface to the WiiMot which is connected to the software via *Bluetooth*. The WiiMot can be used as a remote control utilizing the divers buttons on it, or an input positioning device through its infrared position sensing capability in 3 DOF and its gravitational rotation sensor only in 2 DOF.

Falcon settings
This is the graphical interface to the Falcon haptic device which is connected to the environment via *USB*. The Falcon is a haptic device which could display forces in all three directions while simultaneously sending the position data. The forces displayed on it are not limited by the forces from the dynamic/haptic thread, but can also be read from an external device.

Digital I/O
This window is the graphical interface to a digital I/O relay card which is connected via *USB* to the PC. Since the card can only process digital signals

it sends and receives boolean signals. Such a device is conveniently used to interface process-specific accessories e.g. grippers.

Figure 6.3 shows a screenshot of the software displaying all the components defined in run-time for a cooperating robot application. The components are [1] Manual Inputs, [2.a] Recorders, [2.b] Channel Editor in a recorder, [3] HMI (Novint FALCON and Nintendo WiiMot), [4] Graphical views, [5.a] Signal Matrix, [5.b] Switch-boxes derived from the signal matrix, [5.c] Object signal I/O linked to the signal matrix, [6] Monitors, [7] External communication through UDP/IP, [8] Saved configurations for the signal matrix.

6.4 Signal management

6.4.1 Transparency prerequisite

As a requirement of any signal management scheme, transparency has to be ensured. Transparency in this sense means that at any arbitrary time during operation, the operator has access to all measured, observed or fed signals. In this regard, signals denote any values that are imperative for control (at any level) of the system. To do this, one must start by identifying the major signals in a robotic work-cell and accordingly design the signal scheme bearing in mind the latter prerequisite.

Position (r) This signal consists of 3 double values that indicate the position of a body w.r.t the world coordinate system.

Rotation (θ) This signal denotes the rotation of a body w.r.t the world coordinate system in the form of a rotation matrix requiring 9 double values.

Boolean (B) This signal can take on *true/false* values. It is mainly used to trigger or enable certain functionalities in a digital control manner.

Force (F) This signal consists of 3 double values that indicate the force acting at the body's center of gravity w.r.t the world coordinate system. This vector could be generated from the dynamic simulation of bodies or as real values from an FTS attached to the real objects. This also applies to the torque signal.

Torque (τ) This signal consists of 3 double values that indicate the torque at the body's center of gravity w.r.t the world coordinate system.

The latter signals can be easily expanded to include other time dependent or independent values specific to a class of devices e.g. laser sensors or multi-port pneumatic grippers.

6.4.2 Signal management scheme

Based on the transparency prerequisite and the decision to use the `Qt` framework for user-interface functionalities, a concept for sophisticated signal management was developed. Thus ensuring an expandable and bi-directional information flow between all components of the system. At its core the scheme employs the signals and slots concept from `Qt` (Nokia 2011a). Two objects were built upon this concept, namely; `emitters` and `receivers`. Emitters are a generic class representing outgoing signals from a component, hence it maps an output port for a specific type of signal. Receivers on the other hand represent incoming signals and therefore map an input port. By default, an `emitter` can be connected only to a receiver from its own signal type[1]. Built-in error detection prevents illegal connections which could result in false or unintended signal exchanges. Each component is assigned a number of `emitters` and/or `receivers` according to its function. They are either embedded in the definition of the component or are dynamically defined in run-time with the environment definition. Table 6.1 shows the components and their corresponding `emitter` and `receiver` ports.

Component	Emitters	Receivers	Dynamic/Static	User-defined
UDP Connection	r, θ, B, F, τ	r, θ, B, F, τ	static	yes
Graphical Object	r, θ, F	r	static	no
Joint	r, θ	r, θ	static	no
Manual Input	r, θ	-	static	no
Recorder	r, θ, B, F, τ	r, θ, B, F, τ	dynamic	yes
Switch-box	r, θ, B, F, τ	r, θ, B, F, τ	dynamic	yes
Monitor	-	r, θ	static	no
WiiMot	θ, B	B	static	no
Falcon	r, B	F	static	no
Digital I/O	B	B	static	no

Table 6.1: Components and their assigned emitter and receiver ports

It is worth noting that during run-time the number and type of `emitter` and `receiver` ports could be fixed (static) or could be configured by the user (dynamic). The user interfaces this management scheme through a signal matrix which is continuously updated in run-time. As shown in Figure 6.4 it could be considered a skewed representation of connections in an interconnected block diagram. This window consists of all existing objects at any given time as rows and columns. The rows exhibit all objects with `emitters` while the columns exhibit all objects with `receivers`.

[1]Even if the signals from a data-type point of view are identical: for instance position and force signals

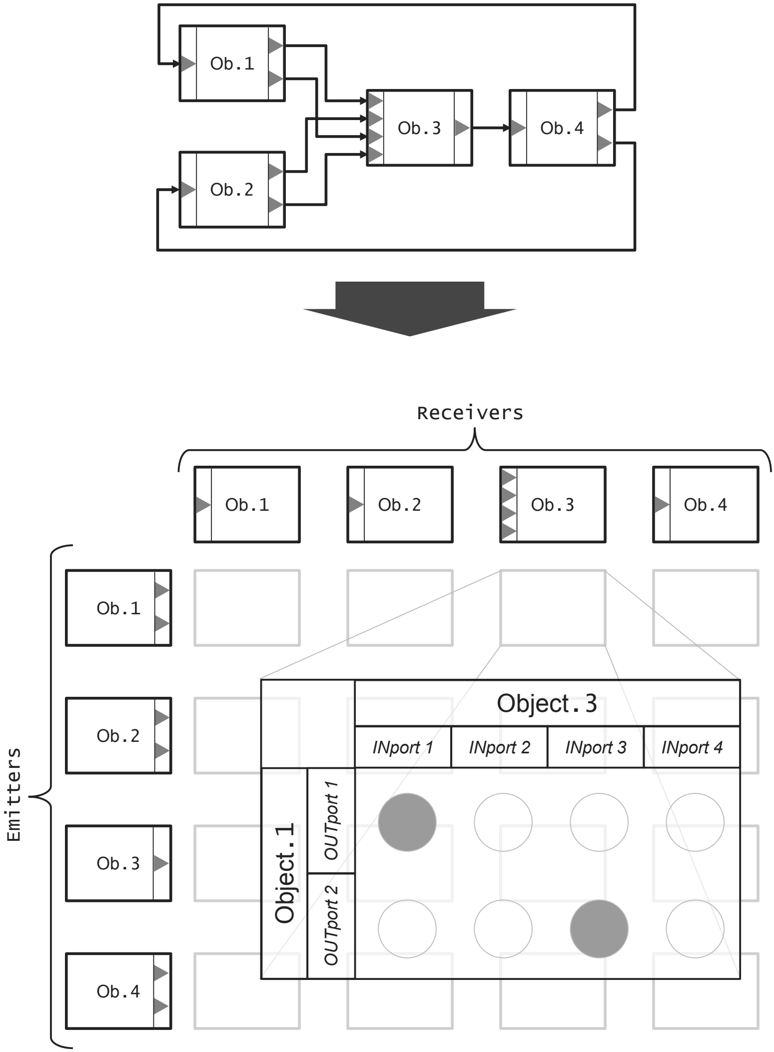


Figure 6.4: Mapping the inputs/outputs signals of interconnected blocks from a block diagram scheme to a signal matrix scheme

Since an object can contain both emitting and receiving ports, it is common to see the same object both in the rows and the columns. By clicking on the intersection between a row and column another matrix-like window pops-up (*channels*) with a check-box corresponding to each emitter port. For instance, if the user wishes to establish a connection between the position from object no.1 to object no.2 in run-time, he only has to tick the check-box corresponding to such a connection in the *channels* windows. It is important to note that, a port can send its signals to several other ports, but it could only receive from one port. However, the signal matrix becomes very complicated and increasingly error-prone for large numbers of objects. Therefore, a signal configuration saving and loading mechanism through the settings window as described in section 6.3.4 is available. Figure 6.5 shows a real example of the proposed mechanism. In the top configuration, the *robot1-wrist3* object -which is a graphical representation of the robot- receives its position and orientation from an external source (the *UDP Connection* object), while in the bottom configuration, the *robot1-wrist3* object receives its positions and rotations directly from manual user input (the *Manual Input* object)

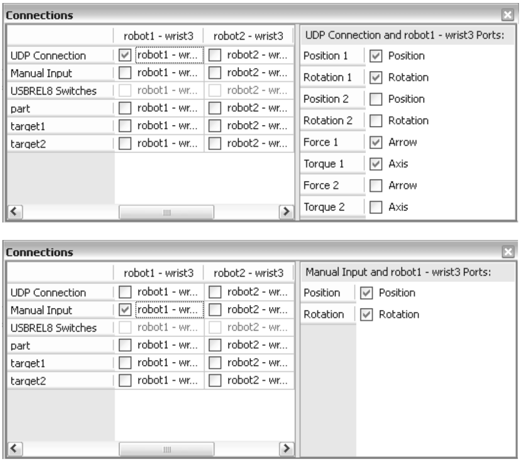


Figure 6.5: An example highlighting the flexibility of the signal matrix

6.4.3 Control module interfacing

To reduce the complexity of interfacing the control functionalities it is imperative to redesign the interfaces between the system components. Since hardware interfaces are usually specified in advance by the vendor, they remain to a large extent unaltered during design and run-time. On the other hand, software and control components depend on the required functionality, architectural and structural design of the system. To accommodate different needs, two types of interfaces are provided:

Signal/parameter interface

Such an interface facilitates a direct and low-level access to the inner-most control loops and its accompanying parameters (Figure 6.6). This gives the operator unrestricted power over the workings of such laws, such that the user dictates the parameters of a certain control law according to the needed response and hence directly influences the outcome of the measured signals. Although this could be preferable for experimenting with control strategies and tuning their parameters, it nevertheless burdens the operator with many low level control decisions which are not of real interest to him.

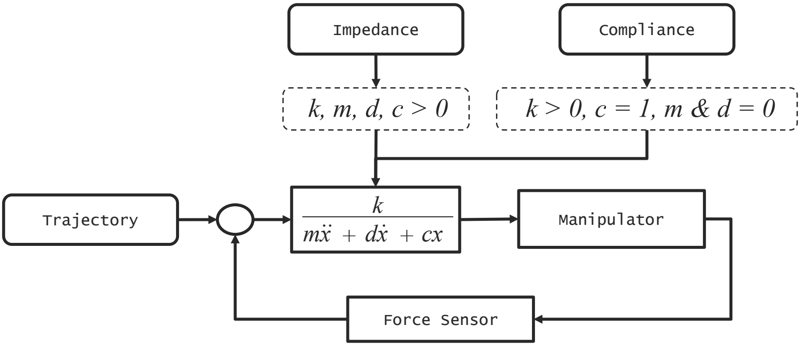


Figure 6.6: Interfacing the control laws through changing the parameters

Switch interface

This interface encapsulates the signals and parameters in blocks which are already tuned and ready to be used (Figure 6.7). Although very beneficial from the user's point of view, it may lead to redundancy in the inner construction of the interface. To ensure a similar level of functionality as that of the latter interface and without threatening the abstract nature of the blocks, a back-door to a limited number of parameters is provided. Hence, an operator is not restricted by the default values of the parameters but is capable in a limited capacity to change them if sufficient knowledge is available.

The functionality in both interfaces does not significantly differ, but their architectural structure and how the human operator or any other intelligent entity perceives them marks how contrasting they are. What makes the second interface more appealing for a human operator is the ability to abstract the required action for a predefined output. Instead of

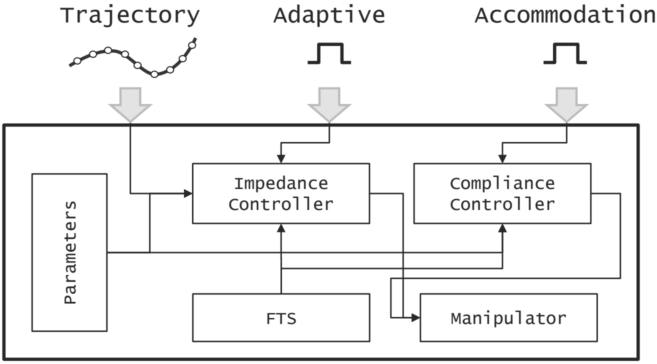


Figure 6.7: Interfacing the control laws through triggering predefined blocks

worrying about the inner workings of the underlying control loops, for instance setting the controller parameters, it defines the controller in term of what the required functionality is from the system. Thus the user only has to trigger or activate/deactivate one or more blocks to achieve a certain functionality. On the other hand, the first interface allows an adaptive parametrization scheme as required different parts of a process. Which is readily consistent with the parametrization scheme mentioned in section 5.3. Thus providing an adequate interface for artificial intelligent units to manipulate the behavior of the control structure through parameter manipulation.

6.4.4 Configuration matrices

Configuration matrices are based on the motion and force requirements for each phase in a cooperative task. The terms stem from the fact that the requirements translate into connections between signals, which is represented by a matrix-like structure (as seen in Figure 6.4). Two sets of configuration matrices are defined here:

Programming Configuration Matrix (PCM)
This configuration matrix sets the control modes during programming of a task. The user influences this by deciding certain actions.

Runtime Configuration Matrix (RCM)
This configuration matrix sets the control modes when a task is in play-back mode i.e. during execution of the task. In this case the user can't enable or trigger any modes (can't influence them) i.e. automatic operation.

The difference between the two matrices depends on several factors. Chiefly is the number of phases in any given task and how variable the force and motion requirements are. This increases the complexity of both the programming and run-time configurations significantly. Furthermore, the flexibility of the system encourages the operator to combine several programming methods together by using functionalities inherent to specific run-time phases during programming. Naturally, such hybrid methods tend to proportionally increase the deviation between the two configurations.

6.5 Device integration

In this section, two classes of device integration will be discussed. The first is the HMI e.g. positioning and haptic devices, while the second is concerned with connecting to external devices through a communication bus. Both rely heavily on the design choices and the implementation details of the software and the accompanying real-time platform.

6.5.1 Human machine interface

The term HMI is used to describe any interaction between a human and a machine during a certain task, literally when humans and machines meet (About.com: Ergonomics 2011). This includes everything from input devices, switch panels to graphical user interfaces and touch screens (Georgia Institute of Technology et al. 2009, p. 82). Human Robot Interaction (HEI) is a related field of study which is dedicated to designing and evaluating interfaces specifically between robots and humans (Goodrich & Schultz 2007)(Reinhart et al. 2010a). Consequently, in this section the discussion is limited to devices that allow the operator to control/navigate the robots in an intuitive and simple manner. Their main advantage is their simplicity, thus no prior tutoring to operate them is necessary. However, they do require some familiarity to be operated efficiently, which mainly depends on the type and structure of the device. The device classes addressed here are divided into:

Positioning Devices
These are input devices that measure the change in their relative position and/or the orientation when the user operates them. They may contain joints and hence exhibit a mechanical structure. Measurement of those variables could be done by two methods, either by independent sensors or calibrated sensors. Gyroscopes for rotational movements and optical encoders for translational movements are common examples of independent sensors usually built in commercial devices (Pongrac et al. 2007)(Neto et al. 2009). Calibrated sensors on the other hand require calibration with external components. Laser and infrared tracking devices are common examples which require external calibration (Hein et al. 2008)(Brecher et al. 2010).

Haptic Devices

Haptic devices can also be considered a subset of positioning devices. The main difference though is that the device acts not only as an input device but also as an output device (Mali & Munih 2003). By incorporating motors in the device, forces and/or torques could be conveyed back to the user in real-time. This complicates the mechanical structure and construction of the device but greatly enhances the user experience when interacting directly with a virtual environment or a physically remote environment (Massie & Salisbury 1994)(Stone 2000). Additionally, it can be used for assisting the operator during off-line programming with haptic cues (Wörn et al. 2005).

To accommodate different classes of devices, an abstraction layer is developed based on the `CHAI3D` device abstraction class; namely `cGenericDevice` (Conti et al. 2011b). As seen in Figure 6.8, each device is imparted to a certain number of inputs and/or outputs. Conventional input signals are for instance position and rotation for both positioning and haptic device classes. On the other hand, some output signals such as forces, are exclusive only for haptic device class. After investigating different devices, other common inputs and outputs were incorporated in the design of the abstraction layer, for example buttons and LEDs. Those inputs/outputs imparted to a certain device are mapped into the signal matrix as ports specific to this class of device.

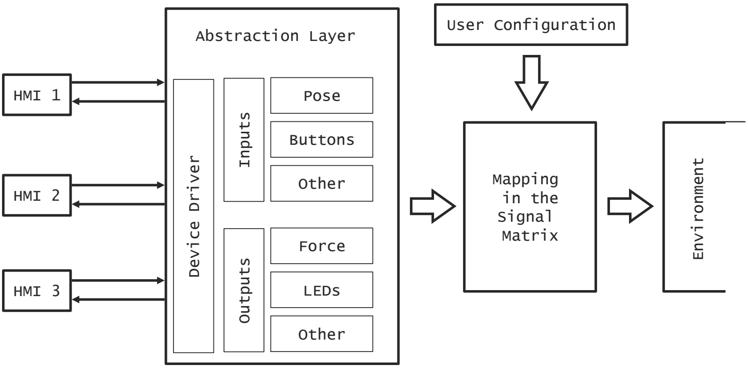


Figure 6.8: HMI abstraction layer

Consequently, defining an object of this class automatically generates the input/output ports for it in the signal matrix. Characteristics for each device are also available in a separate dockable window, from which the user could configure the device accordingly.

For instance the scaling factor of the position signal emitted from a positioning device or parameters of the emitted signal of the force feedback from a haptic device can be directly configured. Two major issues have to be addressed when HMI and especially those including haptic devices are developed; workspace scaling and DOF mapping (RADI ET AL. 2010).

6.5.1.1 Workspace scaling

Due to the restricted mechanical structure of HMI, their effective workspace is usually smaller than that of the manipulators they control (CONTI & KHATIB 2005). This imposes certain geometrical constraints during operation rendering the deployment of HMI useless without adequate workspace scaling techniques. This issue has been extensively studied in the literature and especially in the scope of telepresence applications for both downscaling and upscaling and even mobile manipulation (ZÄH ET AL. 2006)(SCHAUSS ET AL. 2009). The two most common scaling techniques are implemented in the software environment (MALETT ET AL. 2004)(CONTI & KHATIB 2005)(REINHART ET AL. 2010a):

Position navigation

In this arrangement the position signal emitted from the HMI x_{HMI} is interpreted as a scaled position signal x_{output}, where the scaling factor K_p is a run-time variable.

$$x_{output} = K_p.x_{HMI} \tag{6.1}$$

During operation the user presses a button. When the end of the HMI's workspace is reached, the user releases the button and moves the device back to its initial position. Subsequently the button is pressed again and the movement can ensue. This method is similar to the scrolling movement of a computer mouse and is termed *indexing*. Hereby, the index of the device is continuously altered to correspond to the current manipulator position.

Velocity navigation

In this arrangement the position signal emitted from the HMI is interpreted as a scaled velocity signal $\dot{x}_{output}$, where the scaling factor K_v is also a run-time variable.

$$\dot{x}_{output} = K_v.x_{HMI} \tag{6.2}$$

The device in this case behaves like a gas pedal; the more the user steers the device in one direction the quicker the manipulator moves in that direction. Hence, the workspace of the device becomes theoretically infinite. However this method requires calibration to prevent the manipulator from drifting at small values.

Both methods are implemented for each device, making it possible to change the scaling factor and the method of choice in run-time. It has to be mentioned that in the case of a haptic device, scaling is not only limited to the workspace but also to the forces conveyed back to the user. This could in many cases lead to instability issues especially if the time delay in the total control loop is significant (Golle et al. 2003). However, the framework provides the capability of integrating model-based feedback methods or prediction based methods to overcome those issues (Clarke et al. 2006)(Schillhuber & Ulbrich 2008).

6.5.1.2 Degree of freedom mapping

It is not uncommon to use an HMI device with a certain mechanical structure and hence certain DOF to control an object or a manipulator with different DOF. Hence the DOF of an HMI $\Lambda_{HMI} \in \mathbb{R}^l$ have to be mapped to those of the manipulated object $\Lambda_o \in \mathbb{R}^6$ or directly to the manipulator $\Lambda_{rob} \in \mathbb{R}^n$. Several mapping configurations are given in Table 6.2.

HMI DOF	type	Mapping Configurations
2	T	planes X-Y / Y-Z / Z-X
3	T	translation in space XYZ or rotation in space ABC
3	R	rotation in space ABC
6	T+R	translation in space XYZ and rotation in space ABC

Table 6.2: DOF mapping configurations for HMI

Figure 6.9 is an example for DOF mapping for a 6 DOF SENSABLE Phantom (Massie & Salisbury 1994)(Sensable Technologies Inc. 2011) and a 6 DOF KUKA robot (KR 6/2). Since both possess the same DOF, the mapping is relatively straight forward. The device's TCP frame ($\mathcal{T}_{Device}$) w.r.t. its global frame ($\mathcal{W}_{Device}$) is scaled and mapped to the manipulator's TCP frame ($\mathcal{T}_{Manipulator}$) w.r.t. its global frame ($\mathcal{W}_{Manipulator}$).

6.5.2 External communication

To communicate with manipulators, sensors and an RTP, external means of communication are required. The most common method for communication would be a field-bus e.g. ethernet or CAN-bus. For communicating logic signals with process-specific devices or peripheries such as grippers, a digital I/O card would suffice. Both means of communication are implemented in an extensible and intuitive manner in the software environment.

6.5.2.1 Ethernet

Industrial buses are a common characteristic of modern industrial plants. Their standard plug interfaces and underlying protocols make them an attractive technology to connect

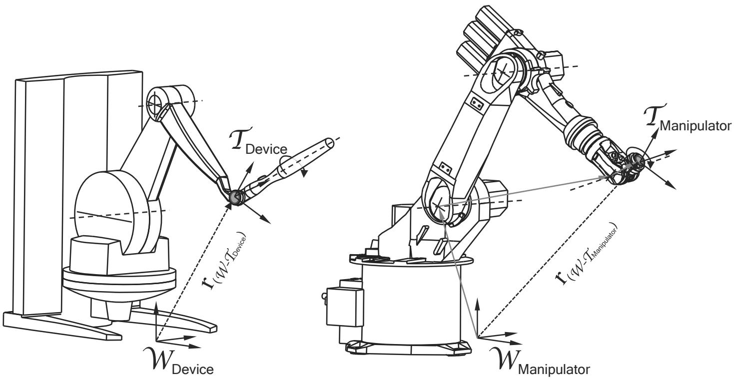


Figure 6.9: Mapping the world and TCP coordinates of a haptic device to an industrial robot

different components from different vendors. To give the environment the flexibility and power to be a central component in the design and deployment of the work-cell, ethernet (IEEE 802.3 ETHERNET WORKING GROUP 2011) was chosen for external communication involving all kinds of data. An ethernet communication scheme could be directly based on the available computer ethernet infrastructure. Thus reducing the cost and time of installing new cables. However, ethernet is not without its own drawbacks. Most important thereof, is its inherent lack of real-time data transfer capability and the long packet switching time[2]. These aspects are accounted for in this work by separating the real-time functionalities and deploying them directly on the real-time platform.

6.5.2.2 Digital I/O

The second type of communication relates to digital signals. Although most modern industrial devices are outfitted with buses, some devices only require direct interfacing through a digital I/O port. The only type of signals that can be connected to this interface are naturally the boolean signals. Provisions for direct connection with stand-alone cards which are not part of the RTP are also taken into consideration. It is important to point out that this doesn't breach the real-time capability of the whole system, since those signals mostly do not affect the real-time operation. For instance controlling the gripper is not a real-time critical operation except if total failure occurs.

[2]This is valid only when utilizing TCP/IP or similar protocols

6.6 Work-piece based virtual environment

To implement the work-piece based approach, a virtual world corresponding to the actual world is created in the software environment. Subsequently both worlds, virtual and real are connected through the real-time platform (will be discussed in section 7.2.4). Given the flexibility of the software, two different virtual worlds are created. As shown in Figure 6.10, the first configuration generates the reference trajectory of the robots through two graphical objects (targets). These are connected with fixed joints to a graphical representation of the work-piece which in turn is controlled by the user through a HMI. Given the relative posture of the target objects to the work-piece object is known, the dynamic engine in the software environment calculates the respective posture of the graphical objects. Subsequently, these positions are directly sent to the real-time platform through a UDP connection component. In the second configuration, the trajectory of the work-piece is sent to the real-time platform directly. This however requires a robot trajectory generator on the platform, which calculates the trajectories of the respective robots. This variation in implementation highlights the flexibility of the software environment and how it is capable of realizing a given approach in a multitude of ways.

6.7 Summary

In this chapter, a software environment was developed representing the technical realization of the software module in the framework. The software assists the user in different capacities during all phases of manufacturing; as a production planner, a robot programmer and a work-cell operator. The software was developed around several open source engines hereby providing simultaneous graphic/dynamic/haptic simulation capabilities. To facilitate description of virtual environments and accompanying signals, an easy-to-understand XML data structure was developed. Additionally, a component based GUI was designed to allow the user to dynamically define graphical components according to the task's needs. Given the large number of signals in a CIR system and the level of flexibility dictated by the work-piece based approach, a powerful but nevertheless flexible signal management scheme was designed and developed. This gives the user unprecedented control over all signals flowing through the system in run-time. Moreover, it represents a continuation of the control architecture in the latter chapter through tight integration with the high-level control layer by providing a reconfigurable control interface. The environment also provides provisions for interfacing external devices whether they are HMI devices, sensors or additional platforms. For intuitive HMI devices an abstract interface was developed and extended to scale the workspace of the devices and map their DOF to the virtual environment and subsequently to the real work-cell. Additionally, interfacing external components was made simple through utilizing a standardized ethernet connection or an industrial input/output signals.

