



LUND UNIVERSITY

Adaptation and Learning for Manipulators and Machining

Sörnmo, Olof

2015

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Sörnmo, O. (2015). *Adaptation and Learning for Manipulators and Machining*. [Doctoral Thesis (monograph), Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:

1

Creative Commons License:

CC0

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Adaptation and Learning for Manipulators and Machining

Olof Sörnmo



LUND
UNIVERSITY

Department of Automatic Control

PhD Thesis
ISRN LUTFD2/TFRT--1110--SE
ISBN 978-91-7623-492-1 (print)
ISBN 978-91-7623-493-8 (web)
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

© 2015 by Olof Sörnmo. All rights reserved.
Printed in Sweden by Media-Tryck.
Lund 2015

Abstract

This thesis presents methods for improving the accuracy and efficiency of tasks performed using different kinds of industrial manipulators, with a focus on the application of machining. Industrial robots offer a flexible and cost-efficient alternative to machine tools for machining, but cannot achieve as high accuracy out of the box. This is mainly caused by non-ideal properties in the robot joints such as backlash and compliance, in combination with the strong process forces that affect the robot during machining operations. In this thesis, three different approaches to improving the robotic machining accuracy are presented. First, a macro/micro-manipulator approach is considered, where an external compensation mechanism is used in combination with the robot, for compensation of high-frequency Cartesian errors. Two different milling scenarios are evaluated, where a significant increase in accuracy was obtained. The accuracy specification of $50\ \mu\text{m}$ was reached for both scenarios. Because of the limited workspace and the higher bandwidth of the compensation mechanism compared to the robot, two different mid-ranging approaches for control of the relative position between the robot and the compensator are developed and evaluated. Second, modeling and identification of robot joints is considered. The proposed method relies on clamping the manipulator end effector and actuating the joints, while measuring joint motor torque and motor position. The joint stiffness and backlash can subsequently be extracted from the measurements, to be used for compensation of the deflections that occur during machining. Third, a model-based iterative learning control (ILC) approach is proposed, where feedback is provided from three different sensors of varying investment costs. Using position measurements from an optical tracking system, an error decrease of up to 84% was obtained. Measurements of end-effector forces yielded an error decrease of 55%, and a force-estimation method based on joint motor torques decreased the error by 38%.

Further investigation of ILC methods is considered for a different kind of manipulator, a marine vibrator, for the application of marine seismic acquisition. A frequency-domain ILC strategy is proposed, in order to

attenuate undesired overtones and improve the tracking accuracy. The harmonics were suppressed after approximately 20 iterations of the ILC algorithm, and the absolute tracking error was reduced by a factor of approximately 50.

The final problem considered in this thesis concerns increasing the efficiency of machining tasks, by minimizing cycle times. A force-control approach is proposed to maximize the feed rate, and a learning algorithm for path planning of the machining path is employed for the case of machining in non-isotropic materials, such as wood. The cycle time was decreased by 14 % with the use of force control, and on average an additional 28 % decrease was achieved by use of a learning algorithm. Furthermore, by means of reinforcement learning, the path-planning algorithm is refined to provide optimal solutions and to incorporate an increased number of machining directions.

Acknowledgments

First of all, I would like to thank my supervisors Anders Robertsson and Rolf Johansson, who have been supportive and provided valuable guidance and encouragement. Further thanks go to Klas Nilsson, who has practically been an additional supervisor and a great source of ideas. Much of my research has been performed in close cooperation with Björn Olofsson, and it has been a pleasure both working and traveling with him. Thanks to Karl Johan Åström for his endless enthusiasm in helping out with everyday control problems. Also, I have enjoyed the numerous trips to Fjellie together with Bo Bernhardsson, iteratively learning how to control "the barrel".

I would like to thank all my colleagues at the Department of Automatic Control for contributing to a very pleasant environment to work in, especially my office colleagues Andreas Stolt, Josefin Berner, and Martin Hast. Thanks to Eva Westin, Ingrid Nilsson, Mika Nishimura, and Monika Rasmusson for helping with administrative matters. Leif Andersson, Anders Nilsson, Rolf Braun, Anders Blomdell, and Pontus Andersson have provided valuable assistance with technical issues. Also, thanks to our Toughest team *Control Freaks* for keeping up the impressive team spirit, and completing the challenge with elegance!

Finally, I would like to thank my wife Tove and my family for support and guidance.

Financial Support

Financial support is acknowledged from the European Commission's Seventh Framework Programme under grant agreements with the acronyms COMET (Ref. #258769) and SMErobotics (Ref. #287787) as well as the Swedish Foundation for Strategic Research within the program ProViking, under grant ProFlexA PV08-0036. The author is part of the LCCC Linnaeus Center, supported by the Swedish Research Council, and the ELLIIT Excellence Center, supported by the Swedish Government.

Contents

1. Introduction	9
1.1 Background and Motivation	9
1.2 Contributions	12
1.3 Publications	12
1.4 Thesis Outline	15
2. Macro/Micro-Manipulator Approach to Machining	16
2.1 Introduction	16
2.2 System Topology	20
2.3 Modeling and Control Design	26
2.4 Micro-Manipulator Identification Results	35
2.5 Results from Prototype Machining Cell	37
2.6 Results from Complete Machining Cell	43
2.7 Discussion	53
2.8 Conclusions	55
3. Adaptive Mid-Ranging Control	57
3.1 Introduction	57
3.2 Method	58
3.3 Simulation Results	65
3.4 Experimental Setup	66
3.5 Experimental Results	67
3.6 Discussion and Conclusions	73
4. Modeling and Identification of Robot Joints	76
4.1 Introduction	76
4.2 Modeling Approach	78
4.3 Clamping Method	79
4.4 Experimental Setup	83
4.5 Experimental Results	84
4.6 Discussion and Conclusions	90

5. Iterative Learning Control for Machining	92
5.1 Introduction	92
5.2 Method	94
5.3 Experimental Setup	103
5.4 Experimental Results	104
5.5 Discussion	111
5.6 Conclusions	113
6. Frequency-Domain Iterative Learning Control	115
6.1 Introduction	115
6.2 Marine Seismic Acquisition	116
6.3 Marine Vibrators	118
6.4 Control System Design Principles	122
6.5 Modeling and Control Design	122
6.6 Experimental Results	128
6.7 Comparison to Time-Domain ILC	137
6.8 Discussion and Conclusions	139
7. Cycle-Time Minimization of Wood Milling	144
7.1 Introduction	144
7.2 Modeling and Control Design	147
7.3 Results	157
7.4 Discussion	167
7.5 Conclusions	171
8. Reinforcement-Learning Approach to Path Planning	173
8.1 Introduction	173
8.2 Method	174
8.3 Results	181
8.4 Discussion and Conclusions	185
9. Conclusions	188
Bibliography	191

1

Introduction

The topic of this thesis is on how to improve the accuracy and efficiency of tasks performed using different kinds of industrial manipulators, by means of adaptation, learning, and control. A major part of the thesis is dedicated to the application of machining with industrial robots.

1.1 Background and Motivation

A common manipulator in today's industry is the industrial robot, such as the one shown in Figure 1.1. Industrial robots offer flexible solutions, in terms of reconfiguration possibilities and the wide range of tasks they can perform. Additionally, for machining applications, industrial robots are cost efficient compared to conventional machine tools, such as computer numerical control (CNC) machines.

Industrial robots can be accurately position controlled when moving in free space, suitable for operations such as pick-and-place and spray painting. However, when performing tasks that require the robot to come into stiff physical contact with the environment, such as machining, high accuracy is harder to achieve. A significant cause of the insufficient accuracy in machining tasks performed with industrial robots, is the combination of comparably low stiffness of the joints in a serial-kinematic manipulator and external process forces affecting the end effector. Also, nonlinearities in the joints, such as backlash and friction, further deteriorate the positioning accuracy. Typically, the robot's motion control is based on distributed feedback from the angle measurements of the individual joint motors. Consequently, the desired arm-side or task-space measurements are not available. Since the major part of the induced position deflections in the machining processes appear on the arm-side of the robot joints, they cannot be detected by the motor-angle measurements. As a result of these issues, machining operations performed with industrial robots are not straightforward to perform [Zhang et al., 2005a; Abele et al., 2011]. The machining tolerances generally cannot



Figure 1.1 A conventional ABB industrial robot.

be met—typical values of the accuracy that can be achieved with CNC machines are in the range of $1\ \mu\text{m}$, and approximately two magnitudes higher for conventional industrial robots. This has limited the usage of industrial robots for performing machining operations, which motivates the need for developing new control strategies for improving the accuracy of robotic machining tasks.

In this thesis, three different approaches to increasing the accuracy of machining operations performed with industrial robots are investigated. First, a macro/micro-manipulator configuration for high-bandwidth online compensation of the measured robot deviation is considered. Two different robot cells are developed and evaluated in a series of milling experiments in aluminum and steel. Also, since the considered type of micro manipulator has a limited workspace, mid-ranging control strategies are developed for controlling the relative position of the manipulators.

Second, the non-ideal properties of the robot joints that lower the position accuracy are modeled, and a method for identifying the unknown joint parameters is presented.

Third, an iterative solution to increasing the absolute accuracy is developed, where feedback from three different sensors are used for the proposed iterative learning control (ILC) algorithms. The proposed solution is evaluated in a series of milling experiments in aluminum.

The subject of ILC is further investigated for a different kind of

manipulator; a *marine vibrator*. By precise position control of the marine vibrator, it can be used to emit well-defined signals in a specified frequency range, which is required for the intended use in marine seismic acquisition. The objective of marine seismic acquisition is to map the ocean floor, which is achieved by analyzing the reflections of the signals emitted by the vibrator. Since it has been found that frequencies above 100 Hz may interfere with marine animal life, the output of the vibrator must be accurately controlled to not emit higher frequencies. However, as marine vibrators are advanced mechanical systems, nonlinear effects and high-frequency harmonics are present, which makes the control problem challenging. Modeling of the marine vibrator, and a subsequent frequency-domain ILC algorithm is presented. Additionally, an adaptive algorithm for reidentification of the vibrator dynamics is presented. Both methods are evaluated using an experimental setup in air.

Another aspect of machining that is considered in this thesis, is how to increase the efficiency of the operation by adjusting the feed rate in order to maximize the material removal. If a certain task is to be performed a large amount of times, reductions in cycle time will generate considerable cost savings over time. In the case of machining tasks, cycle-time reductions can be achieved by removing as much material per time unit as possible. The limiting factor to the material removal rate is the resulting reaction force on the tool during machining, since a too strong force will break the tool or scorch the workpiece. Thus, it is desirable to strive to always applying the maximum allowed force during machining, which can be achieved by adjusting the feed rate of the workpiece. The reaction force does, however, nonlinearly depend on several time-varying parameters, motivating the need for an adaptive control structure that can exercise precise material removal control in varying cutting conditions. Furthermore, for machining tasks performed in non-isotropic materials such as wood, additional cycle-time reductions can be achieved by intelligent path planning.

Modeling of the robot and the machining process dynamics, and a subsequent model-based adaptive control structure for feed-rate control is presented. A learning algorithm for optimizing the machining path, which is independent of *a priori* information, is presented. The methods are combined and verified in both simulations and milling experiments. A subsequent refinement of the path-planning algorithm, based on reinforcement learning, is presented. Furthermore, the problem is extended by considering additional milling directions, and a proposed solution using feature-based Q-learning is presented and evaluated in simulations.

1.2 Contributions

The main contributions of this thesis are:

- Modeling and control of an external position-compensating micro manipulator, experimentally verified using two different robot cells;
- Two different adaptive mid-ranging approaches to position control of the relative position between the macro and micro manipulator;
- Modeling and identification of robot joints using the clamping method;
- An ILC approach to increasing the absolute accuracy for machining, using feedback from three different sensors;
- A frequency-domain ILC algorithm for position control of a marine vibrator, in combination with an adaptive reidentification algorithm;
- A learning approach to cycle-time minimization for machining tasks, based on adaptive force control in combination with a path-planning algorithm;
- A reinforcement-learning approach to minimizing cycle times by intelligent path planning, considering up to four machining directions.

1.3 Publications

Preliminary versions of parts of the research presented in this thesis was published in the Licentiate Thesis by the author:

Sörnmo, O. (2013). *Control Strategies for Machining with Industrial Robots*. Licentiate Thesis ISRN LUTFD2/TFRT--3261--SE. Department of Automatic Control, Lund University, Sweden.

The publications on which this thesis is based are presented in the following:

Olofsson, B., O. Sörnmo, U. Schneider, A. Robertsson, A. Puzik, and R. Johansson (2011). “Modeling and control of a piezo-actuated high-dynamic compensation mechanism for industrial robots”. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Francisco, CA, pp. 4704–4709.

Sörnmo, O., B. Olofsson, U. Schneider, A. Robertsson, and R. Johansson (2012). “Increasing the milling accuracy for industrial robots using a piezo-actuated high-dynamic micro manipulator”. In: *IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics*. Kaohsiung, Taiwan, pp. 104–110.

Schneider, U., B. Olofsson, O. Sörnmo, M. Drust, A. Robertsson, M. Hägele, and R. Johansson (2014). “Integrated approach to robotic machining with macro/micro-actuation”. *J. Robotics and Computer-Integrated Manufacturing* **30**:6, pp. 636–647.

The first two publications were developed as a cooperation between the author and B. Olofsson, and equal contribution is asserted. The author was main responsible for the experimental verification and B. Olofsson was responsible for the modeling and control development. A. Puzik developed the considered micro manipulator and U. Schneider took part in the implementation and the experiments. A. Robertsson and R. Johansson provided feedback on the manuscripts.

The third publication was developed as a cooperation between the author, B. Olofsson and U. Schneider, and equal contribution is asserted. A. Robertsson, M. Hägele, and R. Johansson provided feedback on the manuscript.

Sörnmo, O., B. Olofsson, A. Robertsson, and R. Johansson (2013). “Adaptive internal model control for mid-ranging of closed-loop systems with internal saturation”. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Tokyo, Japan, pp. 4893–4899.

This publication has the author as main contributor. B. Olofsson assisted with theoretical and technical discussions and together with A. Robertsson and R. Johansson provided feedback on the manuscript.

Lehmann, C., B. Olofsson, K. Nilsson, M. Halbauer, M. Haage, A. Robertsson, O. Sörnmo, and U. Berger (2013). “Robot joint modeling and parameter identification using the clamping method”. In: *Proc. IFAC Conf. Manufacturing Modeling, Management, and Control*. St. Petersburg, Russia, pp. 843–848.

This publication is based on an idea proposed by K. Nilsson, which was realized by B. Olofsson with the assistance of the remaining authors. Experiments were performed by C. Lehmann and M. Halbauer.

Cano Marchal, P., O. Sörnmo, B. Olofsson, A. Robertsson, J. Gómez Ortega, and R. Johansson (2014). “Iterative learning control for machining with industrial robots”. In: *IFAC World Congress*. Cape Town, South Africa, pp. 9327–9333.

This publication was developed as a cooperation between the author, P. Cano Marchal and B. Olofsson. The author was main responsible for the experimental verification and P. Cano Marchal was responsible for the

modeling and control development, with assistance from the author and B. Olofsson. A. Robertsson and R. Johansson provided feedback on the manuscript.

Sörnmo, O., B. Olofsson, A. Robertsson, and R. Johansson (2012). “Increasing time-efficiency and accuracy of robotic machining processes using model-based adaptive force control”. In: *Int. IFAC Symp. Robot Control*. Dubrovnik, Croatia, pp. 543–548.

Sörnmo, O., B. Olofsson, A. Robertsson, and R. Johansson (2015). “Learning approach to cycle-time minimization of wood milling using adaptive force control”. *ASME J. Manufacturing Science and Engineering* **138**:1, pp. 011013–011013-11. DOI: 10.1115/1.4030751.

The above publications have the author as main contributor. B. Olofsson assisted with theoretical and technical discussions and together with A. Robertsson and R. Johansson provided feedback on the manuscripts.

Sörnmo, O., B. Bernhardsson, O. Kröling, P. Gunnarsson, and R. TENGHAMN (2015). “Frequency-domain iterative learning control of a marine vibrator”. *J. Control Engineering Practice*. Submitted.

This publication was developed as a cooperation between the author and B. Bernhardsson. O. Kröling and P. Gunnarsson provided technical assistance and discussions, and together with R. TENGHAMN assisted the author in writing the manuscript.

Other Publications

The following publications, in which the author has contributed in related areas, are not part of the thesis:

Olofsson, B., O. Sörnmo, U. Schneider, M. Barho, A. Robertsson, and R. Johansson (2012). “Increasing the accuracy for a piezo-actuated micro manipulator for industrial robots using model-based nonlinear control”. In: *Int. IFAC Symp. Robot Control*. Dubrovnik, Croatia, pp. 277–282.

Sörnmo, O., A. Robertsson, and A. Wanner (2012). “Force controlled knife-grinding with industrial robot”. In: *IEEE Multi-Conference on Systems and Control*. Dubrovnik, Croatia, pp. 1356–1361.

Olofsson, B., O. Sörnmo, A. Robertsson, and R. Johansson (2014). “Continuous-time gray-box identification of mechanical systems using subspace-based identification methods”. In: *IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics*. Besançon, France, pp. 327–333.

All publications are available for download from <http://www.control.lth.se/Publications.html>.

1.4 Thesis Outline

In Chapter 2, two different macro/micro-manipulator setups for increasing milling accuracy are introduced. Modeling and control of the micro manipulator as well as experimental verification in different milling scenarios are presented. Chapter 3 presents an adaptive method for performing mid-ranging control of a macro/micro-manipulator setup, and the method is evaluated using a mock-up setup. Modeling and identification of robot joint properties are found in Chapter 4. Increasing machining accuracy by means of ILC is investigated in Chapter 5, where three different methods are developed and evaluated. Further investigation of ILC is provided in Chapter 6, where a frequency-domain algorithm is developed for position control of a marine vibrator. Cycle-time minimization of machining processes, by means of feed-rate control and path planning, is considered in Chapter 7. An adaptive force control strategy in combination with a learning algorithm for path planning is presented and assessed in milling experiments. In Chapter 8, a reinforcement-learning approach to the path-planning problem is taken, and the problem is extended to incorporate additional machining directions. Finally, conclusions are drawn in Chapter 9.

2

Macro/Micro-Manipulator Approach to Machining

2.1 Introduction

This chapter is based on the publications [Olofsson et al., 2011b], [Sörnmo et al., 2012b], and [Schneider et al., 2014].

In the EU/FP7 project COMET [COMET, 2013], the aim was to increase the accuracy of machining tasks for industrial robots. In particular, milling solutions of accuracies within $50\ \mu\text{m}$, were developed. In order to achieve this, the proposed approach was divided into four parts, as displayed in Figure 2.1. This chapter considers the method represented by the green jigsaw-piece in Figure 2.1; "High Dynamics Compensation Mechanism".

Two different robot cell setups are presented in this chapter; a prototype cell for one-dimensional milling in aluminum and a complete robot cell with

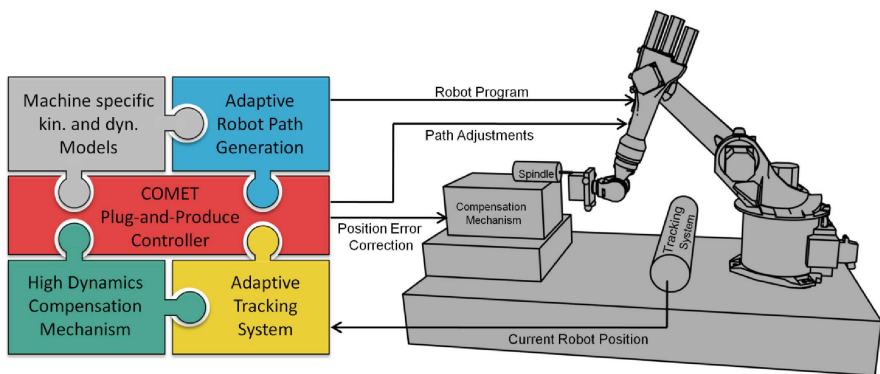


Figure 2.1 The four-part approach to machining of the EU/FP7 COMET-project [COMET, 2013].

an integrated approach to three-dimensional robotic machining in steel. Both cells utilize macro/micro actuation with a conventional industrial robot and an external piezo-actuated 3D compensation mechanism [Puzik et al., 2009; Puzik, 2011].

Modeling of the dynamic properties of the compensation mechanism and how the subsequent control can be designed to obtain satisfactory milling results is presented. It is shown how nonlinear effects in the mechanism can be handled and how oscillations that occur because of the resonant mechanical construction can be reduced by using appropriate control-design methods. Additionally, for the complete robot cell, a mid-ranging control strategy is designed for the integrated macro/micro manipulator control, which is subsequently implemented.

Experimental verifications of the proposed control structures for machining with industrial robots are performed, where results from milling tasks in both aluminum and steel are presented. The experimental verification contrasts the milling accuracy obtained using the compensation mechanism to the standard uncompensated case, as well as to compensation using only the robot.

Background

Motion control for industrial robots has been studied for several decades and the development has thus reached a mature level [Spong et al., 2006; Siciliano et al., 2009]. The joints of the robot are typically position controlled, while the Cartesian end-effector position and orientation are estimated based on the forward kinematic relations, *i.e.*, no explicit workspace sensing is used. While certain methods and commercial products offered by robot manufacturers exist for achieving very high position accuracy for movement of the robot end effector in free space or under constant load, achieving this in the presence of dynamic process forces affecting the end effector is much more challenging. Insufficient position accuracy in machining with industrial robots is a well-known problem in manufacturing; an experimental investigation was presented in [Schneider et al., 2013a]. The accuracy tolerances in manufacturing processes are usually in the range of 50 μm or lower [COMET, 2013]. This can typically not be achieved using a conventional industrial robot in application scenarios where strong process forces are required to execute the desired task.

The fundamental problem of insufficient position accuracy of the robot in the presence of process forces primarily originates from non-ideal joint properties, such as backlash, friction, and compliance of the gear box, see *e.g.*, [Hovland et al., 2002; Ruderman et al., 2009; Bittencourt et al., 2010]. These properties significantly degrade the position accuracy of the robot, and consequently, the accuracy of the machined parts. The feedback control of

the robot joint positions is typically based on sensor data from the motor side of the gear box, whereas the primary interest in applications is the arm-side positions, since they represent the actual workspace position, given that link flexibilities are neglected.

State of the Art

Previous approaches to increasing the position accuracy in robotic machining are primarily based on kinematic calibration [Roth et al., 1987; Schröer et al., 1997; Joubair et al., 2013] and stiffness modeling of the manipulator [Abele et al., 2007; Zhang et al., 2005b; Wang et al., 2009; Pan and Zhang, 2009; Reinl et al., 2011; Dumas et al., 2011; Tyapin et al., 2014]. In the kinematic calibration procedure, the kinematic parameters of the robot, *e.g.*, the Denavit-Hartenberg parametrization [Denavit and Hartenberg, 1955], are determined with high accuracy using, *e.g.*, optical measurement systems for a constant load attached to the robot end effector. A method for modeling and identifying the robot joint properties is considered in Chapter 4.

Other approaches to increasing the position accuracy in robotic machining processes are based on sensor data from high-precision 3D or 6D position-measurement sensors [Schneider et al., 2013b], also referred to as coordinate measuring machines (CMM) [Cuypers et al., 2009]. The sensor data can be used to provide end-effector position feedback for online corrections. The main limitation of these methods in machining is the disturbance rejection bandwidth at the end effector of the robot manipulator, but also communication delays for sensor data and noise in the measurements influence the achievable performance. As a result of the cutting process, high-frequency disturbances on the robot end-effector position are to be expected. Therefore, a high-bandwidth position control of the robot is essential in order to achieve sufficient accuracy of the machined parts. As relatively low feed rates and fast spindle speeds are used for the experiments in this chapter, chatter in the milling process is not considered. For the case of high-speed milling, the methods for detecting and avoiding chatter presented in [Van Dijk et al., 2010] could be used.

Another set of methods proposed for increasing the accuracy of industrial manipulators is based on iterative learning control (ILC). An ILC approach is considered in Chapter 5.

In contrast to the approaches described in the previous paragraphs, the strategy proposed in this chapter comprises workspace sensing using 6D position sensors, combined with macro/micro actuation. The concepts of macro and micro manipulators were suggested in [Egeland and Sagli, 1990; Sharon et al., 1993], together with a control architecture for increased bandwidth of the end-effector position control. The macro actuator has a large workspace, but a limited position-control bandwidth. Typical

values for the bandwidth of the end-effector position control for industrial manipulators are in the range of 10–30 Hz, depending on configuration [Schneider et al., 2013a]. In contrast, the micro manipulator has significantly higher bandwidth, but a geometrically limited workspace. Hence, the micro manipulator is to compensate for the high-frequency position deviations that occur during milling, which the macro manipulator *per se* is unable to compensate for because of its limited disturbance rejection bandwidth at the end effector. The notions of macro/micro actuation and manipulators have been adopted in this chapter.

Piezo-actuated mechanisms based on flexure elements have been proposed for nano manipulation previously, *e.g.*, [Liaw and Shirinzadeh, 2010; Li and Xu, 2011]. Although the compensation mechanisms considered in this chapter utilizes similar components in their mechanical designs, there are certain differences. Previous designs were designed for compensation in micro and nano manipulation, whereas the micro manipulators considered in this chapter are designed for machining processes with industrial robots, where strong process forces are required to fulfill the specified task. Nevertheless, the control design in [Li and Xu, 2011] relates to the control algorithms developed for the micro manipulator in this thesis. Further research in this area is presented in [Eielsen et al., 2015], where control design for high-accuracy tracking using repetitive control in the context of piezo-actuated flexure-based mechanical systems was presented.

Problem Formulation

The objective of the research presented in this chapter is to develop solutions to increase the accuracy of milling operations performed with industrial robots. To this purpose, two different setups are considered. The aim of the first setup is to maximize the absolute accuracy, as well as the workpiece surface accuracy for one-dimensional aluminum millings, performed in each of the Cartesian directions using a prototype of the robot cell. The aim of the second setup is to develop a complete robot cell with an accompanying integrated control architecture, to the purpose of high-accuracy 3D robotic machining in steel. The objective of the second setup is to obtain a mean absolute error below 50 μm , and to maximize the surface accuracy of the machined part.

The long-term goal of the research presented in this chapter is to enable manufacturing with industrial robots, based on CAD specifications, achieving machine-tool accuracy of the produced parts.