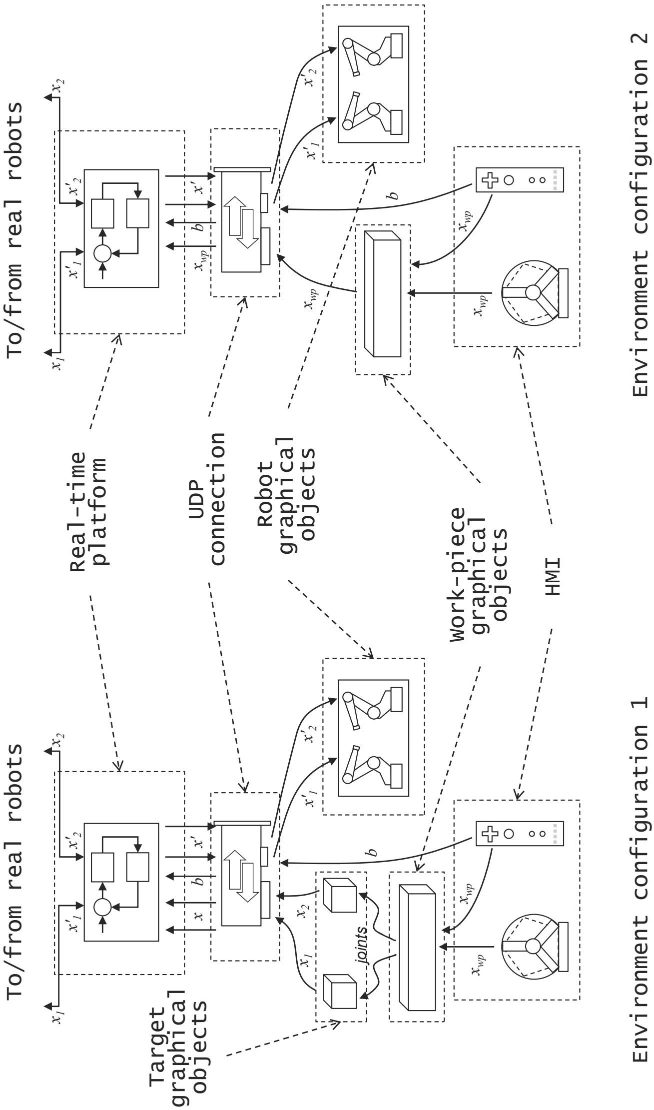


Figure 6.10: Two different configurations for the software environment featuring the WPBA

7 Experimental Test-rig

To invent, you need a good imagination and a pile of junk

THOMAS A. EDISON

7.1 Overview

In this chapter an experimental setup implementing the work-piece based approach will be presented. The technical realization of the first class of components `[functional]` in the framework section 4.4 which included the **control** and **software** module were discussed in chapters 5 and 6 respectively. Due to their heavy reliance on the type and characteristics of the available hardware, the remaining two classes, `[device]` and `[communication architecture]` will be presented here. Thus this chapter explicitly addresses the **sensors module**, **robot module**, **HMI module**, **task-level peripheries** in section 7.2 and the **communication architecture** in section 7.3. The design and development of the test-rig is dictated by a set of requirements which takes into consideration the motivation and objective of this work. Those requirements are listed as follows:

1. To retain the generic nature of this work and to guarantee porting of the methods introduced here to similar robotic work-cells in an industrial setting, only commercially available and off-the-shelf components are selected and used.

2. Since robot manufacturers tend to dictate the type and extent of functionalities available to the user, it was imperative to abstract these interfaces to build control functionalities independent from the commercial controller.

3. Maintaining the design as flexible as possible in order to simplify adding or removing hardware or software and components. Regarding hardware, this meant depending more on PC-based infrastructure to avoid being hampered by different vendor-specific interfaces. Regarding software, most of the user interface is generated in run-time hence it is not hard-coded. Furthermore, those interfaces are subsequently saved in configuration files to avoid repetitive initialization sequences.

4. To prevent hogging of the communication bandwidth and to avoid choking of data at critical locations, dedicated hardware and compact efficient data representation is utilized.

Although those requirements dictated certain design decisions, they nevertheless allowed a continuous improvement of the overlapping aspects of the test-rig. Thus the test-rig was being continuously improved in terms of hardware and communication in order to retain the design concept represented by the requirements.

7.2 Device components

7.2.1 Robot platform

The robotic platform consists of two KR6/2 industrial class robots from KUKA connected to a KRC2 industrial controller from the same manufacturer as shown in Figure 7.1 (KUKA GmbH 2011a)(KUKA GmbH 2011b). Although weighing about 250 Kg each, they have a payload of 6 Kg at a maximum TCP velocity of 2 m/sec, hence their application is limited to pick and place small components or non-contact tasks. The robots are mounted on thin steel plates which in turn are mounted on a custom-made aluminum structure. To avoid any unwarranted vibration during motion, the steel plates are diagonally reinforced from underneath. One of the major advantages of such a structure, is eliminating the need for re-calibration of the whole test-rig during transport. Mechanical stoppage are mounted on the first and second axes of the robots on both sides limit the motors' motion. This is supplemented by software limit switches to decrease their workspace.

7.2.2 Sensors

For force sensing and monitoring, two FTS supplied from SCHUNK are utilized. They are designated as FTC-50-80, where FTC stands for Force-Torque-Compliant (Schunk 2011). They are capable of measuring the forces and torques in 3 translational and 3 rotational DOF. Their compliant nature comes from their internal design, where springs are used to measure the loading while elastically deforming (up to 1.0 mm and 1.4^0 in all translational and rotational directions respectively). Thus they are adequate for executing constrained tasks such as assembly. For convenience, the sensors are mounted such that their coordinate system corresponds to the coordinate system of the robots' TCP. Therefore, simplifying the conversion from one coordinate system to the other. The maximum force that could be measured in all directions is around 300 N. On the other hand the maximum torques in the lateral directions is 7 Nm, and in the perpendicular direction is 15 Nm. An internal controller converts the analogue signal to a digital one and sends it over a CAN-bus[1] with a Baud rate of up to 1 MBaud.

[1]Controller Area Network according to the ISO 11898:2003 standard

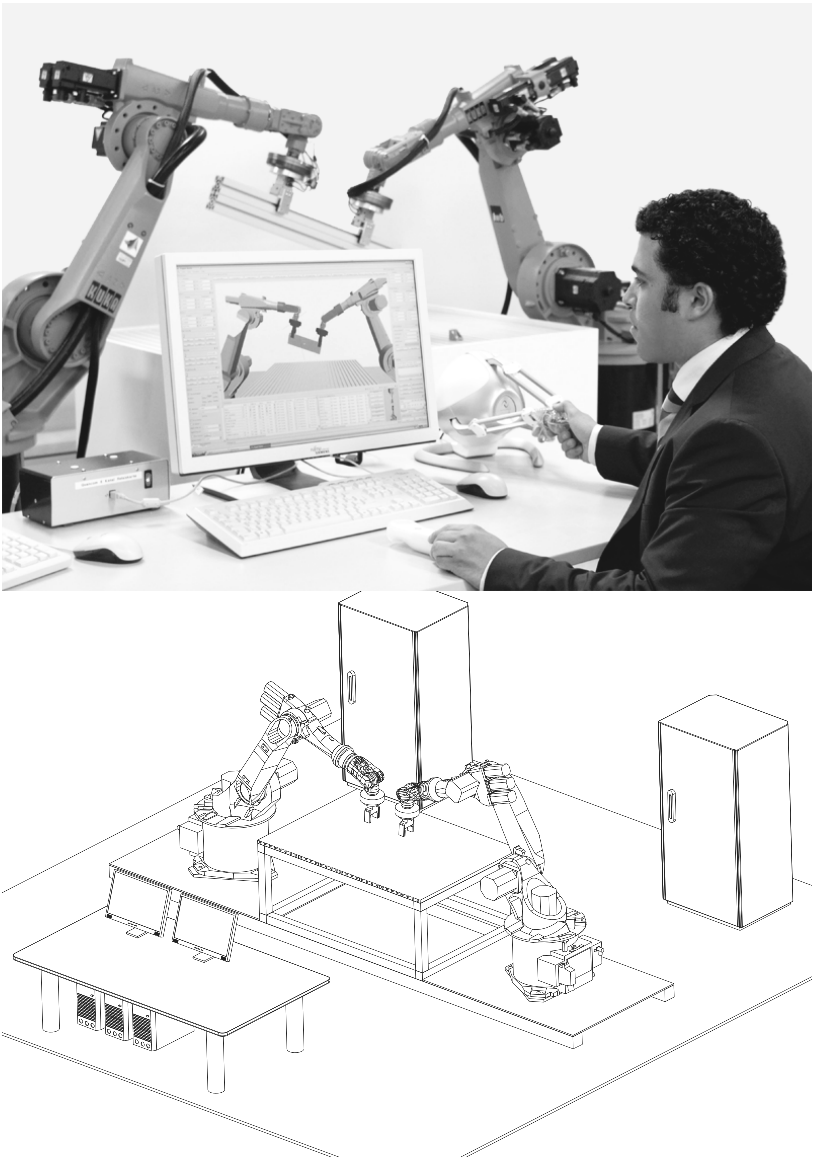


Figure 7.1: The experimental test-rig

7.2.3 Task-level peripheries

Two pneumatic two-finger grippers from *SCHUNK* are mounted on the robots' TCP. They are actuated by a 24 Volt relay controlled by digital relay card from QUANCOM USBREL8 (QUANCOM Informationssysteme GmbH 2011a) connected to the PC running the software environment with a USB connection. Since the digital I/O are independent of any bandwidth control, it was safe to override the RTP and connect it directly to the software environment. The grippers were assigned dedicated boolean signals from the environment and accordingly mapped to the signal matrix.

7.2.4 Real-time platform

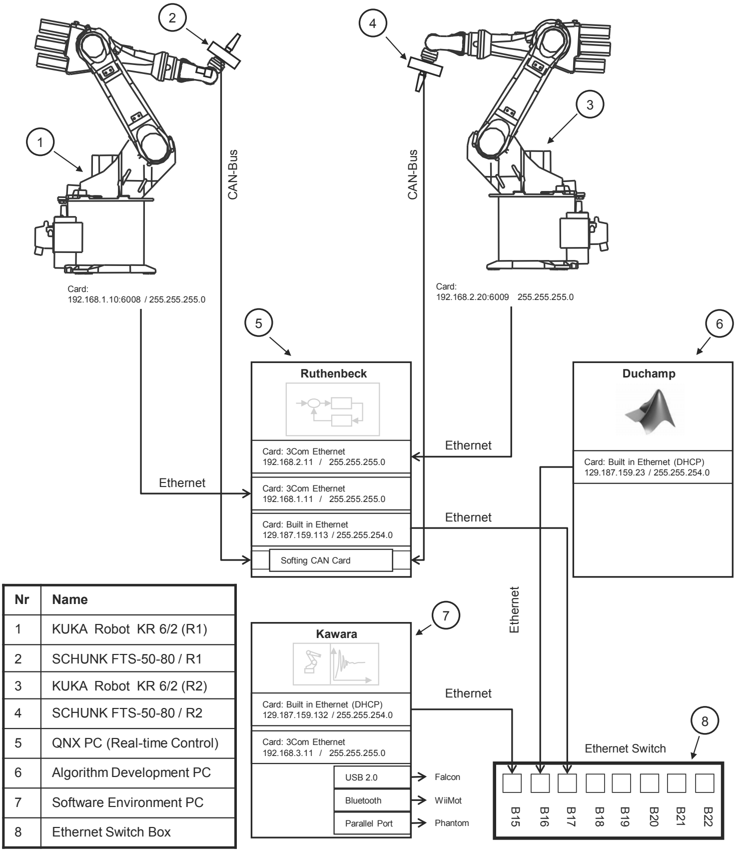
In this set-up, the real-time platform (RTP) consists of a PC running Microsoft Windows for developing all high bandwidth control algorithms and a PC running the RTOS `QNX` (QNX Software Systems 2011) for code execution and communication with hardware actuators and sensors. The arrangement in this work is similar to an Hardware-In-the-Loop configurations commonly utilized for testing and validating control algorithms in automotive applications (Sanvido 2002)(Niedermayr 2005)(Pretschner et al. 2007). On the first PC (control development) control algorithms and other functionalities are developed on *Matlab-Simulink* (MathWorks 2011) in tandem with `RT-LAB` (OPAL-RT Technologies Inc. 2011). `RT-LAB` is a third party software which ensures smooth and quick deployment of a RTP in HIL arrangements as well as rapid prototyping of real-time control algorithms. It is installed on *Matlab-Simulink* and interacts directly with the Real-Time Workshop toolbox. Upon completion of the model design phase, it generates C/C++ code corresponding to the model which is subsequently modified by `RT-LAB` to suit the RTOS and is downloaded on the target platform running `QNX` . A daemon[2] running on the RTP receives the code and compiles it. To start execution, the operator remotely executes and monitors the code on the target platform using remote access tools built in `RT-LAB` .

7.3 Communication architecture

A detailed description of the communication architecture is shown in Figure 7.2. The RTP represents the central communications hub (Component 5). Given that the robot interface (termed RSI and will be discussed in section 7.3.1) dictates a client-server communication model, where the robot controllers are the clients and the external controller is the server, it was decided upon to centralize the model. Thus the RTP serves as the server side, while all other devices communicating with it are clients. In order to guarantee the connection constraints imposed by the RSI interface and the bandwidth constraints imposed by the TCP/IP protocol, each robot (Component 1 and 3) is assigned a dedicated ethernet card

[2] A background process

on the RTP. Another ethernet card is used to communicate with a central ethernet switch (Component 8) which is connected to all the other PCs in the system (Component 6 and 7). A PC CAN-bus card with two ports is used to communicate with the FTS mounted on each robot (Component 2 and 4). The simulation environment running on a PC (Component 7) communicates with the RTP using the UDP/IP protocol via the ethernet switch (Component 8).



Nr	Name
1	KUKA Robot KR 6/2 (R1)
2	SCHUNK FTS-50-80 / R1
3	KUKA Robot KR 6/2 (R2)
4	SCHUNK FTS-50-80 / R2
5	QNX PC (Real-time Control)
6	Algorithm Development PC
7	Software Environment PC
8	Ethernet Switch Box

Figure 7.2: Communication architecture

7.3.1 Robot interface

The robots are outfitted with I/O cards to allow direct access to the positional controller in real-time. A vendor-specific interface termed KUKA Ethernet RSI (Remote Sensor Interface) communicates internally with the controller's hardware and externally with the real-time platform through an ethernet connection. This external connection is based on an XML data structure[3] which has to be exchanged between the robot controller and the external platform in less 12 ms which represents the communication cycle time. As shown in Figure 7.3 the cycle starts when the external controller sending a data structure containing the reference data to the robot controller. These values could either be correction values for the posture or the correction values for the joint angles. Since these values have to be sent, processed and received in a specific time interval, they could be considered as velocity commands. Given that the RSI interface uses the TCP/IP protocol, it is guaranteed that the packets sent are received on the other end. In addition to those reference values, a variable integer value (IPOC) is also sent to the robot which denotes the number of the internal processing cycle in the robot. During 4 ms from the start of the cycle, the structure has to be completely received by the robot (x to x+4). If the sending operation was successful, the robot processes those correction values and drives the motors to execute the required motion in the second phase i.e. the second 4 ms period. Subsequently, the robot increments the IPOC variable and according to the users settings sends it back in another XML data structure containing the actual posture and joint angle values in the last 4 ms. Hence, the total cycle time amounts to 12 ms. Any delay in any one of the 4 ms phases immediately causes the connection to be terminated. As the IPOC variable represents a type of hand-shake protocol, a false or delayed data structure would also cause the interface to terminate the connection due to corrupt data. Although the latter data exchange mechanism uses the normal TCP/IP protocol, the connection could safely be considered real-time, given the nature of the time constraints and the IPOC handshake technique already described.

7.3.2 Asynchronous communication

According to timing between two communicating nodes, two categories of transmission could be distinguished; synchronous and asynchronous (Stallings 2006, P. 182). In synchronous communication the two nodes send and receive data at the same rate and in theory are perfectly harmonious, while in asynchronous communication the rate is different (Bates & Gregory 2006, P. 256). Connecting different hardware devices to each other will usually result in asynchronous data exchange. Variable time latencies occur between the different components whether they are software or hardware. Those latencies

[3] This is independent from the XML data structures used in defining the software environment

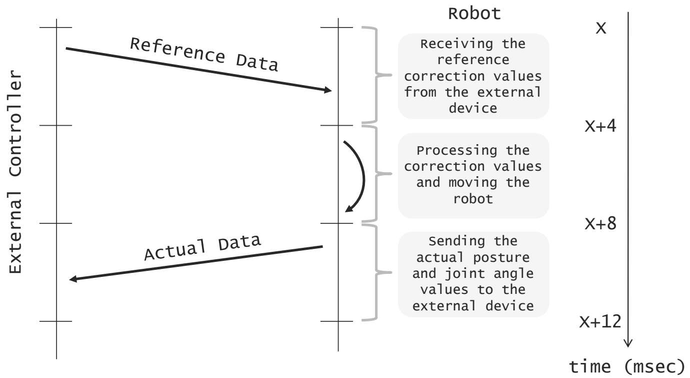


Figure 7.3: Phases in an RSI communication cycle

depend on the type, speed and underlying communication protocols of the connections. Most of the ethernet connections to the RTP were asynchronous due to:

- the variant nature of devices connected to the RTP, hence each device has its own sample time and bandwidth requirements as given in table 7.1.
- the unconstrained nature of the ethernet connection and its underlying protocols i.e. no time constraints apart from the bandwidth of the infrastructure

A common mechanism for implementing interprocess communication on an RTOS makes use of the shared memory principle (QNX Software Systems 2011)(Stevens 1999, P. 303-305). This mechanism is inherently suited for asynchronous communication due to its independence from time constraints (The Open Group 1997).

Device	**Bus**	**Freq. (Hz)**
Real-time platform	Real-time control loop	250
Force/torque sensor	CAN-bus	500
Software environment	Ethernet (UDP/IP)	100
Robot RSI-interface	Ethernet(TCP/IP)	83

Table 7.1: Sample rates in the control loop

7.3.3 Data Transfer Rate

One of the main issues encountered while designing complex robotic systems is the data transfer rate through communication buses. Aside from the CAN-bus connection between the FTS and the RTP, ethernet is used to connect most components with each other. As already discussed in section 7.2.1 the connection between the robots and the RTP is dictated by vendor specific time and data constraints. Furthermore, the connection between the algorithm development PC and the RTP is only functional during design time and not during runtime as already covered in section 7.2.4. Based on the latter facts the only connection not dictated by hardware constraints lies between the RTP and the simulation environment, which is of paramount importance during run-time. Owing to the high-level control role the software environment plays in run-time, data flow must be ensured at all times. Although an ASCII-based XML structure for data representation was contemplated, it was bypassed for the more communication-efficient binary-based representation. The data transfer rate could be readily determined by analyzing the signals transferred between the components. According to Figure 7.4, the signals sent from the RTP to the software environment need a total of 200 `bytes` , while those flowing the way other way round need 104 `bytes` .

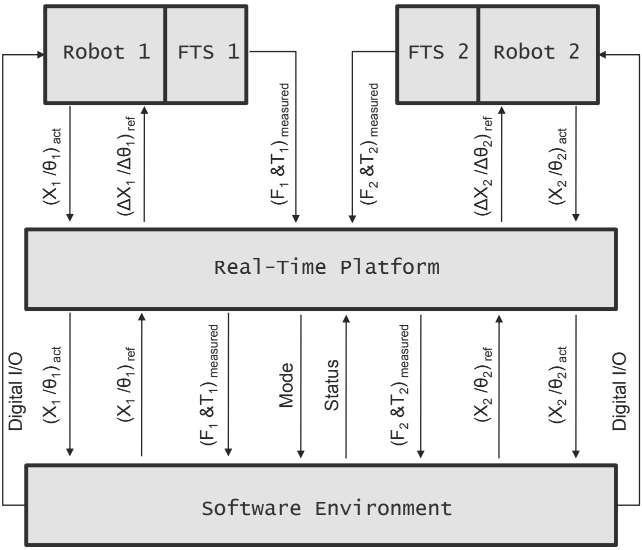


Figure 7.4: Signal architecture

Additionally the UDP and IP protocol headers require 8 and 20 bytes respectively and also an additional number of bytes from the physical layer (Postel 1980)(Postel 1981). This brings the total to about 400 bytes in both directions. Thus the size of the packet is approximately 2000 bits in each direction. To ensure that the robot actuation loop (i.e. the Robot RSI-interface in Table 7.1) runs smoothly, the above amount of information has to be exchanged with a minimum of 83 Hz (12 ms). Hence, the maximum data rate would be 2000 Bits X 83 = 166 Kbit/sec, which is clearly under the allowable rate of the underlying ethernet infrastructure (around 1 Mbit/sec). A UDP send/receive test between the RTP and the simulation environment showed that for the simulation environment to run at the same rate of the real-time control loop (compare Table 7.1), a maximum time latency of 4.55 ms and an average of 2.44 ms is to be expected.

7.4 Summary

In this chapter, an experimental test-rig implementing the work-piece based approach was presented. The emphasis was on the two remaining component classes in the framework, namely the device components and the communication architecture. A robot platform consists of two industrial robots fitted with sensors and grippers on their respective TCP was built. A real-time platform was utilized to execute all real-time functionalities in the control architecture. The communication architecture consists of several different bus connections according to the available hardware and interfaces. The experimental test-rig proved capable of handling the data exchanged between the different components, hereby fulfilling the required flexibility and functionality of the proposed approach.

8 Assessment

An ounce of practice is worth more than tons of preaching

MAHATMA GANDHI

8.1 Overview

In this chapter the implementation of the WPBA on an industrial test-rig will be assessed from technical and economical point of views. The validity of the proposed approach will be highlighted by several experiments outlining its capabilities. Subsequently a qualitative assessment based on evaluating the approach w.r.t. production-specific, programming specific and robot-specific criteria is introduced. Finally, a business case is made for deployment of the approach for CIR through an economical case study.

8.2 Experimental validation

Beside spraying and welding, material handling is considered one of the major applications of robotic-based production. This not only includes the classical class of pick and place operations, but extends to include assembly and mounting operations. Given their importance to industrial production, these classes of applications will be represented in the experiments through three programming methods for three different cooperative tasks. Each task serves to highlight a certain aspect of the proposed approach and hence its applicability for a wide range of cooperative tasks with different programming paradigms (Table 8.1).

Exp.	**Programming technique**	**Feature**
1	On-line	intuitive human robot interaction
2	Off-line	sensor-based on-line adaptation
3	Autonomous	intelligent event-based control

Table 8.1: Basic programming technique and features of each experiment

The first experiment is derived from the traditional lead-through on-line programming technique by allowing the programmer to manually guide both the robots *and* the work-piece (refer to section 2.2.1.2). On the other hand, the second experiment utilizes force sensing and control to automatically adapt the code generated in an off-line simulation environment (refer to section 2.2.2.2). The third experiment employs an event-based controller for autonomous assembly to demonstrate the intelligent layer of the control architecture discussed in section 5.6. Deployment of the WPBA to any cooperative tasks entails going through three distinctive phases (Figure 8.1). The first phase is configuration, where the programmer analyzes and defines the cooperative task (refer to section 4.2). Based on this analysis the phases and their respective requirements can be accordingly derived.

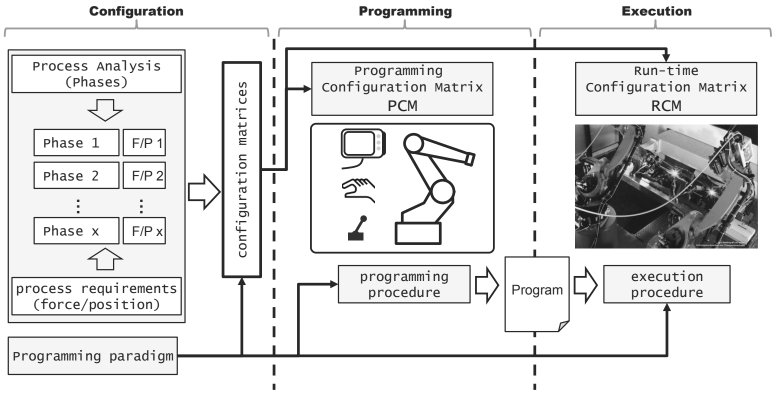


Figure 8.1: Experiment description overview

Process requirements are vital to specify which control structures should be activated during the process phases and the corresponding control law parameters. These are translated into two configuration matrices as detailed in section 6.4.4. However, these configurations are not only determined by the process phases but are mainly influenced by the programming paradigm which the operator dictates during configuration. In the second phase, the robot programmer can utilize the PCM to program the task according to the dictated programming paradigm. In order to successfully utilize this, the programmer has to comply with a programming procedure designed by the operator and tied to the PCM. The outcome of this phase is an instruction list i.e. a robot program, which could be executed in the third phase by applying the RCM. Although program execution doesn't normally entail any human intervention, this is sometimes necessary during execution. Hence an execution procedure similar to that of programming could be designed specifically for such cases.

8.2.1 Hand-guided programming

8.2.1.1 Overview

One of the most popular on-line programming methods as outlined in section 2.2.1.2 provides the user with the ability to manually guide the robot through the required path. Nowadays this is considered a subset of an increasing tendency towards safer and easier human-robot interaction (Krüger et al. 2009)(Arai et al. 2010). Furthermore, by combining the intelligence of the human with the sophistication and accuracy of the machine this method can be utilized for intelligent assist systems (Krüger et al. 2006)(Krüger & Surdilovic 2008). By utilizing the readings of an FTS mounted between the robot and the gripper, the robot can easily be jogged from its TCP through the task space (Schraft & Meyer 2006). Another variant of the scheme allows manual movement of the robots' joints by utilizing the readings of motor currents at each joint (Winkler 2006, P. 113-115). This however depends on the availability of an interface for the motor currents. In this experiment, the first variant utilizing FTS will be investigated in the work-piece based context on a simple pick and place task. Hence, the programmer moves each robot separately until the work-piece is gripped. Subsequently, a switch to work-piece space allows the programmer to move the work-piece. The PCM was configured to map the required functionalities to the buttons on the WiiMot as shown in Figure 8.2.

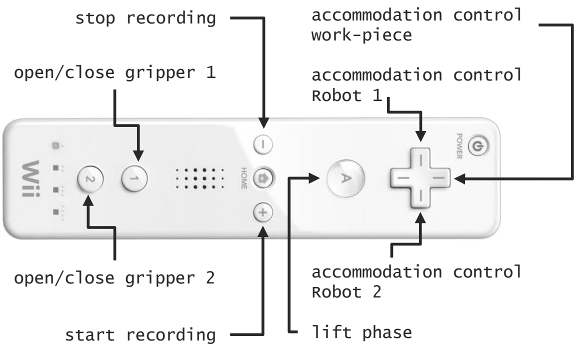


Figure 8.2: Mapping different control functionalities to the WiiMot

8.2.1.2 Programming procedure

In order to program a task by hand-guiding the robots and work-piece, the programmer has to execute the following steps (Figure 8.3 & 8.4):

1. Start recording (plus button)

2. Move the first robot until it reaches its gripping position (left arrow)
3. Grip the work-piece with the first robot (button 1)
4. Move the second robot until it reaches its gripping position (right arrow)
5. Grip the work-piece with the second robot (button 2)
6. Activate lift mode for 2-4 seconds (button A)
7. Trigger the gravity compensation (button B)
8. Start moving the work-piece to its target location (up arrow)
9. When the work-piece reaches its target position release the robots grip (button 1,2)
10. Activate lift mode for 2-4 seconds (button A)
11. Trigger the gravity compensation (button B)
12. Move the first robot until it reaches its park position (left arrow)
13. Move the second robot until it reaches its park position (right arrow)
14. Stop recording (plus button)

8.2.1.3 Experiments

After the programmer receives instructions to the different functionalities triggered by the buttons on the WiiMot and to the steps in the programming procedure, he is allowed to program a simple pick and place task. The choice of task is representative of several other tasks common with CIR, for instance load sharing or load transport. Programming lasted on average about 1 min, which is the time of the task itself. To compare it to conventional programming, the functionalities inherent to a teach-pendant was emulated in the software environment. Programming using this technique lasted on average about 2.5 times longer as the hand-guided method. However, it has to be noted here that programming time depends heavily on the complexity of the given task. During execution the recorded data (positions, triggers etc.) are played back, either with the same speed or by increasing the speed of execution. If it is desired that forces during execution be minimized, force control could be activated during certain phases by adjusting the RCM.

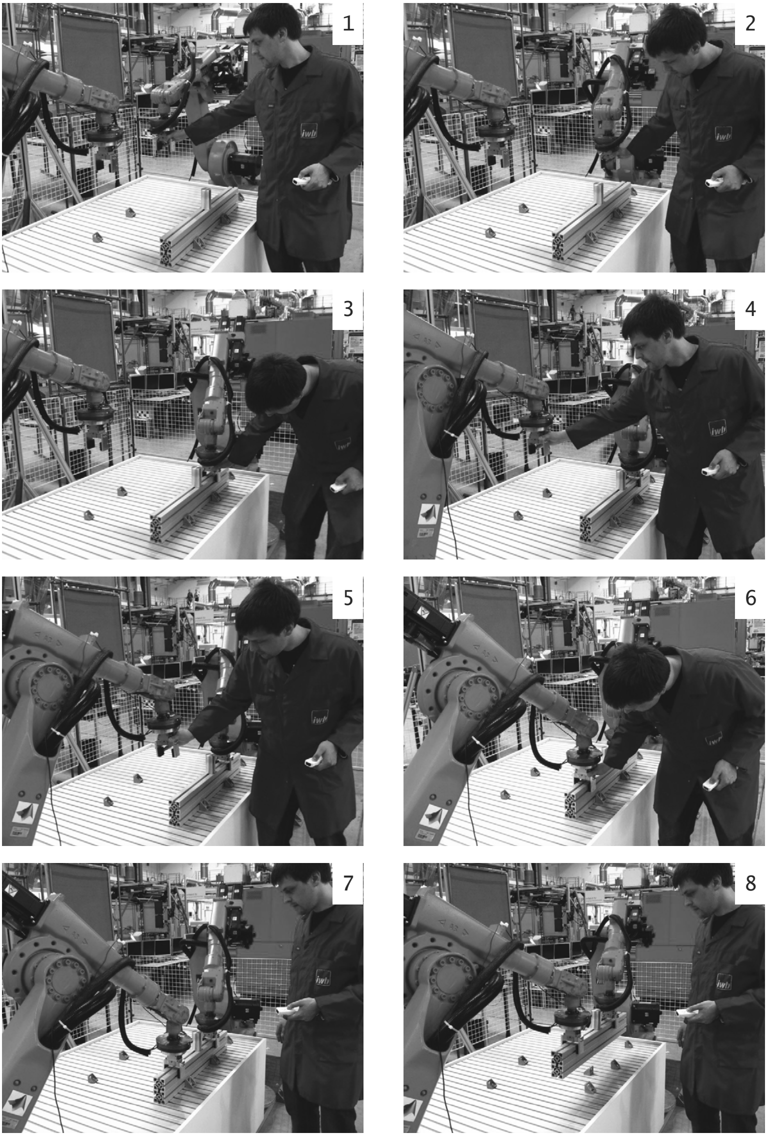


Figure 8.3: Hand guided programming sequence: Part 1

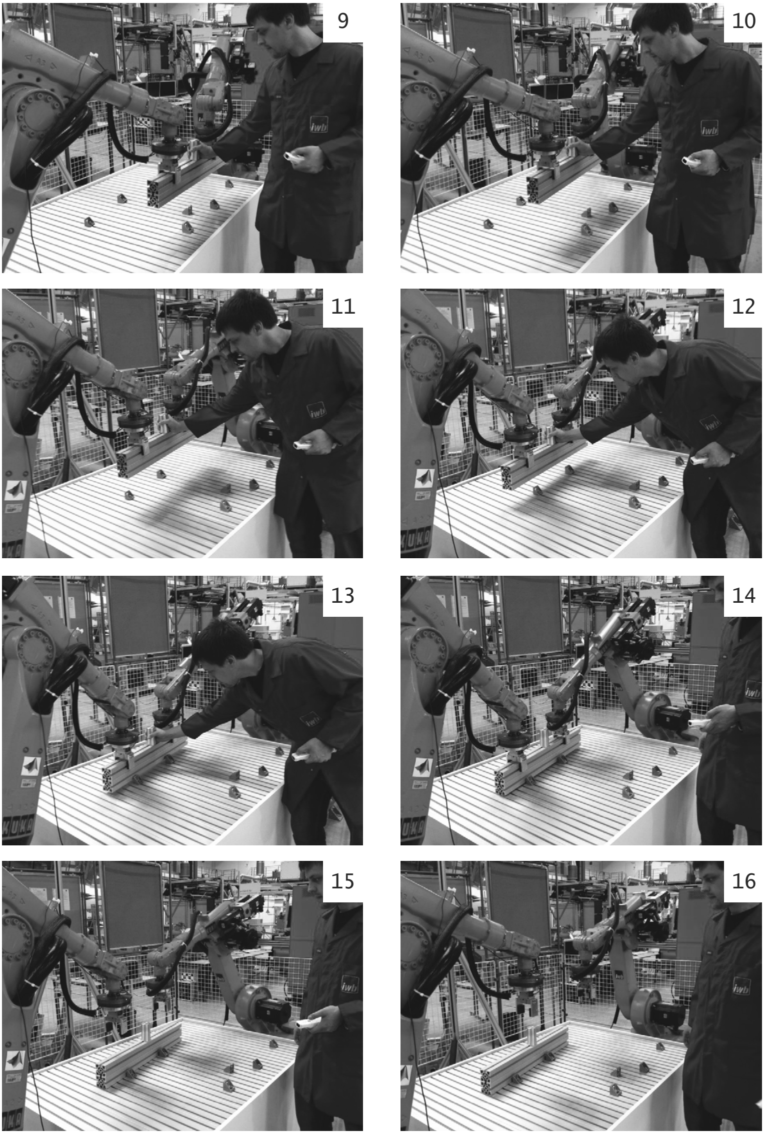


Figure 8.4: Hand guided programming sequence: Part 2

8.2.2 Off-line simulation

8.2.2.1 Overview

The second class of robot programming techniques common in industrial practice is off-line simulation and programming. However, as already mentioned in section 2.2.2 this technique is considered cost intensive given the fact that discrepancies between the simulation and real-world have to be compensated for. Mostly, this is manually executed on the work-cell directly. Other approaches have advocated using sensors or calibration devices to minimize these differences. In this experiment the FTS will be utilized to automatically adapt the behavior of the program during execution on a pick and place task similar to the first experiment. After the task is simulated and code for the robots is automatically generated from the software environment, different force controllers (single robot and/or work-piece) are activated according to the phase requirements. Accordingly, manual calibration and compensation is no more deemed necessary.

8.2.2.2 Programming procedure

Before starting, extra care has to be taken to calibrate the virtual environment w.r.t. to the real world, thereby attempting to attain a near perfect virtual model of the real world in the software environment. This could be done by manually measuring the different coordinate systems in the real world w.r.t. each other. For instance, the distance and rotation of both robots relative to each other is important to determine whether the virtual coordinate systems are accurate or not. In this case, the programming procedure looks as follows:

1. Model the real world in the software environment using the 3D-models available from the manufacturer
2. Model the joints of all manipulators according to their kinematic relations
3. Model all additional motors and moving objects
4. Calibrate the virtual world w.r.t. the real world
5. Define 6 task phases: approach, grasp, lift, transport, release, withdraw
6. Define the motion and force requirement of each phase
7. Simulate the movements of the robots and the work-piece in each phase
8. Simulate process-specific actions, for instance gripper control
9. Record the movements in all phases
10. Simulate the whole process in the simulation environment
11. Configure the RCM to activate certain control structures during execution

12. Connect to the real robots and execute the program

8.2.2.3 Experiments

The experiment investigated with this approach is very similar to section 8.2.1. Despite very careful calibration between the software environment and the real world, forces arose on the work-piece. This is seen in Figure 8.5, which illustrates the magnitude of calibrated[1] forces on the work-piece during the six phases of the task during two runs. In the first run (the dotted line) the movements simulated in the the simulation environment were directly executed on the robots. While in the second run, the RCM was adjusted to activate interaction control during the *lift* and *transport* phase. However, the reduction of forces on the work-piece caused a shift in the absolute work-piece trajectory during movement. The maximum divergence was calculated to be approximately 4 mm measured from the work-piece's center of gravity. Therefore, forces arising due to discrepancies are controlled albeit affecting the quality of the work-piece path. Such a consequence has to be accounted for from the beginning of the process by allowing the programmer to define maximum force limits and maximum trajectory deviations in the phase requirements. These requirements are later achieved by choosing suitable control parameters to keep them within the given limits.

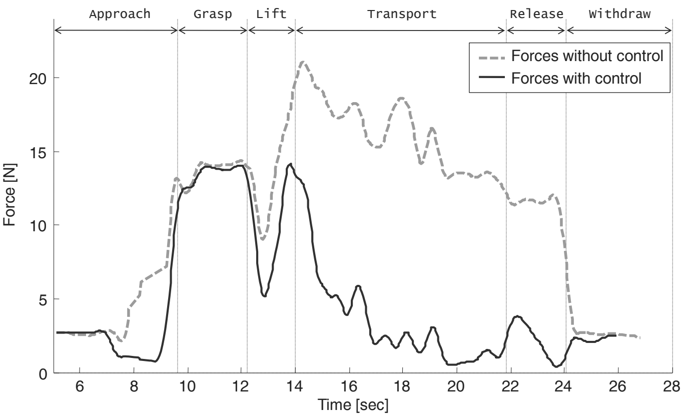


Figure 8.5: Force amplitudes with and without sensor-based adaptation of off-line programming

[1] without gravitational forces

8.2.3 Autonomous assembly

8.2.3.1 Overview

Assembly has been one of two main application domains for interaction research in robotics, the other being machining (refer to section 2.3.5). Several researchers developed control approaches and experimental test-rigs to investigate assembly with CIR. Despite early experiments in the field of high-level cooperative manipulation (Schneider & Cannon 1993)(Schneider & Cannon 1992), most subsequent research concentrated on planning methods and validity of control schemes for dual-arm robots (Nof & Rajan 1993)(Surdilovic et al. 2001). Furthermore the programming aspect in most research efforts was largely ignored. In this section, assembly of a peg-in-hole task will be investigated given its representative status for a wide range of assembly tasks. The method used here is based on the work done by (Newman et al. 2001)(Newman & Wei 2002) albeit for single robotic manipulators. By extending their work in the context of the WPBA, assembly with CIR could be readily achieved. Different process phases are mapped into states which are subsequently realized as a state-based controller. Internally, the states are time-based components which are triggered according to predefined conditions. A common condition would be for instance one that forces in specific directions to reach certain thresholds. Another condition would be that a trajectory has finished execution. In a control sense this corresponds to the high-level control layer and the intelligent layer discussed in section 5.6. Thus the state-controller replaces the human programmer as the intelligent entity responsible for decision making based on the given task requirements.

8.2.3.2 Programming procedure

The four states required for an autonomous assembly task are described as follows:

1. **Inclination**: Position control is switched on to record the absolute inclination of the work-piece relative to the mounting surface. If the two surfaces are nearly parallel, an artificial inclination is executed. This is deemed necessary to prevent the tip of the work-piece from making large area contact with the surface resulting in intangible forces for the controller.

2. **Surface detection**: To force the robots to move in a downward movement, the reference force superimposed on the impedance controller is set to an arbitrary value in the negative z-direction (refer to section 5.3). This forces the robot to move in a downward movement until resistance is encountered i.e. the mounting surface is found. Hereby the surface position is recorded as a reference and passed to the next states. Consequently, these conditions trigger the next state.

3. **Hole search**: A spiral search function is utilized to search for the hole (Guldner et al. 2003). To guarantee that the work-piece stays in contact with the mounting surface at all times, the reference force utilized in the latter state is kept on, albeit

with a reduced value. It simulates a downward pull on the work-piece w.r.t. the surface and thus maintains the contact between them while guaranteeing a stable contact behavior. The condition for ending this state depends on the difference between the actual work-piece position and the reference surface position. Once the peg sinks into the hole this difference exceeds a certain limit and triggers the next state.

4. **Insertion**: This state starts by changing the impedance parameters into pure damping while simultaneously executing a reverse inclination using the value from the first state. Two conditions can be used in tandem to terminate this state and with it the whole task. The first depends on a reverse force when the peg hits the bottom of the hole. While the second depends on the length of insertion if defined in advance.

During each state, the interaction controller is fully parameterized in all directions. This amounts to 18 parameters for the impedance controller and 6 parameters for the force controller (refer to section 5.3). Jumping between states entails fulfilling the trigger conditions between the states and therefore changing the interaction parameters through the mechanism discussed in section 6.4.3.

8.2.3.3 Experiments

The latter procedure was executed using 5 different initial positions (Figure 8.6). In 4 out of 5 runs the task was successfully completed while for the remainder, the hole was not found. This however was attributed to the fact that the search algorithm i.e. the spiral trajectory, is valid only for a definite range of initial positions. Hence, the search result is a function of both the search parameters *and* the initial position. Another observation was the time of successful completion. Naturally, the farther away the initial position was w.r.t. the hole position the longer it took to successfully complete the task as shown in the correlation between *distance* and *time* in Table 8.2. This is also directly related to the number of *spiral turns* required to find the hole. Given the reduced bandwidth of the control loop due to its dependence on positional interface on industrial robots, the velocity during all phases was reduced. This, in turn, contributed to the prolonged time needed for each trial.

Run	Time [sec]	Distance [mm]	Spiral turns
1	34.5	79.5	5.5
2	28.4	55.8	4.5
3	25.1	41.3	3.5
4	21.5	29.8	2.5

Table 8.2: Four runs of the autonomous assembly experiment with different initial positions

Figure 8.6: The states during an autonomous assembly process

8.3 Qualitative assessment

A single method to technically assess the proposed programming approach would be misleading given its interdisciplinary scope. Hence, it is beneficial to consider the assessment in light of application-specific criteria. The discussion here will be based on the basic criteria outlined in Eigner & Stelzer (2004) and in for assessing computer aided methods (CAx) in manufacturing systems; namely *time*, *quality*, *reuse/expandability synergy* and additionally acceptance as i (Rashidy 2008). These will be correspondingly extended to incorporate other aspects pertaining to this specific approach and its technical nature. The criteria will be clustered in the following groups:

- *Production-specific criteria*: flexibility, reliability and quality
- *Programming-specific criteria*: time, user interaction and acceptance
- *Robot-specific criteria*: maintenance, reuse/expandability and compatibility

Figure 8.7 summarizes the latter criteria in a graphical form. The conventional approach is illustrated as the standard form, while each criteria for the proposed approach is illustrated above or under this form, denoting an improvement or degradation of the corresponding criteria respectively.

Flexibility The WPBA increases the operation's flexibility in two ways. The first is the enhanced capability of the system to be easily reconfigured and reprogrammed, thereby quickly adopting to changes dictated by the product. The second utilizes sensor-based operation to react to disturbances appearing during execution. Consequently, the latter aspects compound together to increase the bandwidth of possible applications and their corresponding tasks.

Reliability

Sensor-based operation introduces uncertainty in the process, which might very well lead to work-cell downtime and hence decrease reliability. This could be partially attributed to the sensors themselves. Additionally the interaction of sensors and control algorithms to produce positional offsets results in unpredictable performance if the output is not bounded in advance which necessitates rigorous testing before deployment. Furthermore, external communication and non-standard interfaces could result in errors aversely affecting the overall system reliability.

Quality Any tightly coupled CIR task has two conflicting aspects; trajectory accuracy and stress loading. In reality these aspects reflect the force/position dilemma discussed in section 2.6.1 along with the quality of the process/task. Controlling the stress loading on the robots and the work-piece entails affecting the trajectory. Whether this adversely affects the accuracy or not depends on the process requirements and correspondingly its quality analyzed in section

4.2. However, the approach facilitates control of both simultaneously or separately and hence increases the likelihood of success w.r.t. given process requirements.

Time As mentioned in section 2.2.1, on-line programming contributes to decreased downtime of the system given the non-productive state of the system while programming. The WPBA facilitates off-line programming with on-line adaptation, hence the need for a real work-cell is diminished. This is directly translated into less downtime if the programming and testing takes place on an active production line. An additional advantage thereof is the ability to train personal or experiment with different scenarios without actually needing a real work-cell.

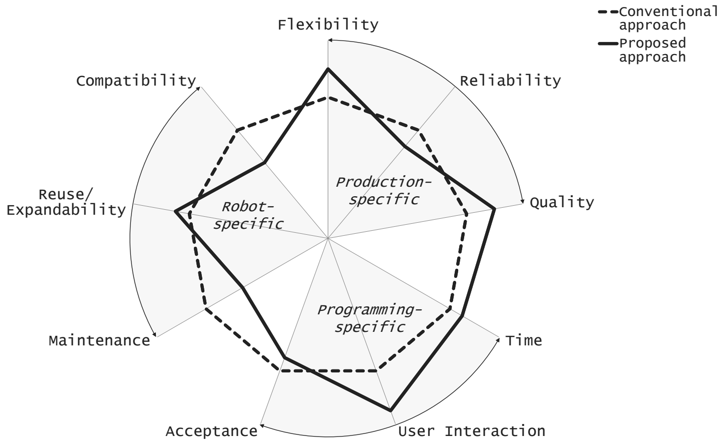


Figure 8.7: A qualitative comparison of the technical assessment

User interaction

The approach inherently reduces the skill set requirements from an operator and hence enhances the user interaction experience while requiring little technical expertise. By moving the programming task to the work-piece space i.e. the object space of the robots, the user avoids complexities of the separate manipulators. Additionally, the ability to utilize simple input devices such as joysticks to control the movement of the robots and/or the work-piece tremendously simplifies the user interaction. Certain functionalities can be activated by mapping them to buttons on those devices. While feedback to the operator can be conveyed with visual or force feedback signals, hereby enriching the interaction experience.

Acceptance

Introduction of any new approach or technical system in a production facility requires acceptance on both the management and operational level. Given the complicated nature of the approach and its wide range of utilization, acceptance on the management level is expected to be initially conservative. On an operational level reluctance to learn a new system which might incur an extra learning curve adversely affects initial productivity gains. The approach however advocates intuitiveness and user-friendliness and therefore acceptance on both levels is expected to be quicker than for comparable industrial machines.

Maintenance

The high degree of integration between the components represents a headache in terms of maintenance. This is exacerbated by the fact that low-level integration between different hardware and software components geared toward user-friendly functionality makes the underlying design fairly complicated. Furthermore, recalibration and testing are absolutely necessary after any maintenance operation whether on specific components or on the whole system. The latter reasons render maintenance a very delicate and complex operation.

Reuse/expandability

Porting the approach to new hardware involves delicate re-calibration of sensors. This could sometimes be troublesome given the highly interconnected nature of any technical realization of the approach, as evident by the complexity of the test-rig in chapter 7. Regarding expandability, the WPBA was specifically designed to address the issue of programing CIR independent of the number of robots involved in the operation by moving the focus of programming from the process enablers i.e. robots to the task i.e. the work-piece. This renders the approach scalable for a theoretically infinite number of robots, irregardless of their kinematic structure.

Compatibility

Although interfaces are to a great extent standardized, low-level functionalities such as external real-time control are only available through vendor-specific interfaces. Additionally, control performance is mainly dictated by the type and bandwidth of both components and interfaces. This is further complicated by the fact that robots from different manufacturers have different programming languages, which raises significant issue relating to compatibility when the approach is ported from one work-cell to another.

8.4 Economic assessment

The aim here is to quantify the economical or financial gains expected if the proposed approach is applied. This benefit is usually determined by comparing two technical systems offering the same functionality (Vogl 2009)(Munzert 2009). Furthermore the efficiency and success rate of the task on both systems has to be almost identical if not the same. Notwithstanding the latter aspects, several other factors play a significant role in determining the financial gains. First and foremost is the type of the application under investigation, which in turn defines the process and its requirements. This not only entails profound knowledge of the application but also and experienced insight into the application domain and its interconnections with other processes. The resultant product and its variants represent the second major factor. If batch sizes and production time are known, the frequency of (re)programming and therefore the costs linked to it can be accurately estimated. Although the question posed here is *how do we economically evaluate the work-piece based programming method?* it would be more beneficial to consider it in a greater context by asking *is applying CIR with the WPBA tied to specific economical gains?*. By considering the latter question, the former is implicitly answered. As a result, three scenarios for economical evaluation could be envisioned here:

1. **Replacement Scenario**: In this scenario static jigs and fixtures are replaced with CIR. Hence, the assessment here is of two completely different systems.
2. **Upgrade Scenario 1**: In this scenario, one or more robots are added to a robotic work-cell and subsequently allowing them to function in a cooperative manner.
3. **Upgrade Scenario 2**: In this scenario, force control is integrated in an existing CIR and subsequently upgrading it to implement the WPBA.

An intricate relation runs through the three scenarios. They actually represent the chain of evolution of any CIR deployment. As already noted in Chapter 1, CIR are mainly utilized as flexible jigs and fixtures. Hence, the first scenario explores the financial benefit when the case for industrial robots or even CIR instead of static jigs is made. In the second scenario, industrial robots are utilized but reaches its limits regarding loading and dexterous manipulation which in turn makes the case for CIR. And finally, the last scenario takes the CIR deployment to the last step by applying the WPBA. An application-oriented assessment would compare the gains between the initial situation of the first scenario (static jigs and fixtures) and solution in the last scenario (CIR with the WPBA). This however doesn't isolate the effect of applying the WPBA, and hence the calculation here will be limited to the third scenario.

In this scenario an existing CIR system will be upgraded to utilize the WPBA. An upgrade starts by purchasing all the devices and components -whether they are hardware or software- for the required operation. As dictated in Table 8.3, these fall under the *investment costs* category. It is noteworthy to point out that some components such as software have to be custom built compared to other components that could be directly purchased.

Annual costs comprise of calculating depreciation and maintenance costs incurred throughout a year. While depreciation and imputed interest are calculated according to a fairly standard formula, maintenance is assumed to cost 6000 €/yr. The **benefit calculation** commences with the following assumptions:

- A CIR programmer costs on average 100 €/hr (Vogl 2009). To (re-)program the required tasks, he is hired for a total of 250 hr/yr.
- Time saved by the programmer when utilizing the WPBA amounts to 45% of the total programming time.

Investment Costs				
Component	**Number**	**Part Cost in €**	**Cost in €**	**Com.**
FTS	2	5000	10000	-
RTP	1	2500	2500	-
Software	1	10000	10000	Custom
PC	1	2000	2000	-
Input devices	2	250	500	Low cost
Total			25000	-
Annual Costs				
Expense	**Calculation**	**Value**	**Unit**	**Com.**
Investment	-	25000	€	-
Service life	T	5	yr	-
Depreciation	$DC = I/T$	5000	€/yr	-
Interest	$I = 0.09I/2$	1125	€/yr	9%/ rate
Maintenance	M	6000	€/yr	-
Total	$AC = D + I + M$	12125	€/yr	-
Benefit Calculation				
Expense	**Calculation**	**Value**	**Unit**	**Com.**
Programmer	P	100	€/hr	-
Time saved (%)	Tp	45	%	-
Time saved (hr)	$Ts = (1/1 - Tp) - 1$	0.82	hr	-
Usage	U	250	hr/yr	-
Cost benefit	$BC = P.Ts.U$	20455	€/yr	-
Payback period	$I/(BC - AC + DC)$	1.87	yr	-

Table 8.3: Details of a benefit calculation based on the third scenario

The latter assumptions are translated to an annual financial benefit of 20,455 €. Hence the payback period amounts to 1.87 years. This however is only an example of the cost benefit expected from utilizing the approach. A more detailed and representative calculation should take several other factors into consideration. For instance how complicated is the given task in terms of motion and force requirements? Is the task planned for batch production or small size lots? This would result in a very good estimate of the frequency of reprogramming i.e. the yearly *usage* and therefore the economical benefit. Additionally, it is imperative to determine whether a specialized robot programmer (for CIR) is required? or is it possible that a normal robot programmer do the job?. Such factors constitute the backbone of any benefit calculation and significantly influence the decision whether or not to apply the WPBA.

8.5 Programming, sensors and intelligent robots - The big picture

Industrial robots are known for their flexibility compared to other manufacturing machines. This potential however was restricted by the class of tasks envisioned for them from the start. By limiting their application domain to spray painting and welding, which require accurate positional capabilities, their full potential as a human replacement in many other tasks was not realized. The historical development of industrial class robots set the stage for the current situation of industrial robots in production facilities. One of the main factors playing a role in said development was the fact that they were only affordable for companies producing large volumes (Bilbao et al. 2005). However, the constraints on the development and deployment of industrial robots are beginning to fade, primarily due to two reasons; the decreasing costs of robots (compared to human workers) and the increasing demand on customized products . These factors play an implicit role in the trends sweeping the robotics research nowadays. Decreasing costs -of automation in general while increased costs of manual work,- render robots more attractive for a new segment of production facilities which was not conceivable a decade ago (International Federation of Robotics - IFR Statistical Department 2009). Additionally, the fact that they will be utilized to replace manual work is forcing the manufacturers to develop smaller, more compact and safe robots, in order to allow for direct human-robot collaboration (Krüger & Surdilovic 2008)(Krüger et al. 2009). The second factor outlines a major challenge facing production engineering worldwide. Customers hungry for tailored products are increasingly pushing the boundaries of production lines and their capability to manage the accompanying complexity (Scholzreiter & Freitag 2007). Despite these factors, the gap between what is currently needed and what is available w.r.t. deployment of robot-based production lines, has yet to decrease. By providing simpler user interfaces and hence enhanced programming capabilities this gap could definitely be bridged. However, it is the author's view that the interfaces should not be limited to over-designed GUI or to abstracting a limited number of functionalities to an input device. But rather, the designers should -in a holistic view- reconsider the hardware and software components and their interactions together. In other words, reevaluating their basic approach to industrial robots as pure positional machines or as enablers of mechanical tasks. This way, designers can tightly integrate different sensors in simple and reliable ways to increase the bandwidth of possible tasks. Additionally, this would allow a sensor to be utilized in different capacities during programming and operation. For instance, a camera can be used during programming to acquire the shape of the object, while during operation to detect the existence of the same object in a given area. Furthermore, enhanced sensoric capacity also means implicitly increasing the robots ability to interact to uncertain conditions during execution in a feel-reason-act loop (Caccavale et al. 2005). Such a development would lead to blurring the line between programming and operation while providing the robots with self-optimization capabilities and more independence. By extrapolating this development path, intelligence is eventually achieved by creating robotic systems capable of reacting in an autonomous and flexible manner to continuously changing environmental conditions[2] (Gausemeier 2005)(Gausemeier et al. 2009). What the author advocates here is attaining intelligence through an incremental approach which takes into consideration the complexities of standardized interfaces and operation procedures in manufacturing facilities and not by attempting to achieve intelligent robots with one step. Despite that the approach and the test-rig developed in this thesis pertain to a certain class of robot tasks (tightly coupled CIR), the discussion here formed the spirit of the research done in this work.