Accuracy Definitions

The terms used in this thesis to describe the accuracy of machining tasks are defined here. Figure 2.2 illustrates the concepts of absolute and surface

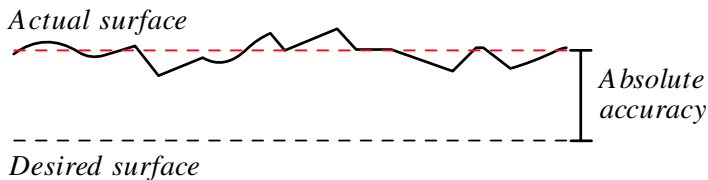


Figure 2.2 Illustration of absolute and surface accuracy. The dashed black line represents the desired surface, whereas the solid black line shows the actual surface obtained from milling. The dashed red line is the mean of the actual surface.

accuracy. The absolute accuracy is a measure of how close to the desired surface the actual obtained surface is, in this thesis measured by the *mean absolute error* (MAE). The surface accuracy describes the smoothness of the actual surface, *i.e.*, the deviation around the mean of the actual surface.

2.2 System Topology

In this section, the system topology for both the prototype and the complete robot cell are described. In addition, the communication interfaces and execution aspects for the different hardware components are discussed.

Two different configurations can be considered for robotic machining. First, the robot can carry the spindle, while the workpiece is attached to a fixed base. In this configuration, the robot benefits from its large workspace and can machine large-scale workpieces. In the second configuration, the robot holds the workpiece, whereas the spindle is in a fixed position in the cell. This setup is suitable for smaller and lighter workpieces, as the workpiece geometry and mass are limited by the workspace and the payload of the robot. Also, this configuration allows the robot to have access to different machining operations by moving to different fixed tools within the workcell. Additionally, handling operations can be performed, facilitating the integration in automated production lines. In the research presented in this chapter, the second configuration was chosen; thus the spindle was attached to the micro manipulator and the workpiece was held by the robot. This was motivated by the heavy weight of the micro manipulator together with the machining spindle.

Prototype Robotic Machining Cell

The prototype machining cell was developed in the lab at Fraunhofer IPA, Stuttgart, Germany. The cell consists of a REIS industrial robot of model RV40 [Reis GmbH, 2011] with a maximum payload of 40 kg, as the macro

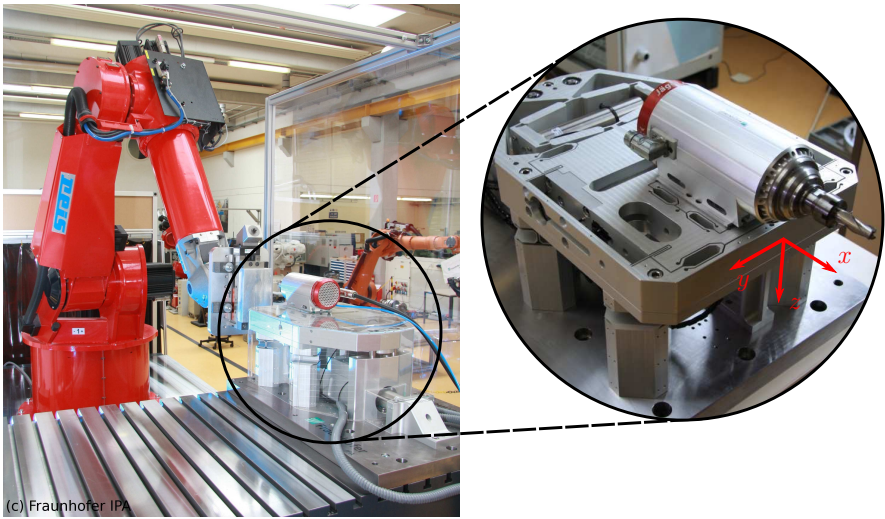


Figure 2.3 The prototype machining cell for real-time compensation of position errors during machining operations, where the robot holds the workpiece and the milling spindle is attached to the micro manipulator. A close-up of the micro manipulator, as seen from the opposite side, is displayed to the right in the figure. Photo courtesy of Fraunhofer IPA, Stuttgart, Germany.

manipulator. The spindle was attached to the micro manipulator and the macro manipulator held the workpiece, which for this setup was a block of aluminum (AlMg3,5). The setup is such that both face milling and peripheral milling, also referred to as radial milling, can be performed. An overview of the prototype machining cell is displayed in Figure 2.3.

Prototype Micro Manipulator Accurate positioning and high actuation bandwidth—high relative to the robot end-effector bandwidth—are key features for the design of an external compensation mechanism, *i.e.*, the micro manipulator, for robot machining [Sharon et al., 1993]. The objective is to use the micro manipulator for keeping the relative position between tool and workpiece according to the specified reference. A prototype micro manipulator with three translatory axes [Puzik et al., 2009], and a serial structure enabling position compensation in a Cartesian coordinate system, was used for the prototype machining cell. The axes were driven by piezo actuators, whose extensions are changed by applying voltages and the extensions are measured using strain gauges, attached to the actuators. Piezo actuators provide a high bandwidth from applied voltage to extension, and can handle forces up to 30 kN. The process forces for the applications

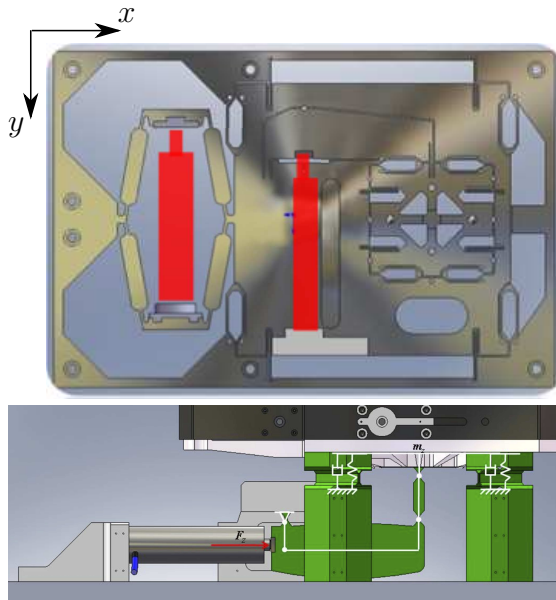


Figure 2.4 Actuation principle for the x -, y -, and z -axes of the prototype micro manipulator. The piezo actuators are marked in red [Puzik et al., 2009].

targeted with the proposed machining robot cells range up to 1 kN. Hence, the micro manipulator can withstand the process forces expected in a machining task. The major benefit of choosing solid-state joints is the lack of backlash and friction, which is essential for micrometer positioning accuracy. Furthermore, the design incorporates solid-state flexure elements for increasing the compensation range, as compared to the piezo-actuator extension. A schematic drawing of the actuation principle of the micro manipulator is provided in Figure 2.4.

The flexure mechanism is designed such that the gear ratio of the displacement of the spindle and the extension of the piezo actuator is approximately five in each direction. This realizes a compensation range for the machining spindle of approximately 400–500 μm in each Cartesian direction. Referring to Figure 2.3, the Cartesian axes for this setup are hereafter called x , y , and z , respectively. The Cartesian displacement of the micro-manipulator end-effector plate is measured using capacitive sensors, one in each direction. For further details on the mechanical design of the prototype micro manipulator, see [Puzik et al., 2009; Puzik et al., 2010; Puzik, 2011].

In order to develop the control structure for the micro-manipulator unit, all sensors and actuators were integrated using a dSPACE system of model DS1103 [dSPACE GmbH, 2007]. The proposed control design was implemented in MATLAB Simulink, then converted to C-code using the *Real-Time Workshop* toolbox [MathWorks Inc., 2010]. The compiled C-code was installed in the dSPACE system and executed at a sampling frequency of 10 kHz.

Optical Tracking System To the purpose of measuring the deflections of the macro manipulator, that are to be compensated by the micro manipulator, a Keyence laser sensor of model LK-G87 [Keyence Corp., 2006], with a resolution of 0.2 μm was used as tracking system.

Compensated and Uncompensated Milling In order to illustrate the benefit of the micro manipulator, the milling experiments were performed both in a setting where compensation with the micro manipulator was utilized, and in a setting with the spindle rigidly attached to a fixed base. In the latter setup, no compensation was performed. The two experimental settings are displayed in Figure 2.5.

In the experiments without compensation, the robot configuration was mirrored, with respect to the center plane of the robot, as compared to the configuration chosen in the experiments with compensation. Consequently, the compliance properties of the macro manipulator in the two configurations are equivalent, which is important in order to make the compensated and uncompensated milling results comparable.

Complete Robotic Machining Cell

In this subsection, the hardware components of the proposed complete machining cell and their characteristics are described in detail. The cell was developed in the lab at Fraunhofer IPA, Stuttgart, Germany. An overview of the cell is shown in Figure 2.6, which contains an industrial robot, the external compensation mechanism, an optical 6D tracking system, a machining spindle, and a CNC controller.

Robot and CNC Controller The foundation of the robot cell was a KUKA industrial robot of model KR125 [KUKA Roboter GmbH, 2013]. The joint positions of the robot were controlled using a Beckhoff TwinCAT CNC controller [Beckhoff Automation GmbH, 2013]. Based on the kinematics of the robot, Cartesian end-effector control could be achieved. Compared to a conventional robot controller, the CNC controller offers several advantages. First, path-planning algorithms, which are optimized for machining tasks, are available. Second, from an implementation point-of-view, the most important feature is the open high-speed interface, which enables online position and orientation corrections based on external sensor data. The implemented

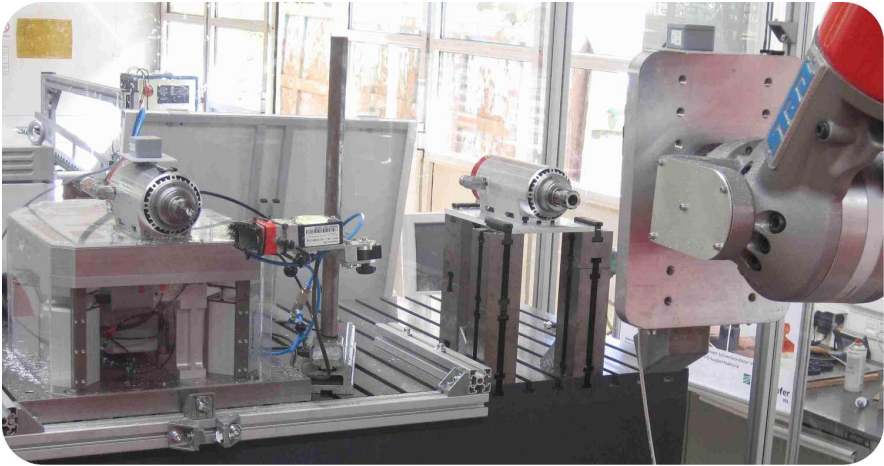


Figure 2.5 Prototype machining cell for evaluation of the effectiveness of the proposed micro manipulator, which is seen to the left. The machining spindle to the right is rigidly attached to the base, and is utilized for milling experiments without compensation.

robot controller is executed on a programmable logic controller (PLC). The integrated Codesys PLC programming environment provides additional possibilities for the implementation of advanced controller structures, such as the architecture proposed in this research.

Micro Manipulator A revised design of the micro manipulator described previously, comprising a parallel actuation principle, was shown to improve the dynamic properties of the mechanism significantly [Schneider et al., 2013c]. In particular, as a result of the reduction of the end-effector mass and a modified geometry of the solid-state flexure elements, higher bandwidth could be achieved. A photo of the mechanical design of the mechanism and the actuation axes are displayed in Figure 2.7. This version of the micro manipulator was used in the complete machining cell.

Similar to the prototype version, the revised micro manipulator was equipped with integrated strain-gauge sensors in the piezo actuators, and capacitive sensors to measure the 3D position of the end-effector plate.

Optical Tracking System In order to acquire workspace position and orientation measurements, a Nikon Metrology K600 optical tracking system [Nikon Metrology, 2010] was chosen because of its high-speed data streaming, large measurement volume, and the possibility to track both position and orientation of multiple Cartesian coordinate systems simultaneously. As the optical measurements were subject to disturbances

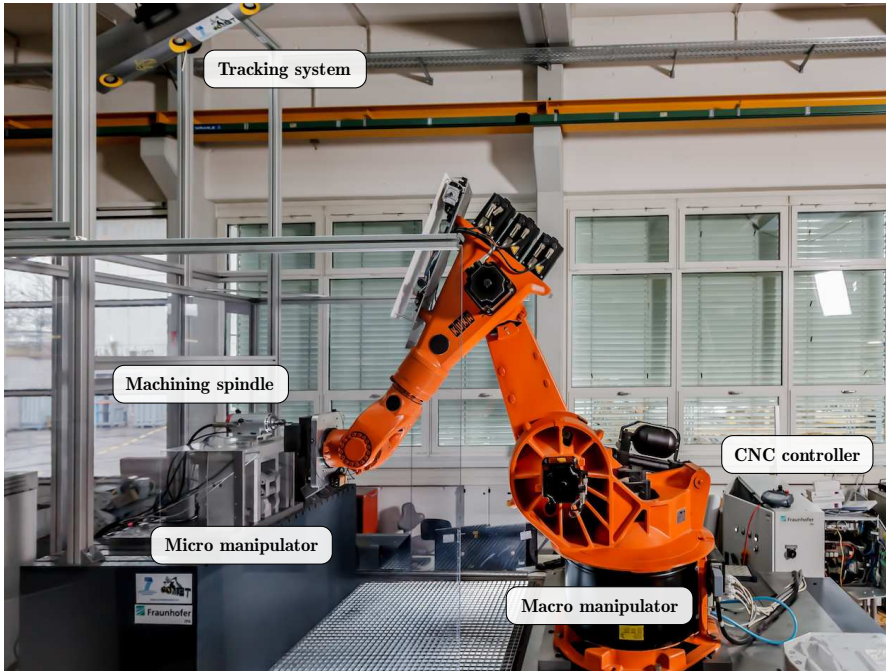


Figure 2.6 The developed complete robot cell for high-accuracy robotic machining, including a macro/micro manipulator and an optical tracking system.

from the milling process, such as emitted metal chips and dust, the coordinate systems were tracked using a redundant number of light-emitting diode (LED) units, which were attached to the rigid bodies to be tracked. For the considered machining application, one coordinate system on the spindle was measured with three LED units and one coordinate system on the macro manipulator end effector was measured with four LED units. The measurement frequency of the tracking system with this configuration was 440 Hz and the absolute accuracy for each LED unit measured was approximately 50–75 μm . The sensor data were retrieved using a transmission control protocol data stream, connected to the CNC controller.

Machining Spindle The milling tool was attached to a Jäger Chopper 3300 high-frequency machining spindle [Alfred Jäger GmbH, 2013], which had a concentricity of less than 2 μm . The spindle speed was set to 28 000 rpm, which was justified by the fact that high rotation velocities reduce process forces and the risk of exciting eigenmodes of the robot during the machining task.

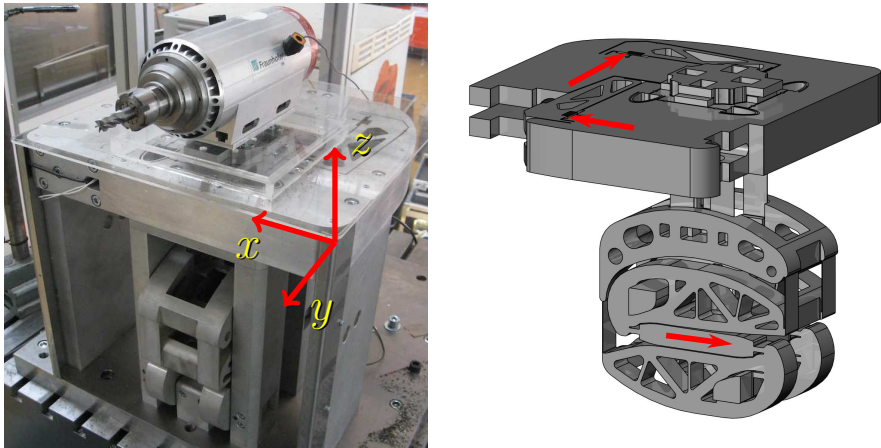


Figure 2.7 Revised design of the micro manipulator with parallel actuation. The Cartesian actuation coordinate system is indicated in red. A machining spindle, with the tool and attached LED units for optical tracking, is mounted on the micro-manipulator end effector. The actuation principle of the micro manipulator is illustrated to the right, where the red arrows represent the location of the piezo actuators.

Communication and Controller Execution Communication and timing are critical for the proposed approach with a macro manipulator and additional external sensors and actuators, since delays between control units reduce the system performance significantly and might cause instability. Figure 2.8 shows the relevant control units with digital to analog converters (DAC) and analog to digital converters (ADC). For each control unit, the execution frequency is specified. The micro manipulator controller was executed on a dSPACE signal processing system of model DS1103 [dSPACE GmbH, 2007]. The closed-loop control of the robot requires a high-speed interface for specifying joint-position references in real-time, which the used CNC controller provides.

2.3 Modeling and Control Design

In this section, the revised micro manipulator system is modeled and a subsequent model-based control structure is proposed. Since identical control structures were used for both micro manipulators, specific details for the prototype micro manipulator are omitted. Additionally, exclusively for the complete machining cell, the micro manipulator controller is integrated with the macro manipulator controller using a mid-ranging control strategy.

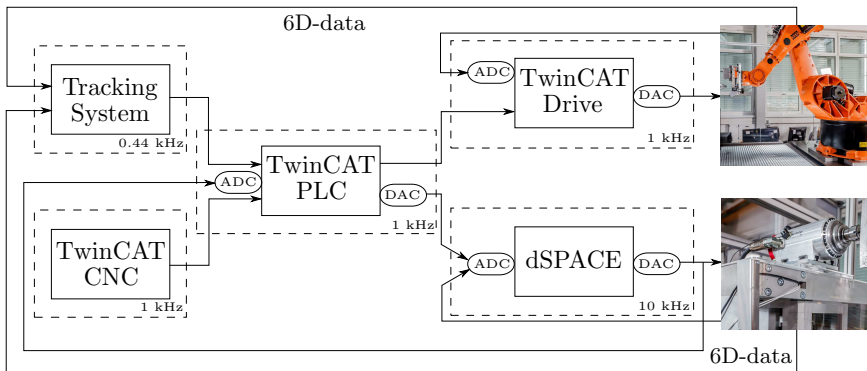


Figure 2.8 Hardware control units in the complete robot cell for machining, and their execution frequencies.

Dynamic Characterization of the Micro Manipulator

Because of the inherent resonant character of the mechanical design—which is a result of the utilized solid-state flexure elements—and the nonlinear dynamics which appear in piezo actuators, accurate position control of the micro manipulator without oscillations under milling-process disturbances is a challenging control problem.

Nonlinear Dynamics in Piezo Actuators It is well-known that piezo actuators exhibit a nonlinear relationship between the applied input voltage and the corresponding position extension [Al Janaideh et al., 2009; Sun and Yang, 2009]. Experiments were performed on the micro manipulator in order to quantify the effect of the nonlinear dynamics in the piezo actuators, on the position accuracy. The experiments indicated that the main nonlinearities that needed to be handled were hysteresis and the creep phenomenon, where the latter means that the extension of the piezo actuator increases over time for a constant input voltage. Results from experiments where the input voltages to the piezo actuators were alternately linearly increasing and decreasing are shown in Figure 2.9. It is clear that the hysteresis needs to be handled actively for accurate positioning [Olofsson et al., 2012]. Further, experiments showed that the nonlinear creep phenomenon in the actuators is a much slower process, and thus easier to handle. The creep effect was quantified to an approximate rate of $0.02 \mu\text{m/s}$ for the respective piezo actuator. Although different in nature, both of these nonlinear effects can be reduced by using high-gain feedback, combined with integral feedback.

Frequency Characterization of the Mechanical Design In order to characterize the frequency properties of the mechanical design of the

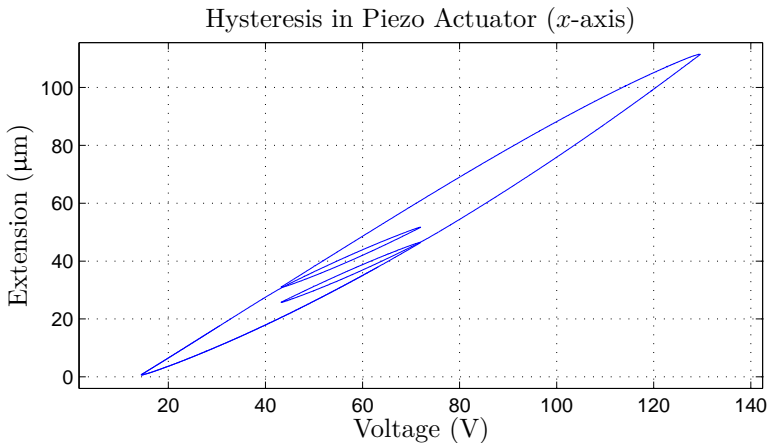


Figure 2.9 Characterization of hysteresis in the piezo-actuator dynamics along the x -axis of the micro manipulator in the case of a linear, alternately increasing and decreasing, input with changing amplitude. Note the complex behavior of the hysteresis, which exhibits both rate and amplitude dependency.

micro manipulator, several frequency-response experiments were performed. The power spectral densities of the output positions when exciting the inputs of the piezo actuators with a chirp signal, for the different actuation directions, are displayed in Figure 2.10. The spectra were estimated using Welch's method [Ljung, 1987]. An important property of the system from a control point-of-view is the location of the first natural eigenfrequency. It is noted that the characteristics are different in the three Cartesian actuation directions. In particular, in the interesting frequency range of 0–150 Hz, three major natural eigenfrequencies are visible for the dynamics along the x -axis, whereas two along the y -axis, and only one along the z -axis. The first eigenfrequency appears in the frequency range 50–80 Hz for all of the three axes. The locations of the eigenfrequencies are important since they limit the achievable bandwidth in the final closed-loop control system. Increasing the bandwidth beyond poorly damped eigenfrequencies requires high control actuation and the sensitivity to model errors is increased significantly.

Modeling of the Mechanical Design

In order to design position-control algorithms, it is advantageous to perform modeling of the micro manipulator prior to the design. Motivated by a control-oriented aim of the micro manipulator modeling, the approach chosen in this chapter is to consider black-box input-output models, without modeling the internal mechanical relationships. Moreover, an analytical

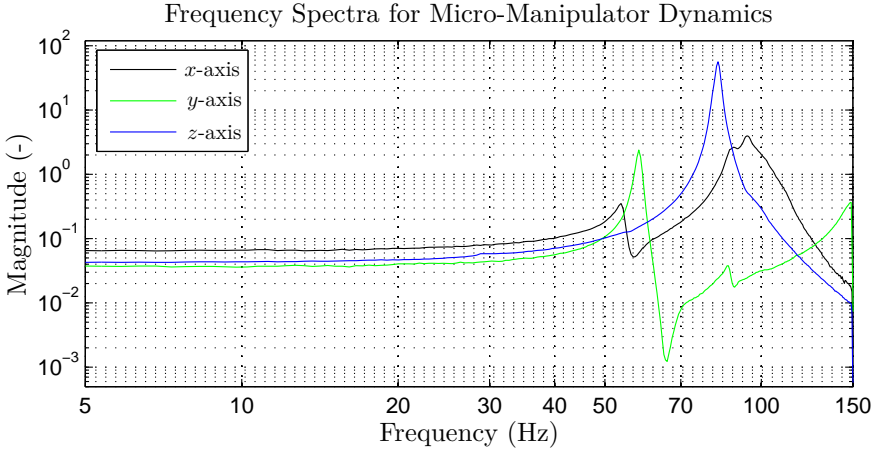


Figure 2.10 Estimated power spectral densities of the output signal in the Cartesian actuation directions of the micro manipulator when exciting with a chirp signal.

modeling approach is intractable for the current setup, since there is no straightforward mathematical description of the dynamics of the solid-state flexure elements used in the micro-manipulator design.

Identification Based on Black-Box Models Using system identification methods [Johansson, 1993], mathematical models describing the dynamics of the micro manipulator can be determined. Discrete-time state-space models of the innovation form were considered, stated as

$$\begin{cases} x_{k+1} = \Phi x_k + \Gamma u_k + K e_k \\ y_k = C x_k + e_k \end{cases}, \quad (2.1)$$

where $u_k \in \mathbb{R}^m$ is the input, $x_k \in \mathbb{R}^n$ is the state vector, $y_k \in \mathbb{R}^p$ is the output, K is the corresponding Kalman filter gain matrix, and e_k is a white-noise sequence. The system model matrices $\mathcal{S} : \{\Phi, \Gamma, C\}$ in the state-space representation are identified using one of the available implementations of subspace-based identification methods. The computer tools used were the System Identification Toolbox [Ljung, 2010] in MATLAB and the State-Space Model Identification (SMI) Toolbox [Haverkamp and Verhaegen, 1997]. In particular, the N4SID method [Overschee and De Moor, 1994] and the MOESP algorithm [Verhaegen and Dewilde, 1992] were utilized. During the identification of the models, the gain matrix K in a Kalman filter [Kalman, 1960] for a minimum-variance estimation of the states in the model was also determined based on the noise properties of the identification data.

The subspace-based identification methods were found to result in models with superior fit to experimental data. In particular, the natural eigenfrequencies of the micro manipulator were identified with significantly higher accuracy with subspace methods, compared to identification of time-series models [Johansson, 1993]. For further details on the system identification procedure, see [Olofsson et al., 2011a; Sörnmo et al., 2012b].

Position Control of the Micro Manipulator

The position-control problem of the micro manipulator is divided into two parts. First, the effects of the nonlinear dynamics in the piezo actuators need to be reduced. Second, the oscillations that arise as a result of the mechanical design need to be damped to achieve accurate position control. The control structure that was chosen to handle these challenges is described next.

Inner Piezo-Actuator Control Loops As the extensions of the piezo actuators in the micro manipulator are available for measurement with the strain-gauge sensors, an inner feedback loop can be closed around the nonlinear actuator and thus achieve sufficient control accuracy of the micro manipulator. The utilized controller is a PID controller, whose most basic continuous-time control law is stated as

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t), \quad (2.2)$$

where $e(t)$ is the control error, *i.e.*, the difference between the reference value and the measured extension, and K_p , K_i , and K_d are controller parameters to be determined as part of the design procedure. The derivative part in the controller is low-pass filtered, in order to reduce the amplification of high-frequency noise contaminating the measured signal from the strain-gauge sensors. Considering the intended application scenario for the micro manipulator—*i.e.*, milling and other machining tasks—disturbances from the cutting process are to be expected. The PID controller is also accompanied by an anti-windup scheme [Åström and Hägglund, 2005], to handle the case when the controller saturates the actuators. Discretization of the continuous-time controller (2.2) for subsequent implementation in a digital signal processor is straightforward [Åström and Wittenmark, 1997].

In order to reduce the effects of the nonlinear dynamics in the piezo actuators, the proportional gain K_p and the integral gain K_i should be increased as much as possible, while not resulting in a too high sensitivity to disturbances that occur during the milling. Further, the derivative part is important since it contributes with lead compensation in the system dominated by hysteresis. An alternative method for explicit model-based control combining feedforward and feedback was considered in [Olofsson et

al., 2012], based on the generalized Prandtl-Ishlinskii model [Al Janaideh et al., 2009].