[2]This can be also termed self-optimizing mechatronic systems

9 Conclusion

Real knowledge is to know the extent of one's ignorance

CONFUCIUS

Summary

Cooperative robotic manipulation represents the path to attain human-like manipulation capabilities, for instance superior load sharing ability and increased dexterity. Such capabilities are realized in industrial practices by replacing static jigs and fixtures with co-operating industrial robots that share a common work-piece. Consequently, this translates into direct gains in productivity by investing less in more compact systems, streamlining material flow through the work-cell and overall building more flexible production systems. However, cooperating industrial robots face tremendous issues regarding application and deployment. These mainly stem from general issues pertaining to industrial robots, but are also compounded with issues specific to cooperative manipulation and its intricate technicalities. A detailed analysis of scientific research in this field and commercial solutions on the market have revealed the inherent flaws in their design and operation namely; lack of integrated interaction control, lack of intuitive programming techniques and inflexible system architectures. Based on the latter discussion, the objective of the thesis was to investigate an approach for simplified programming of cooperating industrial robots. The approach encompasses a methodology to define and satisfy practical task objectives formulated in the work-piece space by imparting to the programmer full control of the work-piece. This includes spatial positioning of the work-piece and stress loading on the work-piece during all phases of a cooperative task which was accordingly termed the work-piece based approach. In order to achieve such functionality, a conceptual framework for programming cooperating industrial robots was developed and subsequently realized on an experimental test-rig. The framework employs classical control concepts in an encompassing control architecture combined with a flexible software environment facilitating a powerful and versatile architecture, which in turn offers the programmer numerous programming paradigms. Hence, one might say that the front-end of this story is simplifying programming of cooperating industrial robots by giving the operator full control of the work-piece during all phases of a cooperative task. While the back-end is one of tightly integrating force control in programming and operation of cooperating industrial robots through designing and implementing a flexible cooperative architecture. The technical realization of two components of the framework, namely the control module and the software module, constitute the backbone of the thesis. The control module builds upon common control practices in cooperative manipulation. Armed with such knowledge a multi-layered control architecture is proposed. In the undermost level, functionalities with real-time constraints are implemented. Taking into consideration the geographical location of its respective constituents, local (for single manipulators) and global (for the work-piece) control loops have been built to cater for the task's needs. These loops contain fully parameterized interaction controllers utilizing a force-augmented impedance scheme, which is additionally superimposed on the position control. Moreover, functionalities required for the different phases are triggered from the high-level control layer by utilizing real-time components through two mechanisms working in tandem: signal rerouting and interaction controller parameterization. These result in control structures, which are time-dependent arrangements of blocks attaining certain control behaviors according to the imposed requirements. The software module is represented by a software environment specifically designed and developed to fulfill predefined requirements. Hereby enabling the human operator to take more than one role during all phases of manufacturing. As a production planner, the operator simulates and tests the work-cell in the virtual world. As a robot programmer, he is capable of applying different programming paradigms to generate code and validate the operation on the real work-cell. And finally as a work-cell operator, he can monitor the safety and reliability of operation during runtime. The environment is characterized by its flexibility which is attributed to several features. Chief among them is the capability of simultaneous graphical, physical and haptic simulation. Additionally, a comprehensive signal switching mechanism allows the operator access to all signals in the system and hence guarantees a transparent information flow. To assist the human-machine interface, several devices are integrated for simple and intuitive interaction and programming. To validate the approach, three cooperative manipulation scenarios were investigated, each one of them employing a different programming paradigm. The scenarios cover on-line, off-line programming and additionally autonomous programming. Hereby validating the versatility of the proposed method and hence its applicability for a wide range of cooperative tasks. Compared to conventional CIR systems, the approach promises reduced programming time and safer operation. Additionally, it fits and complements the major trends advocated in modern production facilities towards developing flexible and reconfigurable machines.

Contributions

The contributions of this research work can be summarized as follows:

1. Developing a programming approach for tightly coupled cooperating industrial robots based on imparting to the programmer full control of the work-piece during all phases of a cooperative task. Basically, this allows the programmer to shift his focus from programming the robots to focusing on programming the work-piece. This in turn simplifies the programming effort.

2. Developing a conceptual framework to implement the proposed approach for programming cooperating industrial robots. This provided the ground work for its

technical realization by incorporating all the required components within one framework.

3. Designing and implementing the two main constituents of the framework, namely the control and the software module. The control module was represented by a multi-layered control architecture and the software module by a flexible software environment.
4. Developing an experimental test-rig featuring a flexible sensor-based architecture using the developed control and software components.
5. Experimental validation of the approach's technical realization on three cooperative manipulation tasks featuring three different programming paradigms. Consequently, the experiments highlight the flexibility of the proposed approach and thereby its applicability to a wide range of cooperative tasks utilizing different programming paradigms

Future research
Based on the research done in this thesis, two research directions are suggested. These are not particularly limited to CIR but are more generic in nature and cover the topic of intuitive programming of industrial robots in general. The first is *hybrid sensor-based programming*. Hybrid here alludes to several programming paradigms applied together to program one task. Hereby exploiting the advantages of each paradigm during certain phases while avoiding its pitfalls during other phases by utilizing another paradigm. The second research direction is *flexible sensor integration*, which represents a natural extension to the work done in this thesis to cover several other classes of sensors. Moreover, the research should focus on reliable and safe application during both programming and operation of a robotic work-cell, while simultaneously complying with industry standards. These directions would greatly improve the programming experience rendering it more user-friendly and hence reduce the learning curve and general acceptance on both operational and management levels. Additionally, they confirm with current research trends in developing robots which should be more responsive to humans. Consequently, they should be capable of understanding programming instructions from multiple sources and utilize intelligent architectures to understand and adapt their behavior. Thus, a significant step toward fully autonomous cognitive robots could be achieved.

List of Figures

List of Tables

Bibliography

[ABB-Group 2011]

ABB-Group: ABB Robot Studio, 2011, http://www.abb.com/product/ap/seitp327/6230bcade7a9d7d8c12573f50042d0a9.aspx, Date Accessed: 22.02.2011, 2011.

[Abele et al. 2005]

Abele, E.; Weigold, M.; Kulok, M.: Spanende Bearbeitung mit Industrierobotern. wt Werkstattstechnik Online 95 (2005) 9, pp. 666–671.

[Abele et al. 2008a]

Abele, E.; Bauer, J.; Bertsch, C.; Laurischkat, R.; Meier, H.; Reese, S.; Stelzer, M.; Von Stryk, O.: Comparison of Implementations of a Flexible Joint Multibody Dynamics System Model for an Industrial Robot. In: 6th CIRP International Conference on Intelligent Computation in Manufacturing Engineering. Naples, Italy: Citeseer 2008a. pp. 23–25.

[Abele et al. 2008b]

Abele, E.; Rothenbücher, S.; Weigold, M.: Cartesian Compliance Model for Industrial Robots Using Virtual Joints. Production Engineering 2 (2008b) 3, pp. 339–343.

[About.com: Ergonomics 2011]

About.com: Ergonomics: Human-Machine Interface, 2011, http://ergonomics.about.com/od/glossary/g/man_machine.htm, Date Accessed: 01.03.2011, 2011.

[Adli et al. 1991]

Adli, M.; Nagai, K.; Miyata, K.; Hanafusa, H.: Apparent Structural Stiffness of Closed Mechanisms Under the Effect of Internal Forces During Dynamic Motion. In: IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91. Osaka, Japan: IEEE 1991. 91, pp. 773–778.

[Al-Jarrah & Zheng 1997]

Al-Jarrah, O.; Zheng, Y.: Arm-Manipulator Coordination for Load Sharing Using Variable Compliance Control. In: IEEE International Conference on Robotics and Automation, 1997. Proceedings., Albuquerque, New Mexico, USA 1997, Volume 1. pp. 895–900.

[Al-Qasimi et al. 1995]

Al-Qasimi, A.; Akyurt, M.; Dehlawi, F.: On Robot Programming and Languages. In: The Fourth Saudi Engineering Conference, 1995. pp. 249–258.

[Alford & Belyeu 1984]

Alford, C.; Belyeu, S.: Coordinated Control of Two Robot Arms. In: 1984 IEEE International Conference on Robotics and Automation, 1984. pp. 468–473.

[Anderson et al. 2008]

Anderson, E.; Engel, S.; Comninos, P.; McLoughlin, L.: The Case for Research in Game Engine Architecture. In: Proceedings of the 2008 Conference on Future Play: Research, Play, Share. New York, New York, USA: ACM 2008. pp. 228–231.

[Ang Jr & Andeen 1995]

Ang Jr, M.; Andeen, G.: Specifying and Achieving Passive Compliance Based on Manipulator Structure. IEEE Transactions on Robotics and Automation 11 (1995) 4, pp. 504–515.

[Arai et al. 1995]

Arai, T.; Osumi, H.; Fukuoka, T.; Moriyama, K.: A Cooperative Assembly System Using Two Manipulators with Precise Positioning Devices. CIRP Annals-Manufacturing Technology 44 (1995) 1, pp. 23–26.

[Arai et al. 2006]

Arai, T.; Yamanobe, N.; Maeda, Y.; Fujii, H.; Kato, T.; Sato, T.: Increasing Efficiency of Force-Controlled Robotic Assembly-Design of Damping Control Parameters Considering Cycle Time. CIRP Annals-Manufacturing Technology 55 (2006) 1, pp. 7–10.

[Arai et al. 2010]

Arai, T.; Kato, R.; Fujita, M.: Assessment of Operator Stress Induced by Robot Collaboration in Assembly. CIRP Annals-Manufacturing Technology 59 (2010) 1, pp. 5–8.

[Arcara & Melchiorri 2002]

Arcara, P.; Melchiorri, C.: Control Schemes for Teleoperation with Time Delay: A Comparative Study. Robotics and Autonomous Systems 38 (2002) 1, pp. 49–64.

[Arnaud & Parisi 2007]

Arnaud, R.; Parisi, T.: Developing Web Applications with Collada and x3d, 2007.

[Barnes et al. 2008]

Barnes, M.; Finch, E.; Sony Computer Entertainment Inc.: COLLADA - Digital Asset Schema Release 1.5.0 :Specification. Technical Report April, 2008.

[Bates & Gregory 2006]

Bates, R.; Gregory, D.: Voice & Data Communications Handbook. 5. Edition. McGraw-Hill Osborne Media 2006.

[Bekalarek et al. 2001]

Bekalarek, J.; Zawal, R.; Kozlowski, K.: Hybrid Force/Position Control for Two Cooperative Staubli RX60 Industrial Robots. In: Proceedings of the Second International Workshop on Robot Motion and Control. RoMoCo'01. Poznan Univ. Technol 2001. pp. 305–310.

[Bernhard et al. 2002]

Bernhard, R.; Schreck, G.; Willnow, C.: Development of Virtual Robot Controllers and Future Trends. In: Cost oriented automation (low cost automation 2001): A proceedings volume from the 6th IFAC Symposium on Cost OrientedAutomation (Low Cost Automation 2001), 2002.

[Biggs & Macdonald 2003]

Biggs, G.; Macdonald, B.: A Survey of Robot Programming Systems. In: Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia 2003.

[Bigras et al. 2007a]

Bigras, P.; Lambert, M.; Perron, C.: New Formulation for an Industrial Robot Force Controller: Real-time Implementation on a KUKA Robot. 2007 IEEE International Conference on Systems, Man and Cybernetics (2007a), pp. 2794–2799.

[Bigras et al. 2007b]

Bigras, P.; Lambert, M.; Perron, C.: New Formulation for an Industrial Robot Impedance Controller: Real-time Implementation on a KUKA Robot. 2007 IEEE International Conference on Systems, Man and Cybernetics (2007b), pp. 2782–2787.

[Bilbao et al. 2005]

Bilbao, A.; Bischoff, R.; Brogardh, T.; Christensen, H.; Dresselhaus, M.; Haegele, M.; Lacello, L.; Nilsson, K.; Tomatis, N.; Sagert, S.; Skordas, T.: DR. 14.1 White Paper - Industrial Robot Automation. Technical Report, IPA, 2005.

[Bischoff et al. 2002]

Bischoff, R.; Kazi, A.; Seyfarth, M.: The MORPHA Style Guide for Icon-based Programming. In: Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication, 2002. pp. 482–487.

[Bogue 2009]

Bogue, R.: Finishing Robots: A Review of Technologies and Applications. Industrial Robot: An International Journal 36 (2009) 1, pp. 6–12.

[Böhm 1994]

Böhm, J.: Kraft- und Positionsregelung von Industrierobotern mit Hilfe von Motorsignalen. VDI-Verlag 1994.

[Bray et al. 2008]

Bray, T.; Paoli, J.; Sperberg-McQueen, C.; Maler, E.; Yergeau, F.: Extensible Markup Language (XML) 1.0, World Wide Web Consortium, Recommendation REC-xml-20081126. 5. Edition. 2008, http://www.w3.org/TR/2008/REC-xml-20081126/.

[Brecher et al. 2010]

Brecher, C.; Roß mann, C.; Schlette, C.; Herfs, W.; Ruf, H.; Göbel, M.: Intuitive Robot Programming in the Automated Assembly - A Hybrid Technique for Programming by Demonstration. wt Werkstattstechnik 100 (2010), pp. 681–686.

[Caccavale & Villani 2000]

Caccavale, F.; Villani, L.: Impedance Control for Multi-Arm Manipulation. In: Proceedings of the 39th IEEE Conference on Decision and Control. Sydney, Australia: IEEE 2000. pp. 3465–3470.

[Caccavale et al. 2000]

Caccavale, F.; Chiacchio, P.; Chiaverini, S.: Task-space Regulation of Cooperative Manipulators. Automatica 36 (2000) 6, pp. 879–887.

[Caccavale et al. 2005]

Caccavale, F.; Natale, C.; Siciliano, B.; Villani, L.: Integration for the Next Generation: Embedding Force Control into Industrial Robots. IEEE Robotics & Automation Magazine 12 (2005) 3, pp. 53–64.

[Caccavale et al. 2008]

Caccavale, F.; Chiacchio, P.; Marino, A.; Villani, L.: Six-DOF Impedance Control of Dual-Arm Cooperative Manipulators. IEEE/ASME TRANSACTIONS ON MECHATRONICS 13 (2008) 5, pp. 576–586.

[Cardarelli et al. 1995]

Cardarelli, G.; Pelagagge, P.; Palumbo, M.: Assembly with Co-operating Robots. Assembly Automation 15 (1995) 2, pp. 36–40.

[Castuera et al. 2004]

Castuera, J. C.; Lopez-juarez, I.; Industrial, P.; Quintana, B.: Intelligent Task Level Planning for Robotic Assembly : Issues and Experiments. In: MICAI 2004: Advances in Artificial Intelligence, Mexico City, Mexico 2004. pp. 872–881.

[Chen et al. 2007]

Chen, H.; Zhang, G.; Zhang, H.; Fuhlbrigge, T.: Integrated Robotic System for High Precision Assembly in a Semi-structured Environment. Assembly Automation 27 (2007) 3, pp. 247–252.

[Chen 2005]

Chen, J. R.: Constructing Task-Level Assembly Strategies in Robot Programming by Demonstration. The International Journal of Robotics Research 24 (2005) 12, pp. 1073–1085.

[Choi et al. 1999]

Choi, J.-D.; Kang, S.; Kim, M.; Lee, C.; Song, J.: Two-arm Cooperative Assembly using Force-guided Control with Adaptive Accommodation. In: Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE 1999. pp. 1253–1258.

[Christensen et al. 2009]

Christensen, H.; Goldberg, K.; Kumar, V.; Trinkle, J.: Whitepaper on Robotics and Automation Research Priorities for U.S. Manufacturing. Technical Report, 2009.

[Chun 1992]

Chun, W.: Adjustable Compliant Robotic Wrist Using an Electro-rheological Fluid. US Patent 5125759 (1992).

[Ciblak & Lipkin 1996]

Ciblak, N.; Lipkin, H.: Remote Center of Compliance Reconsidered. In: Proc. ASME Design Engineering Technical Conference and Computers in Engineering Conference. Irvine, California: Citeseer 1996.

[Ciblak & Lipkin 2003]

Ciblak, N.; Lipkin, H.: Design and Analysis of Remote Center of Compliance Structures. Journal of Robotic Systems 20 (2003) 8, pp. 415–427.

[Clarke et al. 2006]

Clarke, S.; Schillhuber, G.; Zäh, M.; Ulbrich, H.: Telepresence Across Delayed Networks: A Combined Prediction and Compression Approach. IEEE International Workshop on Haptic Audio Visual Environments and their Applications HAVE (2006) November, pp. 171–175.

[Collett et al. 2005]

Collett, T.; MacDonald, B.; Gerkey, B.: Player 2.0: Toward a Practical Robot Programming Framework. In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005). Citeseer 2005.

[Conti & Khatib 2005]

Conti, F.; Khatib, O.: Spanning Large Workspaces Using Small Haptic Devices. In: First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. IEEE 2005. pp. 183–188.

[Conti et al. 2003]

Conti, F.; Barbagli, F.; Balaniuk, R.; Halg, M.; Lu, C.; Morris, D.; Sentis, L.; Vileshin, E.; Warren, J.; Khatib, O.: The CHAI Libraries. Proceedings of Eurohaptics 2003 (2003), pp. 496–500.

[Conti et al. 2005]

Conti, F.; Barbagli, F.; Morris, D.; Sewell, C.: CHAI 3D: An Open-Source Library for the Rapid Development of Haptic Scenes. In: IEEE World Haptics, Pisa, Italy 2005.

[Conti et al. 2011a]

Conti, F.; Barbagli, F.; Morris, D.; Sewell, C.; Grange, S.: CHAI3D, 2011a, `http://www.chai3d.org/`, Date Accessed: 28.02.2011, 2011a.

[Conti et al. 2011b]

Conti, F.; Barbagli, F.; Morris, D.; Sewell, C.; Grange, S.: cGenericDevice Class Reference, 2011b, `http://www.chai3d.org/doc/classc_generic_device.html`, Date Accessed: 01.03.2011, 2011b.

[Craig 2005]

Craig, J.: Introduction to Robotics: Mechanics and Control. 3. Edition. Pearson Education Inc. 2005.

[Dassault Systems 2011]

Dassault Systems: DS DELMIA, 2011, `http://www.3ds.com/products/delmia/welcome/`, Date Accessed: 22.02.2011, 2011.

[De Fazio & Whitney 1984]

De Fazio, T.; Whitney, D.: Adjustable Remote Center Compliance Device. US Patent 4477975 (1984).

[Deisenroth & Krishnan 1999]

Deisenroth, M.; Krishnan, K.: On-Line Programming, John Wiley & Sons Inc. 1999, Chapter 18. 2. Edition.

[Denkena et al. 2004]

Denkena, B.; Hein, B.; Apitz, R.; Kowalski, P.; Mages, D.: Vereinfachte Programmierung von Industrierobotern. wt Werkstattstechnik online - Industrieroboter, CAM (Computer Aided Manufacturing), Virtual Reality 94 (2004), pp. 442–446.

[Dillmann 2010]

Dillmann, R.: Robotik II - Programmieren von Robotern. Vorlesungsskriptum - Universität Karlsruhe. Technical Report, Universität Karlsruhe, 2010.

[DIN EN ISO 8373 1996]

DIN EN ISO 8373: DIN EN ISO 8373: 1994 Industrieroboter - Wörterbuch, 1996, Berlin: Beuth 1996 1996.

[Dorf & Bishop 2007]

Dorf, R.; Bishop, R.: Modern Control Systems. 9. Edition. Prentice Hall 2007.

[Drake & Simunovic 1979]

Drake, S.; Simunovic, S.: Compliant Assembly System Device. US Patent 4155169 (1979).

[Duelen et al. 1991]

Duelen, G.; Munch, H.; Zhang, Y.: A Comparison of Control Strategies for Force Constrained Cooperating Robots. In: Proceedings of the 30th IEEE Conference on Decision and Control. Brighton, England: IEEE 1991. pp. 708–709.

[Eidson 2006]

Eidson, J.: Measurement, Control, and Communication Using IEEE 1588 (Advances in Industrial Control). Springer-Verlag New York, Inc. 2006.

[Eigner & Stelzer 2004]

Eigner, M.; Stelzer, R.: Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management. Springer Verlag 2004.

[Elbestawi et al. 1992]

Elbestawi, M.; Bone, G.; Tern, P.: An Automated Planning, Control, and Inspection System for Robotic Deburring. CIRP Annals-Manufacturing Technology 41 (1992) 1, pp. 397–401.

[Eppinger & Seering 1987]

Eppinger, S.; Seering, W.: Introduction to Dynamic Models for Robot Force Control. IEEE Control Systems Magazine 7 (1987) 2, pp. 48–52.

[FANUC Robotics America Corporation 2011]

FANUC Robotics America Corporation: ROBOGUIDE, 2011, `http://www.fanucrobotics.com/Products/vision-software/simulation-software.aspx`, Date Accessed: 22.02.2011, 2011.

[Fonseca Ferreira & Tenreiro Machado 2007]

Fonseca Ferreira, N.; Tenreiro Machado, J.: Fractional Control of Coordinated Manipulators. Journal of Advanced Computational Intelligence and Intelligent Informatics 11 (2007) 9.

[Fortell et al. 2009]

Fortell, H.; Johansson, S.-E.; Lindstrom, S.-E.: Control System Method and Computer Program for Synchronizing several Robots. US Patent 7620478 (2009).

[Fuchs 2011]

Fuchs, C.: Cafu Engine, 2011, http://www.cafu.de/, Date Accessed: 28.02.2011, 2011.

[Gao et al. 1992]

Gao, X.; Dawson, D.; Qu, Z.: On the Robust Control ofTwo Manipulators Holding a Rigid Object. Journal of Intelligent and Robotic Systems 6 (1992) 1, pp. 107–119.

[Garlan & Shaw 1993]

Garlan, D.; Shaw, M.: An Introduction to Software Architecture. Advances in Software Engineering and Knowledge Engineering 1 (1993) January, pp. 1–40.

[Gausemeier 2005]

Gausemeier, J.: From Mechatronics to Self-Optimizing Concepts and Structures in Mechanical Engineering: New Approaches to Design Methodology. International Journal of Computer Integrated Manufacturing 18 (2005) 7, pp. 550–560.

[Gausemeier et al. 2009]

Gausemeier, J.; Frank, U.; Donoth, J.; Kahl, S.: Specification Technique for the Description of Self-Optimizing Mechatronic Systems. Research in Engineering Design 20 (2009) 4, pp. 201–223.

[Gausemeier et al. 2011]

Gausemeier, J.; Echterhoff, N.; Kokoschka, M.; Wall, M.: Thinking Ahead the Future of Additive Manufacturing - Analysis of Promising Industries. Technical Report, Direct Manufacturing Research Center, 2011.

[Georgia Institute of Technology et al. 2009]

Georgia Institute of Technology; University Of Southern California; Johns Hopkins University; University of Pennsylvania; University of California; Rensselaer Polytechnic Institute; University of Massachusetts; University of Utah; Carnegie Mellon University; Tech Collaborative: A Roadmap for US Robotics From Internet to Robotics. Technical Report, Georgia Institute of Technology, Computing Community Consortium,The Computing Research Association, 2009.

[Gerkey et al. 2003]

Gerkey, B.; Vaughan, R.; Howard, A.: The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In: Proceedings of the International Conference on Advanced Robotics (ICAR 2003). Coimbra, Portugal: Citeseer 2003. Icar, pp. 317–323.

[Gerstenberger et al. 2005]

Gerstenberger, M.; Gmeiner, C.; Müller, S.: Application of IEEE 1588 to Synchronize Multiple Robot Controllers, 2005, 2005.

[Glosser & Newman 1994]

Glosser, G.; Newman, W.: The Implementation of a Natural Admittance Controller on an Industrial Manipulator. In: 1994 IEEE International Conference on Robotics and Automation, 1994. Proceedings., 1994. pp. 1209–1215.

[Golle et al. 2003]

Golle, A.; Ulbrich, H.; Pfeiffer, F.: Real-time Simulation of Non-Smooth Contacts in Telepresence. In: IEEE International Conference on Industrial Technology, 2003. 2, pp. 675–679.

[Goodrich & Schultz 2007]

Goodrich, M.; Schultz, A.: Human-Robot Interaction: A Survey. Foundations and Trends in Human-Computer Interaction 1 (2007) 3, pp. 203–275.

[Graf et al. 2006a]

Graf, S.; Hagenauer, A.; Chaffee, M.: Method and System for Controlling Robots. US Patent 7010390 (2006a).

[Graf et al. 2006b]

Graf, S.; Hagenauer, A.; Chaffee, M.; Stoddard, K.: Method and Apparatus for the Synchronous Control of Manipulations. US Patent 7024250 (2006b).

[Gravel & Newman 2001]

Gravel, D.; Newman, W.: Flexible Robotic Assembly Efforts at Ford Motor Company. In: International Symposium on Intelligent Control, Mexico City, Mexico 2001. pp. 173–182.

[Grunwald et al. 2003]

Grunwald, G.; Schreiber, G.; Albu-Schaffer, A.; Hirzinger, G.: Programming by Touch: The Different Way of Human-Robot Interaction. IEEE Transactions on Industrial Electronics 50 (2003) 4, pp. 659–666.

[Gudiño Lau & Arteaga 2005]

Gudiño Lau, J.; Arteaga, M.: Dynamic Model and Simulation of Cooperative Robots: A Case Study. Robotica 23 (2005) 05, p. 615.

[Gueaieb et al. 2007a]

Gueaieb, W.; Alsharhan, S.; Bolic, M.: Robust Computationally Efficient Control of Cooperative Closed-chain Manipulators with Uncertain Dynamics. Automatica 43 (2007a) 5, pp. 842–851.

[Gueaieb et al. 2007b]

Gueaieb, W.; Karray, F.; Al-Sharhan, S.: A Robust Hybrid Intelligent Position/Force Control Scheme for Cooperative Manipulators. IEEE ASME Transactions on Mechatronics 12 (2007b) 2, pp. 109–125.

[Guldner et al. 2003]

Guldner, J.; Stelter, J.; Verl, A.: Kraft-/Momentenregelung für Industrieroboter zur Durchführung komplexer Montageapplikationen. wt Werkstattstechnik online 93 (2003), pp. 632–635.

[Haegele et al. 2008]

Haegele, M.; Nilsson, K.; Pires, J.: Industrial Robots, Springer 2008, Chapter 42.

[Han & Sun 2008]

Han, G.; Sun, M.: Optimum Path Planning of Robotic Free Abrasive Polishing Process. In: Proceedings of the First International Conference on Intelligent Robotics and Applications: Part II. Springer-Verlag 2008. p. 915.

[Hashimoto & Ohya 2008]

Hashimoto, Y.; Ohya, T.: Robot Coordinated Control Method and System. US Patent 7330777 (2008).

[Haun 2007]

Haun, M.: Handbuch der Robotik - Programmieren und Einsatz intelligenter Roboter. Springer Verlag 2007.

[Hayati 1986]

Hayati, S.: Hybrid Position/Force Control of Multi-Arm Cooperating Robots. In: 1986 IEEE International Conference on Robotics and Automation. IEEE 1986, Volume 3. pp. 82–89.

[Hayati 1988]

Hayati, S.: Position and Force Control of Coordinated Multiple Arms. IEEE Transactions on Aerospace and Electronic 24 (1988) 5.

[Hein et al. 2008]

Hein, B.; Hensel, M.; Worn, H.: Intuitive and Model-based On-line Programming of Industrial Robots: A Modular On-line Programming Environment. 2008 IEEE International Conference on Robotics and Automation (2008), pp. 3952–3957.

[Hesse & Malisa 2010]

Hesse, S.; Malisa, V.: Taschenbuch: Robotik-Montage-Handhabung. Hanser Verlag 2010.

[Hoffmann et al. 2009]

Hoffmann, A.; Angerer, A.; Ortmeier, F.; Vistein, M.; Reif, W.: Hiding Real-time: A New Approach for the Software Development of Industrial Robots. In: IEEE/RSJ International Conference on Intelligent Robots and Systems 2009. IEEE 2009. pp. 2108–2113.

[Hogan 1985]

Hogan, N.: Impedance Control: An Approach to Manipulation Parts I,II and III. ASME Journal of Dynamic Systems, Measurement and Control 107 (1985) 1, pp. 1–24.

[Hogan 1989]

Hogan, N.: Controlling Impedance at the Man/Machine Interface. In: 1989 IEEE International Conference on Robotics and Automation, 1989. Proceedings. IEEE Comput. Soc. Press 1989. pp. 1626–1631.

[Hokayem & Spong 2006]

Hokayem, P.; Spong, M.: Bilateral Teleoperation: An Historical Survey. Automatica 42 (2006) 12, pp. 2035–2057.

[Holzbock 1996]

Holzbock, W.: Robotic Technology Principles and Practice. Van Nostrand Reinhold Company 1996.

[Homebrew Channel 2011]

Homebrew Channel: WiiUse Library, 2011, `http://wiibrew.org/wiki/Wiiuse`, Date Accessed: 02.03.2011, 2011.

[Hoshizaki & Bopp 1990]

Hoshizaki, J.; Bopp, E.: Robot Applications Design Manual. John Wiley & Sons Inc. 1990.

[Hsu 1989]

Hsu, P.: Control of Multi-manipulator Systems - Trajectory Tracking, Load Distribution, Internal Force Control, and Decentralized Architecture. In: 1989 IEEE International Conference on Robotics and Automation, 1989. pp. 1234–1239.

[Huang et al. 2004]

Huang, L.; Ge, S.; Lee, T.: An Adaptive Impedance Control Scheme for Constrained Robots. International Journal of Computers, Systems and Signals 5 (2004) 2, pp. 17–26.

[Hubele 2009]

Hubele, J.: Implementierung und Validierung einer Steuerungskonzeption von kooperierenden Robotern. Diplomarbeit, Technische Universität München (TUM), 2009.

[Humburger 1997]

Humburger, R.: Konzeption eines Systems zur aufgabenorientierten Roboterprogrammierung. PhD Thesis, RWTH Aachen. Aachen, Shaker Verlag, 1997.

[id Software 2011]

id Software: id Tech 3, 2011, `http://www.idsoftware.com/`, Date Accessed: 28.02.2011, 2011.

[IEEE 802.3 ETHERNET WORKING GROUP 2011]

IEEE 802.3 ETHERNET WORKING GROUP: IEEE 802.3 ETHERNET, 2011, `http://www.ieee802.org/3/`, Date Accessed: 01.03.2011, 2011.

[Inaba & Sakakibara 2009]

Inaba, Y.; Sakakibara, S.: Intelligent Industrial Robots, Springer 2009.

[International Federation of Robotics - IFR Statistical Department 2007]

International Federation of Robotics - IFR Statistical Department: World Robotics Industrial Robots Report 2007. Technical Report, IFR Statistical Department / VDMA Robotics & Automation, 2007.

[International Federation of Robotics - IFR Statistical Department 2009]

International Federation of Robotics - IFR Statistical Department: World Robotics Industrial Robots Report 2009. Technical Report, IFR Statistical Department / VDMA Robotics & Automation, 2009.

[Ishida 1977]

Ishida, T.: Force Control in Coordination of Two Arms. In: Proc. 5th Int. Conf. on Artificial Intelligence. Citeseer 1977. pp. 411–415.

[ISO 15187 2002]

ISO 15187: ISO 15187: 2000 Manipulating Industrial Robots - Graphical User Interfaces for Programming and Operation of Robots (GUI-R), 2002, 2002.

[Jankowski et al. 1993]

Jankowski, K.; ElMaraghy, H.; ElMaraghy, W.: Dynamic Coordination of Multiple Robot Arms With Flexible Joints. The International Journal of Robotics Research 12 (1993) 6, pp. 505–528.

[Jorg et al. 2000]

Jorg, S.; Langwald, J.; Stelter, J.; Hirzinger, G.; Natale, C.: Flexible Robot-Assembly Using a Multi-Sensory Approach. Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (2000), pp. 3687–3694.

[Kamnik et al. 1998]

Kamnik, R.; Matko, D.; Bajd, T.: Application of Model Reference Adaptive Control to Industrial Robot Impedance Control. Journal of Intelligent and Robotic Systems (1998) 1, pp. 153–163.

[Kanitani 1993]

Kanitani, K.: Industrial Robot Synchronous Control Method and Apparatus. US Patent 5254923 (1993).

[Kawasaki et al. 2006]

Kawasaki, H.; Ueki, S.; Ito, S.: Decentralized Adaptive Coordinated Control of Multiple Robot Arms without using a Force Sensor. Automatica 42 (2006) 3, pp. 481–488.

[Kettu et al. 2009]

Kettu, T.; Wall, A.; Mustapic, G.; Eriksson, P.: Control System for Controlling the Movements of a Plurality of Mechanical Units. US Patent Application Publication 20090132057 (2009).

[Khalil & Dombre 2002]

Khalil, W.; Dombre, E.: Modeling, Identification and Control of Robots. 3. Edition. Taylor & Francis, Inc. 2002.

[Khatib 1987]

Khatib, O.: A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE Journal on Robotics and Automation 3 (1987) 1, pp. 43–53.

[Khatib et al. 1996a]

Khatib, O.; Yokoi, K.; Chang, K.; Ruspini, D.; Holmberg, R.; Casal, A.: Decentralized Cooperation Between Multiple Manipulators. In: 5th IEEE International Workshop on Robot and Human Communication, 1996., 1996a. pp. 183–188.

[Khatib et al. 1996b]

Khatib, O.; Yokoi, K.; Chang, K.; Ruspini, D.; Holmberg, R.; Casal, A.; Baader, A.: Force Strategies for Cooperative Tasks in Multiple Mobile Manipulation Systems. In: Robotics Research International Symposium, 1996b, Volume 7.

[Kim et al. 1997]

Kim, C.; Park, S.; Park, C.: Robust Position, Motion-Inducing Force, and Internal Force Control for Multi-Robot System. In: IEEE International Workshop on Robot and Human Communication, RO-MAN'97, 1997. pp. 82–87.

[Kim & Zheng 1991]

Kim, K.; Zheng, Y.: Unknown Load Distribution of Two Industrial Robots. In: IEEE International Conference on Robotics and Automation, 1991. pp. 992–997.

[Koivo & Unseren 1991]

Koivo, A.; Unseren, M.: Reduced Order Model and Decoupled Control Architecture for Two Manipulators Holding a Rigid Object. Journal of Dynamic Systems, Measurement, and Control 113 (1991) 4, p. 646.

[Kosuge & Hirata 2005]

Kosuge, K.; Hirata, Y.: Coordinated Motion Control of Multiple Manipulators, CRC Press 2005, Chapter 20.

[Kosuge et al. 1994]

Kosuge, K.; Yoshida, H.; Fukuda, T.; Sakai, M.; Kanitani, K.; Hariki, K.: Unified Control for Dynamic Cooperative Manipulation. In: IEEE/RSJ/GI International Conference on Intelligent Robots and Systems Advanced Robotic Systems and the Real World, IROS'94, 1994, Volume 2.

[Kosuge et al. 1997]

Kosuge, K.; Hashimoto, S.; Takeo, K.: Coordinated Motion Control of Multiple Robots Manipulating a Large Object. In: Intelligent Robots and Systems, IROS., 1997. pp. 208–213.

[Kreuzer et al. 1994]

Kreuzer, E.; Lugtenburg, J.-B.; Meiß ner, H.-G.; Truckenbrodt, A.: Industrieroboter Technik, Berechnung und anwendungsorientierte Auslegeung. Springer Verlag 1994.

[Krüger & Surdilovic 2008]

Krüger, J.; Surdilovic, D.: Robust Control of Force-coupled Human-Robot-Interaction in Assembly Processes. CIRP Annals-Manufacturing Technology 57 (2008) 1, pp. 41–44.

[Krüger et al. 2006]

Krüger, J.; Bernhardt, R.; Surdilovic, D.: Intelligent Assist Systems for Flexible Assembly. CIRP Annals-Manufacturing Technology 55 (2006) figure 1.

[Krüger et al. 2009]

Krüger, J.; Lien, T.; Verl, A.: Cooperation of Human and Machines in Assembly Lines. Annals of the CIRP 58 (2009) 2.

[Kuc et al. 1995]

Kuc, T.; Lee, J.; Park, B.: Learning Control of Cooperating Robot Manipulators Handling an Unknown Object. In: IEEE International Conference on Man and Cybernetics. IEEE 1995, Volume 4. pp. 3334–3339.

[KUKA GmbH 2011a]

KUKA GmbH: KUKA Robot KR 6-2, 2011a, http://www.kuka-robotics.com/germany/de/products/industrial_robots/low/kr6_2/, Date Accessed: 01.03.2011, 2011a.

[KUKA GmbH 2011b]

KUKA GmbH: KUKA Controller KR C2, 2011b, http://www.kuka-robotics.com/germany/de/products/controllers/kr_c2_ed05/?WBCMODE=PresentationUnpublishedprofilecsSeminarDetail, Date Accessed: 01.03.2011, 2011b.

[KUKA GmbH 2011c]

KUKA GmbH: KUKA.SIM, 2011c, http://www.kuka-robotics.com/en/products/software/kuka_sim/, Date Accessed: 22.02.2011, 2011c.

[Kumar & Garg 2005]

Kumar, M.; Garg, D.: Neuro-fuzzy Control Applied to Multiple Cooperating Robots. Industrial Robot: An International Journal 32 (2005) 3, pp. 234–239.

[Kumar & Gardner 1990]

Kumar, V.; Gardner, J.: Kinematics of Redundantly Actuated Closed Chains. IEEE Transactions on Robotics and Automation 6 (1990) 2, pp. 269–274.

[Kume et al. 2002a]

Kume, Y.; Hirata, Y.; Wang, Z.; Kosuge, K.: Decentralized Control of Multiple Mobile Manipulators Handling a Single Object in Coordination. In: IEEE/RSJ International Conference on Intelligent Robots and System, 2002, Lausanne, Switzerland 2002a, Volume 3.

[Kume et al. 2002b]

Kume, Y.; Hirata, Y.; Wang, Z.; Kosuge, K.: Decentralized Control of Multiple Mobile Manipulators Handling a Single Object in Coordination. IEEE/RSJ International Conference on Intelligent Robots and System (2002b), pp. 2758–2763.

[Kurfess 2005]

Kurfess, T. (Editor): Robotics and Automation Handbook. CRC Press 2005.

[Kurth 2005]

Kurth, J.: Flexible Produktionssysteme durch kooperierende Roboter. wt Werkstattstechnik Online - Industrieroboter, Automatisierung 95 (2005), pp. 81–84.

[Lange & Hirzinger 2005]

Lange, F.; Hirzinger, G.: Stability Preserving Sensor-Based Control for Robots with Positional Interface. In: IEEE International Conference on Robotics and Automation ICRA 2005. Barcelona, Spain: IEEE 2005, Volume 2. p. 1700.

[Lapham 2005]

Lapham, J.: Automation equipment control system. US Patent 6922611 (2005).

[Laurig & Vedder 1998]

Laurig, W.; Vedder, J.: Ergonomics, Geneva : International Labor Office 1998, Chapter 29.

[Lee & Spong 2005]

Lee, D.; Spong, M.: Bilateral Teleoperation of Multiple Cooperative Robots over Delayed Communication Networks: Theory. In: IEEE International Conference on Robotics and Automation, ICRA 2005, Barcelona, Spain 2005. p. 05.

[Lee et al. 2005]

Lee, D.; Martinez-Palafox, O.; Spong, M.: Bilateral Teleoperation of Multiple Cooperative Robots over Delayed Communication Networks: Application. In: Proceedings of the IEEE International Conference on Robotics and Automation. Barcelona, Spain: IEEE 2005. April, pp. 366–371.

[Lee & Lee 1991]

Lee, S.; Lee, H.: Generalized Impedance of Manipulators: Its Application to Force and Position Control. In: Fifth International Conference on Advanced Robotics Robots in Unstructured Environments. IEEE 1991. pp. 1477–1480.

[Lego Group 2011]

Lego Group: LEGO Mindstorms NXT, 2011, `http://mindstorms.lego.com/en-us/Default.aspx`, Date Accessed: 12.03.2011, 2011.

[Lewis et al. 2003]

Lewis, F.; Dawson, D.; Abdallah, C.: Robot Manipulator Control Theory and Practice. 2. Edition. CRC Press 2003.

[Lewis & Jacobson 2007]

Lewis, M.; Jacobson, J.: Game Engines in Scientific Research. Communications of the ACM 50 (2007) 7, p. 30.

[Lin et al. 2000]

Lin, J.; Wu, Y.; Huang, T.: Modeling the Constraints of Human Hand Motion. In: Workshop on Human Motion. Los Alamitos, CA , USA: IEEE Comput. Soc 2000. pp. 121–126.

[Lin & Huang 1997]

Lin, S.; Huang, A.: Position-Based Fuzzy Force Control for Dual Industrial Robots. Journal of Intelligent and Robotic Systems (1997), pp. 393–409.

[Liu & Abdel-Malek 2000]

Liu, J.-F.; Abdel-Malek, K.: Robust Control of Planar Dual-Arm Cooperative Manipulators. Robotics and Computer-Integrated Manufacturing 16 (2000) 2-3, pp. 109–119.

[Lopes & Almeida 2006]

Lopes, A.; Almeida, F.: Acceleration Based Force-Impedance Control of a 6-DOF Parallel Robotic Manipulator. In: Conference on Computational Cybernetics, 2006. ICCC, 2006.

[Luh & Zheng 1987]

Luh, J.; Zheng, Y.: Constrained Relations between Two Coordinated Industrial Robots for Motion Control. The International Journal of Robotics Research 6 (1987) 3, pp. 60–70.

[MacDonald et al. 2003]

MacDonald, B.; Yuen, D.; Wong, S.; Woo, E.; Gronlund, R.; Collett, T.; Trepanier, F.; Biggs, G.: Robot Programming Environments. In: ENZCon2003 10th Electronics New Zealand Conference, 2003. pp. 1–2.

[Maeda et al. 2008]

Maeda, Y.; Ushioda, T.; Makita, S.: Easy Robot Programming for Industrial Manipulators by Manual Volume Sweeping. 2008 IEEE International Conference on Robotics and Automation (2008), pp. 2234–2239.

[Malett et al. 2004]

Malett, J.; Chang, D.; Rosenberg, L.; Braun, A.; Martin, K.; Beamer, J.: Enhanced Cursor Control Using Interface Devices. US Patent 6816148 (2004).

[Mali & Munih 2003]

Mali, U.; Munih, M.: Real-time control of haptic device using personal computer. In: The IEEE Region 8 EUROCON 2003. Computer as a Tool. Ieee 2003. pp. 401–404.

[Massie & Salisbury 1994]

Massie, T.; Salisbury, J.: The Phantom Haptic Interface: A Device for Probing Virtual Objects. Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems 55 (1994) 1, pp. 295–300.

[Mastascusa 2005]

Mastascusa, E.: Exploring Classical Control Systems, 2005, http://www.facstaff.bucknell.edu/mastascu/eControlHTML/CourseIndex.html, Date Accessed: 06.03.2011, 2005.

[MathWorks 2011]

MathWorks: Matlab-Simulink, 2011, http://www.mathworks.com/products/simulink/, Date Accessed: 01.03.2011, 2011.

[Matko et al. 1999]

Matko, D.; Kamnik, R.; Bajd, T.: Adaptive Impedance Force Control of an Industrial Manipulator. In: IEEE International Symposium on Industrial Electronics ISIE ’99. IEEE 1999. pp. 129–133.

[Meister et al. 1998]

Meister, D.; Pokrandt, P.; Both, A.: Milling Accuracy in Robot Assisted Orthopaedic Surgery. IECON ’98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (1998), pp. 2502–2505.

[Microsoft 2011]

Microsoft: Microsoft Robotics Developer Studio, 2011, http://www.microsoft.com/robotics/, Date Accessed: 27.02.2011, 2011.

[Microsoft Corporation 2011]

Microsoft Corporation: Microsoft Robotics Developer Studio 2008 R3, 2011, http://www.microsoft.com/robotics/, Date Accessed: 12.03.2011, 2011.

[Moosavian & Ashtiani 2008]

Moosavian, S.; Ashtiani, H.: Cooperation of Robotic Manipulators using Non-Model-Based Multiple Impedance Control. Industrial Robot: An International Journal 35 (2008) 6, pp. 549–558.

[Morrow et al. 1995]

Morrow, J.; Nelson, B.; Khosla, P.: Vision and Force Driven Sensorimotor Primitives for Robotic Assembly Skills. Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots. 3 (1995), pp. 234–240.

[Munzert 2009]

Munzert, U.: Bahnplanungsalgorithmen für das robotergestützte Remote-Laserstrahlschweißen. PhD Thesis, Technische Universität München (TUM), 2009.

[Nagchaudhuri & Garg 2001]

Nagchaudhuri, A.; Garg, D.: Adaptive Control and Impedance Control for Dual Robotic Arms Manipulating a Common Heavy Load. 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (2001) July, pp. 683–688.

[Neto et al. 2009]

Neto, P.; Pires, J.; Moreira, A.: High-level Programming for Industrial Robotics: Using Gestures, Speech and Force Control, 2009.

[Newman & Wei 2002]

Newman, W.; Wei, J.: Improving Robotic Assembly Performance Through Autonomous Exploration. In: Proceedings 2002 IEEE International Conference on Robotics and Automation. Washington DC, USA: IEEE 2002. pp. 3303–3308.

[Newman et al. 1999]

Newman, W.; Branicky, M.; Podgurski, H.; Chhatpar, S.; Swaminathan, J.: Force-responsive Robotic Assembly of Transmission Components. Proceedings 1999 IEEE International Conference on Robotics and Automation (1999) May, pp. 2096–2102.

[Newman et al. 2001]

Newman, W.; Branicky, M.; Pao, Y.: Intelligent Strategies for Compliant Robotic Assembly. In: 11th Yale Workshop on Adaptive and Learning Systems, 2001. pp. 139–146.

[Newton 1846]

Newton, I.: The Principia: Mathematical Principles of Natural Philosophy [English Translation]. Daniel Adee 1846.

[Niedermayr 2005]

Niedermayr, C.: Hauptseminar Automotive Software Engineering Testen , Rapid Prototyping und X - in the loop, 2005.

[Nikolaus Gebhardt et al. 2011]

Nikolaus Gebhardt et al.: IRRLICHT, 2011, http://irrlicht.sourceforge.net/, Date Accessed: 28.02.2011, 2011.

[Nintendo 2011]

Nintendo: Wii, 2011, http://www.nintendo.com/wii, Date Accessed: 01.03.2011, 2011.

[Nof & Rajan 1993]

Nof, S.; Rajan, V.: Automatic Generation of Assembly Constraints and Cooperation Task Planning. CIRP Annals-Manufacturing Technology 42 (1993) 1, pp. 1–4.

[Nokia 2011a]

Nokia: QT 4.7 Documentation: Signals and Slots, 2011a, http://doc.qt.nokia.com/4.7/signalsandslots.html, Date Accessed: 01.03.2011, 2011a.

[Nokia 2011b]

Nokia: QT Application Framework, 2011b, http://qt.nokia.com/products, Date Accessed: 22.02.2011, 2011b.

[Novint 2011]

Novint: Falcon, 2011, http://www.novint.com/index.php?option=com_content&view=article&id=39&Itemid=123, Date Accessed: 01.03.2011, 2011.

[OPAL-RT Technologies Inc. 2011]

OPAL-RT Technologies Inc.: RT-LAB Professional, 2011, http://www.opal-rt.com/product/rt-lab-professional, Date Accessed: 01.03.2011, 2011.

[Osumi & Arai 1994]

Osumi, H.; Arai, T.: Cooperative Control Between Two Position-Controlled Manipulators. In: IEEE International Conference on Robotics and Automation. IEEE Comput. Soc. Press 1994. pp. 1509–1514.

[Osumi et al. 1995]

Osumi, H.; Arai, T.; Fukuoka, T.; Moriyama, K.; Torii, H.: Cooperative Control of Two Industrial Robots with Force Control Devices. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, 1995. pp. 550–555.

[Otera et al. 1993]

Otera, N.; Nishimura, T.; Takahashi, T.: Robot Control System for Controlling a Set of Industrial Robots for Cooperative Operation. US Patent 5204942 (1993).

[Owen et al. 2008]

Owen, W.; Croft, E.; Benhabib, B.: Stiffness Optimization for Two-Armed Robotic Sculpting. Industrial Robot: An International Journal 35 (2008) 1, pp. 46–57.

[Park & Park 2008]

Park, C.; Park, K.: Design and Kinematics Analysis of Dual Arm Robot Manipulator for Precision Assembly. 2008 6th IEEE International Conference on Industrial Informatics (2008), pp. 430–435.

[Park et al. 2008]

Park, K.-T.; Park, C.-H.; Shin, Y.-J.; Mo, Y.-S.; Jeong, D.-C.; Shin, Y.: Robot Application for Assembly Process of Engine Part. 2008 International Conference on Control, Automation and Systems (2008), pp. 2107–2110.

[Pelletier & Doyon 1994]

Pelletier, M.; Doyon, M.: On the Implementation and Performance of Impedance Control on Position Controlled Robots. IEEE International Conference on Robotics and Automation (1994), pp. 1228–1233.

[Pfeffer & Cannon 1993]

Pfeffer, L.; Cannon, R.: Experiments with a Dual-armed, Cooperative, Flexible-Drivetrain Robot System. In: IEEE International Conference on Robotics and Automation, 1993. pp. 601–608.

[Pfeiffer 1992]

Pfeiffer, F.: Combined Path and Force Control for Elastic Manipulators. Mechanical Systems and Signal Processing 6 (1992) 3, pp. 237–249.

[Pires 2008a]

Pires, J.: New Challenges for Industrial Robotic Cell Programming. Industrial Robot: An International Journal 36 (2008a) 1.

[Pires 2008b]

Pires, J.: From the Coworker to the Cognitive Factory Scenario: New Developments for SME Manufacturing. In: 4th International Congress on Welding Technology and Manufacturing, Saltillo, Mexico 2008b.

[Pires et al. 2002]

Pires, J.; Ramming, J.; Rauch, S.; Araújo, R.: Force/Torque Sensing Applied to Industrial Robotic Deburring. Sensor Review 22 (2002) 3, pp. 232–241.

[Pongrac et al. 2007]

Pongrac, H.; Reiter, A.; Faerber, B.; Zaeh, M.: Evaluation of Joystick Sensitivity in Telemanipulative Microassembly. In: 2007 IEEE International Workshop on Haptic, Audio and Visual Environments and Games. Ottawa, Canada: IEEE 2007. October, pp. 71–75.

[Postel 1980]

Postel, J.: RFC 768: User Datagram Protocol (UDP), 1980, ftp://ftp.math.utah.edu/pub/rfc/rfc768.txt, Date Accessed: 02.03.2011, 1980.

[Postel 1981]

Postel, J.: RFC 791: Internet Protocol (IP), 1981, ftp://ftp.math.utah.edu/pub/rfc/rfc760.txt,ftp://ftp.math.utah.edu/pub/rfc/rfc791.txt, Date Accessed: 02.03.2011, 1981.

[Pretschner et al. 2007]

Pretschner, A.; Broy, M.; Kruger, I.; Stauner, T.: Software Engineering for Automotive Systems: A Roadmap. Future of Software Engineering (FOSE '07) (2007), pp. 55–71.

[Qiao & Zhu 1999]

Qiao, B.; Zhu, J.: Learning Force Control for Position Controlled Robotic Manipulator. CIRP Annals-Manufacturing Technology 48 (1999), pp. 7–10.

[QNX Software Systems 2011]

QNX Software Systems: QNX Neutrino RTOS, 2011, http://www.qnx.com/products/neutrino-rtos/neutrino-rtos.html, Date Accessed: 01.03.2011, 2011.

[QUANCOM Informationssysteme GmbH 2011a]

QUANCOM Informationssysteme GmbH: USBREL8 - QUANCOM USB Relaiskarte, 2011a, http://www.quancom.de/qprod01/homed.htm, Date Accessed: 02.03.2011, 2011a.

[QUANCOM Informationssysteme GmbH 2011b]

QUANCOM Informationssysteme GmbH: QLIB Library, 2011b, http://www.quancom.de/qprod01/homed.htm, Date Accessed: 02.03.2011, 2011b.

[Radi et al. 2010]

Radi, M.; Reiter, A.; Zaidan, S.; Nitsch, V.; Färber, B.; Reinhart, G.: Telepresence in Industrial Applications: Implementation Issues for Assembly Tasks. Presence 19 (2010) 5.

[Raibert & Craig 1981]

Raibert, M.; Craig, J.: Hybrd position/force control of Manipulators. ASME Journal of Dynamic Systems, Measurement and Control 102 (1981) 2, pp. 126–133.

[Rampersad 1995]

Rampersad, H.: State of the Art in Robotic Assembly. Industrial Robot: An International Journal 22 (1995) 2, pp. 10–13.

[Rashidy 2008]

Rashidy, H.: Knowledge-based Quality Control in Manufacturing Processes with Application to the Automotive Industry. PhD Thesis, Technische Universität München (TUM), Munich, 2008.

[Reinhart & Zaidan 2009]

Reinhart, G.; Zaidan, S.: A Generic Framework for Workpiece-based Programming of Cooperating Industrial Robots. In: Mechatronics and Automation 2009, Changchun, China 2009. pp. 37–42.

[Reinhart et al. 2007]

Reinhart, G.; Vogl, W.; Kresse, I.: A Projection-based User Interface for Industrial Robots. In: Virtual Environments, Human-Computer Interfaces, 2007. June, pp. 25–27.

[Reinhart et al. 2008a]

Reinhart, G.; Munzert, U.; Vogl, W.: A Programming System for Robot-based Remote-Laser-Welding with Conventional Optics. CIRP Annals-Manufacturing Technology 57 (2008a) 1, pp. 37–40.

[Reinhart et al. 2008b]

Reinhart, G.; Werner, J.; Lange, F.: Robot Based System for the Automation of Flow Assembly Lines. Production Engineering 3 (2008b) 1, pp. 121–126.

[Reinhart et al. 2008c]

Reinhart, G.; Zaidan, S.; Radi, M.; Wittmann, W.: On The Feasibility of a PBD Approach for a Force Controlled Robot Production Cell. In: 2nd CIRP Conference on Assembly Technologies & Systems, Toronto, Canada 2008c. pp. 368–376.

[Reinhart et al. 2010a]

Reinhart, G.; Radi, M.; Zaidan, S.: Workspace Scaling Techniques for Haptic Robot Guidance Systems. In: Lien, T. K. (Editor): 3rd CIRP Conference on Assembly Technologies and Systems (CATS2010), Trondheim, Norway 2010a.

[Reinhart et al. 2010b]

Reinhart, G.; Zaidan, S.; Hubele, J.: Is Force Monitoring in Cooperating Industrial Robots Necessary? In: ISR/ROBOTIK 2010. Munich, Germany: VDE Verlag 2010b.

[Robertsson et al. 2006]

Robertsson, A.; Olsson, T.; Johansson, R.; Blomdell, A.; Nilsson, K.; Haage, M.; Lauwers, B.; Baerdemaeker, H.; Brogardh, T.; Brantmark, H.: Implementation of Industrial Robot Force Control Case Study: High Power Stub Grinding and Deburring. IEEE/RSJ International Conference on Intelligent Robots and Systems (2006), pp. 2743–2748.

[Rückel 2006]

Rückel, V.: Verkürzung von Montageprozessen durch den Einsatz CAD-gestützter Ablaufplanung und kooperierender Industrieroboter. PhD Thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2006.