Model-Based Feedback Control of the Micro Manipulator By utilizing the identified state-space models of the linear dynamics, a state-feedback control loop can be designed for each of the three Cartesian actuation directions of the micro manipulator. However, new models need to be identified after closing the inner feedback control loop for the piezo actuators, where the reference signal to the inner PID control loop is considered as the input signal instead. With this approach, the effect of the nonlinear dynamics of the system is reduced and the nonlinear components are thus not degrading the performance of the identification of the linear systems.

State feedback is an appropriate control structure for this kind of system, since damping of the resonant modes in the micro manipulator can be introduced by suitable control design. The control law for state-feedback control of the system (2.1) can be written as

$$u_k = L(x_k^r - x_k) + u_k^f, \quad (2.3)$$

where k represents a time index, the feedback gain matrix $L \in \mathbb{R}^{m \times n}$ is to be chosen, $x_k^r \in \mathbb{R}^n$ is the vector with reference values for the states, and $u_k^f \in \mathbb{R}^m$ is the feedforward control signal. The design procedure is to determine the gain matrix L by linear-quadratic (LQ) optimal control [Åström and Wittenmark, 1997], which provides a suitable parametrization of the trade-off between attenuation of the resonances in the system and the utilization of control signal.

Since all states in the state-space models of the micro manipulator are not available for direct measurement, a Kalman filter [Kalman, 1960] is introduced for estimation of the states, based on the measured position signal, the system input, and the identified model. Consequently, the proposed control law is a linear-quadratic Gaussian (LQG) controller. The Kalman filter is organized as

$$\begin{cases} \hat{x}_{k+1} &= \Phi \hat{x}_k + \Gamma u_k + K(y_k - C \hat{x}_k) \\ \hat{y}_k &= C \hat{x}_k \end{cases}, \quad (2.4)$$

where the estimated states \hat{x}_k and the estimated output \hat{y}_k have been introduced [Åström and Wittenmark, 1997]. As previously mentioned, the Kalman filter gain matrix K for minimum-variance estimation of the states in the model is obtained from the identification procedure, since the noise model component is also included in the innovation model structure (2.1). In particular, the subspace-based identification methods employed provide the Kalman filter gain matrix for each of the models.

In order to eliminate stationary errors in the position-control loop, integral action is also introduced in the state-feedback controller. The integral

action is achieved by extending the state vector with the integral state x_k^i according to

$$x_{k+1}^i = x_k^i + h(r_k - y_k), \quad x_0^i = 0, \quad (2.5)$$

where the position reference signal r_k and the sample time h have been introduced. With this extra state, it is required that the state-feedback gain vector L is augmented with one element and subsequently redesigned to accommodate the introduced integral state. Various approaches can be chosen to handle the feedforward control signal. In the control architecture presented in this chapter, the feedforward control u_k^f is chosen as a direct term from the reference signal, *i.e.*, $u_k^f = l_r r_k$. The parameter l_r influences the response of the controller to changes in the reference signal.

To summarize the design of the position controller for the micro manipulator, the proposed structure is displayed in Figure 2.11.

Control Architecture

To the purpose of establishing a macro/micro-manipulator configuration for milling with industrial robots, a controller for the micro manipulator tool position was developed in the previous section. In this section, the macro-manipulator controller is described, and both of the manipulator controllers are integrated using a mid-ranging control strategy.

Macro-Manipulator Controller The geometric path to be tracked is planned offline using appropriate computer-aided manufacturing (CAM) software. The position and orientation deviations occurring during the milling are computed based on the measurements from the optical tracking system. The orientation deviations are compensated by the macro manipulator solely, whereas the position deviations are fed to a mid-ranging control structure, described later in this section, which separates the error in the frequency domain for subsequent compensation by the macro and micro manipulators jointly. As the position deviations during the milling are measured in Cartesian space, the macro manipulator corrections calculated by the mid-ranging control structure need to be transformed to joint space. To this purpose, the Jacobian matrix, expressing the differential kinematics for the end-effector coordinate system of the robot, is used. Assuming small position and orientation corrections, the Cartesian deviation $\delta x \in \mathbb{R}^6$ is transformed to joint space using the inverse of the Jacobian matrix $J(q) \in \mathbb{R}^{6 \times 6}$, which is dependent on the joint-angle configuration $q \in \mathbb{R}^6$, according to

$$\delta q = J^{-1}(q)\delta x \quad (2.6)$$

where δq is the corresponding joint-angle correction. Given the expected range of the Cartesian position corrections, which are smaller than 1 mm,

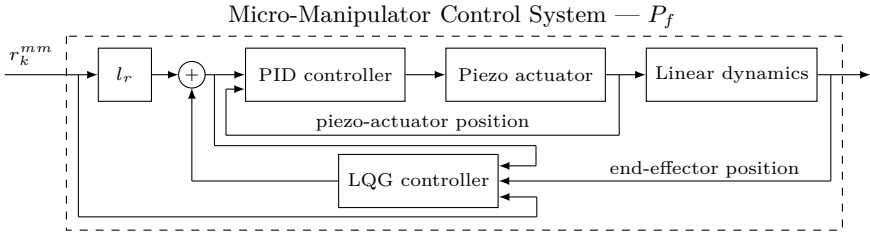


Figure 2.11 Control architecture for the micro manipulator. The figure illustrates the control strategy for one axis, but the control structures are identical (except parameters) for all three Cartesian actuation axes.

the approximation using the Jacobian matrix stated in (2.6) can be justified [Schneider et al., 2013b].

Micro-Manipulator Controller The micro manipulator is controlled and actuated in Cartesian space, using the strategy presented previously in this section for each axis separately. The complete model-based solution is illustrated for one axis in Figure 2.11, with the reference value r_k^{mm} as input and the corresponding end-effector position as output. The reference position r_k^{mm} for each actuation axis is determined by a mid-ranging control architecture, described in the next paragraph.

Mid-Ranging Control As mentioned earlier, the micro manipulator has a limited workspace of approximately 0.5 mm along each Cartesian axis, and thus the manipulator may reach its actuation limits when performing advanced milling tasks, where strong process forces are required. Therefore, a mid-ranging approach is considered for the complete machining cell, in order to control the relative position between the macro and micro manipulator. In the implementation, each Cartesian axis is considered and controlled separately. The mid-ranging control strategy makes use of the higher bandwidth of the micro manipulator, while striving to keep the position of the micro manipulator close to the midpoint of its workspace. The purpose is that the actuation limits of the micro manipulator should not be reached [Allison and Isaksson, 1998]. The control approach can be seen as a separation of the position error, computed from the tracking sensor data, in the frequency domain, which lets the fast micro manipulator handle the high-frequency deviations and the comparably slow macro manipulator handle the low-frequency deviations. Several mid-ranging control architectures have previously been proposed, such as the valve position control (VPC), modified valve position control (MVPC) [Allison and Isaksson, 1998; Allison and Ogawa, 2003], and internal model control (IMC) [Gayadeen and Heath, 2009]. In this chapter, the MVPC control structure is

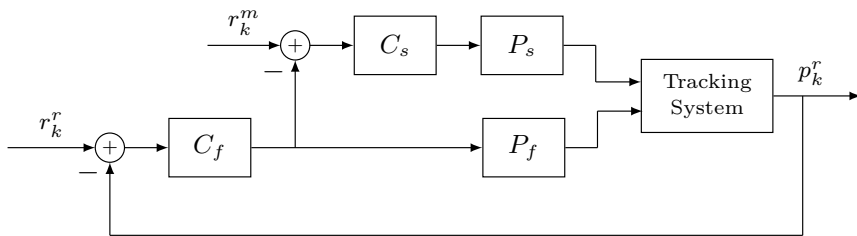


Figure 2.12 The complete closed-loop system, for each Cartesian actuation axis, of the proposed complete milling cell, using the mid-ranging MVPC control architecture.

considered because of its widespread presence in applications in literature and its appealing properties in terms of low complexity and robustness to model uncertainty. A further investigation on mid-ranging control of macro/micro-manipulator setups, based on adaptive IMC, is presented in Chapter 3.

The complete closed-loop control system for the integrated macro/micro-manipulator control, using the MVPC control architecture, is displayed in Figure 2.12, where C_f and C_s are controllers, here chosen as PI controllers. The systems P_f (displayed in Figure 2.11) and P_s represent the closed-loop Cartesian position control systems of the micro and macro manipulator, respectively. The relative position between the manipulators along each Cartesian axis, which was measured by the optical tracking system, is denoted p_k^r , the relative position reference is denoted r_k^r , and the desired setpoint for the mid-ranged input is denoted r_k^m . The latter is for the current setup chosen such that the micro manipulator should operate in the middle of its working range, for each actuation axis. The tuning guidelines that are provided in [Allison and Ogawa, 2003] are based on *lambda*-tuning for first-order models of P_f and P_s . Here, however, the models of the micro-manipulator dynamics in the three Cartesian directions ranges from order 2–7, see Section 2.4, and the dynamics contain several poorly damped resonances. Even though a significant damping of the resonances in the micro manipulator can be achieved with the LQG controller, the closed-loop system naturally cannot be described using first-order systems. Hence, using the tuning guidelines proposed in the mentioned references will not result in satisfactory performance. Therefore, the parameters obtained from the proposed design procedure in [Allison and Ogawa, 2003] were used as a starting point, and were subsequently experimentally tuned in order to obtain the desired performance.

2.4 Micro-Manipulator Identification Results

Discrete-time state-space models of the form (2.1) for the linear dynamics along the x -, y -, and z -axis of the micro manipulator were estimated based on experimental data. First, models of the open-loop system were determined, and subsequently models with the inner piezo-actuator control loops closed were identified.

SIMO-Model Identification To the purpose of quantifying the cross couplings between the actuation axes of the micro manipulator, identification of single-input multiple-output (SIMO) models with the structure in (2.1) was performed. Experimental data were collected when exciting the piezo actuators with chirp signals and measuring the corresponding Cartesian end-effector positions. The model input u_k was the the input voltage to the actuator, scaled to a nominal interval, and the three outputs y_k were the end-effector positions. The magnitude plot of the Bode diagram for the identified model of order nine, obtained from the data when actuating along the x -axis, is provided in Figure 2.13. From the model, minor cross coupling between the x - and z -axes can be concluded, in particular when exciting close to the natural eigenfrequencies. This might lead to vibrations transversal to the actuation direction during machining. The corresponding Bode diagrams for the SIMO-models obtained with actuation along the y - and z -axes verify the coupling between the z - and x -axes, whereas they indicate that the y -axis is decoupled from both the x - and z -axes.

Model Identification for Control Design In the modeling phase, the axes can be assumed to be decoupled provided that the mechanical design is made such that the motions of the different axes are sufficiently independent. During the dynamic characterization of the micro manipulator and the SIMO-model identification in the previous paragraph, minor cross couplings between the x - and z -axes were observed. However, since the influence of the coupling was minor, the assumption on decoupling was made in the subsequent model-based control design. Instead, the cross coupling was handled as a disturbance in the controller. Consequently, for the control design, single-input single-output (SISO) models were considered for each direction. The collection of experimental input-output data was performed such that the input u_k was the input voltage to the piezo actuator, scaled to the same nominal interval for all axes, whereas the output y_k was the Cartesian position of the micro-manipulator end effector, as measured by the capacitive sensor. For excitation of the system, chirp signals were chosen as inputs. The model orders vary in the different actuation directions, reflecting the number of natural eigenfrequencies, *cf.* the power spectral densities in Figure 2.10. The determined model orders for the axes of the micro manipulator are 7, 6, and 2, for the x -, y -, and z -axis, respectively.

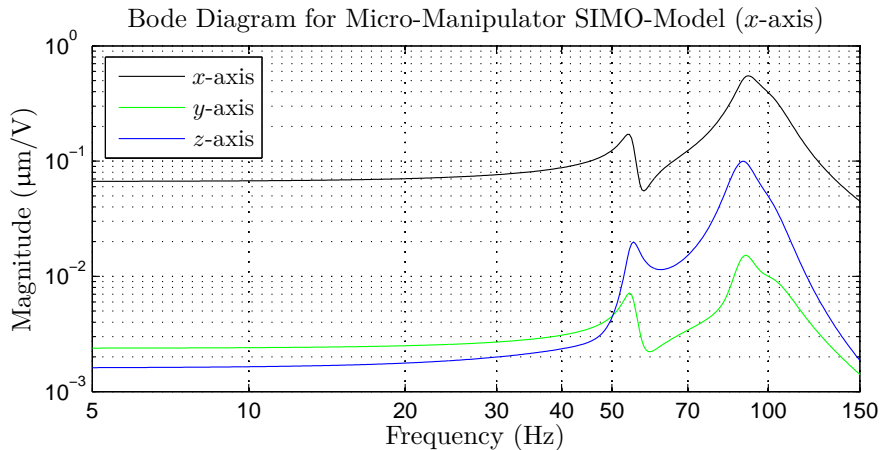


Figure 2.13 Bode magnitude plot of the discrete-time state-space SIMO-model of the micro manipulator, with the scaled voltage to the *x*-axis piezo actuator as input and the three Cartesian end-effector positions as output.

The model-order selection was based on the singular values analysis of the Hankel matrices computed as part of the identification procedure. The Bode diagrams of the identified models are shown in Figure 2.14. It is noted that there is good correspondence with the estimated power spectral densities of the output in Figure 2.10, when comparing the location of the natural eigenfrequencies. The normalized root mean square errors (NRMSE), $\tau_{\text{NRMSE}} \in [-\infty, 100]\%$, is a measure of the fit of the models to the experimentally collected identification data, where 100% indicates complete model fit. Given N data points, the NRMSE is given by

$$\tau_{\text{NRMSE}} = 100 \times \left(1 - \frac{\|W_N - \widehat{W}_N\|_2}{\|W_N - \overline{W}_N\|_2} \right) \%, \quad (2.7)$$

where W_N is the validation output data, \widehat{W}_N is the output from the estimated model and \overline{W}_N is the mean of the validation output data. The NRMSE values for the identified models in the *x*-, *y*-, and *z*-directions were 92.8%, 88.7%, and 96.0%, respectively. These values indicate that the identified models capture the essential dynamics of the micro manipulator.

Furthermore, dynamic models were identified with the inner piezo-actuator control loops closed. Since the differences compared to the open-loop models presented in Figure 2.14 are visually negligible, the models with the closed inner loop are not presented here.

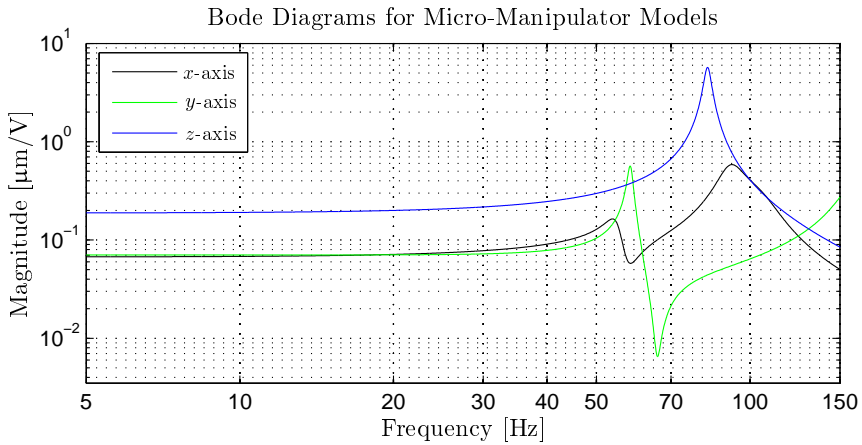


Figure 2.14 Bode magnitude plot of the discrete-time state-space models identified using subspace identification for the dynamics along the x -, y -, and z -axes of the micro manipulator. The location of the natural eigenfrequencies can be compared to those in the power spectral densities in Figure 2.10. The differences in the low-frequency gain between the axes are explained by the different types of the respective piezo actuator and the gear ratios.

2.5 Results from Prototype Machining Cell

In this section, experiments performed using the prototype machining cell are presented. Results from milling experiments in aluminum, both for the uncompensated and compensated cases, are provided.

Milling Experiments with Compensation

With the prototype experimental setup described in the first part of Section 2.2, milling experiments in aluminum were performed. The macro manipulator can be reconfigured such that milling can be executed in all three directions of the micro manipulator. Results obtained during face milling in the x -direction and peripheral milling in the y - and z -directions of the micro manipulator are presented here. The experiments were performed with a feed rate of 7.5 mm/s, a spindle speed of 28 000 rpm and a 1 mm depth of cut in the face millings, and $1 \times 10 \text{ mm}^2$ in the peripheral millings.

X -direction In the first setting, face milling was performed, where the surface orthogonal to the x -axis of the micro manipulator was machined. Consequently, the micro manipulator was controlled in this direction. The result of the milling experiment is displayed in Figure 2.15. The control

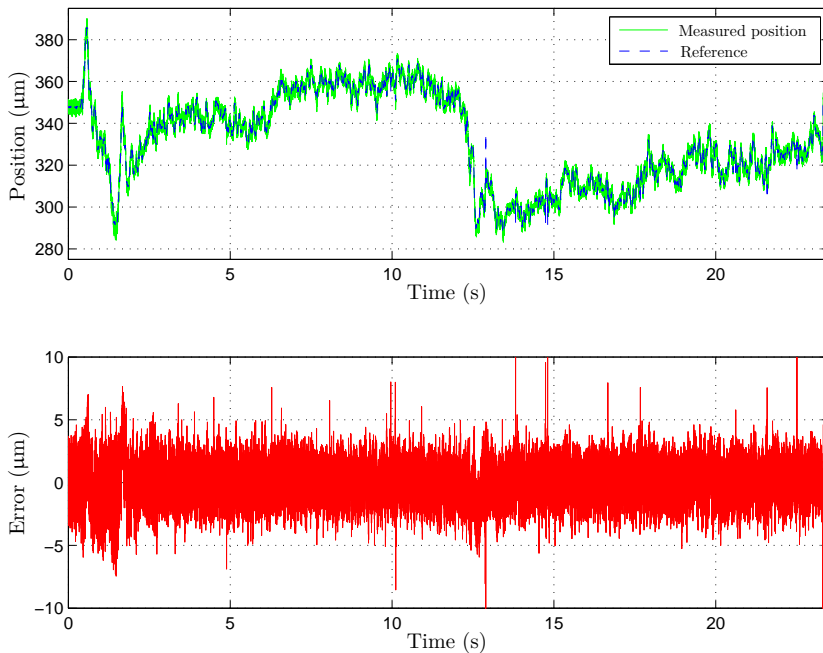


Figure 2.15 Reference and measured position of the micro manipulator during a face milling experiment in the x -direction (upper panel) and the corresponding control error (lower panel).

error is defined as the difference between the reference value to the micro manipulator control system and the measurement from the capacitive sensor in the x -direction of the micro manipulator.

Y-direction The milling accuracy was further tested in a peripheral milling scenario, where the compensation was performed along the y -axis of the micro manipulator. It should be noted that this milling task is different from the face milling presented in the previous paragraph, in the sense that the process forces affect the macro manipulator differently. Furthermore, the experiment was designed such that the macro manipulator was not moving perpendicularly to the compensation direction. This situation can be considered as a result of a poorly calibrated workpiece or industrial robot. By utilizing the micro manipulator, this effect can be compensated since the movement of the macro manipulator is tracked in real-time. The result of the milling experiment is displayed in Figure 2.16. The displayed control error is defined analogously to the case with face milling in the x -direction of the micro manipulator.

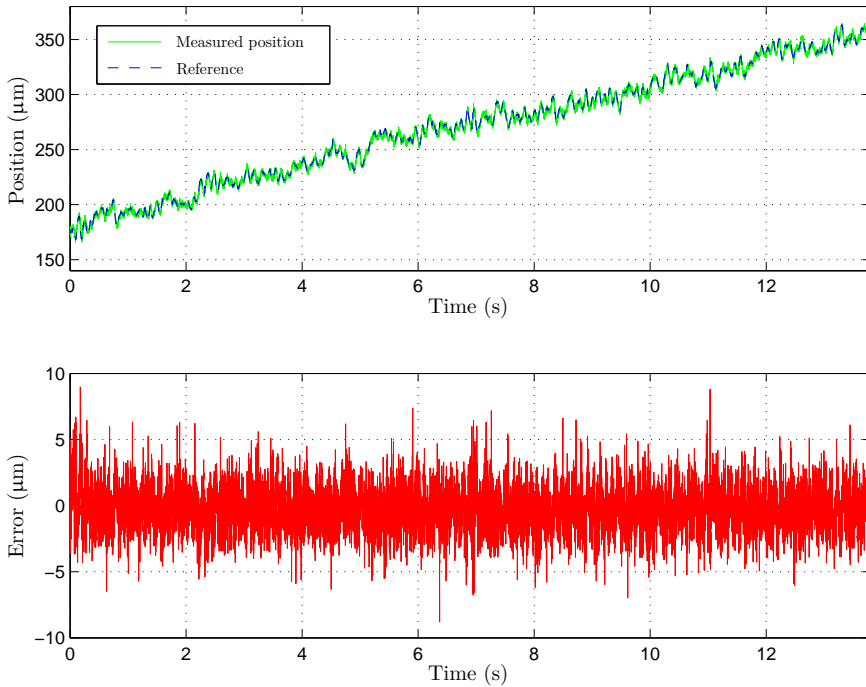


Figure 2.16 Reference and measured position of the micro manipulator during a peripheral milling experiment in the y -direction (upper panel) and the corresponding control error (lower panel).

Z -direction The third experiment was a peripheral milling along the z -axis of the micro manipulator. The control performance of the micro manipulator in the milling experiment is displayed in Figure 2.17.

Milling Experiments without Compensation

The same milling experiments described and presented in the previous subsection were repeated, but with the machining spindle rigidly attached—*i.e.*, no online compensation was active. The results of the experiments are evaluated in the subsequent sections.

Position-Control Results

A requirement for achieving accurately machined workpieces using the proposed robot cell, is to have precise position control of the micro manipulator. The control performance is evaluated from the millings performed in each of the Cartesian directions, displayed in Figures 2.15–2.17.

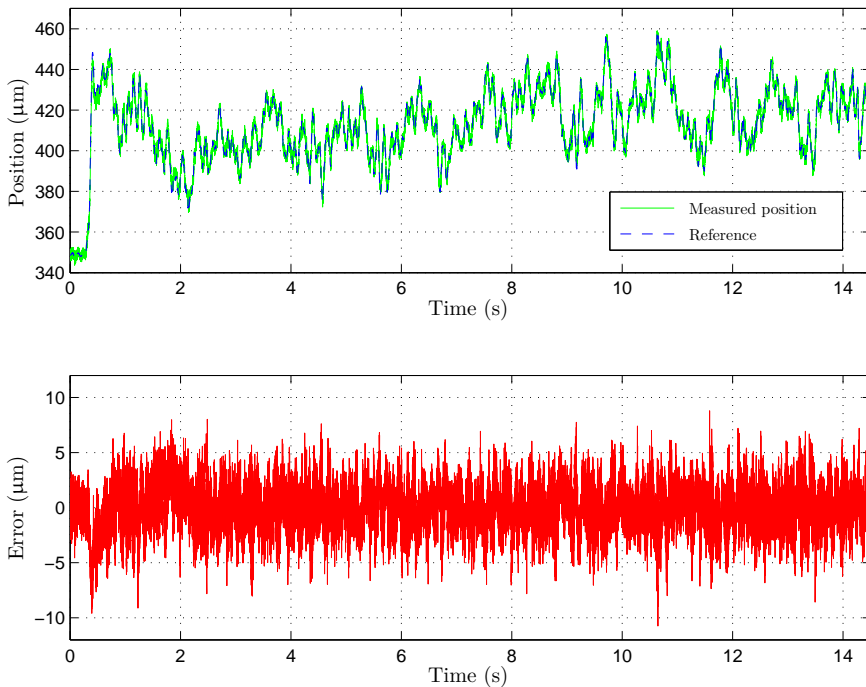


Figure 2.17 Reference and measured position of the micro manipulator during a peripheral milling experiment in the z -direction (upper panel) and the corresponding control error (lower panel).

From the lower panels of the plots it is noted that the control error is approximately within $\pm 5 \mu\text{m}$ in all directions, which is well within the desired accuracy of $50 \mu\text{m}$.

Absolute-Accuracy Results

The absolute accuracy of the milling experiment results, *i.e.*, how close to the desired absolute position the milling was performed, is considered here. As mentioned previously, the process forces that arise during the milling result in deflection of the robot position, which is measured by the optical tracking system. In order to compare the uncompensated to the compensated case, the MAE is considered. The MAE is defined as

$$\text{MAE} = \frac{1}{N} \sum_{k=1}^N |e_k|, \quad (2.8)$$

Table 2.1 MAE of tracking error, given in μm .

	x -axis	y -axis	z -axis
MAE uncomp.	23.4	138.4	67.0
MAE comp.	1.6	1.9	1.9
Ratio	14.6	72.8	35.3

where k is the sample index, N is the number of measurement points, and e_k is the tracking error, calculated as the difference between the nominal and measured position. The MAE values provide an indication of the obtained control-accuracy performance.

In Table 2.1, MAE values for all Cartesian axes are presented, for both uncompensated and compensated milling experiments. It is noted that a substantial increase in absolute accuracy was obtained for all axes. Because of the calibration along the y -axis mentioned previously, the performance increase in the y -direction is notably larger than for the other axes. This is to be expected, as in this case, the error during milling does not solely depend on deflections caused by process forces. The error during uncompensated milling will thus continuously increase when moving along the y -axis, as seen in Figure 2.16.

Surface-Accuracy Results

In addition to the absolute accuracy, it is of interest to achieve a high accuracy of the machined surface on the workpiece. In order to evaluate the surface accuracy, a Mahr measurement device of model M400 SD26 [Mahr GmbH, 2011] was utilized to measure the obtained workpiece profiles. The measurement device was calibrated such that it had a measurement accuracy of $1 \mu\text{m}$.

Milling with Compensation The results of the surface roughness measurements, for the three milling experiments with online compensation, are displayed in Figure 2.18. The measured profiles indicate that the surface accuracy in the x - and y -directions is within $\pm 7 \mu\text{m}$ and within approximately $\pm 12 \mu\text{m}$ in the z -direction of the micro manipulator. Furthermore, it is noted that the measured profiles correspond well to the measurements from the capacitive sensors attached to the micro manipulator which are used for feedback. This correspondence indicates that the measured position of the compensation mechanism agrees with the actual position of the milling tool. Photos of the milled surfaces for the experiments in the x -, y -, and z -directions are provided in Figures 2.20–2.22.