[Rückel & Feldmann 2005]

Rückel, V.; Feldmann, K.: Komplettmontage mit kooperierenden Robotern. wt Werkstattstechnik online 95 (2005), pp. 85–90.

[Russell & Norvig 2009]

Russell, S.; Norvig, P.: Artificial Intelligence: A Modern Approach. 3. Edition. Pearson 2009.

[Sanvido 2002]

Sanvido, M.: Hardware-in-the-loop Simulation Framework. PhD Thesis, Swiss Federal Institute of Technology, 2002.

[Sastry & Bodson 1994]

Sastry, S.; Bodson, M.: Adaptive Control: Stability, Convergence, and Robustness. Prentice-Hall Advanced Reference Series 1994.

[Schauss et al. 2009]

Schauss, T.; Unterhinninghofen, U.; Buss, M.: Unlimited Workspace - Coupling a Mobile Haptic Interface with a Mobile Teleoperator, Springer Berlin Heidelberg 2009, Chapter 4.

[Scheibl 2003]

Scheibl, H.-J.: Visual C++.NET. Für Einsteiger und Fortgeschrittene. Hanser Fachbuchverlag 2003.

[Schillhuber & Ulbrich 2008]

Schillhuber, G.; Ulbrich, H.: Haptic Simulation of Deformable Bodies with Consideration of the Human Sensation of Continuous Forces. In: The 16th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems, Reno, Nevada, USA 2008.

[Schmitt et al. 2008]

Schmitt, R.; Pavim, A.; Brecher, C.; Pyschny, N.; Loosen, P.; Funck, M.; Dolkemeyer, J.; Morasch, V.: Flexibel automatisierte Montage von Festkörperlasern. wt Werkstattstechnik 98 (2008), pp. 955–960.

[Schmitt et al. 2010]

Schmitt, R.; Schönberg, A.; Damm, B.: Indoor-GPS Based Robots as a Key Technology for Versatile Production. In: ISR/ROBOTIK 2010. Munich, Germany: VDE Verlag 2010.

[Schneider & Cannon 1992]

Schneider, S.; Cannon, R.: Object Impedance Control for Cooperative Manipulation: Theory and Experimental Results. IEEE Transactions on Robotics and Automation 8 (1992) 3, pp. 383–394.

[Schneider & Cannon 1993]

Schneider, S.; Cannon, R.: Experimental Object- Level Strategic Control With Cooperating Manipulators. The International Journal of Robotics Research 12 (1993) 4, pp. 338–350.

[Scholzreiter & Freitag 2007]

Scholzreiter, B.; Freitag, M.: Autonomous Processes in Assembly Systems. CIRP Annals-Manufacturing Technology 56 (2007) 2, pp. 712–729.

[Schraft & Meyer 2006]

Schraft, R.; Meyer, C.: The Need for an Intuitive Teaching Method for Small and Medium Enterprises. VDI BERICHTE 1956 (2006), p. 95.

[Schreck 1998]

Schreck, G.: Advanced Man-Machine Interfaces for Robot System Applications. In: 24th Annual Conference of the IEEE industrial Electronics Society, 1998. IECON '98., Aachen, Germany 1998. pp. 1914 – 1917.

[Schunk 2011]

Schunk: Schunk FTC-50-80, 2011, http://www.de.schunk.com/, Date Accessed: 01.03.2011, 2011.

[Sciavicco & Siciliano 2005]

Sciavicco, L.; Siciliano, B.: Modelling and Control of Robot Manipulators. 5. Edition. Springer 2005.

[Sensable Technologies Inc. 2011]

Sensable Technologies Inc.: PHANTOM Desktop Haptic Device, 2011, http://www.sensable.com/haptic-phantom-desktop.htm, 2011.

[Seraji 1988]

Seraji, H.: Dual-Arm Manipulators with Adaptive Control. US Patent 5023808 (1988).

[Setoodeh et al. 2006]

Setoodeh, P.; Sirouspour, S.; Shahdi, S.: Discrete-time Multi-Model Control for Cooperative Teleoperation Under Time Delay. In: IEEE International Conference on Robotics and Automation, ICRA, Orlando, Florida 2006. May, pp. 2921–2926.

[Siemens AG 2011]

Siemens AG: Tecnomatix Robcad, 2011, http://www.plm.automation.siemens.com/en_us/products/tecnomatix/robotics_automation/robcad/index.shtml, Date Accessed: 22.02.2011, 2011.

[Silicon Graphics 2011a]

Silicon Graphics: GLUT - The OpenGL Utility Tool, 2011a, http://www.opengl.org/resources/libraries/glut/, Date Accessed: 28.02.2011, 2011a.

[Silicon Graphics 2011b]

Silicon Graphics: OpenGL, 2011b, http://www.opengl.org/, Date Accessed: 28.02.2011, 2011b.

[Sirouspour 2005a]

Sirouspour, S.: Modeling and Control of Cooperative Teleoperation Systems. IEEE Transactions on Robotics 21 (2005a) 6, pp. 1220–1225.

[Sirouspour 2005b]

Sirouspour, S.: Robust Control Design for Cooperative Teleoperation. In: IEEE International Conference on Robotics and Automation, ICRA, Barcelona, Spain 2005b, Volume pp. pp. 1133–1138.

[Sirouspour & Setoodeh 2005a]

Sirouspour, S.; Setoodeh, P.: Adaptive Nonlinear Teleoperation Control in Multi-Master/Multi-Slave Environments. In: IEEE Conference on Control Applications. Ieee 2005a. pp. 1263–1268.

[Sirouspour & Setoodeh 2005b]

Sirouspour, S.; Setoodeh, P.: Multi-operator/Multi-robot Teleoperation: An Adaptive Nonlinear Control Approach. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. Ieee 2005b. pp. 2506–2511.

[Smith 2006]

Smith, R.: Manual: Open Dynamics Engine V0.5 User Guide. ODE 2006, http://www.ode.org/.

[Smith 2011]

Smith, R.: ODE- Open Dynamics Engine, 2011, http://www.ode.org/, Date Accessed: 28.02.2011, 2011.

[Sony Computer Entertainment Inc. & Khronos Group 2011]

Sony Computer Entertainment Inc.; Khronos Group: COLLADA - Digital Asset and FX Exchange Schema, 2011, https://collada.org/mediawiki/index.php/COLLADA_-_Digital_Asset_and_FX_Exchange_Schema, Date Accessed: 28.02.2011, 2011.

[Spiller & Verl 2010]

Spiller, A.; Verl, A.: Superimposed Force/Torque-Control of Cooperating Robots. In: ISR/ROBOTIK 2010. Munich, Germany: VDE Verlag 2010.

[Spillner et al. 2008]

Spillner, R.; Thiemann, G.; Schilp, J.; Reinhart, G.: Challenges of the Demographic Change in the Field of Assembly. wt Werkstattstechnik online (2008).

[Spong et al. 2006]

Spong, M.; Hutchinson, S.; Vidyasagar, M.: Robot Modelling and Control. John Wiley & Sons Inc. 2006.

[SRI International 1988]

SRI International: Robot Design Handbook. McGraw-Hill Book Company 1988.

[Stallings 2006]

Stallings, W.: Data and Computer Communications. 8. Edition. Prentice Hall 2006.

[Stelzer et al. 2008]

Stelzer, M.; von Stryk, O.; Abele, E.; Bauer, J.; Weigold, M.: High Speed Cutting with Industrial Robots: Towards Model Based Compensation of Deviations. In: Proceedings of Robotik 2008, Munich, Germany 2008. pp. 1–6.

[Stevens 1999]

Stevens, W.: UNIX Network Programming. 2. Edition. Prentice Hall 1999.

[Stoddard et al. 2004]

Stoddard, K.; Kneifel, R.; Martin, D.; Mirza, K.; Chaffee, M.; Hagenauer, A.; Graf, S.: Method and Control System for Controlling a Plurality of Robots. US Patent 6804580 (2004).

[Stone 2000]

Stone, R.: Haptic Feedback: A Potted History, from Telepresence to Virtual Reality. In: The First International Workshop on Haptic Human-Computer Interaction, 2000. pp. 1–7.

[Suh & Shin 1988]

Suh, I.; Shin, K.: Coordination of Dual Robot Arms using Kinematic Redundancy. In: 1988 IEEE International Conference on Robotics and Automation, 1988. pp. 504–509.

[Sun & Mills 2002a]

Sun, D.; Mills, J.: Manipulating Rigid Payloads with Multiple Robots Using Compliant Grippers. IEEE/ASME Transactions on Mechatronics 7 (2002a) 1, pp. 23–34.

[Sun & Mills 2002b]

Sun, D.; Mills, J.: Adaptive Synchronized Control for Coordination of Multirobot Assembly Tasks. IEEE Transactions on Robotics and Automation 18 (2002b) 4, pp. 498–510.

[Sun et al. 1996]

Sun, D.; Shi, X.; Liu, Y.: Modeling and Cooperation of Two-arm Robotic System Manipulating a Deformable Object. In: EEE International Conference on Robotics and Automation. Minneapolis, Minnesota, USA: IEEE 1996, Volume 3. pp. 2346–2351.

[Sunnanbo 2003]

Sunnanbo, A.: Laser Feedback Control for Robotics in Aircraft Assembly. Master thesis, Linköpings Universitet, 2003.

[Surdilovic & Vukobratović 2002]

Surdilovic, D.; Vukobratović, M.: Control of Robotic Systems in Contact Tasks, CRC Press 2002, Chapter 23.

[Surdilovic et al. 2001]

Surdilovic, D.; Grassini, F.; De Bartolomei, M.: Synthesis of Impedance Control for Complex Co-operating Robot Assembly Task. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Como, Italy 2001. July, pp. 1181–1187.

[Szewczyk et al. 1996]

Szewczyk, J.; Morel, G.; Bidaud, P.: Optimization of Cooperating System Performances under Distributed Impedance Control. In: Computational Engineering in Systems Applications Multiconference, Lille, France 1996. section 3.

[Takahashi & Kosaka 2001]

Takahashi, H.; Kosaka, T.: Control System for Synchronously Cooperative Operation of Plurality of Robots. US Patent 6330493 (2001).

[Tan et al. 1997]

Tan, T.; Ang Jr, M.; Teo, C.: Non-model-based impedance control of an industrial robot. In: The 1997 8 th International Conference on Advanced Robotics, ICAR'97, Monterey, California 1997. pp. 407–412.

[Tanaka & Tsuji 2004]

Tanaka, Y.; Tsuji, T.: On-line Learning of Robot Arm Impedance Using Neural Networks. 2004 IEEE International Conference on Robotics and Biomimetics (2004), pp. 941–946.

[Tanner 2005]

Tanner, N.: Improving Perception in Time-delayed Telerobotics. The International Journal of Robotics Research 24 (2005) 8, pp. 631–644.

[Tarn et al. 1986]

Tarn, T.; Bejczy, A.; Yun, X.: Coordinated Control of Two Robot Arms. In: IEEE International Conference on Robotics and Automation, 1986. pp. 1193 – 1202.

[Terada 1998]

Terada, H.: Collision Free Control of a Multi-Arm Testing Robot. CIRP Annals-Manufacturing Technology 47 (1998), pp. 1–4.

[The OGRE Team 2011]

The OGRE Team: OGRE : Object-Oriented Graphics Rendering Engine, 2011, http://www.ogre3d.org/, Date Accessed: 28.02.2011, 2011.

[The Open Group 1997]

The Open Group: Shared Memory: The Single UNIX Specification - Version 2, 1997, http://pubs.opengroup.org/onlinepubs/007908799/xsh/shm_open.html, Date Accessed: 04.03.2011, 1997.

[Tzafestas et al. 1998]

Tzafestas, C.; Prokopiou, P.; Tzafestas, S.: Path Planning and Control of a Cooperative Three-Robot System Manipulating Large Objects. Journal of Intelligent and Robotic Systems 22 (1998) 2, pp. 99–116.

[Tzeranis et al. 2005]

Tzeranis, D.; Ishijima, Y.; Dubowsky, S.: Manipulation of Large Flexible Structural Modules by Space Robots Mounted on Flexible Structures. In: 8th International Symposium on Artifical Intelligence, Robotics and Automation in Space, Munich, Germany 2005, Volume 1.

[Uchiyama & Dauchez 1988]

Uchiyama, M.; Dauchez, P.: A Symmetric Hybrid Position/Force Control Scheme for the Coordination of Two Robots. IEEE International Conference on Robotics and Automation (1988), pp. 350–356.

[Uchiyama & Nakamura 1988]

Uchiyama, M.; Nakamura, Y.: Symmetric Hybrid Position/Force Control Of Two Cooperating Robot Manipulators. In: IEEE International Workshop on Intelligent Robots. IEEE 1988. pp. 515–520.

[Venkatesan & Karnan 2009]

Venkatesan, S.; Karnan, M.: Industrial Automation, 2009, http://www.articlesbase.com/college-and-university-articles/industrial-automation-1569437.html, Date Accessed: 01.03.2011, 2009.

[Vogl 2009]

Vogl, W.: Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern. PhD Thesis, Technische Universität München (TUM), 2009.

[von Collani et al. 1999]

von Collani, Y.; Scheering, C.; Zhang, J.; Knoll, A.: A Neuro-Fuzzy Solution for Integrated Visual and Force Control. In: IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems MFI'99. Taipei , Taiwan: Ieee 1999. pp. 135–140.

[Voyles et al. 1994]

Voyles, R.; Morrow, J.; Khosla, P.: A Comparison of Force Sensors. Technical Report, Advanced Manipulators Laboratory, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[Vukobratović et al. 2003]

Vukobratović, M.; Potkonjak, V.; Matijevic, V.: Dynamics of Robots with Contact Tasks. Kluwer 2003.

[Vukobratović et al. 2008]

Vukobratović, M.; Surdilovic, D.; Ekalo, Y.; Katic, D.: Dynamics and Robust Control of Robot-environment Interaction. World Scientific Publishing Co., Inc. 2008.

[Walker et al. 1989]

Walker, M.; Kim, D.; Dionise, J.: Adaptive Coordinated Motion Control of Two Manipulator Arms Dynamic Model of Two Coordinated. In: IEEE International Conference on Robotics and Automation 1989, 1989. pp. 1084 –1090.

[Weck & Brecher 2006]

Weck, M.; Brecher, C.: Werkzeugmaschinen 4: Automatisierung von Maschinen und Anlagen. Springer-Verlag Berlin Heidelberg 2006.

[Weck & Dammertz 1995]

Weck, M.; Dammertz, R.: OPERA - A New Approach to Robot Programming. CIRP Annals-Manufacturing Technology 44 (1995) 1, pp. 389–392.

[Whitney 1985]

Whitney, D.: Remote Center Compliance Gripper System. US Patent 4537557 (1985).

[Whitney 2004]

Whitney, D.: Mechanical Assemblies Their Design, Manufacture, and Role in Product Development. Oxford University Press, USA 2004.

[William Kneifel II & Stoddard 2004]

William Kneifel II, R.; Stoddard, K.: Coordinated Robot Control from Multiple Remote Instruction Sources. US Patent 6807461 (2004).

[Winkler 2006]

Winkler, A.: Ein Beitrag zur kraftbasierten Mensch-Roboter-Interaktion. PhD Thesis, Technische Universität Chemnitz, 2006.

[Winkler & Suchy 2007a]

Winkler, A.; Suchy, J.: Dynamic Force/Torque Measurement Using a 12DOF Sensor. IEEE/RSJ International Conference on Intelligent Robots and Systems (2007a), pp. 1870–1875.

[Winkler & Suchy 2007b]

Winkler, A.; Suchy, J.: Position Based Force Control of an Industrial Manipulator. In: 52nd Internationales Wissenschaftliches Kolloquium. Technische Universität Ilmenau 2007b.

[World Health Organization (WHO) 2007]

World Health Organization (WHO): International Classification of Functioning, Disability and Health (ICF), 2007, `http://www.who.int/classifications/icf/en/`, Date Accessed: 02.03.2011, 2007.

[World Wide Web Consortium 2011]

World Wide Web Consortium: XML - Extensible Markup Language, 2011, `http://www.w3.org/TR/REC-xml/`, Date Accessed: 28.02.2011, 2011.

[Wörn et al. 2005]

Wörn, H.; Hein, B.; Mages, D.; Denkena, B.; Apitz, R.; Kowalski, P.; Reimer, N.: Combining Manual Haptic Path Planning Of Industrial Robots With Automatic Path Smoothing. In: Proceedings of ICINCO, Barcelona, Spain 2005.

[YASKAWA Motoman Robotics 2011]

YASKAWA Motoman Robotics: MotoSim EG, 2011, http://www.motoman.com/products/software/program/MotoSimEG.php, Date Accessed: 22.02.2011, 2011.

[Yin et al. 1997]

Yin, Y.; Zhu, J.; Wei, Z.: Intelligent Strategy of Force & Position Parallel Control for a Robot. CIRP Annals-Manufacturing Technology 46 (1997) 1, pp. 279–282.

[Yong & Bonney 1999]

Yong, Y.; Bonney, M.: Off-Line Programming, John Wiley & Sons Inc. 1999, Chapter 19. 2. Edition.

[Yoshikawa 1991]

Yoshikawa, T.: Coordinated Dynamic Control for Multiple Robot Manipulators Handling an Object. In: Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments. Pisa , Italy: IEEE 1991, Volume 1. pp. 579–584.

[Yoshikawa et al. 1988]

Yoshikawa, T.; Sugie, T.; Tanaka, M.: Dynamic Hybrid Position/Force Control of Robot Manipulators - Controller Design and Experiment. IEEE Journal of Robotics and Automation 4 (1988) 6, pp. 699–705.

[Yun 1991]

Yun, X.: Modeling and Control of Two Constrained Manipulators. Journal of Intelligent and Robotic Systems 4 (1991) 4, pp. 363–377.

[Yun et al. 1989]

Yun, X.; Tarn, T.; Bejczy, A.: Dynamic Coordinated Control of Two Robot Manipulators. In: 28th Conference on Decision and Control, Tempa, Florida, USA 1989. December.

[Zäh et al. 2006]

Zäh, M.; Reiter, A.; Clarke, S.: Design, Construction and Analysis of a Telepresent Microassembly System. In: International Workshop on Human-Centered Robotic Systems (HCRS'06), Munich, Germany 2006.

[Zhang 2005]

Zhang, G.: Machining with Flexible Manipulator: Toward Improving Robotic Machining Performance. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (2005), pp. 1127–1132.

[Zhang et al. 2004]

Zhang, J.; Lumia, R.; Starr, G.; Wood, J.: Sharing Inertia Load Between Multiple Robots with Active Compliant Grippers Using Trajectory Pre-shaping. In: IEEE International Conference on Robotics and Automation, ICRA '04. New Orleans, USA: IEEE 2004. pp. 2574–2581.

[Zheng & Luh 1988]

Zheng, Y.; Luh, J.: Optimal Load Distribution for Two Industrial Robots Handling a Single Object. In: IEEE International Conference on Robotics and Automation, 1988. pp. 344–349.

[Zhu 2005]

Zhu, W.-H.: On Adaptive Synchronization Control of Coordinated Multirobots With Flexible Rigid Constraints. IEEE Transactions on Robotics 21 (2005) 3, pp. 520–525.

[Zivanovic & Vukobratović 2006]

Zivanovic, M.; Vukobratović, M.: Multi-Arm Cooperating Robots: Dynamics and Control. Springer-11647 2006.

[Zribi & Ahmad 1992]

Zribi, M.; Ahmad, S.: Adaptive Control for Multiple Cooperative Robot Arms. In: The 31st IEEE Conference on Decision and Control, Tucson, Arizona, USA 1992. pp. 1392–1398.

Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Arbeit selbstständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche gekennzeichnet.

Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt.

Garching, den 13.10.2011

Sherif Zaidan

iwb Forschungsberichte Band 1–121

Herausgeber: Prof. Dr.-Ing. J. Milberg und Prof. Dr.-Ing. G. Reinhart, Institut für Werkzeugmaschinen und Betriebswissenschaften der Technischen Universität München

Band 1–121 erschienen im Springer Verlag, Berlin, Heidelberg und sind im Erscheinungsjahr und den folgenden drei Kalenderjahren erhältlich im Buchhandel oder durch Lange & Springer, Otto-Suhr-Allee 26–28, 10585 Berlin

1 *Streifinger, E.*
Beitrag zur Sicherung der Zuverlässigkeit und Verfügbarkeit moderner Fertigungsmittel
1986 · 72 Abb. · 167 Seiten · ISBN 3-540-16391-3

2 *Fuchsberger, A.*
Untersuchung der spanenden Bearbeitung von Knochen
1986 · 90 Abb. · 175 Seiten · ISBN 3-540-16392-1

3 *Maier, C.*
Montageautomatisierung am Beispiel des Schraubens mit Industrierobotern
1986 · 77 Abb. · 144 Seiten · ISBN 3-540-16393-X

4 *Summer, H.*
Modell zur Berechnung verzweigter Antriebsstrukturen
1986 · 74 Abb. · 197 Seiten · ISBN 3-540-16394-8

5 *Simon, W.*
Elektrische Vorschubantriebe an NC-Systemen
1986 · 141 Abb. · 198 Seiten · ISBN 3-540-16693-9

6 *Büchs, S.*
Analytische Untersuchungen zur Technologie der Kugelbearbeitung
1986 · 74 Abb. · 173 Seiten · ISBN 3-540-16694-7

7 *Hunzinger, I.*
Schneiderodierte Oberflächen
1986 · 79 Abb. · 162 Seiten · ISBN 3-540-16695-5

8 *Pilland, U.*
Echtzeit-Kollisionsschutz an NC-Drehmaschinen
1986 · 54 Abb. · 127 Seiten · ISBN 3-540-17274-2

9 *Barthelmeß, P.*
Montagegerechtes Konstruieren durch die Integration von Produkt- und Montageprozeßgestaltung
1987 · 70 Abb. · 144 Seiten · ISBN 3-540-18120-2

10 *Reithofer, N.*
Nutzungssicherung von flexibel automatisierten Produktionsanlagen
1987 · 84 Abb. · 176 Seiten · ISBN 3-540-18440-6

11 *Diess, H.*
Rechnerunterstützte Entwicklung flexibel automatisierter Montageprozesse
1988 · 56 Abb. · 144 Seiten · ISBN 3-540-18799-5

12 *Reinhart, G.*
Flexible Automatisierung der Konstruktion und Fertigung elektrischer Leitungssätze
1988 · 112 Abb. · 197 Seiten · ISBN 3-540-19003-1

13 *Bürstner, H.*
Investitionsentscheidung in der rechnerintegrierten Produktion
1988 · 74 Abb. · 190 Seiten · ISBN 3-540-19099-6

14 *Groha, A.*
Universelles Zellenrechnerkonzept für flexible Fertigungssysteme
1988 · 74 Abb. · 153 Seiten · ISBN 3-540-19182-8

15 *Riese, K.*
Klipsmontage mit Industrierobotern
1988 · 92 Abb. · 150 Seiten · ISBN 3-540-19183-6

16 *Lutz, P.*
Leitsysteme für rechnerintegrierte Auftragsabwicklung
1988 · 44 Abb. · 144 Seiten · ISBN 3-540-19260-3

17 *Klippel, C.*
Mobiler Roboter im Materialfluß eines flexiblen Fertigungssystems
1988 · 86 Abb. · 164 Seiten · ISBN 3-540-50468-0

18 *Rascher, R.*
Experimentelle Untersuchungen zur Technologie der Kugelherstellung
1989 · 110 Abb. · 200 Seiten · ISBN 3-540-51301-9

19 *Heusler, H.-J.*
Rechnerunterstützte Planung flexibler Montagesysteme
1989 · 43 Abb. · 154 Seiten · ISBN 3-540-51723-5

20 *Kirchknopf, P.*
Ermittlung modaler Parameter aus Übertragungsfrequenzgängen
1989 · 57 Abb. · 157 Seiten · ISBN 3-540-51724-3

21 *Sauerer, Ch.*
Beitrag für ein Zerspanprozeßmodell Metallbandsägen
1990 · 89 Abb. · 166 Seiten · ISBN 3-540-51868-1

22 *Karstedt, K.*
Positionsbestimmung von Objekten in der Montage- und Fertigungsautomatisierung
1990 · 92 Abb. · 157 Seiten · ISBN 3-540-51879-7

23 *Peiker, St.*
Entwicklung eines integrierten NC-Planungssystems
1990 · 66 Abb. · 180 Seiten · ISBN 3-540-51880-0

24 *Schugmann, R.*
Nachgiebige Werkzeugaufhängungen für die automatische Montage
1990 · 71 Abb. · 155 Seiten · ISBN 3-540-52138-0

25 *Wrba, P.*
Simulation als Werkzeug in der Handhabungstechnik
1990 · 125 Abb. · 178 Seiten · ISBN 3-540-52231-X

26 *Eibelshäuser, P.*
Rechnerunterstützte experimentelle Modalanalyse mittels gestufter Sinusanregung
1990 · 79 Abb. · 156 Seiten · ISBN 3-540-52451-7

27 *Prasch, J.*
Computerunterstützte Planung von chirurgischen Eingriffen in der Orthopädie
1990 · 113 Abb. · 164 Seiten · ISBN 3-540-52543-2

28 *Teich, K.*
Prozeßkommunikation und Rechnerverbund in der Produktion
1990 · 52 Abb. · 158 Seiten · ISBN 3-540-52764-8

29 *Pfrang, W.*
Rechnergestützte und graphische Planung manueller und teilautomatisierter Arbeitsplätze
1990 · 59 Abb. · 153 Seiten · ISBN 3-540-52829-6

30 *Tauber, A.*
Modellbildung kinematischer Strukturen als Komponente der Montageplanung
1990 · 93 Abb. · 190 Seiten · ISBN 3-540-52911-X

31 *Jäger, A.*
Systematische Planung komplexer Produktionssysteme
1991 · 75 Abb. · 148 Seiten · ISBN 3-540-53021-5

32 *Hartberger, H.*
Wissensbasierte Simulation komplexer Produktionssysteme
1991 · 58 Abb. · 154 Seiten · ISBN 3-540-53326-5

33 *Tuczek, H.*
Inspektion von Karosseriepreßteilen auf Risse und Einschnürungen mittels Methoden der Bildverarbeitung
1992 · 125 Abb. · 179 Seiten · ISBN 3-540-53965-4

34 *Fischbacher, J.*
Planungsstrategien zur stömungstechnischen Optimierung von Reinraum-Fertigungsgeräten
1991 · 60 Abb. · 166 Seiten · ISBN 3-540-54027-X

35 *Moser, O.*
3D-Echtzeitkollisionsschutz für Drehmaschinen
1991 · 66 Abb. · 177 Seiten · ISBN 3-540-54076-8

36 *Naber, H.*
Aufbau und Einsatz eines mobilen Roboters mit unabhängiger Lokomotions- und Manipulationskomponente
1991 · 85 Abb. · 139 Seiten · ISBN 3-540-54216-7

37 *Kupec, Th.*
Wissensbasiertes Leitsystem zur Steuerung flexibler Fertigungsanlagen
1991 · 68 Abb. · 150 Seiten · ISBN 3-540-54260-4

38 *Maulhardt, U.*
Dynamisches Verhalten von Kreissägen
1991 · 109 Abb. · 159 Seiten · ISBN 3-540-54365-1

39 *Götz, R.*
Strukturierte Planung flexibel automatisierter Montagesysteme für flächige Bauteile
1991 · 86 Abb. · 201 Seiten · ISBN 3-540-54401-1

40 *Koepfer, Th.*
3D-grafisch-interaktive Arbeitsplanung - ein Ansatz zur Aufhebung der Arbeitsteilung
1991 · 74 Abb. · 126 Seiten · ISBN 3-540-54436-4

41 *Schmidt, M.*
Konzeption und Einsatzplanung flexibel automatisierter Montagesysteme
1992 · 108 Abb. · 168 Seiten · ISBN 3-540-55025-9

42 *Burger, C.*
Produktionsregelung mit entscheidungsunterstützenden Informationssystemen
1992 · 94 Abb. · 186 Seiten · ISBN 3-540-55187-5

43 *Hoßmann, J.*
Methodik zur Planung der automatischen Montage von nicht formstabilen Bauteilen
1992 · 73 Abb. · 168 Seiten · ISBN 3-540-5520-0

44 *Petry, M.*
Systematik zur Entwicklung eines modularen Programmbaukastens für robotergeführte Klebeprozesse
1992 · 106 Abb. · 139 Seiten · ISBN 3-540-55374-6

45 *Schönecker, W.*
Integrierte Diagnose in Produktionszellen
1992 · 87 Abb. · 159 Seiten · ISBN 3-540-55375-4

46 *Bick, W.*
Systematische Planung hybrider Montagesysteme unter Berücksichtigung der Ermittlung des optimalen Automatisierungsgrades
1992 · 70 Abb. · 156 Seiten · ISBN 3-540-55377-0

47 *Gebauer, L.*
Prozeßuntersuchungen zur automatisierten Montage von optischen Linsen
1992 · 84 Abb. · 150 Seiten · ISBN 3-540-55378-9

48 *Schrüfer, N.*
Erstellung eines 3D-Simulationssystems zur Reduzierung von Rüstzeiten bei der NC-Bearbeitung
1992 · 103 Abb. · 161 Seiten · ISBN 3-540-55431-9

49 *Wisbacher, J.*
Methoden zur rationellen Automatisierung der Montage von Schnellbefestigungselementen
1992 · 77 Abb. · 176 Seiten · ISBN 3-540-55512-9

50 *Garnich, F.*
Laserbearbeitung mit Robotern
1992 · 110 Abb. · 184 Seiten · ISBN 3-540-55513-7

51 *Eubert, P.*
Digitale Zustandesregelung elektrischer Vorschubantriebe
1992 · 89 Abb. · 159 Seiten · ISBN 3-540-44441-2

52 *Glaas, W.*
Rechnerintegrierte Kabelsatzfertigung
1992 · 67 Abb. · 140 Seiten · ISBN 3-540-55749-0

53 *Helml, H.J.*
Ein Verfahren zur On-Line Fehlererkennung und Diagnose
1992 · 60 Abb. · 153 Seiten · ISBN 3-540-55750-4

54 *Lang, Ch.*
Wissensbasierte Unterstützung der Verfügbarkeitsplanung
1992 · 75 Abb. · 150 Seiten · ISBN 3-540-55751-2

55 *Schuster, G.*
Rechnergestütztes Planungssystem für die flexibel automatisierte Montage
1992 · 67 Abb. · 135 Seiten · ISBN 3-540-55830-6

56 *Bomm, H.*
Ein Ziel- und Kennzahlensystem zum Investitionscontrolling komplexer Produktionssysteme
1992 · 87 Abb. · 195 Seiten · ISBN 3-540-55964-7

57 *Wendt, A.*
Qualitätssicherung in flexibel automatisierten Montagesystemen
1992 · 74 Abb. · 179 Seiten · ISBN 3-540-56044-0

58 *Hansmaier, H.*
Rechnergestütztes Verfahren zur Geräuschminderung
1993 · 67 Abb. · 156 Seiten · ISBN 3-540-56053-2

59 *Dilling, U.*
Planung von Fertigungssystemen unterstützt durch Wirtschaftssimulationen
1993 · 72 Abb. · 146 Seiten · ISBN 3-540-56307-5

60 *Strohmayr, R.*
Rechnergestützte Auswahl und Konfiguration von Zubringeeinrichtungen
1993 · 80 Abb. · 152 Seiten · ISBN 3-540-56652-X

61 *Glas, J.*
Standardisierter Aufbau anwendungsspezifischer Zellenrechnersoftware
1993 · 80 Abb. · 145 Seiten · ISBN 3.540-56890-5

62 *Stetter, R.*
Rechnergestütze Simulationswerkzeuge zur Effizienzsteigerung des Industrierobotereinsatzes
1994 · 91 Abb. · 146 Seiten · ISBN 3-540-56889-1

63 *Dirndorfer, A.*
Robotersysteme zur förderbandsynchronen Montage
1993 · 76 Abb. · 144 Seiten · ISBN 3-540-57031-4

64 *Wiedemann, M.*
Simulation des Schwingungsverhaltens spanender Werkzeugmaschinen
1993 · 81 Abb. · 137 Seiten · ISBN 3-540-57177-9

65 *Woenckhaus, Ch.*
Rechnergestütztes System zur automatisierten 3D-Layoutoptimierung
1994 · 81 Abb. · 140 Seiten · ISBN 3-540-57284-8

66 *Kummetsteiner, G.*
3D-Bewegungssimulation als integratives Hilfsmittel zur Planung manueller Montagesysteme
1994 · 62 Abb. · 146 Seiten · ISBN 3-540-57535-9

67 *Kugelmann, F.*
Einsatz nachgiebiger Elemente zur wirtschaftlichen Automatisierung von Produktionssystemen
1993 · 76 Abb. · 144 Seiten · ISBN 3-540-57549-9

68 *Schwarz, H.*
Simulationsgestützte CAD/CAM-Kopplung für die 3D-Laserbearbeitung mit integrierter Sensorik
1994 · 96 Abb. · 148 Seiten · ISBN 3-540-57577-4

69 *Viethen, U.*
Systematik zum Prüfen in flexiblen Fertigungssystemen
1994 · 70 Abb. · 142 Seiten · ISBN 3-540-57794-7

70 *Seehuber, M.*
Automatische Inbetriebnahme geschwindigkeitsadaptiver Zustandsregler
1994 · 72 Abb. · 155 Seiten · ISBN 3-540-57896-X

71 *Amann, W.*
Eine Simulationsumgebung für Planung und Betrieb von Produktionssystemen
1994 · 71 Abb. · 129 Seiten · ISBN 3-540-57924-9

72 *Schöpf, M.*
Rechnergestütztes Projektinformations- und Koordinationssystem für das Fertigungsvorfeld
1997 · 63 Abb. · 130 Seiten · ISBN 3-540-58052-2

73 *Welling, A.*
Effizienter Einsatz bildgebender Sensoren zur Flexibilisierung automatisierter Handhabungsvorgänge
1994 · 66 Abb. · 139 Seiten · ISBN 3-540-580-0

74 *Zetlmayer, H.*
Verfahren zur simulationsgestützten Produktionsregelung in der Einzel- und Kleinserienproduktion
1994 · 62 Abb. · 143 Seiten · ISBN 3-540-58134-0

75 *Lindl, M.*
Auftragsleittechnik für Konstruktion und Arbeitsplanung
1994 · 66 Abb. · 147 Seiten · ISBN 3-540-58221-5

76 *Zipper, B.*
Das integrierte Betriebsmittelwesen - Baustein einer flexiblen Fertigung
1994 · 64 Abb. · 147 Seiten · ISBN 3-540-58222-3

77 *Raith, P.*
Programmierung und Simulation von Zellenabläufen in der Arbeitsvorbereitung
1995 · 51 Abb. · 130 Seiten · ISBN 3-540-58223-1

78 *Engel, A.*
Strömungstechnische Optimierung von Produktionssystemen durch Simulation
1994 · 69 Abb. · 160 Seiten · ISBN 3-540-58258-4

79 *Zäh, M. F.*
Dynamisches Prozeßmodell Kreissägen
1995 · 95 Abb. · 186 Seiten · ISBN 3-540-58624-5

80 *Zwanzer, N.*
Technologisches Prozeßmodell für die Kugelschleifbearbeitung
1995 · 65 Abb. · 150 Seiten · ISBN 3-540-58634-2

81 *Romanow, P.*
Konstruktionsbegleitende Kalkulation von Werkzeugmaschinen
1995 · 66 Abb. · 151 Seiten · ISBN 3-540-58771-3

82 *Kahlenberg, R.*
Integrierte Qualitätssicherung in flexiblen Fertigungszellen
1995 · 71 Abb. · 136 Seiten · ISBN 3-540-58772-1

83 *Huber, A.*
Arbeitsfolgenplanung mehrstufiger Prozesse in der Hartbearbeitung
1995 · 87 Abb. · 152 Seiten · ISBN 3-540-58773-X

84 *Birkel, G.*
Aufwandsminimierter Wissenserwerb für die Diagnose in flexiblen Produktionszellen
1995 · 64 Abb. · 137 Seiten · ISBN 3-540-58869-8

85 *Simon, D.*
Fertigungsregelung durch zielgrößenorientierte Planung und logistisches Störungsmanagement
1995 · 77 Abb. · 132 Seiten · ISBN 3-540-58942-2

86 *Nedeljkovic-Groha, V.*
Systematische Planung anwendungsspezifischer Materialflußsteuerungen
1995 · 94 Abb. · 188 Seiten · ISBN 3-540-58953-8

87 *Rockland, M.*
Flexibilisierung der automatischen Teilbereitstellung in Montageanlagen
1995 · 83 Abb. · 168 Seiten · ISBN 3-540-58999-6

88 *Linner, St.*
Konzept einer integrierten Produktentwicklung
1995 · 67 Abb. · 168 Seiten · ISBN 3-540-59016-1

89 *Eder, Th.*
Integrierte Planung von Informationssystemen für rechnergestützte Produktionssysteme
1995 · 62 Abb. · 150 Seiten · ISBN 3-540-59084-6

90 *Deutschle, U.*
Prozeßorientierte Organisation der Auftragsentwicklung in mittelständischen Unternehmen
1995 · 80 Abb. · 188 Seiten · ISBN 3-540-59337-3

91 *Dieterle, A.*
Recyclingintegrierte Produktentwicklung
1995 · 68 Abb. · 146 Seiten · ISBN 3-540-60120-1

92 *Hechl, Chr.*
Personalorientierte Montageplanung für komplexe und variantenreiche Produkte
1995 · 73 Abb. · 158 Seiten · ISBN 3-540-60325-5

93 *Albertz, F.*
Dynamikgerechter Entwurf von Werkzeugmaschinen - Gestellstrukturen
1995 · 83 Abb. · 156 Seiten · ISBN 3-540-60608-8

94 *Trunzer, W.*
Strategien zur On-Line Bahnplanung bei Robotern mit 3D-Konturfolgesensoren
1996 · 101 Abb. · 164 Seiten · ISBN 3-540-60961-X

95 *Fichtmüller, N.*
Rationalisierung durch flexible, hybride Montagesysteme
1996 · 83 Abb. · 145 Seiten · ISBN 3-540-60960-1

96 *Trucks, V.*
Rechnergestützte Beurteilung von Getriebestrukturen in Werkzeugmaschinen
1996 · 64 Abb. · 141 Seiten · ISBN 3-540-60599-8

97 *Schäffer, G.*
Systematische Integration adaptiver Produktionssysteme
1996 · 71 Abb. · 170 Seiten · ISBN 3-540-60958-X

98 *Koch, M. R.*
Autonome Fertigungszellen - Gestaltung, Steuerung und integrierte Störungsbehandlung
1996 · 67 Abb. · 138 Seiten · ISBN 3-540-61104-5

99 *Moctezuma de la Barrera, J.L.*
Ein durchgängiges System zur computer- und rechnergestützten Chirurgie
1996 · 99 Abb. · 175 Seiten · ISBN 3-540-61145-2

100 *Geuer, A.*
Einsatzpotential des Rapid Prototyping in der Produktentwicklung
1996 · 84 Abb. · 154 Seiten · ISBN 3-540-61495-8

101 *Ebner, C.*
Ganzheitliches Verfügbarkeits- und Qualitätsmanagment unter Verwendung von Felddaten
1996 · 67 Abb. · 132 Seiten · ISBN 3-540-61678-0

102 *Pischeltsrieder, K.*
Steuerung autonomer mobiler Roboter in der Produktion
1996 · 74 Abb. · 171 Seiten · ISBN 3-540-61714-0

103 *Köhler, R.*
Disposition und Materialbereitstellung bei komplexen variantenreichen Kleinprodukten
1997 · 62 Abb. · 177 Seiten · ISBN 3-540-62024-9

104 *Feldmann, Ch.*
Eine Methode für die integrierte rechnergestützte Montageplanung
1997 · 71 Abb. · 163 Seiten · ISBN 3-540-62059-1

105 *Lehmann, H.*
Integrierte Materialfluß- und Layoutplanung durch Kopplung von CAD- und Ablaufsimulationssystem
1997 · 96 Abb. · 191 Seiten · ISBN 3-540-62202-0

106 *Wagner, M.*
Steuerungsintegrierte Fehlerbehandlung für maschinennahe Abläufe
1997 · 94 Abb. · 164 Seiten · ISBN 3-540-62656-5

107 *Lorenzen, J.*
Simulationsgestützte Kostenanalyse in produktorientierten Fertigungsstrukturen
1997 · 63 Abb. · 129 Seiten · ISBN 3-540-62794-4

108 *Krönert, U.*
Systematik für die rechnergestützte Ähnlichteilsuche und Standardisierung
1997 · 53 Abb. · 127 Seiten · ISBN 3-540-63338-3

109 *Pfersdorf, I.*
Entwicklung eines systematischen Vorgehens zur Organisation des industriellen Service
1997 · 74 Abb. · 172 Seiten · ISBN 3-540-63615-3

110 *Kuba, R.*
Informations- und kommunikationstechnische Integration von Menschen in der Produktion
1997 · 77 Abb. · 155 Seiten · ISBN 3-540-63642-0

111 *Kaiser, J.*
Vernetztes Gestalten von Produkt und Produktionsprozeß mit Produktmodellen
1997 · 67 Abb. · 139 Seiten · ISBN 3-540-63999-3

112 *Geyer, M.*
Flexibles Planungssystem zur Berücksichtigung ergonomischer Aspekte bei der Produkt- und Arbeitssystemgestaltung
1997 · 85 Abb. · 154 Seiten · ISBN 3-540-64195-5

113 *Martin, C.*
Produktionsregelung - ein modularer, modellbasierter Ansatz
1998 · 73 Abb. · 162 Seiten · ISBN 3-540-64401-6

114 *Löffler, Th.*
Akustische Überwachung automatisierter Fügeprozesse
1998 · 85 Abb. · 136 Seiten · ISBN 3-540-64511-X

115 *Lindermaier, R.*
Qualitätsorientierte Entwicklung von Montagesystemen
1998 · 84 Abb. · 164 Seiten · ISBN 3-540-64686-8

116 *Koehrer, J.*
Prozeßorientierte Teamstrukturen in Betrieben mit Großserienfertigung
1998 · 75 Abb. · 185 Seiten · ISBN 3-540-65037-7

117 *Schuller, R. W.*
Leitfaden zum automatisierten Auftrag von hochviskosen Dichtmassen
1999 · 76 Abb. · 162 Seiten · ISBN 3-540-65320-1

118 *Debuschewitz, M.*
Integrierte Methodik und Werkzeuge zur herstellungsorientierten Produktentwicklung
1999 · 104 Abb. · 169 Seiten · ISBN 3-540-65350-3

119 *Bauer, L.*
Strategien zur rechnergestützten Offline-Programmierung von 3D-Laseranlagen
1999 · 98 Abb. · 145 Seiten · ISBN 3-540-65382-1

120 *Pfob, E.*
Modellgestützte Arbeitsplanung bei Fertigungsmaschinen
1999 · 69 Abb. · 154 Seiten · ISBN 3-540-65525-5

121 *Spitznagel, J.*
Erfahrungsgeleitete Planung von Laseranlagen
1999 · 63 Abb. · 156 Seiten · ISBN 3-540-65896-3

Seminarberichte iwb

herausgegeben von Prof. Dr.-Ing. Gunther Reinhart und Prof. Dr.-Ing. Michael Zäh,
Institut für Werkzeugmaschinen und Betriebswissenschaften
der Technischen Universität München

Seminarberichte iwb sind erhältlich im Buchhandel oder beim
Herbert Utz Verlag, München, Fax 089-277791-01, info@utz.de

1 Innovative Montagesysteme - Anlagengestaltung, -bewertung und -überwachung
115 Seiten · ISBN 3-931327-01-9
2 Integriertes Produktmodell - Von der Idee zum fertigen Produkt
82 Seiten · ISBN 3-931327-02-7
3 Konstruktion von Werkzeugmaschinen - Berechnung, Simulation und Optimierung
110 Seiten · ISBN 3-931327-03-5
4 Simulation - Einsatzmöglichkeiten und Erfahrungsberichte
134 Seiten · ISBN 3-931327-04-3
5 Optimierung der Kooperation in der Produktentwicklung
95 Seiten · ISBN 3-931327-05-1
6 Materialbearbeitung mit Laser · von der Planung zur Anwendung
86 Seiten · ISBN 3-931327-76-0
7 Dynamisches Verhalten von Werkzeugmaschinen
80 Seiten · ISBN 3-931327-77-9
8 Qualitätsmanagement · der Weg ist das Ziel
130 Seiten · ISBN 3-931327-78-7
9 Installationstechnik an Werkzeugmaschinen · Analysen und Konzepte
120 Seiten · ISBN 3-931327-79-5
10 3D-Simulation - Schneller, sicherer und kostengünstiger zum Ziel
90 Seiten · ISBN 3-931327-10-8
11 Unternehmensorganisation - Schlüssel für eine effiziente Produktion
110 Seiten · ISBN 3-931327-11-6
12 Autonome Produktionssysteme
100 Seiten · ISBN 3-931327-12-4
13 Planung von Montageanlagen
130 Seiten · ISBN 3-931327-13-2
14 Nicht erschienen – wird nicht erscheinen
15 Flexible fluide Kleb/Dichtstoffe · Dosierung und Prozeßgestaltung
80 Seiten · ISBN 3-931327-15-9
16 Time to Market - Von der Idee zum Produktionsstart
80 Seiten · ISBN 3-931327-16-7
17 Industriekeramik in Forschung und Praxis - Probleme, Analysen und Lösungen
80 Seiten · ISBN 3-931327-17-5
18 Das Unternehmen im Internet - Chancen für produzierende Unternehmen
165 Seiten · ISBN 3-931327-18-3
19 Leittechnik und Informationslogistik - mehr Transparenz in der Fertigung
85 Seiten · ISBN 3-931327-19-1
20 Dezentrale Steuerungen in Produktionsanlagen - Plug & Play - Vereinfachung von Entwicklung und Inbetriebnahme
105 Seiten · ISBN 3-931327-20-5
21 Rapid Prototyping - Rapid Tooling - Schnell zu funktionalen Prototypen
95 Seiten · ISBN 3-931327-21-3
22 Mikrotechnik für die Produktion - Greifbare Produkte und Anwendungspotentiale
95 Seiten · ISBN 3-931327-22-1
24 EDM Engineering Data Management
195 Seiten · ISBN 3-931327-24-8
25 Rationelle Nutzung der Simulationstechnik - Entwicklungstrends und Praxisbeispiele
152 Seiten · ISBN 3-931327-25-6
26 Alternative Dichtungssysteme - Konzepte zur Dichtungsmontage und zum Dichtmittelauftrag
110 Seiten · ISBN 3-931327-26-4
27 Rapid Prototyping · Mit neuen Technologien schnell vom Entwurf zum Serienprodukt
111 Seiten · ISBN 3-931327-27-2
28 Rapid Tooling · Mit neuen Technologien schnell vom Entwurf zum Serienprodukt
154 Seiten · ISBN 3-931327-28-0
29 Installationstechnik an Werkzeugmaschinen · Abschlußseminar
156 Seiten · ISBN 3-931327-29-9
30 Nicht erschienen – wird nicht erscheinen
31 Engineering Data Management (EDM) · Erfahrungsberichte und Trends
183 Seiten · ISBN 3-931327-31-0
32 Nicht erschienen – wird nicht erscheinen
33 3D-CAD · Mehr als nur eine dritte Dimension
181 Seiten · ISBN 3-931327-33-7
34 Laser in der Produktion · Technologische Randbedingungen für den wirtschaftlichen Einsatz
102 Seiten · ISBN 3-931327-34-5
35 Ablaufsimulation · Anlagen effizient und sicher planen und betreiben
129 Seiten · ISBN 3-931327-35-3
36 Moderne Methoden zur Montageplanung · Schlüssel für eine effiziente Produktion
124 Seiten · ISBN 3-931327-36-1
37 Wettbewerbsfaktor Verfügbarkeit · Produktivitätsteigerung durch technische und organisatorische Ansätze
95 Seiten · ISBN 3-931327-37-X
38 Rapid Prototyping · Effizienter Einsatz von Modellen in der Produktentwicklung
128 Seiten · ISBN 3-931327-38-8
39 Rapid Tooling · Neue Strategien für den Werkzeug- und Formenbau
130 Seiten · ISBN 3-931327-39-6
40 Erfolgreich kooperieren in der produzierenden Industrie · Flexibler und schneller mit modernen Kooperationen
160 Seiten · ISBN 3-931327-40-X
41 Innovative Entwicklung von Produktionsmaschinen
146 Seiten · ISBN 3-89675-041-0
42 Stückzahlflexible Montagesysteme
139 Seiten · ISBN 3-89675-042-9
43 Produktivität und Verfügbarkeit · ...durch Kooperation steigern
120 Seiten · ISBN 3-89675-043-7
44 Automatisierte Mikromontage · Handhaben und Positionieren von Mikrobauteilen
125 Seiten · ISBN 3-89675-044-5
45 Produzieren in Netzwerken · Lösungsansätze, Methoden, Praxisbeispiele
173 Seiten · ISBN 3-89675-045-3
46 Virtuelle Produktion · Ablaufsimulation
108 Seiten · ISBN 3-89675-046-1

47 Virtuelle Produktion · Prozeß- und Produktsimulation
131 Seiten · ISBN 3-89675-047-X

48 Sicherheitstechnik an Werkzeugmaschinen
106 Seiten · ISBN 3-89675-048-8

49 Rapid Prototyping · Methoden für die reaktionsfähige Produktentwicklung
150 Seiten · ISBN 3-89675-049-6

50 Rapid Manufacturing · Methoden für die reaktionsfähige Produktion
121 Seiten · ISBN 3-89675-050-X

51 Flexibles Kleben und Dichten · Produkt-& Prozeßgestaltung, Mischverbindungen, Qualitätskontrolle
137 Seiten · ISBN 3-89675-051-8

52 Rapid Manufacturing · Schnelle Herstellung von Klein- und Prototypenserien
124 Seiten · ISBN 3-89675-052-6

53 Mischverbindungen · Werkstoffauswahl, Verfahrensauswahl, Umsetzung
107 Seiten · ISBN 3-89675-054-2

54 Virtuelle Produktion · Integrierte Prozess- und Produktsimulation
133 Seiten · ISBN 3-89675-054-2

55 e-Business in der Produktion · Organisationskonzepte, IT-Lösungen, Praxisbeispiele
150 Seiten · ISBN 3-89675-055-0

56 Virtuelle Produktion – Ablaufsimulation als planungsbegleitendes Werkzeug
150 Seiten · ISBN 3-89675-056-9

57 Virtuelle Produktion – Datenintegration und Benutzerschnittstellen
150 Seiten · ISBN 3-89675-057-7

58 Rapid Manufacturing · Schnelle Herstellung qualitativ hochwertiger Bauteile oder Kleinserien
169 Seiten · ISBN 3-89675-058-7

59 Automatisierte Mikromontage · Werkzeuge und Fügetechnologien für die Mikrosystemtechnik
114 Seiten · ISBN 3-89675-059-3

60 Mechatronische Produktionssysteme · Genauigkeit gezielt entwickeln
131 Seiten · ISBN 3-89675-060-7

61 Nicht erschienen – wird nicht erscheinen

62 Rapid Technologien · Anspruch – Realität – Technologien
100 Seiten · ISBN 3-89675-062-3

63 Fabrikplanung 2002 · Visionen – Umsetzung – Werkzeuge
124 Seiten · ISBN 3-89675-063-1

64 Mischverbindungen · Einsatz und Innovationspotenzial
143 Seiten · ISBN 3-89675-064-X

65 Fabrikplanung 2003 – Basis für Wachstum · Erfahrungen Werkzeuge Visionen
136 Seiten · ISBN 3-89675-065-8

66 Mit Rapid Technologien zum Aufschwung · Neue Rapid Technologien und Verfahren, Neue Qualitäten, Neue Möglichkeiten, Neue Anwendungsfelder
185 Seiten · ISBN 3-89675-066-6

67 Mechatronische Produktionssysteme · Die Virtuelle Werkzeugmaschine: Mechatronisches Entwicklungsvorgehen, Integrierte Modellbildung, Applikationsfelder
148 Seiten · ISBN 3-89675-067-4

68 Virtuelle Produktion · Nutzenpotenziale im Lebenszyklus der Fabrik
139 Seiten · ISBN 3-89675-068-2

69 Kooperationsmanagement in der Produktion · Visionen und Methoden zur Kooperation – Geschäftsmodelle und Rechtsformen für die Kooperation – Kooperation entlang der Wertschöpfungskette
134 Seiten · ISBN 3-98675-069-0

70 Mechatronik · Strukturdynamik von Werkzeugmaschinen
161 Seiten · ISBN 3-89675-070-4

71 Klebtechnik · Zerstörungsfreie Qualitätssicherung beim flexibel automatisierten Kleben und Dichten
ISBN 3-89675-071-2 · vergriffen

72 Fabrikplanung 2004 Ergfolgsfaktor im Wettbewerb · Erfahrungen – Werkzeuge – Visionen
ISBN 3-89675-072-0 · vergriffen

73 Rapid Manufacturing Vom Prototyp zur Produktion · Erwartungen – Erfahrungen – Entwicklungen
179 Seiten · ISBN 3-89675-073-9

74 Virtuelle Produktionssystemplanung · Virtuelle Inbetriebnahme und Digitale Fabrik
133 Seiten · ISBN 3-89675-074-7

75 Nicht erschienen – wird nicht erscheinen

76 Berührungslose Handhabung · Vom Wafer zur Glaslinse, von der Kapsel zur aseptischen Ampulle
95 Seiten · ISBN 3-89675-076-3

77 ERP-Systeme - Einführung in die betriebliche Praxis · Erfahrungen, Best Practices, Visionen
153 Seiten · ISBN 3-89675-077-7

78 Mechatronik · Trends in der interdisziplinären Entwicklung von Werkzeugmaschinen
155 Seiten · ISBN 3-89675-078-X

79 Produktionsmanagement
267 Seiten · ISBN 3-89675-079-8

80 Rapid Manufacturing · Fertigungsverfahren für alle Ansprüche
154 Seiten · ISBN 3-89675-080-1

81 Rapid Manufacturing · Heutige Trends – Zukünftige Anwendungsfelder
172 Seiten · ISBN 3-89675-081-X

82 Produktionsmanagement · Herausforderung Variantenmanagement
100 Seiten · ISBN 3-89675-082-8

83 Mechatronik · Optimierungspotenzial der Werkzeugmaschine nutzen
160 Seiten · ISBN 3-89675-083-6

84 Virtuelle Inbetriebnahme · Von der Kür zur Pflicht?
104 Seiten · ISBN 978-3-89675-084-6

85 3D-Erfahrungsforum · Innovation im Werkzeug- und Formenbau
375 Seiten · ISBN 978-3-89675-085-3