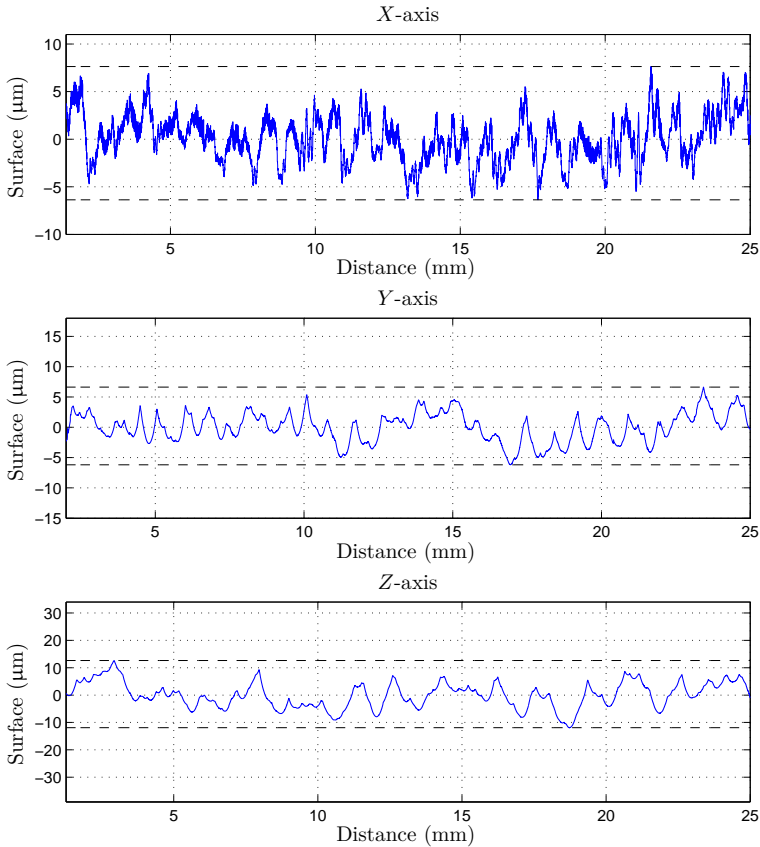


Figure 2.18 Workpiece profiles after face milling in x -direction and peripheral milling in y - and z -directions of the micro manipulator. In all experiments, online compensation with the micro manipulator was utilized.

Milling without Compensation The resulting surface accuracy of the profiles from the uncompensated milling experiments, as measured by the Mahr device, are displayed in Figure 2.19. To evaluate the quality of the measured profiles from the experiments with online compensation compared to the profiles obtained in milling without compensation, the standard deviation σ_e of the profiles are calculated and displayed in Table 2.2.

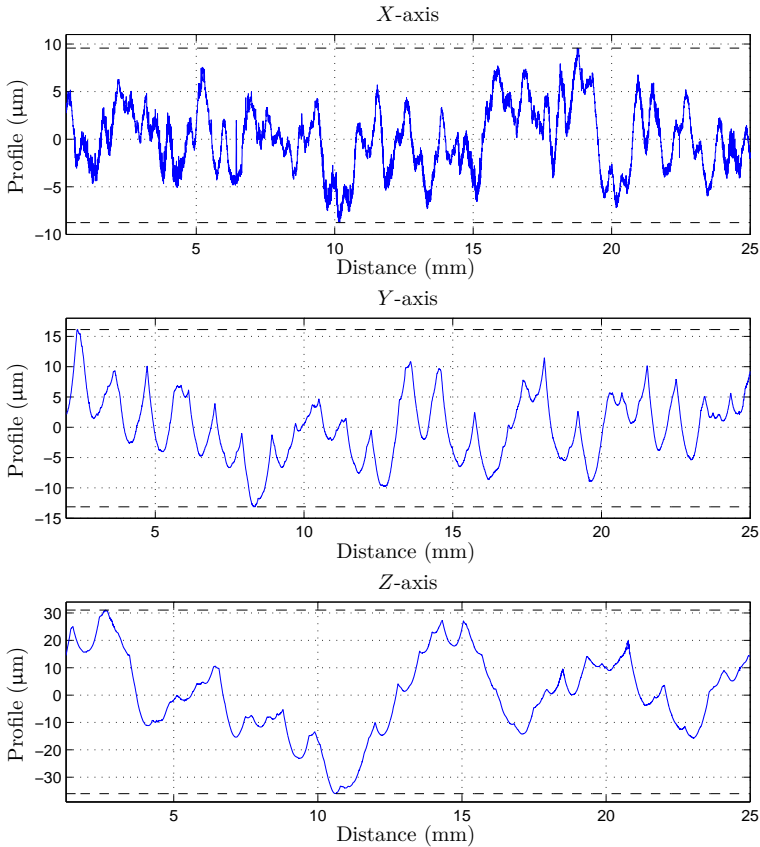


Figure 2.19 Workpiece profiles after uncompensated milling in the x -, y -, and z -directions of the micro manipulator, respectively.

2.6 Results from Complete Machining Cell

In this section, results from machining in steel with the use of the complete machining cell are presented. The process forces, and hence the position deviations of the macro manipulator, are increased approximately 3–5 times in steel milling compared to aluminum milling for the considered milling geometry and tool diameter. Also, the experiments were performed with the mid-ranging control architecture. Consequently, the performance of the complete system is here demonstrated in machining applications with three simultaneously actuated compensation axes in an industrially relevant scenario. A toroid milling geometry was chosen, since this geometry was

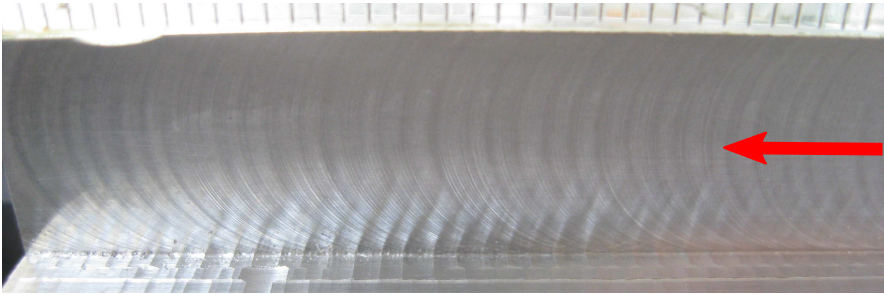


Figure 2.20 Workpiece after face milling on the surface indicated by the red arrow, with compensation in the x -direction of the micro manipulator.

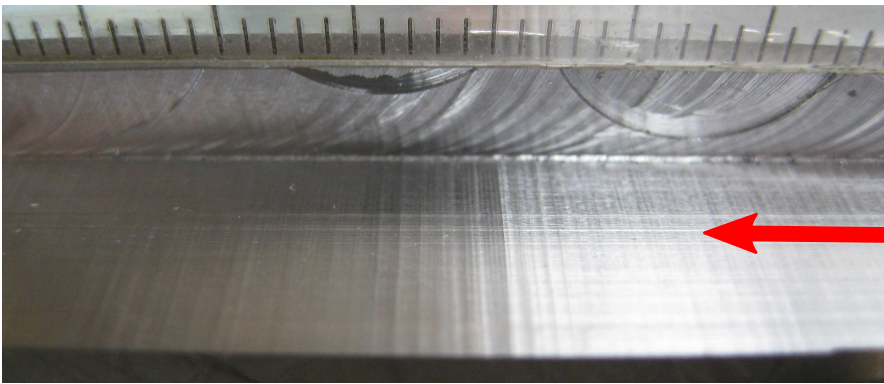


Figure 2.21 Workpiece after peripheral milling on the surface indicated by the red arrow, with compensation in the y -direction of the micro manipulator.

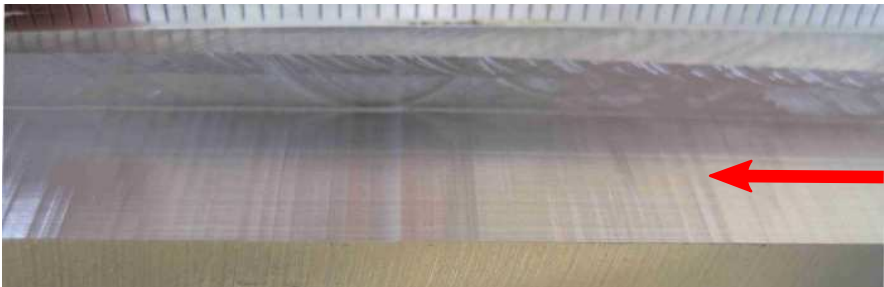


Figure 2.22 Workpiece after peripheral milling on the surface indicated by the red arrow, with compensation in the z -direction of the micro manipulator.

Table 2.2 Standard deviation of the obtained milling surfaces, given in μm .

	<i>x</i> -axis	<i>y</i> -axis	<i>z</i> -axis
σ_e uncomp.	7.6	5.6	14.9
σ_e comp.	2.8	2.5	4.7
Ratio	2.7	2.2	3.2

deemed appropriate for demonstrating and quantifying the accuracy of the proposed milling strategy, as it provides easily measurable information in two dimensions.

Experimental Validation

Using the complete robot cell described in the second part of Section 2.2, a toroid with a rectangular cross section and a mid-diameter of 70 mm was machined with a depth of 0.5 mm in full width cut, see Figure 2.23. A Horex solid carbide milling tool with four teeth and a diameter of 8 mm was used. Further, a feed rate of 5 mm/s and a spindle speed of 28 000 rpm were used. The magnitude of the process forces required for this milling was approximately 100 N. Because of the optical tracking of the spindle and the robot end effector, no lubrication or cooling, potentially obstructing the line-of-sight, were used. To the purpose of evaluation of the method and comparison to previous approaches, three different configurations of the setup were considered:

- A. Position-controlled macro manipulator with the CNC controller, and the spindle attached fixed to a rigid base.
- B. Position-controlled macro manipulator with the CNC controller, the spindle fixed to a rigid base, and online workspace position and orientation corrections of the macro manipulator based on feedback from measurements obtained with the optical tracking system.
- C. The complete setup proposed in the second part of Section 2.2, including the macro and micro manipulators, the optical tracking system, and the mid-ranging control architecture.

As for the prototype cell, two separate spindles were used during the course of the experimental validation; one was fixed to a rigid base and one was attached to the micro manipulator. In Figure 2.6, the spindle attached to the micro manipulator is shown.

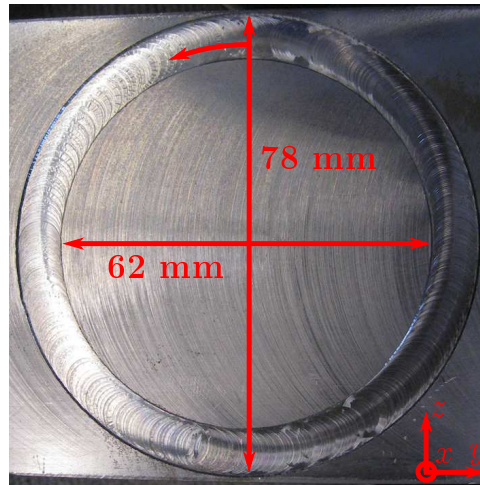


Figure 2.23 The milling geometry considered for evaluation of the proposed complete robotic machining cell. The milling path starts in the upper center point and continues in the counterclockwise direction.

Milling Results

The toroid milling experiment was performed for each of the Setups A–C. In this section, the results from Setup C are presented. The results from Setups A and B are further evaluated and compared later in this section. The results obtained from the milling experiment with Setup C are displayed in Figures 2.24 and 2.25. The tracking performance of the micro manipulator position control during a representative part of ten seconds of the toroid milling is displayed in Figure 2.24. The definition of the axes is according to Figure 2.7 and the error is defined as the difference between the reference value r_k^{mm} sent to the micro manipulator from the mid-ranging control architecture, see Figure 2.11, and the corresponding position of the micro-manipulator end effector. Further, the relative Cartesian errors, defined as the difference between the tool coordinate system and the workpiece coordinate system and computed from the measurements from the optical tracking system, are displayed in Figure 2.25 for the first 20 seconds of the milling task.

As a first observation, the initial transient when the tool enters the workpiece can be seen in Figure 2.25 at approximately $t = 1.2$ s. Moreover, it should be noted that the x -direction corresponds to the face direction of the milling tool, see Figure 2.23, and is therefore orthogonal to the machined surface. Hence, the required compensation is smaller than in the other two Cartesian directions. Because of the alignment of the tool axis and the

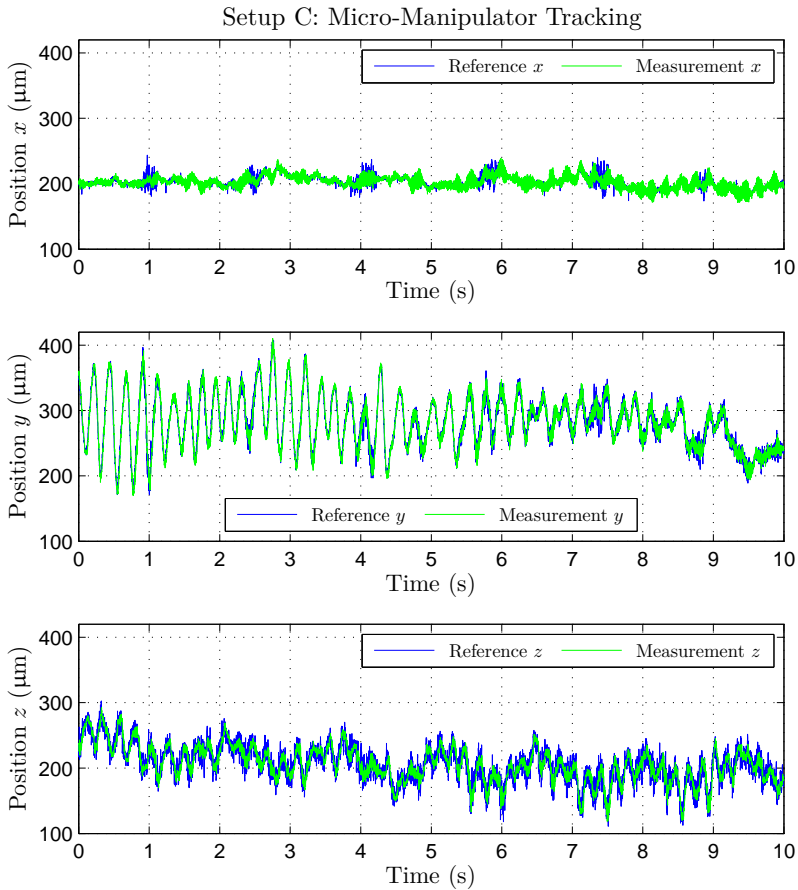


Figure 2.24 Micro manipulator tracking performance during machining with compensation using Setup C, along each Cartesian axis x , y , and z .

compensation axis, the x -axis is sensitive to milling process disturbances and as a result, low-amplitude oscillations were present.

The major part of the compensation with the micro manipulator was performed along the y - and z -axes. The reference in the y -direction exhibits a low-frequency oscillation of a comparably high amplitude, which was induced by a deficiency in the mechanics of the robot. It is also to be noted that the effects of the robot mechanics exhibit configuration dependency. Nevertheless, the micro manipulator was effective in compensating for the oscillation, which would be impossible to eliminate with the macro manipulator solely. Considering the alignment of joint two, three, and five of

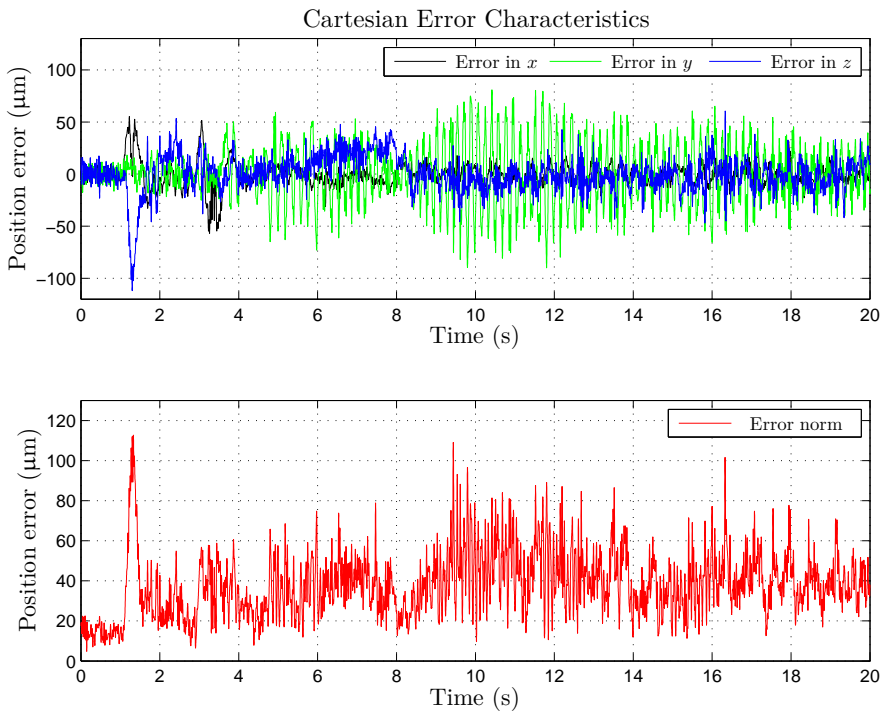


Figure 2.25 Relative position error between tool and workpiece during machining with macro and micro manipulator (Setup C), computed from the measurements by the optical tracking system. Note that the part of the milling showed in this plot is not the same as the one in Figure 2.24.

the robot, see Figure 2.6 for the robot configuration in the milling process, the robot exhibits lower stiffness along the z -axis compared to the x - and y -axes. However, the micro manipulator shows satisfactory tracking behavior, with only minor vibrations of the micro manipulator.

Error Analysis The experimental results obtained with Setup C were evaluated using statistical methods. The amplitudes of the error characteristics as well as the corresponding frequency spectra were investigated. First, the performance of the micro-manipulator tracking was considered. Referring to Figure 2.24, it can be observed that the micro manipulator exhibits good tracking behavior along all three Cartesian axes. Second, the power spectral densities of the Cartesian position errors in Figure 2.25 were investigated. The frequency spectra are displayed in Figure 2.26. Considering that the mechanical design of the micro manipulator comprises solid-state flexure elements, the damping of the oscillatory modes

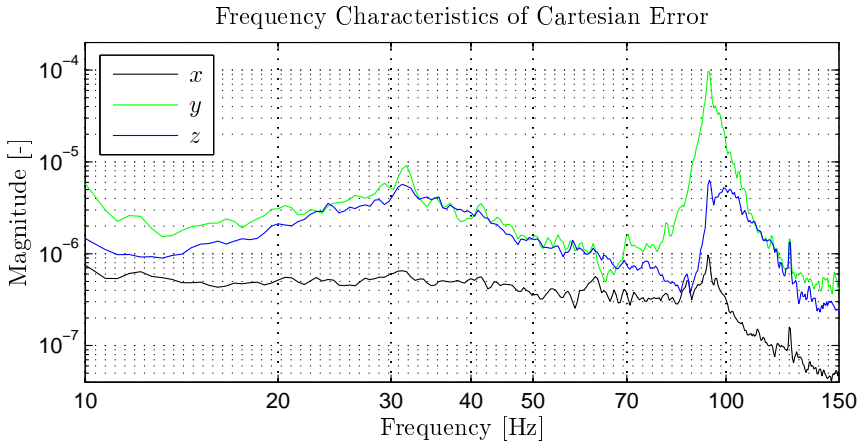


Figure 2.26 Power spectral densities of the Cartesian position errors—displayed in Figure 2.25—obtained during the machining of the toroid with Setup C.

of the system is challenging. As a result, the micro manipulator is sensitive to process disturbances with frequencies near the eigenfrequencies along the respective actuation axis. The first natural eigenfrequency of the micro manipulator, along each actuation direction, falls in the range from 50–100 Hz, see Section 2.3. It should be noted in the power spectral density plots in Figure 2.26, that a significant damping of the resonances was obtained with the proposed model-based state-feedback controller. The major resonances of the micro manipulator were clearly damped by the control design, even though a significant peak at approximately 95 Hz is visible in the error spectrum. The sensitivity of the micro manipulator around the eigenfrequencies led to low-amplitude oscillations of the machining tool with an amplitude of 10–30 μm , as can be observed in the plot in Figure 2.24. In turn, these tool oscillations result in surface undulations on the machined surfaces. The control design is here limited by the mechanical design of the micro manipulator. Additionally, a periodicity with a frequency of approximately 30 Hz can be observed in the error spectrum. As this frequency is below 50 Hz and appears in all three Cartesian directions, it can be traced back to excitation of an eigenmode of the robot, which can be verified by modal analysis of the same. The observed peak in the frequency spectra is found within 10–30 Hz, which is the range of typical eigenfrequencies of industrial robots.

Workpiece Geometry Measurements In order to validate and quantify the milling results, the machined surfaces of the workpieces obtained using

Setups A–C were measured. The 2D contours of the machined toroids were analyzed using a Werth CMM of model Videocheck HA400, which provides measurements with an accuracy of $0.5\ \mu\text{m}$ [Werth GmbH, 2013]. By performing raster scanning, point clouds of the inner and the outer circular contours were captured and subsequently compared to the desired reference circles. A least-squares matching, assuming additive Gaussian noise on the error, was used to locate the measured CMM data acquired in a local coordinate system, and the reference circles relative to each other, in order to enable comparison.

The results of the surface measurements for Setups A–C are collected in Figure 2.27. The plots display the computed difference of the CMM-measured outer circle from the nominal reference circle. The circumference measure starts at the location where the milling starts and traverses the toroid in the milling direction, see Figure 2.23. In addition, the measured inner and outer circular contours of the surfaces measured by the CMM are visualized and compared to the nominal circles in Figure 2.28. As a first observation, it can be noted in Figure 2.28 that the position deviations with Setup A are not constant during the machining of the circle. This is a result of the local stiffness properties of the robot combined with the changing direction of the process forces while traversing the toroid. Furthermore, when comparing the error of the toroid machined with Setup C to the corresponding results obtained using Setup A or Setup B, the significance of the different hardware and control components of the setups can be identified. A feedback loop for the robot controller, based on external sensor data from online position and orientation tracking of the workpiece, is sufficient for compensation of static calibration errors and low-frequency errors. This can be concluded by comparing the top and middle panels of Figure 2.27, where it is noted that the major part of the error, *i.e.*, the low-frequency part, has been compensated. However, the high-frequency part of the error remained in the results from Setup B, which is obvious from the substantial number of peaks in the error plot. With the addition of a high-bandwidth micro manipulator, the high-frequency process disturbances were compensated, reducing the machining error significantly, as seen in the bottom panel of Figure 2.27.

Surface-Roughness Measurements The CMM measurements of the circular contours on the machined workpiece can only evaluate the achieved performance along the y - and z -axes of the milling geometry. The performance in the x -direction, however, is also directly reflected on the machined surface. Therefore, the Mahr measurement device was used in order to quantify the machining errors in the face direction of the tool. Figure 2.29 displays the surface roughness over a randomly chosen length of 25 mm of each workpiece. It is to be noted that deviations on the surface were not as significant compared to the circular contours, which can be

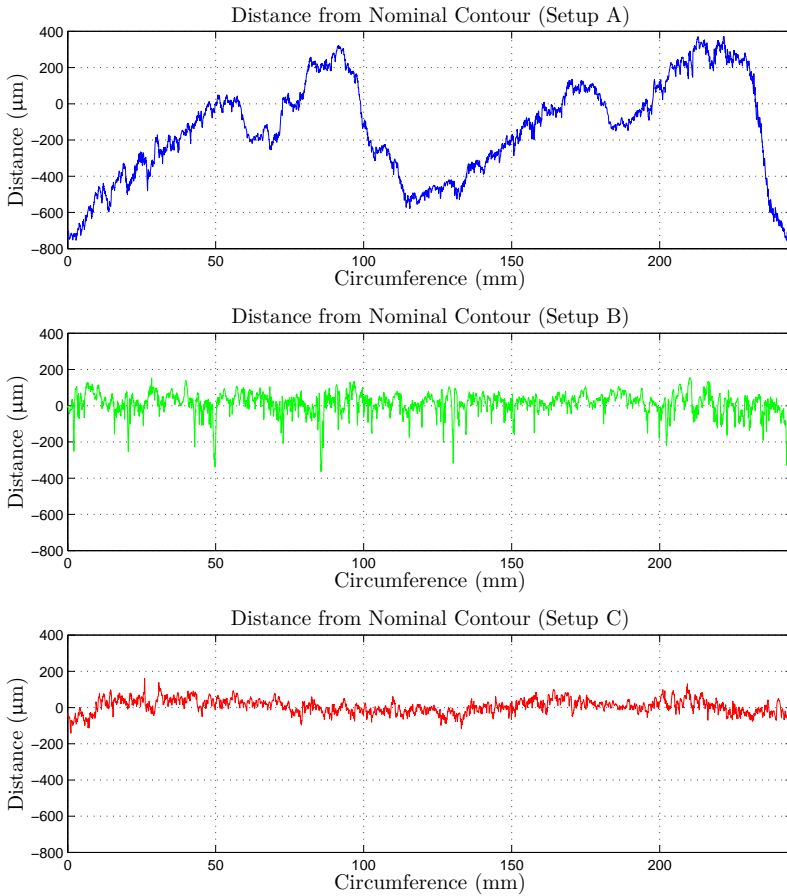


Figure 2.27 CMM measurements of the outer circular contour deviation, with the workpiece machined with Setups A–C. The circumference measure starts at the location where the milling starts, see Figure 2.23.

explained by the cutting process dynamics. Since the major part of the material was removed in the y - and z -directions, lower process forces were obtained in the x -direction, and hence reduced position deviations in this direction. An interesting result is that the surface obtained with Setup B is significantly worse than that of Setup A. It is plausible that this is related to the limited compensation bandwidth of the robot—*i.e.*, the high-frequency disturbances in position in the face direction are not possible to compensate with this approach because of phase lag and, to some extent, communication delays. As a consequence, the resulting machined surface becomes worse than without online compensation.

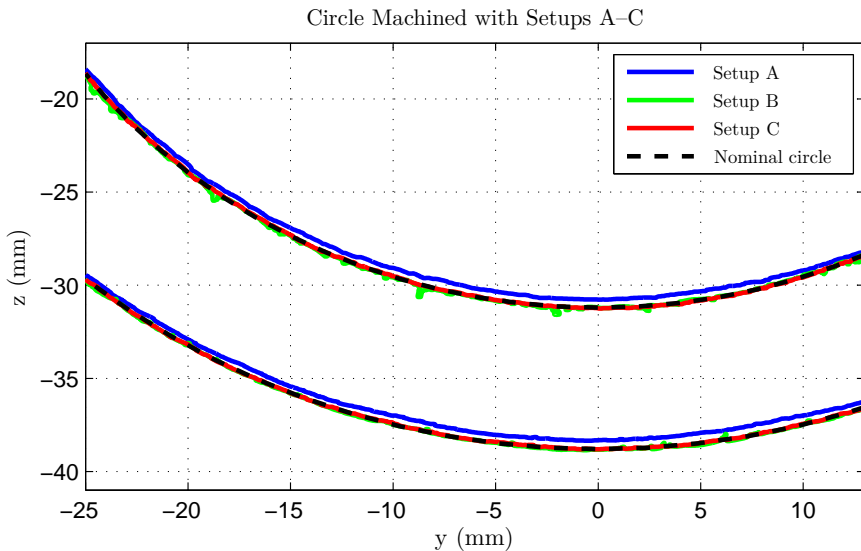


Figure 2.28 CMM measurement of the circular outer and inner contours, for the workpieces machined with the different Setups A–C. In this plot, the bottom part of the circle is displayed.

Table 2.3 Accuracy results of the machined toroids, displaying MAE from nominal profiles and the standard deviation of the surface measurements. The numbers were computed from machining results obtained with the respective Setup A–C.

Setup	MAE (Circle) [μm]	σ (Circle) [μm]	σ (Surface) [μm]
A	255	276	7
B	47	61	16
C	32	39	7

Summary of Experimental Results In order to quantify and compare the absolute accuracy of the considered Setups A–C, the MAE of the outer circle of the toroid machinings was computed for each. The surface accuracy was quantified for the circle as well as the surface in the x -direction, calculating the standard deviation σ of the obtained surface. The resulting numbers obtained using the Setups A–C are found in Table 2.3.

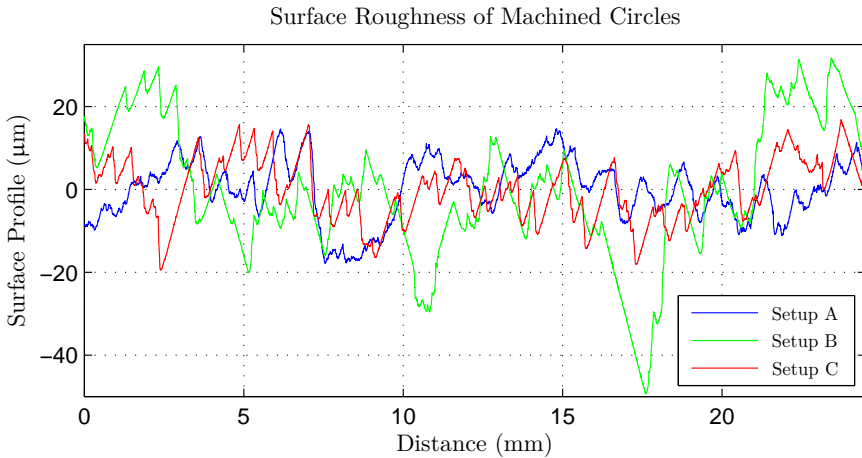


Figure 2.29 Roughness measurements of the machined surfaces for the respective configuration, Setups A–C.

2.7 Discussion

As an extension to joint-space position-controlled industrial robots for machining, workspace sensing combined with online position corrections with a macro/micro-manipulator setup was introduced in this chapter. High accuracy sensing in workspace was realized using an optical tracking system. A prerequisite for this strategy to work well in practice is a high actuation bandwidth of the micro manipulator and low latencies in the communication interfaces, in order to maintain stability during the machining process. Considering the open-loop bandwidth of the micro manipulator, which is approximately 3–4 times higher than the bandwidth of the closed-loop industrial robot at the end effector, it is clear that the capabilities are significantly increased for compensation of high-frequency position errors during machining, as well as other tasks which require contact between the robot and the tool/workpiece. In addition, as pointed out in [Sharon et al., 1993; Fasse and Hogan, 1995], workspace sensing as well as workspace actuation—*i.e.*, collocation of sensing and actuation—is beneficial in contact operations with mechanical manipulators in order to reach high-bandwidth control of the interaction between the tool and the workpiece.

The approach to machining proposed in this chapter is hardware intensive, in particular compared to previous approaches to increasing the position accuracy based on quasi-static and dynamic modeling of the robot stiffness. However, purely model-based approaches exhibit dependency on robot configuration, workpiece characteristics, and machining tool. Hence,

the generality in application of the approach proposed in this chapter is higher, since no assumptions regarding these aspects have to be made. Instead, position deviations are measured online in the workspace during the machining process and subsequently compensated. In addition, the need for an extensive and time-consuming calibration procedure for the force-deflection models, *i.e.*, stiffness models, is eliminated with the proposed approach. However, the investment cost is significantly higher for the proposed approach.

Investigating the costs for the proposed complete robot cell for high-accuracy machining, it can be noted that the high-accuracy optical tracking system and the micro manipulator are comparably expensive components. However, considering the rapid development and cost reduction for, in particular, optical tracking systems, the proposed solution is still competitive as compared to the cost of a machine tool. The cost distribution for the micro manipulator is between piezo actuators (45%), manufacturing of the mechanical parts (35%), sensor equipment (15%), and other auxiliary equipment (5%).

Workspace control of industrial manipulators based on feedback from data obtained with force sensors mounted on the robot end effector has previously been proposed for contact operations [Hogan, 1985; Vuong et al., 2009] in applications such as deburring [Jonsson et al., 2013] and drilling [Olsson et al., 2010]. Further, many major robot manufacturers offer such functionality for their industrial robots. The force-control performance is nevertheless limited by the mechanical bandwidth of the robot manipulator as well as the bandwidth of the internal joint-position controllers of the robot. Hence, the force control must be combined with appropriate mechanical actuation if high-frequency position deviations are to be compensated. In the approach presented in this chapter, only position information was used for the workspace feedback. However, integration of a force sensor in the setup would be beneficial, in particular for monitoring the process forces during the machining process but also for incorporation in the feedback control architecture as a complement to the position information.

For the prototype robot cell, the resulting position-control error during milling experiments was within approximately $\pm 5 \mu\text{m}$ for each direction, and the MAE was below $2 \mu\text{m}$ for each direction, which by far achieves the desired absolute accuracy of $50 \mu\text{m}$ for the complete milling task. Furthermore, the surface accuracy was increased by approximately a factor of three for all directions. From these results, it is evident that online compensation with the micro manipulator has significantly improved both the absolute and surface accuracy of the milling, compared to the uncompensated case.

It is further noted that the increase in absolute accuracy is higher in the y - and z -directions, than in the x -direction. This can be explained by the fact that face milling was performed in the x -direction, as opposed to peripheral

milling in the y - and z -directions, where the milling process forces affect the robot differently.

For the complete robot cell, significantly higher MAE values were obtained for the presented milling experiments, as compared to the prototype robot cell. Nevertheless, the objective of achieving an absolute accuracy of 50 μm was fulfilled using both Setup B and C. The decrease in accuracy compared to the prototype robot cell is not unexpected, for a number of reasons. First and foremost, the accuracies of the optical tracking systems differ significantly between the two cells, limiting the achievable machining accuracy. Second, in the complete robot cell, milling was performed in steel instead of aluminum, and mid-ranged three-dimensional compensation instead of one-dimensional compensation was performed. As mentioned previously, steel is significantly stiffer than aluminum, resulting in stronger process forces and larger machining errors. Because of the large machining errors, the use of the prototype machining cell for milling in steel is not possible, since the micro-manipulator workspace is limited, and no mid-ranging control structure was implemented. Without the proposed mid-ranging control, the outer bounds of the micro-manipulator workspace would quickly be reached. As seen in the top panel of Figure 2.27, the required compensation range for the milling experiments in steel was approximately 1200 μm , which is significantly larger than the micro-manipulator workspace of 400–500 μm , illustrating the need for the mid-ranging control structure. As seen in Figure 2.24, the proposed mid-ranging control for the complete machining cell is successful in keeping the micro-manipulator position in the middle of its workspace.

Based on the results of the two different robot cells, it is likely that an optical tracking system with higher accuracy could improve the machining results further. Thus, for the complete machining cell, the next logical step for future research would be to investigate how much the accuracy could be increased with the use of a new, higher-accuracy optical tracking system.

2.8 Conclusions

An approach to high-accuracy robotic machining was proposed, where a conventional industrial robot in combination with an external piezo-actuated compensation mechanism realizes online correction of errors in the relative position between the tool and the workpiece, beyond the bandwidth of the robot.

For milling in aluminum using the prototype robot cell, the absolute accuracy was significantly increased, the MAE was reduced by a factor of up to 70, to a level lower than 2 μm along each of the three axes. Additionally, the surface accuracy of the millings was increased by up to a factor of three.

For the milling experiments performed in steel using the complete robot cell, one of the harder materials used in production, a significant increase of the absolute accuracy was achieved, up to eight times measured by MAE values. Additionally, the surface accuracy of the machined parts was increased by up to seven times. For all milling experiments, the desired accuracy of 50 μm was achieved. It was further shown that the mid-ranging control structure is essential in order to maintain active compensation throughout the duration of the milling.

These results, and the accompanying hardware and control architecture, are considered as a step toward the goal of having robot cells that are capable of machining with machine-tool accuracy.

3

Adaptive Mid-Ranging Control

3.1 Introduction

This chapter is based on the publication [Sörnmo et al., 2013].

In the setup for performing high-precision milling that was presented in the previous chapter, an MVPC mid-ranging control architecture was proposed. In this chapter, an alternative mid-ranging control approach is presented, which can handle process variation and internal saturations.

In order to obtain a system that is robust to process-parameter variations, which may occur as a result of the strong process forces of the milling process and varying cutting conditions, it is desirable to employ an adaptive control structure. However, in the scenario considered in this chapter, the two manipulators are already controlled closed-loop systems that contain internal input saturations, which does not render the design of an adaptive mid-ranging controller straightforward. Motivated by this, an adaptive internal model control structure for mid-ranging control is presented, with adaptive dynamic reference governors for compensation of internal saturations, making the control approach possible.

A set of different mid-ranging control strategies are evaluated in [Allison and Isaksson, 1998], based on, *e.g.*, Valve Position Control (VPC) and Model Predictive Control (MPC). Design and tuning guidelines of VPC and Modified VPC controllers (MVPC) are presented in [Allison and Ogawa, 2003]. Anti-windup schemes for VPC controllers are introduced in [Haugwitz et al., 2005].

Internal Model Control (IMC) is reviewed and compared with similar control strategies in [Garcia and Morari, 1982], where also several IMC stability theorems are proven and practical tuning guidelines are provided. An extension of IMC to nonlinear systems is presented in [Economou et al., 1986], where it is proven that the properties of linear IMC also applies to the

general nonlinear case. The problem of having a control-signal saturation for an IMC controller is considered in [Zheng et al., 1994]. Design and stability analysis of Adaptive Internal Model Control (AIMC) is provided in [Datta and Ochoa, 1996], and the discrete-time counterpart is described in [Silva and Datta, 1999]. Nonlinear approaches to AIMC are investigated in [Hu and Rangaiah, 1999], as well as in [Hunt and Sbarbaro, 1991], where neural networks are utilized.

The application of mid-ranging control using IMC is investigated in [Gayadeen and Heath, 2009], where design rules are presented and verified through simulation studies.

Discrete-time Dynamic Reference Governors (DRG) for constrained nonlinear systems are considered in [Bemporad, 1998], and reference governors for systems with input and state saturations are presented in [Gilbert et al., 1995].

The method presented in this chapter is based on [Gayadeen and Heath, 2009], which is here extended by introducing adaptivity to the IMC mid-ranging structure, inspired by [Datta and Ochoa, 1996; Silva and Datta, 1999]. Further, the control scheme is modified to account for internal saturations, by introducing a DRG based on the concepts in [Gilbert et al., 1995], but derived using a different approach. Further, in order to maintain performance under parameter variation, the DRG is made adaptive.

Disposition

This chapter is organized as follows. The DRG is derived and the design of the AIMC controller is described in Section 3.2. Section 3.3 presents simulation results obtained with the proposed control structure. The experimental setup is described in Section 3.4, and the experimental results are presented in Section 3.5, followed by a discussion and conclusions in Section 3.6.

3.2 Method

Consider two stable, discrete-time closed-loop systems on the standard feedback form, see Figure 3.1, denoted $H_{cl}^f(z)$ and $H_{cl}^s(z)$, representing the fast micro and slow macro manipulator, respectively. Consequently, the bandwidth of $H_{cl}^f(z)$ is significantly higher than that of $H_{cl}^s(z)$. The controllers in the closed-loop systems are assumed known, and the output signals of the closed-loop systems, denoted y_f and y_s , are measured. The midpoint of the micro manipulator workspace is zero.

The objective to perform mid-ranging control of the two closed-loop systems can be met by standard methods such as VPC control, as was shown in Chapter 2. The block diagram for the VPC structure is constructed as displayed in Figure 3.2, where y_r is the relative position of the manipulators,

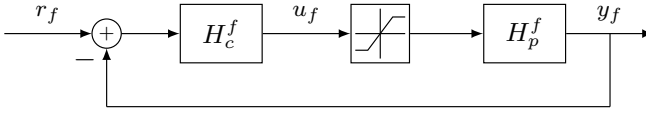


Figure 3.1 Block diagram for the standard feedback form with input saturation to the process. The transfer function from r_f to y_f , *i.e.*, for the non-saturated case, is denoted $H_{cl}^f(z)$.

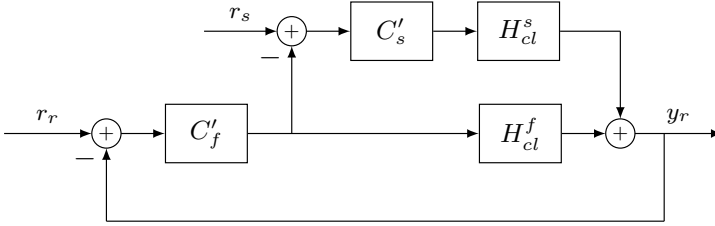


Figure 3.2 Block diagram for the VPC and MVPC mid-ranging control structures.

r_r the desired relative position and r_s the reference value of the mid-ranged input. It is to be noted that in this research, since the midpoint of the micro manipulator is zero, the input r_s is also zero and will hence be omitted. The structure of the controllers $C'_f(z)$ and $C'_s(z)$ can be chosen arbitrarily, but are commonly selected as PI controllers. Experimental tuning of the controllers is tedious work and even with accurate models of the process, arbitrary pole placement is not always possible. IMC is an appealing solution which has been proven to yield satisfactory results in mid-ranging scenarios [Gayadeen and Heath, 2009]. However, as mentioned earlier, the process parameters may change over time and it is therefore desirable to update the IMC controller in order to adapt to process variations.

The block diagram for the mid-ranging IMC is displayed in Figure 3.3, where C_f and C_s are the controllers, \hat{H}_{cl}^f and \hat{H}_{cl}^s the internal models of H_{cl}^f and H_{cl}^s , respectively. It is to be noted that the notation C_f and C_s represent Q_1 and Q_2 respectively, in the Youla parametrization of the IMC according to [Gayadeen and Heath, 2009]. The complementary sensitivity function $T_f(z)$ is introduced as the desired response of the system from r_r to y_r and $T_s(z)$ is the desired response of the system from r_r to y_r with $w_f = 0$. In order to achieve the desired mid-ranging effect, the following conditions for the

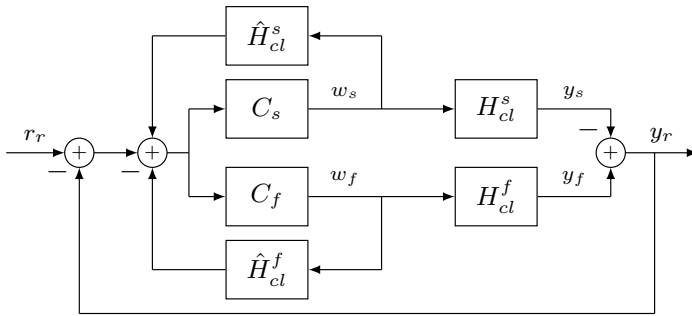


Figure 3.3 Block diagram for the IMC mid-ranging control structure.

controllers must be fulfilled:

$$\begin{aligned} T_f(z) &= C_f(z)H_{cl}^f(z) - C_s(z)H_{cl}^s(z) \\ T_s(z) &= C_s(z)H_{cl}^s(z). \end{aligned} \quad (3.1)$$

The controllers are thus defined as

$$C_f(z) = \frac{T_f(z) + T_s(z)}{H_{cl}^f(z)} \quad (3.2)$$

$$C_s(z) = \frac{T_s(z)}{H_{cl}^s(z)}, \quad (3.3)$$

where the parameters of $H_{cl}^f(z)$ and $H_{cl}^s(z)$ should be updated in order to adapt the controllers. To this purpose, a Recursive Least Squares (RLS) algorithm [Johansson, 1993] with forgetting factor $0 < \lambda < 1$ is introduced, because of its fast convergence for input signals of proper excitation, which are assumed to be present. For systems with low excitation, a Kalman filter may exhibit better performance. The proposed method can easily be modified to incorporate a different estimator. The estimator is used to continuously estimate the process parameters of H_{cl}^f and H_{cl}^s , and consequently update the internal models \hat{H}_{cl}^f and \hat{H}_{cl}^s , as well as the controllers C_f and C_s . The RLS algorithm is stated as [Johansson, 1993; Åström and Wittenmark, 1997]:

$$\begin{aligned} \hat{\theta}_k &= \hat{\theta}_{k-1} + K_{k-1}(y_k - \phi_k^T \hat{\theta}_{k-1}) \\ K_k &= P_{k-1} \phi_k (\lambda + \phi_k^T P_{k-1} \phi_k)^{-1} \\ P_k &= \frac{1}{\lambda} (I - K_k \phi_k^T) P_{k-1}, \end{aligned} \quad (3.4)$$

where P_k is the covariance matrix, K_k a gain matrix, λ the forgetting factor and $\hat{\theta}_k$ is the model parameter estimates from the discrete-time measurement

model of the form

$$y_k = \phi_k^T \theta_k + e_k, \quad (3.5)$$

where e_k is white noise.

However, as displayed in Figure 3.1, H_{cl}^f contains an internal saturation of the inner control signal u_f (denoted $u_{f,k}$ in this section, where k is the current sample), with a given saturation level at $\pm u_{\text{sat}}$. The system H_{cl}^s is assumed not to have an internal saturation. Once the control signal saturates, *i.e.*, when the system leaves its linear region, the linear internal model can no longer accurately describe the process. Further, the estimation of H_{cl}^f will be corrupted as a result of the saturation, since the input/output relation of the plant is no longer linear. This will lead to a false estimate of the system parameters and, in turn, unexpected behavior, which in the worst case results in instability. This problem can be solved by implementing a nonlinear model and applying nonlinear estimation techniques which will become intricate, especially if H_{cl}^f is implemented with anti-windup. Instead, an adaptive DRG is introduced to modify the input to the system, such that the system is never allowed to enter saturation. This approach makes linear modeling still feasible. The dynamics of anti-windup schemes possibly implemented in the closed-loop system can be disregarded, since the system is designed to never enter saturation.

Consider the control structure of H_{cl}^f as displayed in Figure 3.1, where the controller and process dynamics are known and given by the rational, discrete-time transfer functions

$$H_c^f(z) = \frac{n_0 + n_1 z^{-1} + \dots + n_{a-1} z^{-(a-1)} + n_a z^{-a}}{m_0 + m_1 z^{-1} + \dots + m_{b-1} z^{-(b-1)} + m_b z^{-b}} \quad (3.6)$$

$$H_p^f(z) = \frac{q_0 + q_1 z^{-1} + \dots + q_{c-1} z^{-(c-1)} + q_c z^{-c}}{p_0 + p_1 z^{-1} + \dots + p_{d-1} z^{-(d-1)} + p_d z^{-d}}, \quad (3.7)$$

where $[a, b, c, d] \in \mathbb{Z}_0$. The objective is to dynamically modify the reference input $r_{f,k}$ to H_{cl}^f such that $|u_{f,k}| \leq u_{\text{sat}}, \forall k$. Since $y_{f,k}$ is assumed to be measured and $H_c^f(z)$ is known, this can be solved by simply computing the current control signal $u_{f,k}$ and modifying the input to H_{cl}^f accordingly. However, since the mid-ranging controller is meant to control an existing closed-loop system, the output of H_{cl}^f may be delayed from the network connection between the controllers. Therefore, a model-based solution is proposed that is independent of measurements of $y_{f,k}$.

The modified input to the system is denoted $w_{f,k}$, and the dynamic relation between $w_{f,k}$ and $r_{f,k}$ is given by

$$w_{f,k} = w_{f,k-1} + \alpha_k (r_{f,k} - w_{f,k-1}), \quad (3.8)$$

which is a first order low-pass filter with a time-varying parameter α_k . When $\alpha_k = 1$ the filter does not affect the input and $w_{f,k} = r_{f,k}$ holds true and, conversely, when $\alpha_k = 0$, $w_{f,k} = w_{f,k-1}$.

When $|u_{f,k}| \leq u_{\text{sat}}, \forall k$, is satisfied, the control signal $u_{f,k}$ is given by

$$u_{f,k} = \frac{H_c^f}{1 + H_c^f H_p^f} w_{f,k} = \frac{\sum_{i=0}^e q'_i z^{-i}}{f} w_{f,k}, \quad (3.9)$$

where $e = a + d$ and $f = \max(d + b, a + c)$. In order to determine α_k , the predicted control signal with unaltered reference is denoted by $\hat{u}_{f,k}$ and is defined as

$$\hat{u}_{f,k} = u_{f,k}|_{\alpha_k=1}.$$

If $|\hat{u}_{f,k}| \leq u_{\text{sat}}$, there is no need to alter the input and consequently $\alpha_k = 1$. Otherwise, the desired control signal should be as large as possible, *i.e.*, $\pm u_{\text{sat}}$. The desired control signal in the current time-step k is denoted $u_{f,k}^d$ and defined as

$$u_{f,k}^d = \text{sgn}(\hat{u}_{f,k}) u_{\text{sat}},$$

which together with (3.8) and (3.9) gives the expression for α_k :

$$\alpha_k = \begin{cases} \frac{p'_0 u_{f,k}^d + \chi(u_{f,k}, w_{f,k})}{q'_0 (r_{f,k} - w_{f,k-1})}, & |\hat{u}_{f,k}| > u_{\text{sat}} \\ 1, & |\hat{u}_{f,k}| \leq u_{\text{sat}} \end{cases} \quad (3.10)$$

where

$$\chi(u_{f,k}, w_{f,k}) = \sum_{i=1}^f p'_i z^{-i} u_{f,k} - \sum_{i=1}^e q'_i z^{-i} w_{f,k} - q'_0 w_{f,k-1}. \quad (3.11)$$

It is clear from (3.10) that $q'_0 \neq 0$ must be fulfilled, *i.e.*, that $H_c^f(z)$ must have a direct feedthrough path, which can easily be satisfied by a proportional part in the controller.

Since the system $H_{cl}^f(z)$ is likely to be time varying, and the inner controller $H_c^f(z)$ is fixed, the inner process $H_p^f(z)$ must be estimated in order to adapt the DRG to the process changes. Since measurements of $u_{f,k}$ are not available, and only estimated based on time-invariant models, the process dynamics $H_p^f(z)$ cannot be determined based on the estimations. However, under the assumption that $|u_{f,k}| \leq u_{\text{sat}}, \forall k$, holds true, the inner process of

the system can be expressed in terms of $H_{cl}^f(z)$, which is already estimated to adapt the IMC controller, and $H_c^f(z)$:

$$H_p^f(z) = \frac{H_{cl}^f(z)}{H_c^f(z)(1 - H_{cl}^f(z))}. \quad (3.12)$$

Using the fact that $y_{f,k}$ is measured and $H_c^f(z)$ is known, actual values of past control signals are calculated and used to improve the prediction of $u_{f,k}$, reducing errors that occur because of process variation.

The scenario of H_{cl}^f and H_{cl}^s being non-minimum phase systems must be considered, since the systems are inverted according to the control design in (3.2) and (3.3). Inverting non-minimum phase zeros will result in an unstable system. This can be handled by approximating a stable inverse of the system, by mirroring the non-minimum phase zeros in the unit circle, ensuring stability of the resulting controller. The continuous-time version of the IAE optimal approximation is given in [Wiener, 1949]. The discrete-time counterpart is performed by inverting the magnitude of the zero with unaltered argument, an example of this method is shown in Figure 3.4. The described algorithm is implemented and used, if needed, in every sample to mirror the zeros of the models provided by the estimators.

Before activating the AIMC controller, an initial estimation phase is performed using a sufficiently exciting signal as input to the systems, until the estimated models have reached the desired accuracy. During this phase, the requirement $|u_{f,k}| \leq u_{\text{sat}}, \forall k$ is unlikely to be fulfilled by the DRG since its prediction model is being estimated. Therefore it is important to choose the excitation signal such that the system does not saturate during this phase.

The final control architecture for the mid-ranging adaptive internal model control with compensation for internal saturation is displayed in Figure 3.5. Analogously to the method presented in this section, the proposed approach can be extended to systems where internal saturations appear in both H_{cl}^f and H_{cl}^s .

In order to evaluate the proposed control structure, a comparison to the method used in Chapter 2, the MVPC structure, is performed. Following the tuning rules given in [Allison and Ogawa, 2003], the controllers C_f and C_s in Figure 3.2 are chosen as PI controllers and designed using the same desired closed-loop system as for the proposed controller. Since H_{cl}^f contains an internal saturation and the control signal is not available, the PI controllers will undoubtedly suffer from integrator windup problems. However, assuming that the current control signal is available to the controller, a tracking anti-windup algorithm [Haugwitz et al., 2005] can be implemented. Both controllers, with and without anti-windup, denoted MVPC and MVPC+AW, are evaluated in simulation and experiments.

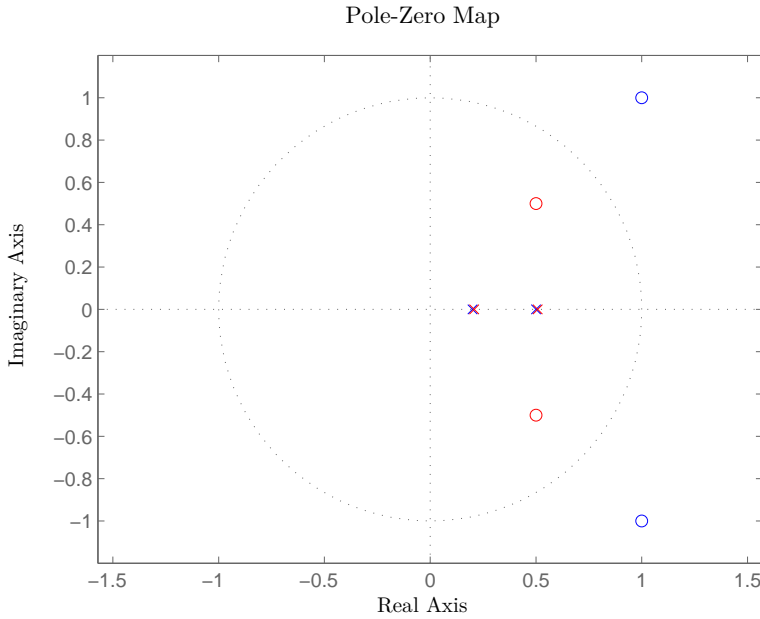


Figure 3.4 Example of mirroring non-minimum phase zeros. The original system is shown in blue and the mirrored system in red.