86 Rapid Manufacturing · Erfolgreich produzieren durch innovative Fertigung
162 Seiten · ISBN 978-3-89675-086-0

87 Produktionsmanagement · Schlank im Mittelstand
102 Seiten · ISBN 978-3-89675-087-7

88 Mechatronik · Vorsprung durch Simulation
134 Seiten · ISBN 978-3-89675-088-4

89 RFID in der Produktion · Wertschöpfung effizient gestalten
122 Seiten · ISBN 978-3-89675-089-1

Forschungsberichte iwb

herausgegeben von Prof. Dr.-Ing. Gunther Reinhart und Prof. Dr.-Ing. Michael Zäh,
Institut für Werkzeugmaschinen und Betriebswissenschaften
der Technischen Universität München

Forschungsberichte iwb ab Band 122 sind erhältlich im Buchhandel oder beim
Herbert Utz Verlag, München, Fax 089-277791-01, info@utz.de

122 Schneider, Burghard
Prozesskettenorientierte Bereitstellung nicht formstabiler Bauteile
1999 · 183 Seiten · 98 Abb. · 14 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-559-5

123 Goldstein, Bernd
Modellgestützte Geschäftsprozeßgestaltung in der Produktentwicklung
1999 · 170 Seiten · 65 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-546-3

124 Mößmer, Helmut E.
Methode zur simulationsbasierten Regelung zeitvarianter Produktionssysteme
1999 · 164 Seiten · 67 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-585-4

125 Gräser, Ralf-Gunter
Ein Verfahren zur Kompensation temperaturinduzierter Verformungen an Industrierobotern
1999 · 167 Seiten · 63 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-603-6

126 Trossin, Hans-Jürgen
Nutzung der Ähnlichkeitstheorie zur Modellbildung in der Produktionstechnik
1999 · 162 Seiten · 75 Abb. · 11 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-614-1

127 Kugelmann, Doris
Aufgabenorientierte Offline-Programmierung von Industrierobotern
1999 · 168 Seiten · 68 Abb. · 2 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-615-X

128 Diesch, Rolf
Steigerung der organisatorischen Verfügbarkeit von Fertigungszellen
1999 · 160 Seiten · 69 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-618-4

129 Lulay, Werner E.
Hybrid-hierarchische Simulationsmodelle zur Koordination teilautonomer Produktionsstrukturen
1999 · 182 Seiten · 51 Abb. · 14 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-620-6

130 Murr, Otto
Adaptive Planung und Steuerung von integrierten Entwicklungs- und Planungsprozessen
1999 · 178 Seiten · 85 Abb. · 3 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-636-2

131 Macht, Michael
Ein Vorgehensmodell für den Einsatz von Rapid Prototyping
1999 · 170 Seiten · 87 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-638-9

132 Mehler, Bruno H.
Aufbau virtueller Fabriken aus dezentralen Partnerverbünden
1999 · 152 Seiten · 44 Abb. · 27 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-645-1

133 Heitmann, Knut
Sichere Prognosen für die Produktionsptimierung mittels stochastischer Modelle
1999 · 146 Seiten · 60 Abb. · 13 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-675-3

134 Blessing, Stefan
Gestaltung der Materialflußsteuerung in dynamischen Produktionsstrukturen
1999 · 160 Seiten · 67 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-690-7

135 Abay, Can
Numerische Optimierung multivariater mehrstufiger Prozesse am Beispiel der Hartbearbeitung von Industriekeramik
2000 · 159 Seiten · 46 Abb. · 5 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-697-4

136 Brandner, Stefan
Integriertes Produktdaten- und Prozeßmanagement in virtuellen Fabriken
2000 · 172 Seiten · 61 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-715-6

137 Hirschberg, Arnd G.
Verbindung der Produkt- und Funktionsorientierung in der Fertigung
2000 · 165 Seiten · 49 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-729-6

138 Reek, Alexandra
Strategien zur Fokuspositionierung beim Laserstrahlschweißen
2000 · 193 Seiten · 103 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-730-X

139 Sabbah, Khalid-Alexander
Methodische Entwicklung störungstoleranter Steuerungen
2000 · 148 Seiten · 75 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-739-3

140 Schliffenbacher, Klaus U.
Konfiguration virtueller Wertschöpfungsketten in dynamischen, heterarchischen Kompetenznetzwerken
2000 · 187 Seiten · 70 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-754-7

141 Sprenzel, Andreas
Integrierte Kostenkalkulationsverfahren für die Werkzeugmaschinenentwicklung
2000 · 144 Seiten · 55 Abb. · 6 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-757-1

142 Gallasch, Andreas
Informationstechnische Architektur zur Unterstützung des Wandels in der Produktion
2000 · 150 Seiten · 69 Abb. · 6 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-781-4

143 Cuiper, Ralf
Durchgängige rechnergestützte Planung und Steuerung von automatisierten Montagevorgängen
2000 · 168 Seiten · 75 Abb. · 3 Tab. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-783-0

144 Schneider, Christian
Strukturmechanische Berechnungen in der Werkzeugmaschinenkonstruktion
2000 · 180 Seiten · 66 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-789-X

145 Jonas, Christian
Konzept einer durchgängigen, rechnergestützten Planung von Montageanlagen
2000 · 183 Seiten · 82 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-870-5

146 Willnecker, Ulrich
Gestaltung und Planung leistungsorientierter manueller Fließmontagen
2001 · 175 Seiten · 67 Abb. · broschiert · 20,5 x 14,5 cm · ISBN 3-89675-891-8

147 Lehner, Christof
Beschreibung des Nd:Yag-Laserstrahlschweißprozesses von Magnesiumdruckguss
2001 · 205 Seiten · 94 Abb. · 24 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0004-X

148 Rick, Frank
Simulationsgestützte Gestaltung von Produkt und Prozess am Beispiel Laserstrahlschweißen
2001 · 145 Seiten · 57 Abb. · 2 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0008-2

149 Höhn, Michael
Sensorgeführte Montage hybrider Mikrosysteme
2001 · 171 Seiten · 74 Abb. · 7 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0012-0

150 Böhl, Jörn
Wissensmanagement im Klein- und mittelständischen Unternehmen der Einzel- und Kleinserienfertigung
2001 · 179 Seiten · 88 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0020-1

151 Bürgel, Robert
Prozessanalyse an spanenden Werkzeugmaschinen mit digital geregelten Antrieben
2001 · 185 Seiten · 60 Abb. · 10 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0021-X

152 Stephan Dürrschmidt
Planung und Betrieb wandlungsfähiger Logistiksysteme in der variantenreichen Serienproduktion
2001 · 914 Seiten · 61 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0023-6

153 Bernhard Eich
Methode zur prozesskettenorientierten Planung der Teilebereitstellung
2001 · 132 Seiten · 48 Abb. · 6 Tabellen · 20,5 x 14,5 cm · ISBN 3-8316-0028-7

154 Wolfgang Rudorfer
Eine Methode zur Qualifizierung von produzierenden Unternehmen für Kompetenznetzwerke
2001 · 207 Seiten · 89 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0037-6

155 Hans Meier
Verteilte kooperative Steuerung maschinennaher Abläufe
2001 · 162 Seiten · 85 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0044-9

156 Gerhard Nowak
Informationstechnische Integration des industriellen Service in das Unternehmen
2001 · 203 Seiten · 95 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0055-4

157 Martin Werner
Simulationsgestützte Reorganisation von Produktions- und Logistikprozessen
2001 · 191 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0058-9

158 Bernhard Lenz
Finite Elemente-Modellierung des Laserstrahlschweißens für den Einsatz in der Fertigungsplanung
2001 · 150 Seiten · 47 Abb. · 5 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0094-5

159 Stefan Grunwald
Methode zur Anwendung der flexiblen integrierten Produktentwicklung und Montageplanung
2002 · 206 Seiten · 80 Abb. · 25 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0095-3

160 Josef Gartner
Qualitätssicherung bei der automatisierten Applikation hochviskoser Dichtungen
2002 · 165 Seiten · 74 Abb. · 21 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0096-1

161 Wolfgang Zeller
Gesamtheitliches Sicherheitskonzept für die Antriebs- und Steuerungstechnik bei Werkzeugmaschinen
2002 · 192 Seiten · 54 Abb. · 15 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0100-3

162 Michael Loferer
Rechnergestützte Gestaltung von Montagesystemen
2002 · 178 Seiten · 80 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0118-6

163 Jörg Fährer
Ganzheitliche Optimierung des indirekten Metall-Lasersinterprozesses
2002 · 176 Seiten · 69 Abb. · 13 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0124-0

164 Jürgen Höppner
Verfahren zur berührungslosen Handhabung mittels leistungsstarker Schallwandler
2002 · 132 Seiten · 24 Abb. · 3 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0125-9

165 Hubert Götte
Entwicklung eines Assistenzrobotersystems für die Knieendoprothetik
2002 · 258 Seiten · 123 Abb. · 5 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0126-7

166 Martin Weißenberger
Optimierung der Bewegungsdynamik von Werkzeugmaschinen im rechnergestützten Entwicklungsprozess
2002 · 210 Seiten · 86 Abb. · 2 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0138-0

167 Dirk Jacob
Verfahren zur Positionierung unterseitenstrukturierter Bauelemente in der Mikrosystemtechnik
2002 · 200 Seiten · 82 Abb. · 24 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0142-9

168 Ulrich Roßgoderer
System zur effizienten Layout- und Prozessplanung von hybriden Montageanlagen
2002 · 175 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0154-2

169 Robert Klingel
Anziehverfahren für hochfeste Schraubenverbindungen auf Basis akustischer Emissionen
2002 · 164 Seiten · 89 Abb. · 27 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0174-7

170 Paul Jens Peter Ross
Bestimmung des wirtschaftlichen Automatisierungsgrades von Montageprozessen in der frühen Phase der Montageplanung
2002 · 144 Seiten · 38 Abb. · 38 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0191-7

171 Stefan von Praun
Toleranzanalyse nachgiebiger Baugruppen im Produktentstehungsprozess
2002 · 250 Seiten · 62 Abb. · 7 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0202-6

172 Florian von der Hagen
Gestaltung kurzfristiger und unternehmensübergreifender Engineering-Kooperationen
2002 · 220 Seiten · 104 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0208-5

173 Oliver Kramer
Methode zur Optimierung der Wertschöpfungskette mittelständischer Betriebe
2002 · 212 Seiten · 84 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0211-5

174 Winfried Dohmen
Interdisziplinäre Methoden für die integrierte Entwicklung komplexer mechatronischer Systeme
2002 · 200 Seiten · 67 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0214-X

175 Oliver Anton
Ein Beitrag zur Entwicklung telepräsenter Montagesysteme
2002 · 158 Seiten · 85 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0215-8

176 Welf Broser
Methode zur Definition und Bewertung von Anwendungsfeldern für Kompetenznetzwerke
2002 · 224 Seiten · 122 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0217-4

177 Frank Breitinger
Ein ganzheitliches Konzept zum Einsatz des indirekten Metall-Lasersinterns für das Druckgießen
2003 · 156 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0227-1

178 Johann von Pieverling
Ein Vorgehensmodell zur Auswahl von Konturfertigungsverfahren für das Rapid Tooling
2003 · 163 Seiten · 88 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0230-1

179 Thomas Baudisch
Simulationsumgebung zur Auslegung der Bewegungsdynamik des mechatronischen Systems Werkzeugmaschine
2003 · 190 Seiten · 67 Abb. · 8 Tab. · 20,5 x 14,5 cm · ISBN 3-8316-0249-2

180 Heinrich Schieferstein
Experimentelle Analyse des menschlichen Kausystems
2003 · 132 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0251-4

181 Joachim Berlak
Methodik zur strukturierten Auswahl von Auftragsabwicklungssystemen
2003 · 244 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0258-1

182 Christian Meierlohr
Konzept zur rechnergestützten Integration von Produktions- und Gebäudeplanung in der Fabrikgestaltung
2003 · 181 Seiten · 84 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0292-1

183 Volker Weber
Dynamisches Kostenmanagement in kompetenzzentrierten Unternehmensnetzwerken
2004 · 210 Seiten · 64 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0330-8

184 Thomas Bongardt
Methode zur Kompensation betriebsabhängiger Einflüsse auf die Absolutgenauigkeit von Industrierobotern
2004 · 170 Seiten · 40 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0332-4

185 Tim Angerer
Effizienzsteigerung in der automatisierten Montage durch aktive Nutzung mechatronischer Produktkomponenten
2004 · 180 Seiten · 67 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0336-7

186 Alexander Krüger
Planung und Kapazitätsabstimmung stückzahlflexibler Montagesysteme
2004 · 197 Seiten · 83 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0371-5

187 Matthias Meindl
Beitrag zur Entwicklung generativer Fertigungsverfahren für das Rapid Manufacturing
2005 · 222 Seiten · 97 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0465-7

188 Thomas Fusch
Betriebsbegleitende Prozessplanung in der Montage mit Hilfe der Virtuellen Produktion am Beispiel der Automobilindustrie
2005 · 190 Seiten · 99 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0467-3

189 Thomas Mosandl
Qualitätssteigerung bei automatisiertem Klebstoffauftrag durch den Einsatz optischer Konturfolgesysteme
2005 · 182 Seiten · 58 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0471-1

190 Christian Patron
Konzept für den Einsatz von Augmented Reality in der Montageplanung
2005 · 150 Seiten · 61 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0474-6

191 Robert Cisek
Planung und Bewertung von Rekonfigurationsprozessen in Produktionssystemen
2005 · 200 Seiten · 64 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0475-4

192 Florian Auer
Methode zur Simulation des Laserstrahlschweißens unter Berücksichtigung der Ergebnisse vorangegangener Umformsimulationen
2005 · 160 Seiten · 65 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0485-1

193 Carsten Selke
Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung
2005 · 137 Seiten · 53 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0495-9

194 Markus Seefried
Simulation des Prozessschrittes der Wärmebehandlung beim Indirekten-Metall-Lasersintern
2005 · 216 Seiten · 82 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0503-3

195 Wolfgang Wagner
Fabrikplanung für die standortübergreifende Kostensenkung bei marktnaher Produktion
2006 · 208 Seiten · 43 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0586-6

196 Christopher Ulrich
Erhöhung des Nutzungsgrades von Laserstrahlquellen durch Mehrfach-Anwendungen
2006 · 178 Seiten · 74 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0590-4

197 Johann Härtl
Prozessgaseinfluss beim Schweißen mit Hochleistungsdiodenlasern
2006 · 140 Seiten · 55 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0611-0

198 Bernd Hartmann
Die Bestimmung des Personalbedarfs für den Materialfluss in Abhängigkeit von Produktionsfläche und -menge
2006 · 208 Seiten · 105 Abb. · 20,5 x 14,5 cm · ISBN 3-8316-0615-3

199 Michael Schilp
Auslegung und Gestaltung von Werkzeugen zum berührungslosen Greifen kleiner Bauteile in der Mikromontage
2006 · 130 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0631-5

200 Florian Manfred Grätz
Teilautomatische Generierung von Stromlauf- und Fluidplänen für mechatronische Systeme
2006 · 192 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0643-9

201 Dieter Eireiner
Prozessmodelle zur statischen Auslegung von Anlagen für das Friction Stir Welding
2006 · 214 Seiten · 20,5 x 14,5 cm · ISBN 3-8316-0650-1

202 Gerhard Volkwein
Konzept zur effizienten Bereitstellung von Steuerungsfunktionalität für die NC-Simulation
2007 · 192 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0668-9

203 Sven Roeren
Komplexitätsvariable Einflussgrößen für die bauteilbezogene Struktursimulation thermischer Fertigungsprozesse
2007 · 224 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0680-1

204 Henning Rudolf
Wissensbasierte Montageplanung in der Digitalen Fabrik am Beispiel der Automobilindustrie
2007 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0697-9

205 Stella Clarke-Griebsch
Overcoming the Network Problem in Telepresence Systems with Prediction and Inertia
2007 · 150 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0701-3

206 Michael Ehrenstraßer
Sensoreinsatz in der telepräsenten Mikromontage
2008 · 160 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0743-3

207 Rainer Schack
Methodik zur bewertungsorientierten Skalierung der Digitalen Fabrik
2008 · 248 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0748-8

208 Wolfgang Sudhoff
Methodik zur Bewertung standortübergreifender Mobilität in der Produktion
2008 · 276 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0749-5

209 Stefan Müller
Methodik für die entwicklungs- und planungsbegleitende Generierung und Bewertung von Produktionsalternativen
2008 · 240 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0750-1

210 Ulrich Kohler
Methodik zur kontinuierlichen und kostenorientierten Planung produktionstechnischer Systeme
2008 · 232 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0753-2

211 Klaus Schlickenrieder
Methodik zur Prozessoptimierung beim automatisierten elastischen Kleben großflächiger Bauteile
2008 · 204 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0776-1

212 Niklas Möller
Bestimmung der Wirtschaftlichkeit wandlungsfähiger Produktionssysteme
2008 · 260 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0778-5

213 Daniel Siedl
Simulation des dynamischen Verhaltens von Werkzeugmaschinen während Verfahrbewegungen
2008 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0779-2

214 Dirk Ansorge
Auftragsabwicklung in heterogenen Produktionsstrukturen mit spezifischen Planungsfreiräumen
2008 · 146 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0785-3

215 Georg Wünsch
Methoden für die virtuelle Inbetriebnahme automatisierter Produktionssysteme
2008 · 224 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0795-2

216 Thomas Oertli
Strukturmechanische Berechnung und Regelungssimulation von Werkzeugmaschinen mit elektromechanischen Vorschubantrieben
2008 · 194 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0798-3

217 Bernd Petzold
Entwicklung eines Operatorarbeitsplatzes für die telepräsente Mikromontage
2008 · 234 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0805-8

218 Loucas Papadakis
Simulation of the Structural Effects of Welded Frame Assemblies in Manufacturing Process Chains
2008 · 260 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0813-3

219 Mathias Mörtl
Ressourcenplanung in der variantenreichen Fertigung
2008 · 210 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0820-1

220 Sebastian Weig
Konzept eines integrierten Risikomanagements für die Ablauf- und Strukturgestaltung in Fabrikplanungsprojekten
2008 · 232 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0823-2

221 Tobias Hornfeck
Laserstrahlbiegen komplexer Aluminiumstrukturen für Anwendungen in der Luftfahrtindustrie
2008 · 150 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0826-3

222 Hans Egermeier
Entwicklung eines Virtual-Reality-Systems für die Montagesimulation mit kraftrückkoppelnden Handschuhen
2008 · 210 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0833-1

223 Matthäus Sigl
Ein Beitrag zur Entwicklung des Elektronenstrahlsinterns
2008 · 185 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0841-6

224 Mark Harfensteller
Eine Methodik zur Entwicklung und Herstellung von Radiumtargets
2009 · 196 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0849-8

225 Jochen Werner
Methode zur roboterbasierten förderbandsynchronen Fließmontage am Beispiel der Automobilindustrie
2009 · 210 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0857-7

226 Florian Hagemann
Ein formflexibles Werkzeug für das Rapid Tooling beim Spritzgießen
2009 · 226 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0861-4

227 Haitham Rashidy
Knowledge-based quality control in manufacturing processes with application to the automotive industry
2009 · 212 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0862-1

228 Wolfgang Vogl
Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern
2009 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0869-0

229 Sonja Schedl
Integration von Anforderungsmanagement in den mechatronischen Entwicklungsprozess
2009 · 160 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0874-4

230 Andreas Trautmann
Bifocal Hybrid Laser Welding – A Technology for Welding of Aluminium and Zinc-Coated Steels
2009 · 268 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0876-8

231 Patrick Neise
Managing Quality and Delivery Reliability of Suppliers by Using Incentives and Simulation Models
2009 · 224 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0878-2

232 Christian Habicht
Einsatz und Auslegung zeitfensterbasierter Planungssysteme in überbetrieblichen Wertschöpfungsketten
2009 · 200 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0891-1

233 Michael Spitzweg
Methode und Konzept für den Einsatz eines physikalischen Modells in der Entwicklung von Produktionsanlagen
2009 · 180 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0931-4

234 Ulrich Munzert
Bahnplanungsalgorithmen für das robotergestützte Remote-Laserstrahlschweißen
2010 · 176 Seiten · 20,5 x 14,5 cm · ISBN 978-3-8316-0948-2

235 Georg Völlner
Rührreibschweißen mit Schwerlast-Industrierobotern
2010 · 232 Seiten · 20,5 x 14,5 cm · 978-3-8316-0955-0

236 Nils Müller
Modell für die Beherrschung und Reduktion von Nachfrageschwankungen
2010 · 270 Seiten · 20,5 x 14,5 cm · 978-3-8316-0992-5

237 Franz Decker
Unternehmensspezifische Strukturierung der Produktion als permanente Aufgabe
2010 · 180 Seiten · 20,5 x 14,5 cm · 978-3-8316-0996-3

238 Christian Lau
Methodik für eine selbstoptimierende Produktionssteuerung
2010 · 200 Seiten · 20,5 x 14,5 cm · 978-3-8316-4012-6

239 Christoph Rimpau
Wissensbasierte Risikobewertung in der Angebotskalkulation für hochgradig individualisierte Produkte
2010 · 200 Seiten · 20,5 x 14,5 cm · 978-3-8316-4015-7

240 Michael Loy
Modulare Vibrationswendelförderer zur flexiblen Teilezuführung
2010 · 169 Seiten · 20,5 x 14,5 cm · 978-3-8316-4027-0

241 Andreas Eursch
Konzept eines immersiven Assistenzsystems mit Augmented Reality zur Unterstützung manueller Aktivitäten in radioaktiven Produktionsumgebungen
2010 · 205 Seiten · 20,5 x 14,5 cm · 978-3-8316-4029-4

242 Florian Schwarz
Simulation der Wechselwirkungen zwischen Prozess und Struktur bei der Drehbearbeitung
2010 · 256 Seiten · 20,5 x 14,5 cm · 978-3-8316-4030-0

243 Martin Georg Prasch
Integration leistungsgewandelter Mitarbeiter in die variantenreiche Serienmontage
2010 · 261 Seiten · 20,5 x 14,5 cm · 978-3-8316-4033-1

244 Johannes Schilp
Adaptive Montagesysteme für hybride Mikrosysteme unter Einsatz von Telepräsenz
2011 · 160 Seiten · 20,5 x 14,5 cm · 978-3-8316-4063-8

245 Stefan Lutzmann
Beitrag zur Prozessbeherrschung des Elektronenstrahlschmelzens
2011 · 222 Seiten · 20,5 x 14,5 cm · 978-3-8316-4070-6

246 Gregor Branner
Modellierung transienter Effekte in der Struktursimulation von Schichtbauverfahren
2011 · 230 Seiten · 20,5 x 14,5 cm · 978-3-8316-4071-3

247 Josef Ludwig Zimmermann
Eine Methodik zur Gestaltung berührungslos arbeitender Handhabungssysteme
2011 · 184 Seiten · 20,5 x 14,5 cm · 978-3-8316-4091-1

248 Clemens Pörnbacher
Modellgetriebene Entwicklung der Steuerungssoftware automatisierter Fertigungssysteme
2011 · 280 Seiten · 20,5 x 14,5 cm · 978-3-8316-4108-6

249 Alexander Lindworsky
Teilautomatische Generierung von Simulationsmodellen für den entwicklungsbegleitenden Steuerungstest
2011 · 300 Seiten · 20,5 x 14,5 cm · 978-3-8316-4125-3

250 Michael Mauderer
Ein Beitrag zur Planung und Entwicklung von rekonfigurierbaren mechatronischen Systemen – am Beispiel von starren Fertigungssystemen
2011 · 150 Seiten · 20,5 x 14,5 cm · 978-3-8316-4126-0

251 Roland Mork
Qualitätsbewertung und -regelung für die Fertigung von Karosserieteilen in Presswerken auf Basis Neuronaler Netze
2011 · 228 Seiten · 20,5 x 14,5 cm · 978-3-8316-4127-7

252 Florian Reichl
Methode zum Management der Kooperation von Fabrik- und Technologieplanung
2011 · 224 Seiten · 20,5 x 14,5 cm · 978-3-8316-4128-4

253 Paul Gebhard
Dynamisches Verhalten von Werkzeugmaschinen bei Anwendung für das Rührreibschweißen
2011 · 220 Seiten · 20,5 x 14,5 cm · 978-3-8316-4129-1

254 Michael Heinz
Modellunterstützte Auslegung berührungsloser Ultraschallgreifsysteme für die Mikrosystemtechnik
2012 · 302 Seiten · 20,5 x 14,5 cm · 978-3-8316-4147-5

255 Pascal Krebs
Bewertung vernetzter Produktionsstandorte unter Berücksichtigung multidimensionaler Unsicherheiten
2012 · 244 Seiten · 20,5 x 14,5 cm · 978-3-8316-4156-7

256 Gerhard Straßer
Greiftechnologie für die automatisierte Handhabung von technischen Textilien in der Faserverbundfertigung
2012 · 290 Seiten · 20,5 x 14,5 cm · 978-3-8316-4161-1

257 Frédéric-Felix Lacour
Modellbildung für die physikbasierte Virtuelle Inbetriebnahme materialflussintensiver Produktionsanlagen
2012 · 222 Seiten · 20,5 x 14,5 cm · 978-3-8316-4162-8

258 Thomas Hensel
Modellbasierter Entwicklungsprozess für Automatisierungslösungen
2012 · 220 Seiten · 20,5 x 14,5 cm · 978-3-8316-4167-3

259 Sherif Zaidan
A Work-Piece Based Approach for Programming Cooperating Industrial Robots
2012 · 210 Seiten · 20,5 x 14,5 cm · 978-3-8316-4175-8

260 Hendrik Schellmann
Bewertung kundenspezifischer Mengenflexibilität im Wertschöpfungsnetz
2012 · 224 Seiten · 20,5 x 14,5 cm · 978-3-8316-4189-5

261 Marwan Radi
Workspace scaling and haptic feedback for industrial telepresence and teleaction systems with heavy-duty teleoperators
2012 · 172 Seiten · 20,5 x 14,5 cm · 978-3-8316-4195-6

262 Markus Ruhstorfer
Rührreibschweißen von Rohren
2012 · 206 Seiten · 20,5 x 14,5 cm · 978-3-8316-4197-0

263 Rüdiger Daub
Erhöhung der Nahttiefe beim Laserstrahl-Wärmeleitungsschweißen von Stählen
2012 · 182 Seiten · 20,5 x 14,5 cm · 978-3-8316-4199-4

264 Michael Ott
Multimaterialverarbeitung bei der additiven strahl- und pulverbettbasierten Fertigung
2012 · 220 Seiten · 20,5 x 14,5 cm · 978-3-8316-4201-4

265 Martin Ostgathe
System zur produktbasierten Steuerung von Abläufen in der auftragsbezogenen Fertigung und Montage
2012 · 278 Seiten · 20,5 x 14,5 cm · 978-3-8316-4206-9

266 Imke Nora Kellner
Materialsysteme für das pulverbettbasierte 3D-Drucken
2013 · 208 Seiten · 20,5 x 14,5 cm · 978-3-8316-4223-6

267 Florian Oefele
Remote-Laserstrahlschweißen mit brillanten Laserstrahlquellen
2013 · 238 Seiten · 20,5 x 14,5 cm · 978-3-8316-4224-3