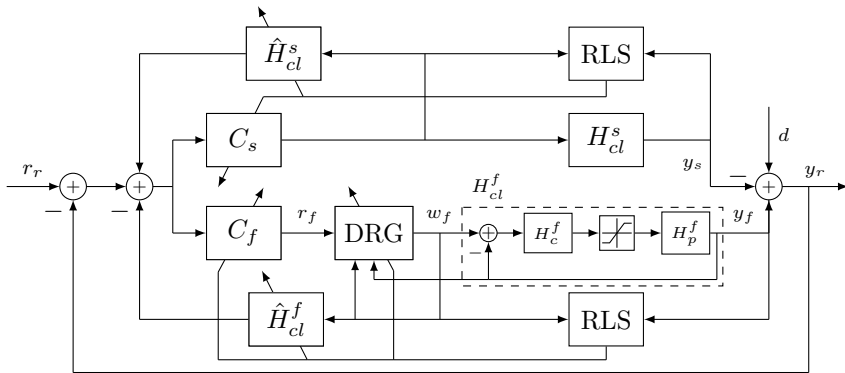


Figure 3.5 Block diagram for the mid-ranging adaptive internal model control, with internal saturation compensation.

3.3 Simulation Results

The proposed control architecture in Figure 3.5 was implemented and tested in MATLAB Simulink, using $u_{\text{sat}} = 10$ and the following systems:

$$H_c^f(z) = 5, \quad H_p^f(z) = \frac{2hz^{-1}}{1-z^{-1}}, \quad H_{cl}^s(z) = \frac{1-e^{-h}}{1-e^{-h}z^{-1}}, \quad (3.13)$$

where h is the sample period of the simulation, in this case $h = 0.004$ s. The desired complementary sensitivity functions are set to

$$T_f(z) = H_{cl}^f(z), \quad T_s(z) = H_{cl}^s(z), \quad (3.14)$$

which corresponds to preserving the bandwidths of the closed-loop systems. This choice is motivated by the assumption that the systems are well-controlled closed-loop systems, ideally having as high bandwidth as possible. An initial guess of the parameters is provided to the estimators and the systems are excited using a low amplitude square wave. In the first simulation, a ramped square wave with a superimposed low frequency sine wave is sent as relative position reference r_r and the relative position y_r is subject to a step position disturbance d . The results of the simulation are displayed in Figure 3.6. The ramped input is used to demonstrate the mid-ranging effect of the micro manipulator system position y_f , which is clearly visible from the bottom panel in Figure 3.6, where the green curve is close to its midpoint. Further, it can be concluded that H_{cl}^f never enters saturation, since the control signal u_f , which is the control signal before entering the saturation, is kept within the saturation bounds. It is also noted that the position disturbance at 18 s is attenuated rapidly, similar to the response of the closed-loop system. This is expected since the disturbance d on y_r can be seen as a disturbance on r_r , thus exhibiting the same dynamics as the closed-loop system from r_r to y_r .

The second simulation focuses on testing the adaptivity of the control, *i.e.*, its robustness to process variations. The simulation is performed using the same input signal as in the first simulation but without the ramp, and also increasing the gain of the plant $H_p^f(z)$ by 50% at time 8 s and subsequently decreasing the gain by 60% at time 16 s. The result of the simulation is displayed in Figure 3.7. It is noted that the gain changes are only visible in the response of the relative position for one period of the square wave. Further, since the gain of $H_p^f(z)$ increases, the system becomes faster and consequently less control signal is needed to achieve the desired response. This leads to less time in saturation and a higher value of α_k .

The simulation results for the MVPC controllers compared to the proposed controller are presented together with the experimental results in Section 3.5, for coherency.

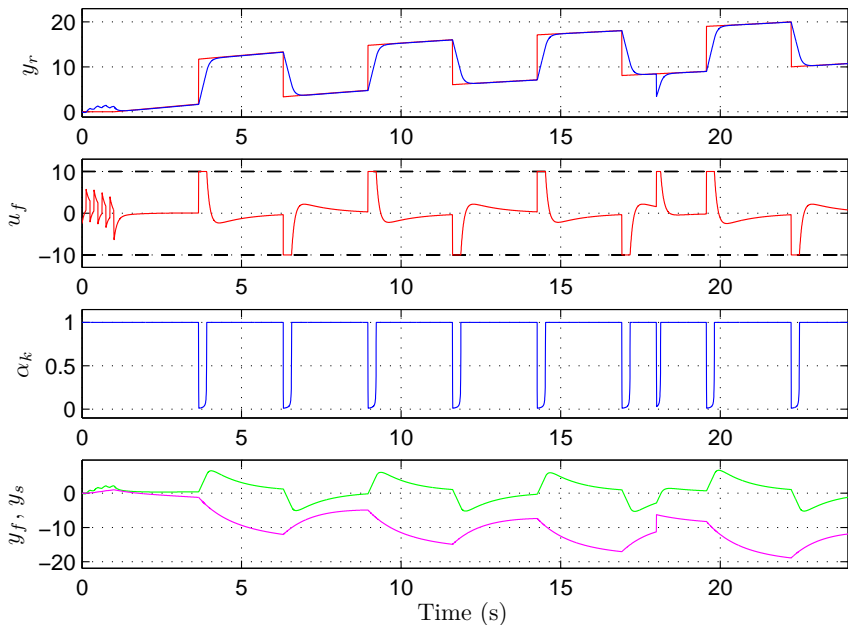


Figure 3.6 Simulation result of the AIMC controller subject to a ramped square wave with a low-frequency sine wave superimposed as reference position (red curve in the top plot). The response of the relative position is displayed in blue in the top panel and the actuator positions are shown in the bottom panel, where y_f is green and y_s is magenta. At time 18 s, the system is affected by a step position disturbance with an amplitude of 5 mm.

3.4 Experimental Setup

The experimental setup used to evaluate the proposed control structure is designed to be a mock-up version of the macro/micro-manipulator setup described in Chapter 2. The setup consists of an ABB IRB2400 robot [ABB Robotics, 2015] with an S4CPlus controller, which acts as the macro-manipulator system, and an ABB IRB120 robot [ABB Robotics, 2015a] with an IRC5 controller, which is the micro-manipulator system with high bandwidth. Naturally, both robots have saturation limits on velocity, but for proof of concept, the high-bandwidth system is set to have an input saturation at ± 80 mm/s. The bandwidth of the macro-manipulator system is assumed to be low enough such that no saturation limits are reached. In addition to the position measurements provided by the robot joint resolvers, the IRB120 robot is equipped with a Heidenhain linear encoder of model ST3078 [Heidenhain, 2013]. The encoder measures the relative distance between the

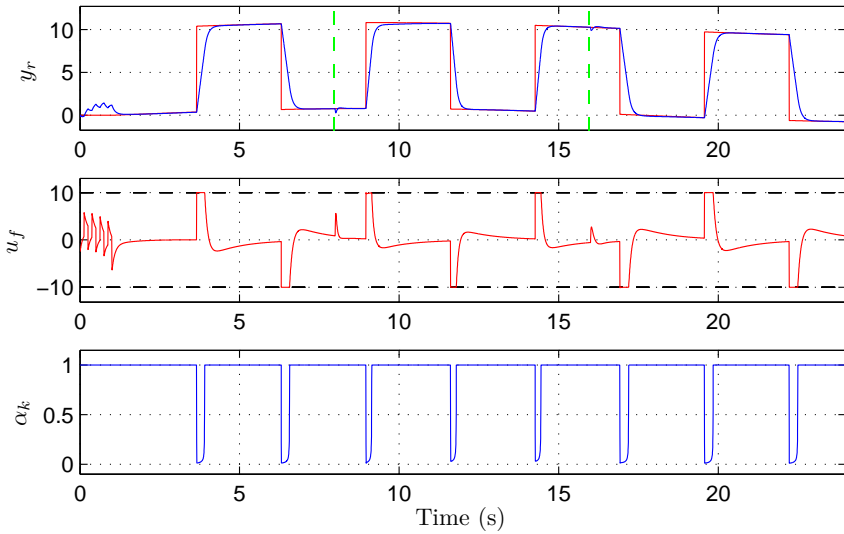


Figure 3.7 Simulation result of the AIMC controller subject to a square wave with a sine wave superimposed as reference position (red curve in the top plot). The response of the relative position is displayed in blue in the top panel. The vertical dashed green lines indicate changes of process gain.

two robot end effectors with a measurement range of 26 mm at an accuracy of 2 μm . This measurement corresponds to the position data obtained using an optical tracking system in Chapter 2. It is essential to measure the relative distance in order to be able to compensate for arm-side position deviations, which cannot be captured by the motor-side measurements. The IRB2400 robot is rigidly attached to the ground, while the IRB120 robot is attached to a base that can move in one direction, in order to introduce disturbances in the position, which frequently appear in the actual milling setup described in Chapter 2.

The robots were interfaced using an open robot control extension of the conventional robot controller, called *ExtCtrl/ORCA* [Blomdell et al., 2010], running at 250 Hz. The MATLAB Simulink models were translated to C code using *Real-Time Workshop* and compiled in order to run them on the extended robot system. A picture of the experimental setup is shown in Figure 3.8.

3.5 Experimental Results

Prior to performing experiments, dynamic models of the two robots, with Cartesian velocity reference as input and Cartesian position as output, were identified along one axis using the Prediction Error Method [Johansson,

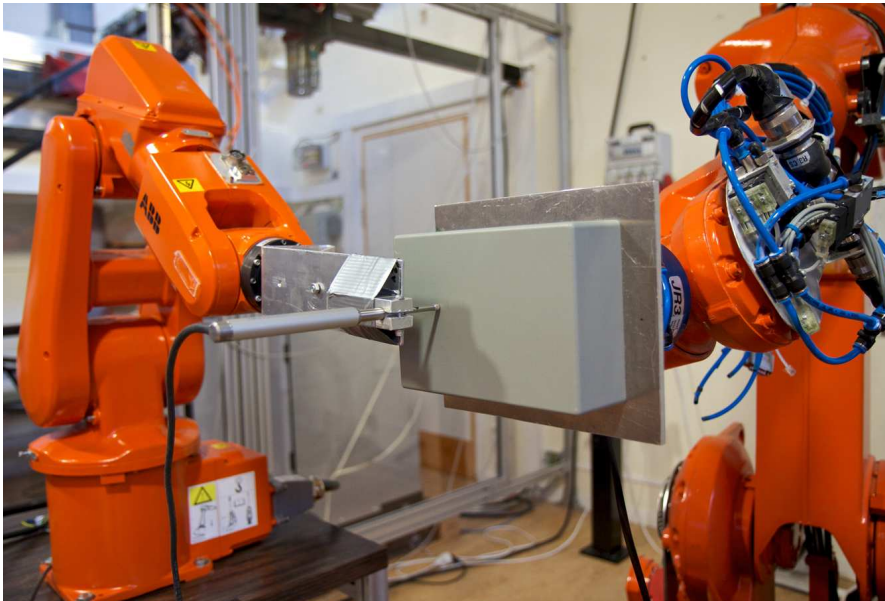


Figure 3.8 Experimental setup for performing mid-ranging control. The IRB120 robot (micro manipulator) with the Heidenhain linear encoder is seen to the left, and the IRB2400 robot (macro manipulator) is seen to the right.

1993]. In order to provide excitation for the identification algorithm, a square wave was used as reference, which is converted to joint motor angle velocity references, using the inverse Jacobian of the robot. The resulting Cartesian position of the robot was computed using forward kinematics, based on measured joint angles, and used as system output. Both robots exhibited similar dynamics and the control loops that form H_{cl}^f and H_{cl}^s , were closed using proportional controllers such that the micro-manipulator system had five times higher bandwidth than the macro-manipulator system. New models of the closed-loop systems were identified, resulting in third-order models, which were used as initial guesses in the RLS estimators. The online estimation of the models was evaluated before initiating the full AIMC control structure, by sending square waves as position references to the two robots. The results of the estimation procedure are displayed in Figure 3.9, where the bandwidth difference of the two systems is clearly illustrated. The desired complementary sensitivity functions T_f and T_s were chosen as first-order systems with bandwidths matching H_{cl}^f and H_{cl}^s , respectively.

The first experiment performed was designed to resemble the simulation in Figure 3.6, but since the linear encoder has limited measurement range,

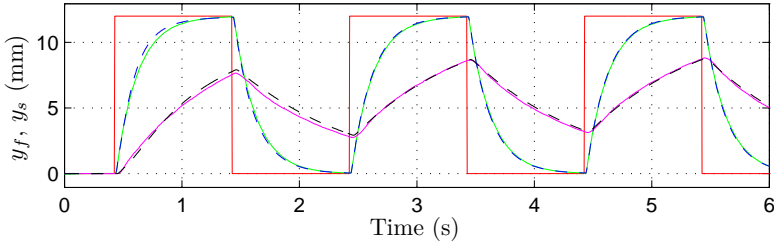


Figure 3.9 Estimation phase of the two systems, where y_f is green and y_s magenta. The estimates of these signals are denoted \hat{y}_f and \hat{y}_s , and are shown in dashed blue and black, respectively.

a ramp signal as input would leave that range rapidly. Thus, the relative position of the robots was instead computed from the resolver measurements of the robots, so that a ramp signal could be used as input. The obtained result is displayed in Figure 3.10. It is evident from the figure that the desired mid-ranging effect is achieved, and in addition the control signal u_f is kept within its boundaries. A zoomed view of a step response from Figure 3.10 is displayed in Figure 3.11, where also the desired response is shown. It is noted that the response for the relative position y_r is close to the desired response. There is, however, an initial discrepancy, which appears because of the fact that the system has an input saturation. As displayed in the lower panel of Figure 3.11, the upper boundary on the control signal u_f has been reached, limiting the achievable bandwidth of the closed-loop system. It is to be noted that given a perfect model of the system, the response of the system with and without the DRG for any input signal, would look the same. The control signal before the saturation (u_f) would, however, not be the same.

Additional experiments were performed in order to test how well the system handles position disturbances. To this purpose, the linear encoder was put into operation, replacing the resolver measurements for the relative position, so that disturbances in position can be measured and compensated for. The experiment was designed such that once the estimation phase finishes, the macro manipulator was controlled to move until the linear encoder is in the middle of its measurement range, which was set to be the zero position. The AIMC controller was then activated, with a square wave as reference signal, while simultaneously moving the base with the micro manipulator in order to introduce position disturbances. As mentioned earlier, the linear encoder only has a measurement range of 26 mm, and thus the amplitude of the relative position reference r_r was chosen to be 5 mm. Since a smaller amplitude of the reference results in less control signal, the desired bandwidth of T_f was increased by a factor of 5. The results of the experiment are shown in Figure 3.12. It is noted that under no disturbances, the system responds rapidly in a well-damped manner. When subject to

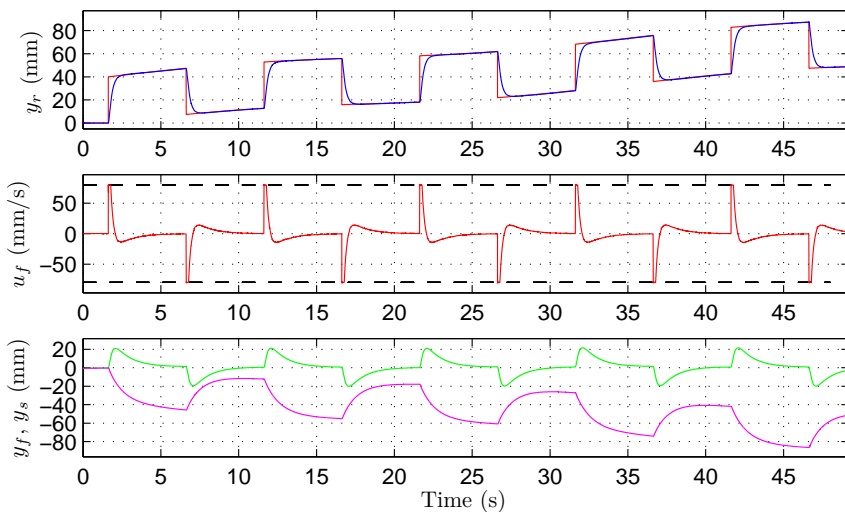


Figure 3.10 Experimental result of the AIMC controller subject to a ramped square wave with a sine wave superimposed as reference position (red curve in top plot). The response of the relative position y_r is displayed in blue in the top panel, and the robot positions are shown in the bottom panel, where y_f is green and y_s is magenta.

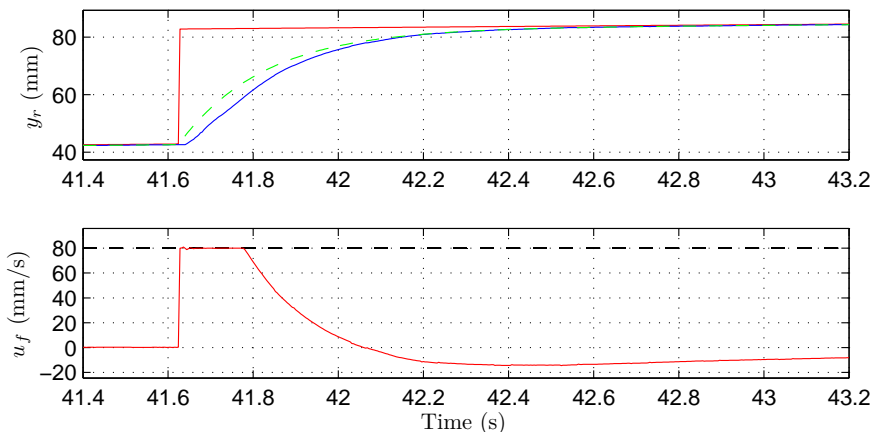


Figure 3.11 Zoomed view of the step change after 40 s in Figure 3.10. The dashed green line shows the desired response of the system, *i.e.*, the response of $T_f(z)$.

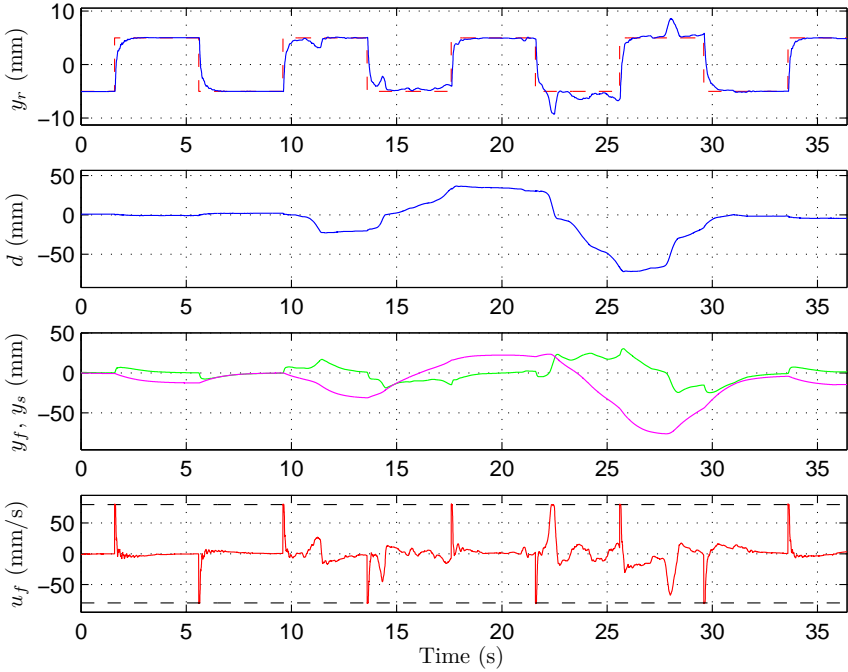


Figure 3.12 Experimental result of the AIMC controller with a square wave as reference position (red curve in the top plot), subject to position disturbances d as displayed in the second panel. The response of the relative position y_r is displayed in blue in the top panel and the robot positions are shown in the third panel where y_f is green and y_s is magenta.

continuous disturbances, the macro manipulator has to deviate further from its desired position, in order to cancel the disturbance. It does, however, eventually return to its midpoint.

In order to quantify the performance of the proposed control structure, the average of the Integrated Absolute Error (IAE) over several step responses was chosen. The discrete-time approximation of the IAE over the interval $[0, k_{max}]$ is defined as

$$\text{IAE} = h \sum_{k=0}^{k_{max}} |r_{r,k} - y_{r,k}|. \quad (3.15)$$

The MVPC controllers were tuned as described in Section 3.2. As the method suggests, model-order reduced versions of the previously identified models of H_{cl}^f and H_{cl}^s were used. The validity of the reduced-order models was investigated by looking at the Bode diagrams, as shown in Figure 3.13. The figure shows close correspondence for both models, in the frequency range of

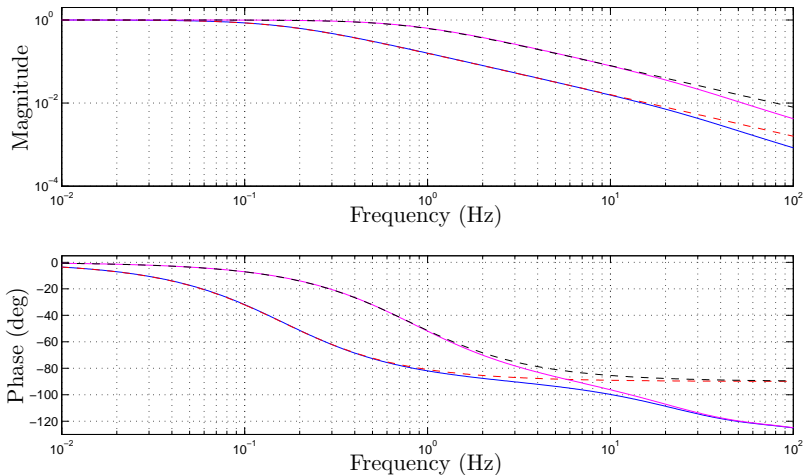


Figure 3.13 Bode diagram of the identified models H_{cl}^f (magenta) and H_{cl}^s (blue) and their reduced-order counterparts, shown in dashed black and red, respectively.

Table 3.1 Normalized IAE value of step responses.

Setup	AIMC	MVPC+AW	MVPC
1. Sim, $r_r = 1.5$	1.000	1.525	1.525
2. Sim, $r_r = 5$	1.000	1.458	1.555
3. Sim, $r_r = 5$, noise	1.000	1.380	1.474
4. Exp, $r_r = 20$	1.000	1.344	1.353
5. Exp, $r_r = 30$	1.000	1.408	1.560

interest. The results of a series of simulations and experiments are presented in Table 3.1, where the IAE values have been normalized by the IAE value of the AIMC controller, in order to simplify comparison. The first setup was designed such that r_r was small enough to fulfill $|u_f| \leq u_{\text{sat}}$, whereas the following four setups were designed more aggressively in order to encounter saturated control signals. In the third setup, measurement noise was added in the simulation. It is to be noted that no process variation was present during the experiments presented in Table 3.1.

Experiments to test the robustness to process variations were performed by attaching a weight to the macro manipulator while running the controller. However, even though the weight was close to being as heavy as the maximum payload of the robot, no significant change in the process dynamics was

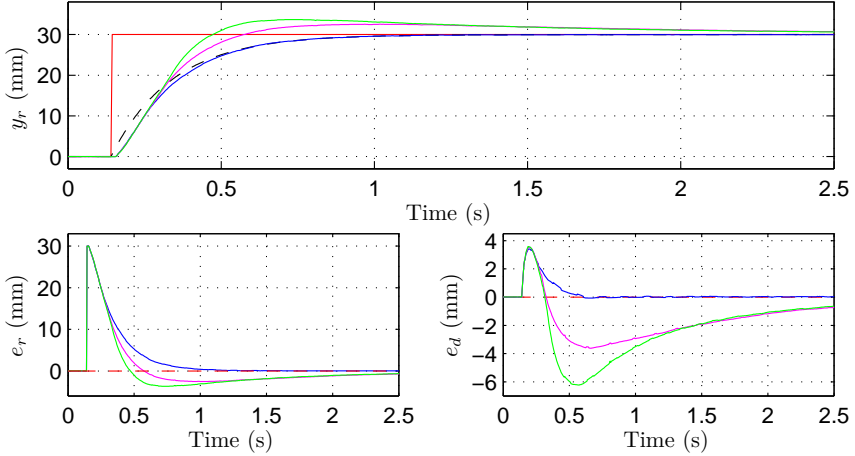


Figure 3.14 Experimental results of responses obtained from experiments with $r_r = 30$. In the top plot, the AIMC, MVPC+AW, and MVPC controllers are shown in blue, magenta, and green, respectively. The desired step response is shown in dashed black. The bottom left plot shows the error between the reference and the responses and the bottom right plot shows the error between the desired response and the measured responses.

observed. Instead, an artificial process variation was introduced by changing the gain at the input of H_{cl}^f . In a similar manner to the simulation studies, the gain was increased by 25% and subsequently decreased by 30%. The results of the experiment are shown in Figure 3.15. It is evident from the figure that the performance is temporarily deteriorated once the process changes, but in approximately three periods of the input signal, the system has adapted and the desired response is obtained. It is also noted that the DRG is efficient in keeping the input constraints, even though considerable model errors occur during the adaptation to the process change.

3.6 Discussion and Conclusions

It was shown in simulations and verified through experiments that the proposed mid-ranging control structure for a macro and micro manipulator setup performs satisfactorily with a response close to the specification, while the desired properties were maintained in the presence of internal saturations, process variations, and position disturbances. It is also noted that the results obtained from the simulations and experiments exhibited close correspondence.

As shown in Table 3.1, the performance of using the proposed controller

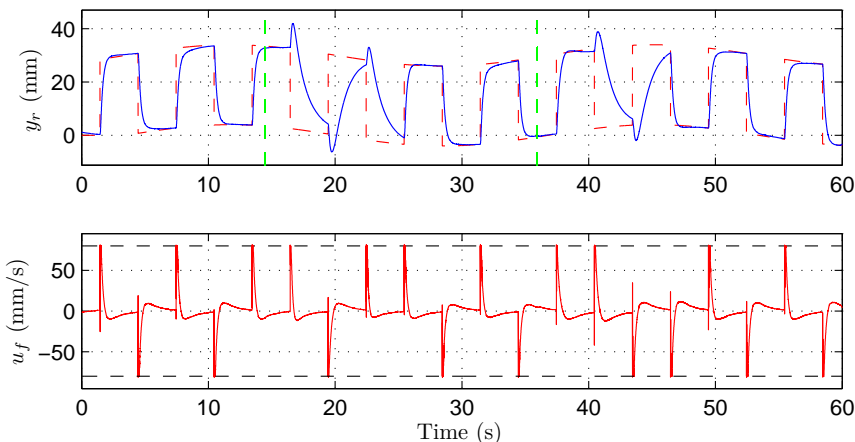


Figure 3.15 Experimental result of the AIMC controller subject to a ramped square wave with a sine wave superimposed as reference position (red curve in the top plot). The response of the relative position y_r is displayed in blue in the top panel. The vertical dashed green lines indicate changes of process gain.

as compared to the MVPC controllers in the case of time-invariant processes is increased by as much as 56%. In the cases where process variations are present, the proposed controller will naturally perform significantly better than the MVPC controllers, because of its adaptivity. Furthermore, looking at the error between the system response and the desired response, the achieved IAE was up to a factor 16 lower than that of the MVPC controller, as is clearly displayed in Figure 3.14. The performance of the MVPC controller is dependent on if it is possible to implement anti-windup schemes or not, but according to the problem formulation in this chapter, it would not be possible. The MVPC controller without anti-windup will naturally perform worse for high-amplitude steps and references with high-frequency content, since the controller will be saturated more frequently. It could be possible to implement a DRG for the MVPC control structure, similar to the proposed control structure, in order to estimate the control signal for the anti-windup scheme. Since the purpose was to compare the developed controller to previously established methods, this option was not considered. Further, it is noted that the performance of the proposed control structure is deteriorated when subject to substantial measurement noise, since it corrupts the estimations of the models. The noise rejection can be improved by increasing the forgetting factor to a value closer to one, but will consequently result in slower adaptation to process variation. However, in terms of IAE, the controller still performs 38%–47% better than the compared controllers.

When process variations were introduced, the system adapted quickly to the new parameters and the effect of the change could only be seen for a few periods of the input signal. It is, however, noted that the experimental results exhibited slightly slower adaptation with more pronounced transients than in simulation. This is caused by the fact that noise is present in the experiments and as discussed earlier, the forgetting factor should be set to a higher value in order to reduce noise sensitivity. Additionally, the transients appeared because of the increased complexity in estimating the parameters for a third-order model, as compared to the first-order models that were used in simulation. The adaptation of the system is dependent on the excitation of the input signal. With an input signal of high excitation, the system will adapt faster. Conversely, if the input signal has low excitation, the system will adapt slowly and the transient performance will be poor.

The proposed adaptive DRG was proven to be effective. As seen in Figures 3.7, 3.11, and 3.15, the control signal was kept within the constraints, even under significant process variations. Even if no variations in the process dynamics are present, the adaptivity of the controller is still beneficial. This was demonstrated in Figure 3.9, where it is noted that the identified models were improved throughout the estimation procedure, thus increasing the performance of the closed-loop system.

The next step of this research will be to implement the proposed approach on the actual milling setup from Chapter 2 and to evaluate it in machining experiments. It is possible that the method needs modification in the case of low excitation in certain machining tasks. Further, it would be desirable to extend the method to account for the endpoints of the micro-manipulator workspace, such that it can be guaranteed that they are never reached.

4

Modeling and Identification of Robot Joints

4.1 Introduction

This chapter extends the publication [Lehmann et al., 2013].

As an alternative to the method presented in Chapter 2, a different approach to achieving the objective of higher machining accuracy is considered in this chapter. A method is proposed that involves developing a model of the non-ideal properties of the robot joints. This model may in turn be used for the development of a model-based path-planning algorithm in order to eliminate position errors in the machining task. The focus of this chapter is to establish a model of the relevant robot characteristics and develop a subsequent method for identification of the model parameters for an arbitrary industrial robot. A method for identifying both joint backlash and stiffness is presented, which relies on attaching the robot end effector to a stiff environment, and has therefore been dubbed the *clamping* method. In the context of performing machining, modeling of the robot joints is of interest in order to be able to predict the position deviations, which occur because of the machining process forces [Reinl et al., 2011; Abele et al., 2011]. In order to significantly improve the machining accuracy, joint models of high quality are required. Additionally, since the models in this chapter are identified in joint space, they are valid in the whole robot workspace, in contrast to local methods based on task-space models.

For most machining processes, comparably low speeds and accelerations are used throughout the task. For this reason, link inertia is not as important as the robot joint properties, characterized by, *e.g.*, backlash and stiffness. These properties account for the major part of the position deviations of the robot end effector that occur during machining tasks, and thus need to be modeled and identified in order to apply compensation. Methods for achieving high position accuracy for an unconstrained robot, *i.e.*, moving in

free space, are commercially available. However, achieving the same accuracy for a constrained robot, which is affected by process forces, is still an open research problem.

The proposed modeling approach, and the subsequent method for identification of the robot joint parameters are presented and evaluated using an experimental setup.

Previous Research

The idea of clamping the end effector to a stiff environment has previously been proposed to the purpose of kinematic calibration, in [Bennett et al., 1992]. Methods for modeling and determining the backlash in robot joints was investigated in [Hovland et al., 2002; Ruderman et al., 2009], whereas stiffness modeling and identification are discussed in [Abele et al., 2007; Abele et al., 2008; Ruderman et al., 2009; Dumas et al., 2011; Schneider et al., 2015]. In the majority of previous joint parameter identification methods, the quantities of robot stiffness and backlash are obtained by applying an external force or a load on the robot tool center point (TCP), and subsequently measuring the resulting end-effector force with a force/torque-sensor. Simultaneously, the corresponding deflection of the robot end effector is measured using high-resolution measurement equipment, such as optical tracking systems or arm-side encoders. One of the methods presented in [Abele et al., 2007] relies on clamping the preceding robot joints in the kinematic chain, in order to neglect their properties when applying loads on the measured joint. Compensation of position errors for robots affected by external process forces was investigated in [Wang et al., 2009; Pan and Zhang, 2009]. In [Tyapin et al., 2014], measuring stiffness of robot joints was considered by combining measured external forces on the end effector with optical measurements of the robot link positions.

Contrary to these methods, the method in this chapter only utilizes measurements already available in the robot system. This reduces both the investment cost in sensors significantly, as well as the time required for calibration of the sensors. Additionally, the excitation for the identification experiments is achieved by using the robot system; no external load is required. This makes it possible to execute the method autonomously.

Disposition

This chapter is organized as follows. A description of the modeling approach is provided in Section 4.2. Section 4.3 presents the clamping method, considering both stiff and compliant environments. The two different platforms used for experiments are described in Section 4.4, and the results are presented in Section 4.5. A discussion of the method and the results, as well as conclusions are provided in Section 4.6.

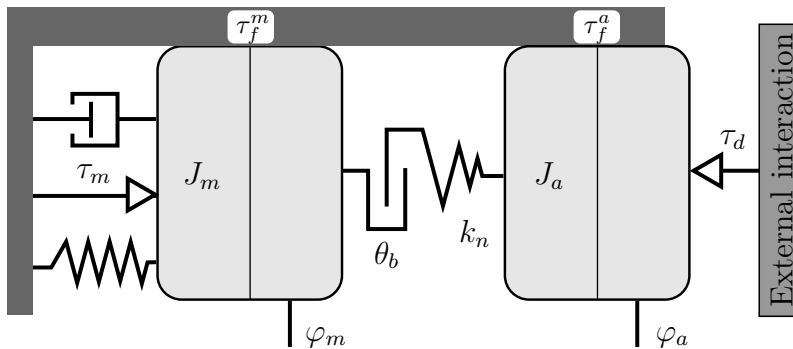


Figure 4.1 Robot joint model with nonlinear dynamics, visualized as a prismatic joint for simplicity. The gear ratio is omitted in the figure.

4.2 Modeling Approach

The model developed in this chapter is opted for describing the characteristics of the robot in a machining scenario. The most dominant effects in the robot joints, influencing the resulting machining accuracy performed with industrial robots, are identified from extensive experiments as the following:

- Backlash in the gearbox of the joint, *i.e.*, lost motion of the gears caused by the clearance between mated gear teeth. The backlash is in this chapter described by an angle, corresponding to the lost motion that occurs when the gear motion is reversed.
- Stiffness of the joint, *i.e.*, the relationship between applied torque and the difference between the motor- and arm-side angle of the robot joint.

Consequently, the model of the robot joints proposed in this chapter contains the elements above. Friction in the joint is not treated in this chapter; details on friction modeling and identification for robots can be found in, *e.g.*, [Bittencourt et al., 2010; Bagge Carlson et al., 2015]. The model components for each of the joints are schematically visualized in Figure 4.1. The corresponding notations for variables and parameters are collected in Table 4.1. Since the link effects are omitted in the proposed model, each joint is modeled independently of the others. However, cross couplings between the joints can be introduced as external disturbances in each joint model. The main model components that are considered in this chapter are discussed next.

Table 4.1 Variables and parameters in the robot joint model.

τ_m	—	Torque from controller to motor
τ_d	—	External disturbance torque
φ_m	—	Joint angle, motor side
φ_a	—	Joint angle, arm side
θ_b	—	Backlash angle
k_n	—	Nonlinear spring function
τ_f^m	—	Friction torque, motor side
τ_f^a	—	Friction torque, arm side
J_m	—	Actuator/motor inertia
J_a	—	Arm-side inertia

4.3 Clamping Method

In this section, a procedure for identifying the main parameters of the robot joint model described in the previous section is presented. The proposed method is based on locking all DoF of the manipulator, by clamping the end effector to a stiff environment, and subsequently actuating each joint sequentially, using the robot control system. In order to achieve accurate estimates of the parameters, two assumptions are required. First, the clamping of the manipulator end effector is assumed to be sufficiently stiff such that any movement of the end effector when the joints are actuated can be neglected. Second, the manipulator is assumed to only be compliant in the rotational direction of each joint, *i.e.*, compliance orthogonal to this direction is neglected. For the cases when these requirements are only partially or not fulfilled, methods for compensating the identified parameters are also presented in this section.

Clamping Procedure

The clamping procedure is initiated by rigidly fixing the manipulator end effector, such that all DoF are locked. This can be done by, *e.g.*, using a tool-changing mechanism, which can easily dock and lock to the environment. Alternatively, the manipulator flange plate can be directly fastened to the stiff environment using bolts. The latter method is more time consuming, but should result in a stiffer clamping. In this chapter, both methods for locking the manipulator end effector were used to evaluate the clamping method.

Once the end effector is in position for clamping, the internal joint position controllers are detuned, a function referred to as *soft servo* [ABB Robotics, 2010]. The detuning is done in two steps; the integral action is deactivated, and the proportional gain of the controller is reduced by a percentage specified by the user. With the end effector locked in place, the contact becomes significantly stiff. If the clamping position is not perfectly aligned,

the position-control supervision system in the robot would activate because of excessive torques on some joint, as a result of the stiff contact. With soft servo activated, the joints become slightly compliant and stiff contacts can be handled. Once the end effector is clamped, all DoF of the manipulator are locked for motion, assuming that the robot has a maximum of six DoF. However, as the robot links are affected by gravity, some of the joint gearboxes may be preloaded even though no external forces are applied on the robot. Therefore, an iterative method for reaching the state where no torques are exerted by the joint motors is employed. The method incrementally alters the position of each joint separately and in a sequence, until the torques in each joint is within a threshold of zero torque. Since soft servo is active for all joints, it is possible to perform this zero-torque search even though the end effector is clamped.

In order to measure the stiffness and backlash properties of a joint, the joint is actuated by applying small deviations in the position reference around the zero-torque position, in alternating directions. Under the assumption that the manipulator end effector is perfectly clamped to a stiff environment, and that the manipulator only is compliant in the rotational directions of its joints, the arm side of the joint is unable to move. Thus, by performing the alternating motions while measuring the joint motor torque and the joint motor position, the relation between the torque and the motor position can be used to extract the desired joint parameters. Consequently, measurements of the joint motor torque and position must be available. However, no high-frequency measurements are required as the motions are performed at low speeds. Because of the assumption that the arm side of the joint is fixed, no external measurement equipment is required to measure the arm-side position. The procedure of performing alternating motions and measuring joint motor torque and position is repeated for each joint separately. When finished, the robot end effector is unclamped, the soft servo is deactivated, and the robot leaves the clamping position. The steps of the clamping method is summarized in the following:

1. Move the robot to the clamping position;
2. Activate the soft servo of the joints, where individual softness values are used for each joint;
3. Clamp the robot end effector using the method of choice;
4. Initiate the zero-torque search, which is run until the applied torque in each joint is within a given threshold of zero torque;
5. Actuate each joint separately and sequentially, using the robot motion controller, by moving it a short distance in alternating

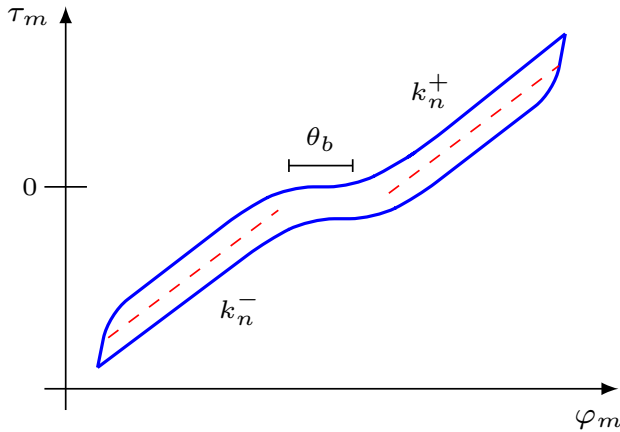


Figure 4.2 Schematic representation of a torque–position map for a robot joint, exhibiting backlash, nonlinear stiffness and friction. The hysteresis of the curve is a result of the joint friction. In this illustration, the stiffness of the joint is modeled as piecewise linear (dashed lines), with a positive and negative stiffness component, denoted k_n^+ and k_n^- , respectively.

directions repeatedly. Record joint motor torques and motor angles simultaneously;

6. Unclamp the end effector, deactivate the soft servo and leave the clamping position.

Parameter Extraction

Using the procedure defined in the previous subsection, data sets of measured joint motor torques and positions are obtained. With these data sets, torque–position maps can be constructed, from which the desired joint parameters can be extracted. An illustration of a schematic torque–position map is shown in Figure 4.2, where the backlash θ_b , which occurs around zero torque after a switch in direction of the gears, is displayed. Additionally, the joint stiffness properties are modeled as piecewise linear, with a positive and negative component, denoted k_n^+ and k_n^- , respectively. However, it is to be noted that any representation of the stiffness measurements can be used, nonlinear or linear. The joint parameters can subsequently be identified by regression of the measured torque/position data used to form the map in Figure 4.2. It should be noted that additional information can be extracted from the torque–position map, such as the joint motor friction. However, as mentioned in Section 4.2, friction is not considered in this chapter.

Clamping in Compliant Environments

As stated previously, the clamping method relies on the assumption that the robot end effector is completely locked for motion. If it is possible for the arm side of the joints to move, the estimated joint stiffness will be underestimated as a result of the joint motion. In practice, it is likely that some setups will exhibit compliance in the clamping of the end effector. If the joint motion can be measured, the underestimated joint parameters can be compensated. However, accurate measurements of joint motion are time consuming and typically require expensive measurement equipment. Therefore, these measurements are used to provide validation measurements for comparison. By measuring the Cartesian positions of discrete points on the robot links using a 3D tracking system, the arm-side angle of each joint with respect to the preceding link can be computed. This is achieved by projecting the measured positions to the joint actuation plane, *i.e.*, the plane with a normal in the direction of the joint rotational axis. In order to obtain the actuation plane, calibration is performed by moving each joint separately, while measuring positions with the tracking system. A subsequent singular-value decomposition of the position data is performed, and the actuation plane is obtained using the left singular vectors corresponding to the two largest singular values. Additionally, the radius from the joint rotational axis to the measurement point on the link is calculated from the tracking system measurements. The radius is required in order to transform the Cartesian position measurements of the link to the corresponding joint angles. The arm-side joint angles are then transformed to motor-side angles using the gear ratio matrix $J_g \in \mathbb{R}^{6 \times 6}$. The motor position data acquired during a clamping experiment can then be compensated for the possible motion obtained using the described method, by subtracting the measured joint motion from the motor position.

As an alternative to this method, measurements of Cartesian space 6D-deflections of the robot end effector can be used for compensation. These measurements are not as demanding to obtain as acquiring the joint angle motion, and can be obtained by using, *e.g.*, dial gauges or linear encoders. Thus, a method to compensate the parameter estimates using measurements of Cartesian space deflections of the robot end effector is proposed. Assuming that the deflections of the end effector $\Delta X_e \in \mathbb{R}^6$ are small, the corresponding arm-side joint angle deflections can be calculated using the relation

$$\Delta \varphi_a = J(\varphi_a)^{-1} \Delta X_e, \quad (4.1)$$

where $J(\varphi_a) \in \mathbb{R}^{6 \times 6}$ is the Jacobian matrix of J of the robot. As for the method using link measurements, the compensated motor angles are computed using the obtained joint motion.

For the setups considered in this chapter, the compliance matrices of the

clamping environments were not available. However, for the case when the compliance matrix $C_e \in \mathbb{R}^{6 \times 6}$ of the clamping environment is known, yet an alternative method is outlined here. The end-effector deflections can be obtained by

$$\Delta X_e = C_e F_e, \quad (4.2)$$

where $F_e \in \mathbb{R}^6$ denotes the external forces and torques applied to the end effector. In order to eliminate the need for additional measurements, the end-effector forces and torques can be estimated from the joint torques using the relation [Spong et al., 2006]

$$F_e = J(\varphi_a)^{-T} J_g^T \tau_m. \quad (4.3)$$

Alternatively, the force-estimation method discussed in Chapter 5 can be used, which takes the joint friction into account. With the estimated end-effector deflection, the corresponding joint angle deflections can be calculated as defined in (4.1), such that

$$\Delta \varphi_a = J(\varphi_a)^{-1} C_e J(\varphi_a)^{-T} J_g^T \tau_m. \quad (4.4)$$

Finally, the same procedure to compensate for the joint motion as for the previous method is applied.

4.4 Experimental Setup

In order to evaluate the proposed method, an experimental platform consisting of a six DoF ABB IRB140-robot [ABB Robotics, 2015b] with an IRC5 controller, in combination with an open robot control extension called *ExtCtrl/ORCA* [Blomdell et al., 2010] was used. The proposed method was implemented in RAPID, and measurements from the robot were collected both through RAPID and ExtCtrl. Measurements of the joint motor torque for each joint cannot be directly accessed from the robot system; the torque reference signal is, however, available. Since the motor current loop is tightly controlled, the reference signal is assumed to sufficiently approximate measurements of the joint motor torque [Stolt, 2015].

For verification purposes, 6D position/orientation measurements were obtained using a Nikon K600 optical tracking system [Nikon Metrology, 2010]. By attaching a large number of LEDs at strategic positions on the robot, position measurements in both Cartesian space and joint space could be achieved with an absolute Cartesian accuracy of 50–75 μm , for the current measurement configuration.

Two different clamping setups were used to perform the experiments. In the first setup, denoted Setup 1, both the robot base and end effector were directly attached to the same heavy steel plate using bolts. In the

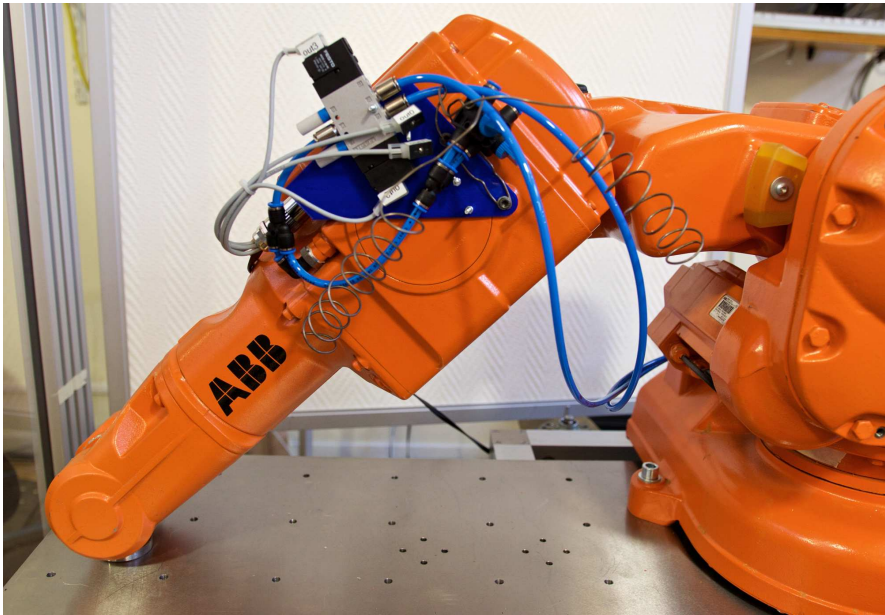


Figure 4.3 Setup 1, where both the robot base and the end effector are rigidly attached to a large steel plate.

second setup, denoted Setup 2, a tool-changing system attached to an aluminum-frame table with a high density fibreboard table top was used, in order to investigate clamping in a compliant environment. The two different setups are displayed in Figures 4.3 and 4.4.

4.5 Experimental Results

In this section, results from clamping experiments using the two setups described in the previous section are presented.

Results from Setup 1

The results of the experiments performed by clamping onto the steel plate are displayed in Figure 4.5, where torque–position maps are shown¹ for each of the six joints of the robot. The means of the motor positions have been subtracted, such that results from different clamping configurations, *i.e.*,

¹ For the torque–position maps presented in the figures of this section, several actuation cycles were performed, and as a result of high repeatability, dashed lines may appear solid.

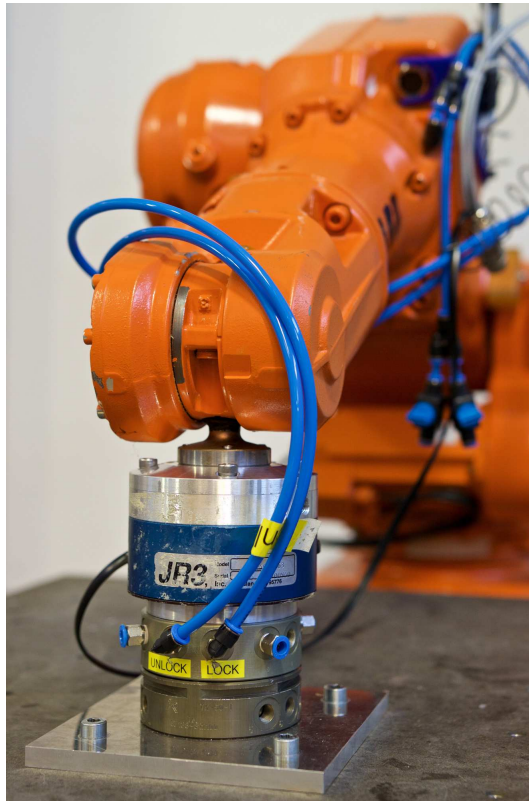


Figure 4.4 Setup 2, where the robot end effector is clamped using a tool-changing system to a compliant table. The force/torque sensor visible in the picture is not used in the experiments.

different positions of the clamping device in the workspace of the robot, can be compared easily. In the figure, the measurement data of torque and position are displayed, as well as the actual transmission characteristics, obtained by compensating the data for any joint motion that occurred during the experiment. To this end, measurements from the optical tracking system and the method described in Section 4.3 were used.

From the plots in Figure 4.5, several observations can be made. First, it is noted that the backlash of each joint is negligible, *cf.* Figure 4.2, and therefore focus is shifted to the stiffness characteristics of the joints. Furthermore, it is noted that the raw measurement data and the data compensated for joint motion exhibit discrepancies of varying size for each joint. This can be explained by the clamping environment not being perfectly stiff and that orthogonal joint compliances are present, *i.e.*, the assumptions

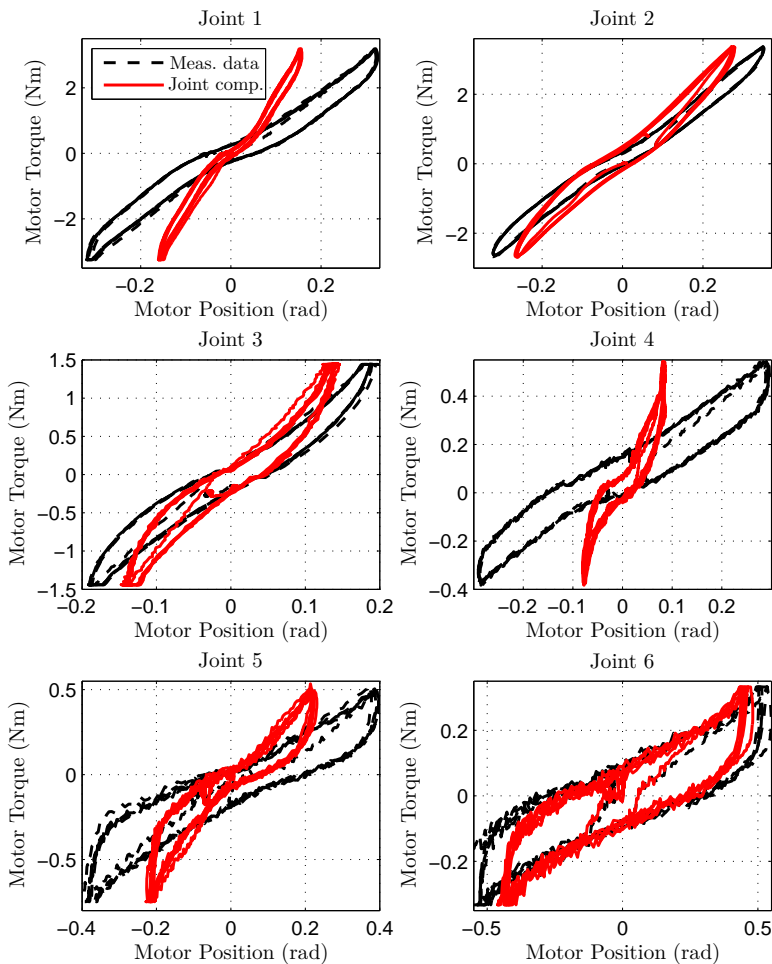


Figure 4.5 Torque–position maps of measurement results for joints 1–6 of the IRB140 robot, obtained using Setup 1. The dashed black curves show the measured data, whereas the red curves show the data compensated for joint motion.

made in Section 4.3 are not completely true. It was found that joint 3 exhibited a significant orthogonal joint compliance, where the bearings were not stiff enough to withhold the forces generated when actuating joint 1. This compliance is most likely a result of the asymmetric design of joint 3 of the considered robot. When actuating joint 1 during a clamping experiment, the robot is 'folded' around the orthogonal direction of joint 3, see Figure 4.6.

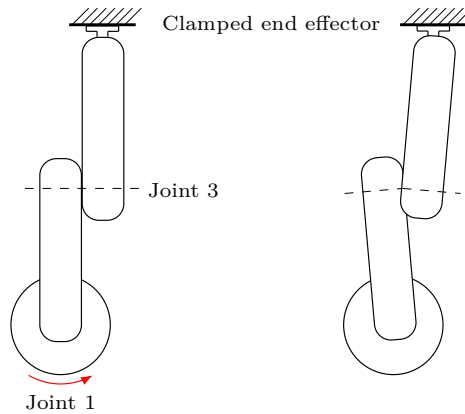


Figure 4.6 Simplified illustration of the considered robot manipulator during a clamping experiment where joint 1 is actuated. To the left, the initial clamping pose of the robot is shown, and the resulting pose of actuating joint 1 is shown to the right. The deflection caused by the orthogonal joint compliance is exaggerated for clarity.

This results in an undesired motion of joint 1, causing the stiffness of the joint to be underestimated. However, this discrepancy was possible to resolve using the optical measurements of the link.

Furthermore, it was found that the clamping configuration influenced the measured transmission dynamics. With a long lever arm, *i.e.*, clamping the robot end effector far away from the actuated joint rotational axis, the stiffness was less underestimated than with a short lever arm. Since the same torque was applied to the joint in both cases, a long lever arm will result in smaller forces at the end effector, which in turn results in smaller deflections of the clamping device and the joints. Thus, for the experimental results presented in this section, the clamping configuration that resulted in the longest lever arm for the respective joint was chosen.

Results from Setup 2

Since significantly stiff environments are not always available in robot cells, clamping experiments in a compliant environment were performed using the second setup described in Section 4.4. The focus of the experiments presented in this subsection was on joints 1 and 2, for comparison to the results of Setup 1. However, the analysis for the remaining joints can be performed analogously. During the clamping experiments, the TCP deflection and the link motion were measured. As for the experiments performed using Setup 1, the link deflections were used to compute the actual joint angle of the actuated joint. Furthermore, the TCP deflection measurements were used

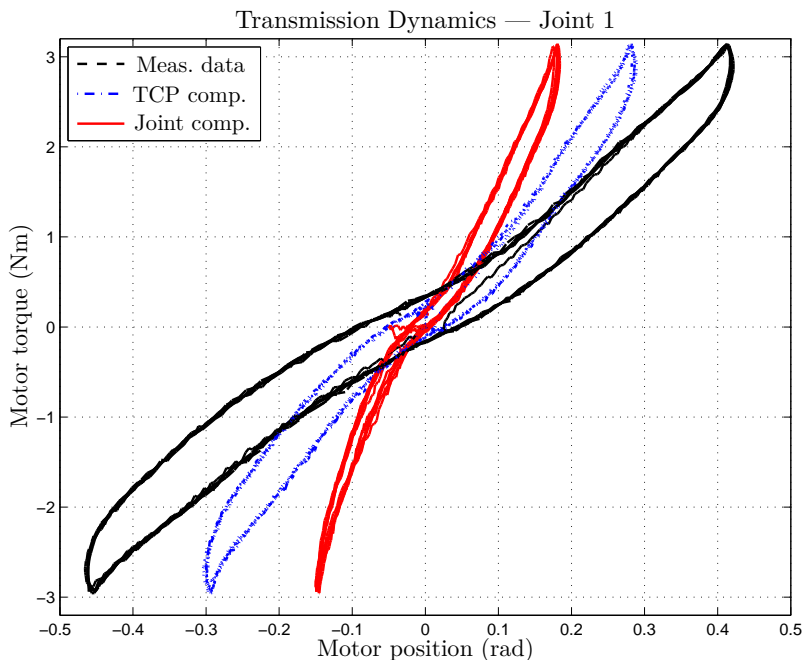


Figure 4.7 Torque–position maps of measurement results for joint 1 of the IRB140 robot, obtained using Setup 2. The figure illustrates the effects of the compliant environment of Setup 2, and the orthogonal joint compliances. The black curve shows the measurement data, the blue curve shows the data compensated for TCP motion, and the red curve shows the data compensated for joint motion.

to calculate the corresponding joint angle deflections, using the relation (4.1). The results of the clamping experiments for joints 1 and 2 are displayed in Figures 4.7 and 4.8, respectively. In the figures, the measurement data are shown, as well as the data compensated using TCP and link deflections. It is clear from both figures that the compensation based on TCP deflection only accounts for a part of the arm-side joint motion. With additional measurements using the optical tracking system, it was again verified that the orthogonal joint compliance of joint 3 was significant in the joint 1 experiments. Additionally, for the joint 2 experiments, it was found that the robot base was moving during the actuation of the joint. Because of these effects, it can be concluded that TCP deflection measurements are not sufficient to obtain the actual transmission characteristics. The error of the estimate could however be reduced by up to 50 % by using the TCP deflection measurements.

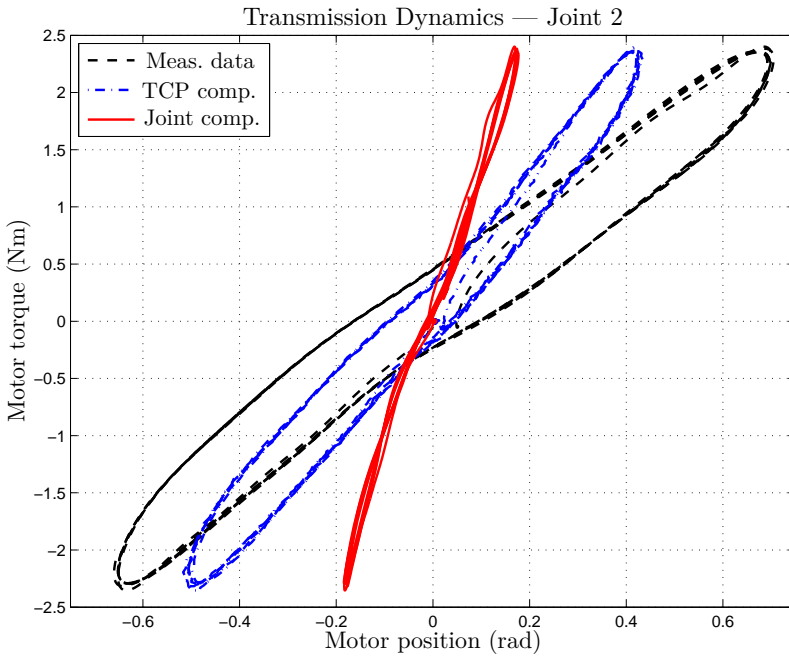


Figure 4.8 Torque–position maps of measurement results for joint 2 of the IRB140 robot, obtained using Setup 2. The figure illustrates the effects of the compliant environment of Setup 2, and the orthogonal joint compliances. The black curve shows the measurement data, the blue curve shows the data compensated for TCP motion, and the red curve shows the data compensated for joint motion.

Consistency of Measurements

In order to verify that the measurements of the joint motion can be used for validation, two different clamping experiments of joint 1 were performed for each of the two setups. The data were compensated using the link measurements and the results of the four experiments are found in Figure 4.9. It is noted from the figure that although the raw measurement data differ significantly, the compensated torque–position maps exhibit a striking correspondence. Furthermore, the previous statement that the joint stiffness is less underestimated when clamping with a long lever arm to the actuated joint rotational axis, is also demonstrated in Figure 4.9. The clamping configurations in each of the setups differ in lever arm length, where the configurations using a longer lever arm correspond to the torque–position maps with a steeper inclination, indicating that the claim of less underestimation of the joint stiffness is correct.

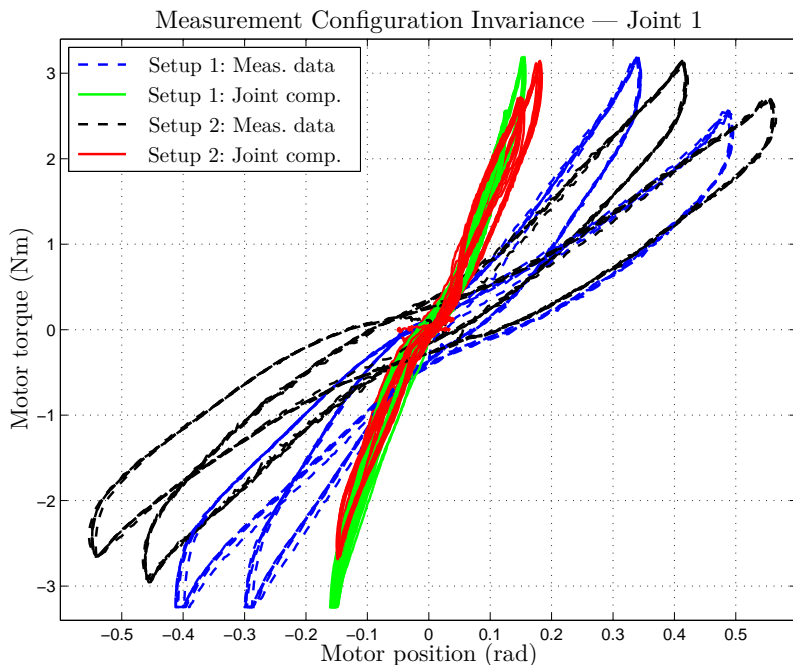


Figure 4.9 Torque–position maps of measurement results for joint 1 on the IRB140 robot, obtained from two different measurement configurations of Setup 2. The blue and black curves show the measurement data obtained using Setups 1 and 2, respectively. The corresponding curves compensated for joint motion are displayed in green and red.

4.6 Discussion and Conclusions

In this chapter, a method for identifying the robot joint characteristics, such as backlash and stiffness, was presented. In contrast to previous methods, where external forces are applied to the robot end effector or to the robot links using complex setups, the proposed method relies on locking all DoF of the robot manipulator by clamping the end effector in a stiff environment. Once locked, the joints are actuated separately in a sequence. Measurements of motor torque and motor position are obtained from internal sensors, and are used to characterize the joint properties. An advantage of this approach is that all joint properties can be measured automatically and the setup does not have to be changed for each joint. Additionally, only measurements from internal sensors that already exist in the robot are used.

The clamping method was verified using two different experimental setups; a stiff and a compliant environment. An optical tracking system

was used to obtain validation data of the transmission dynamics for each joint, as well as deflection measurements of the TCP position during clamping experiments. Two different aspects were investigated; the effect of a compliant clamping environment as well as the orthogonal joint compliances. It was found that the joint stiffness was underestimated when clamping in a compliant environment and as expected, the stiffness was less underestimated when clamping in a significantly stiffer environment. A method for compensating the underestimated stiffness using deflection measurements of the TCP was presented, and was shown to be able to reduce the error by up to 50%. It is to be noted that the TCP position measurements could be obtained using less expensive sensors than used in this chapter, such as linear encoders. The remaining error was found to be caused by orthogonal joint compliances, whose effect could be compensated for with the use of an optical tracking system. Alternatively, it could be possible to attach linear encoders to the manipulator in order to measure the arm-side joint angles directly. While this may require construction of fixtures and extensive calibration, the joint stiffness will be underestimated without access to arm-side measurements.

Even though none of the joints of the considered robot manipulator exhibited significant backlash, the method as such is capable of identifying the backlash.

A possible extension of the method could be to not only use measurement data from a single clamping configuration, but from a series of clamping experiments performed in multiple configurations. It could then be possible to apply a least-squares approximation in order to identify the complete stiffness matrix of the robot manipulator, including the orthogonal joint compliances.

The obvious next step for the proposed method is to utilize the obtained joint models in order to apply compensation to the robot motion. This can be realized in two different ways, either applying online compensation during execution of the robot program, or offline compensation by updating the robot program iteratively after an initial trial of the program. Future research will focus on applying both of these methods, to compare the obtained machining results, both with respect to online/offline compensation, as well as to the other methods proposed in this thesis.

5

Iterative Learning Control for Machining

5.1 Introduction

This chapter extends the method from the publication [Cano Marchal et al., 2014], and presents additional results.

As another alternative to the methods for increasing the machining accuracy presented in Chapters 2 and 4, an iterative learning control (ILC) approach is considered in this chapter. For the application of machining, ILC can be used to adapt the geometric robot path in order to increase the absolute accuracy of the machined parts. The surface accuracy of the machined parts is not considered, as no high-bandwidth compensation can be performed, *cf.* the results of Chapter 2.

The subject of ILC has been comprehensively researched. It was initially introduced in a Japanese journal in 1978 [Uchiyama, 1978], but it did not reach a broad audience until 1984. In that year, three different papers were simultaneously published [Casalino and Bartolini, 1984; Arimoto et al., 1984; Craig, 1984], in which improving position control of robotic manipulators by iterative learning were considered. Later, robots with joint and link flexibilities performing repetitive tasks were considered [Miyazaki et al., 1986; Norrlöf, 2002; Hakvoort et al., 2007]. The idea is to measure the position deviations during the first execution of the task, and subsequently update the control inputs or the specified path and trajectory for the following iterations. Under certain assumptions, convergence of this iterative procedure can be theoretically proven. However, all of the mentioned references consider situations where the robot end effector is moving in free space.

Further development of ILC is presented in the thesis [Norrlöf, 2000]. General surveys of ILC methods, theory, and applications are found in [Bristow et al., 2006; Ahn et al., 2007]. Additionally, real-time algorithms for ILC were proposed in [Xu et al., 2010].

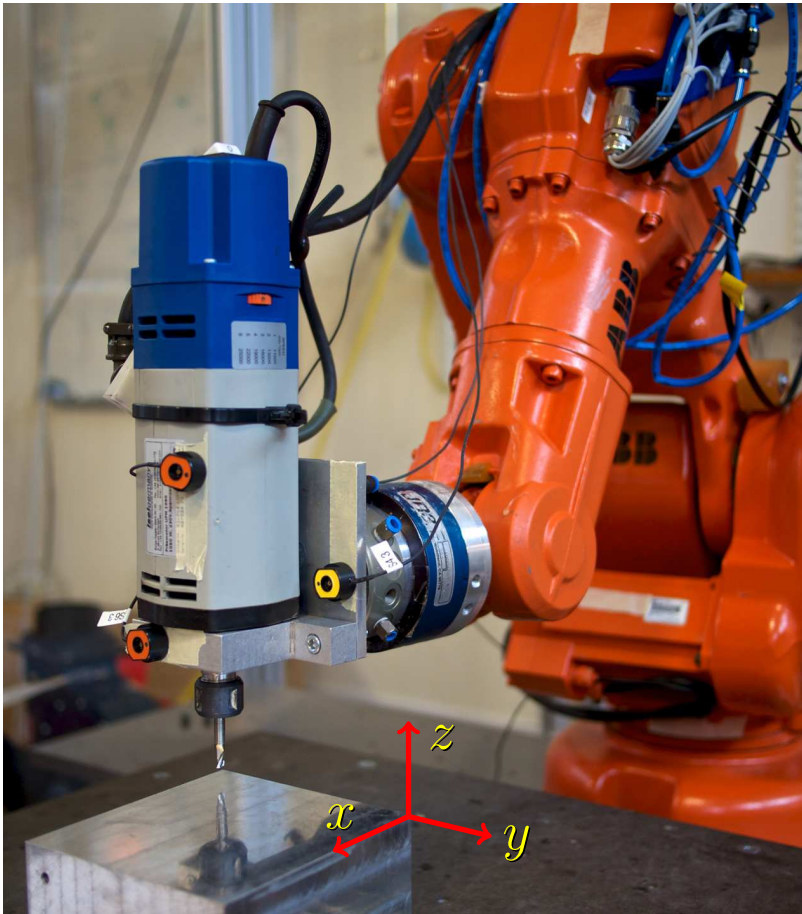


Figure 5.1 The experimental platform used for evaluation of ILC in milling scenarios, with the robot holding the milling spindle and the aluminum workpiece fixed in the robot workspace. The milling spindle is equipped with three LED units, in order to enable high-accuracy tracking of the tool position.

The difference of this research compared to previous applications of ILC in robotics is the contact between the workpiece and the machining spindle, which is required for completion of the machining task. This constitutes a major difficulty in applying the iterative scheme, since the force interaction between the tool and the workpiece must be considered when determining the updated geometric path. The motivation for seeking ILC-based solutions to the problem of increasing the position accuracy in machining, is the common

batch-oriented nature of modern production, which comprises significant repetitiveness in the tasks to be performed.

In this chapter, three different versions of ILC algorithms are developed and subsequently investigated in milling experiments; the difference between the three being the available sensor data. Obviously, with more relevant sensor data available, higher performance can be achieved. First, model-based ILC is considered based on arm-side measurements in task space of the robot using an optical 6D tracking system, in order to demonstrate the effectiveness of the strategy for reducing the position deviations. Second, an ILC algorithm based on joint motor angles and measurements provided by a force/torque sensor attached to the robot end effector, is developed. Finally, an ILC algorithm based on end-effector forces estimated from joint motor torques is developed, *i.e.*, using only sensors that already exist in the majority of conventional industrial robots. The two latter approaches are appealing, since the need for expensive external tracking systems is eliminated.

Disposition

This chapter is organized as follows: Section 5.2 presents the modeling of the robot and the milling process, a subsequent identification procedure for the required models, and the details of the proposed ILC algorithms. The experimental setup and the obtained results from applying ILC to milling in aluminum are presented in Sections 5.3 and 5.4, respectively. The obtained results are analyzed and the method as such is contrasted to other approaches for increasing the robotic machining accuracy in Section 5.5. The chapter is summarized in Section 5.6, where conclusions also are drawn.

5.2 Method

In this section, the robot and the effects of the milling process on the robot are modeled using system identification methods, in order to provide a basis for the design of model-based ILC algorithms. Additionally, the force-estimation method is described, as well as the three different proposed ILC algorithms.

System Modeling and Identification

As mentioned in the introduction, the main difficulty of milling with an industrial robot is the deflection caused by process forces that act on the robot end effector. In addition to this, the accuracy of the position control is limited by the quality of the kinematic calibration of the robot, as well as error caused by the load attached to the robot end effector¹. For a robot

¹ Errors caused by inertial forces are neglected because of the low accelerations that are used when performing milling.

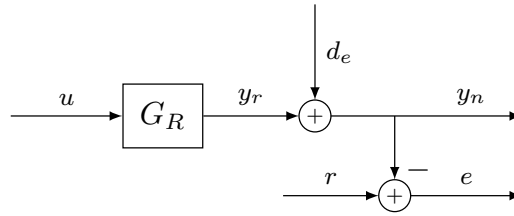


Figure 5.2 Block diagram describing the model of an unconstrained robot.

moving unconstrained in free space, the proposed modeling approach is depicted in Figure 5.2, where the kinematic and load errors are treated as an additive output disturbance. The model variables and subsystems of the block diagram in this chapter are denoted as:

- r : desired position for each spatial coordinate;
- u : control input to the robot, in this case a Cartesian position reference;
- y_r : Cartesian position of the robot end effector, computed from measured joint angles;
- y_n : actual Cartesian position of the robot end effector;
- e : Cartesian position error of the robot;
- d : position deflection caused by the milling process forces;
- d_e : position disturbances caused by kinematic or calibration errors;
- f : measured forces on the end effector along the three Cartesian axes;
- G_R : model of the controlled robot in Cartesian space (including the internal joint-position feedback controllers). More specifically, the transfer function from u to y_r when the robot is moving in free space;
- G_M : model relating y_n to f , *i.e.*, the model describing how process forces arise in the milling task;
- G_K : compliance model relating the process forces f to the deflections d of the robot in Cartesian space.

For the case when the robot is performing milling, the dynamic compliance relation is modeled as two different inherently coupled phenomena; the deflection of the robot when a force is applied to its end effector, and the process forces arising when milling is performed. The model of the first phenomenon aims to link the applied force with the deflection by

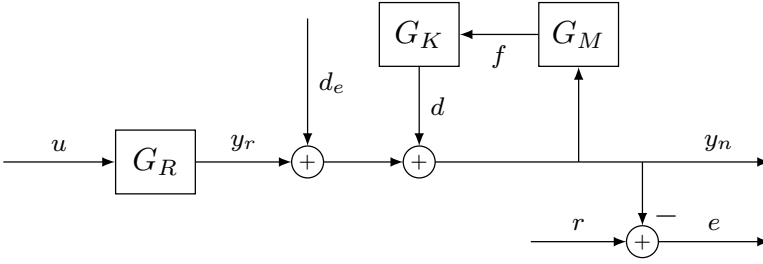


Figure 5.3 Block diagram describing the proposed modeling approach of the milling system model.

the robot—*i.e.*, the relationship which is commonly modeled using Hooke’s Law [Ugural and Fenster, 2003] or generalizations thereof—while the second model is intended to relate the path traversed by the robot end effector to the forces that appear during the milling. A block diagram displaying the proposed modeling approach to the robotic milling system is presented in Figure 5.3.

The following relations formalize the modeling approach described in the previous paragraph and illustrated in Figure 5.3:

$$y_n = G_R u + d + d_e, \quad d = G_K f, \quad f = G_M y_n. \quad (5.1)$$

Robot-Model Identification The model G_R of the controlled robot motion in Cartesian space is estimated using system identification methods. Since the controlled robot motion is considered only in a limited Cartesian workspace in the milling task², linear models can be justified. In particular, a chirp excitation signal is applied in each direction, and the response in position is obtained using the measured joint motor angles and forward kinematics. Using a time-series modeling approach, third-order discrete-time dynamic models are identified and subsequently used in the ILC algorithms. The models capture the inherent resonant character of the mechanical structure, which is a result of the joint and link flexibilities of the robot.

Process-Force Model Identification Intuitively, it is natural to assume that the process forces that occur during a milling experiment are related to the amount of material that is being removed, *i.e.*, related to the depth of cut and the feed rate. Thus, the model G_M is chosen with robot position as input and process forces as output. However, as the robot position does not contain any information on the depth of cut of the milling, it is assumed that the

² Considering that the robot configuration influences the controlled dynamics of the robot end effector, configuration-dependent linear parameter-varying models or nonlinear models (and thus nonlinear ILC methods) explicitly accounting for this property are required if larger workspaces are to be considered.

depth remains constant throughout the milling. Additionally, in order not to restrict assumptions on which directions process forces arise, a multi-input multi-output (MIMO) model structure is considered. This structure is also necessary as process forces were found to appear by cross couplings in all Cartesian directions even when milling was performed in a single direction.

The model can then be identified by recording the following sensor-data signals during a milling experiment:

- The forces acting on the end effector (f);
- The arm-side position of the robot (y_n), measured by an optical tracking system.

The required sensor data can be collected during the initial uncompensated ILC iteration using the nominal geometric path; thus avoiding the need for separate identification experiments prior to application of the ILC iterations.

With the collected sensor data, PEM was used to identify the MIMO model G_M , which is a 3×3 matrix of discrete-time transfer functions. The orders of the different identified transfer functions vary between 2–6. The measured forces and the output of the model for all Cartesian axes are shown in Figure 5.4, where it is noted that the model exhibits satisfactory fit to the experimental data for all axes.

Because of the assumption on constant depth of cut, the process-force model must be reidentified if a different depth of cut is considered. However, should a linear or affine relationship be assumed, the previously identified model could simply be changed correspondingly with respect to the change in depth of cut.

Deflection-Model Identification As stated earlier in this section, the purpose of the model G_K is to relate the deflections exhibited by the robot with the forces applied on its end effector. For the identification of this compliance relation, it is assumed that the deflections that occur during milling are exclusively caused by the milling process forces. The model can then be identified by recording the following sensor-data signals during the initial milling experiment:

- The forces acting on the end effector (f);
- The estimated position of the robot end effector, computed from the joint motor angles using the forward kinematics of the robot (y_r);
- The arm-side position of the robot end effector measured by an external tracking system (y_n).

With the collected data, a MIMO model describing the dynamics in all Cartesian directions was estimated, with the Cartesian forces as input and the

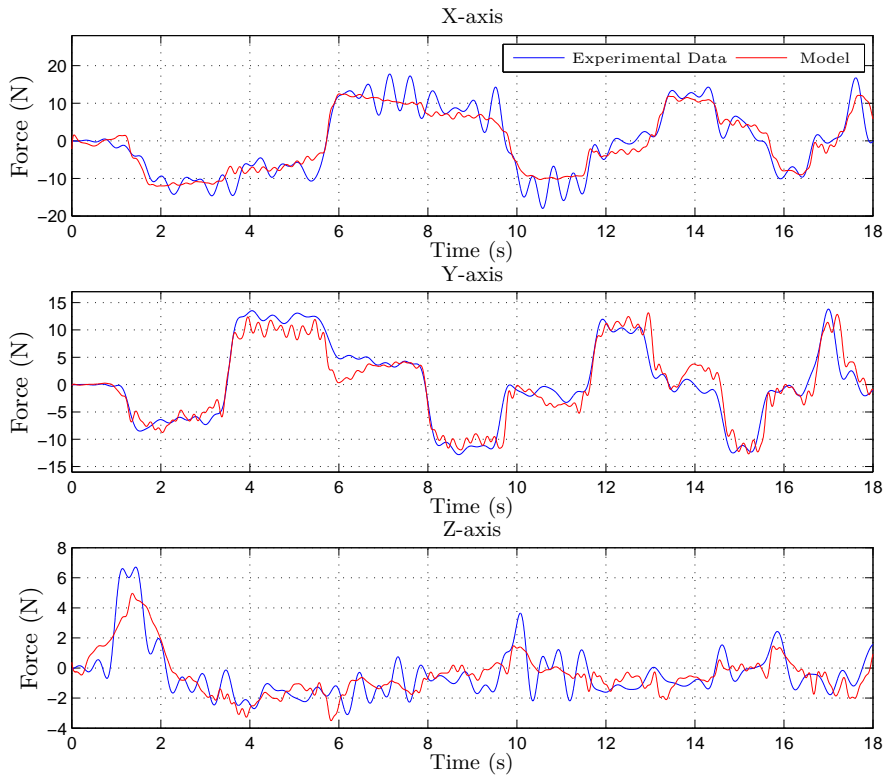


Figure 5.4 Measured process forces and model output (G_M) for all three Cartesian axes.

arm-side deflections as output, using PEM. It is assumed that the orientation of the workpiece is unaltered during the milling, which means that the fixed coordinate system defined in Figure 5.1 was used throughout the experiments. The experimental data and the output of the models for each of the Cartesian axes are shown in Figure 5.5. As can be observed in the figure, the fit of the models to the experimental data is satisfying, capturing the major parts of the compliance dynamics.

In order not to impose any assumptions on the high-frequency properties of the system, dynamic models are considered instead of static models based on Hooke's law. Additionally, the MIMO structure is motivated by investigating the input and output data used for the identification, as displayed in Figure 5.4 and 5.5, respectively. It is noted from the bottom panel of Figure 5.4 that negligible excitation is provided along the z -axis, which is a natural consequence of milling in the xy -plane. However, the measured

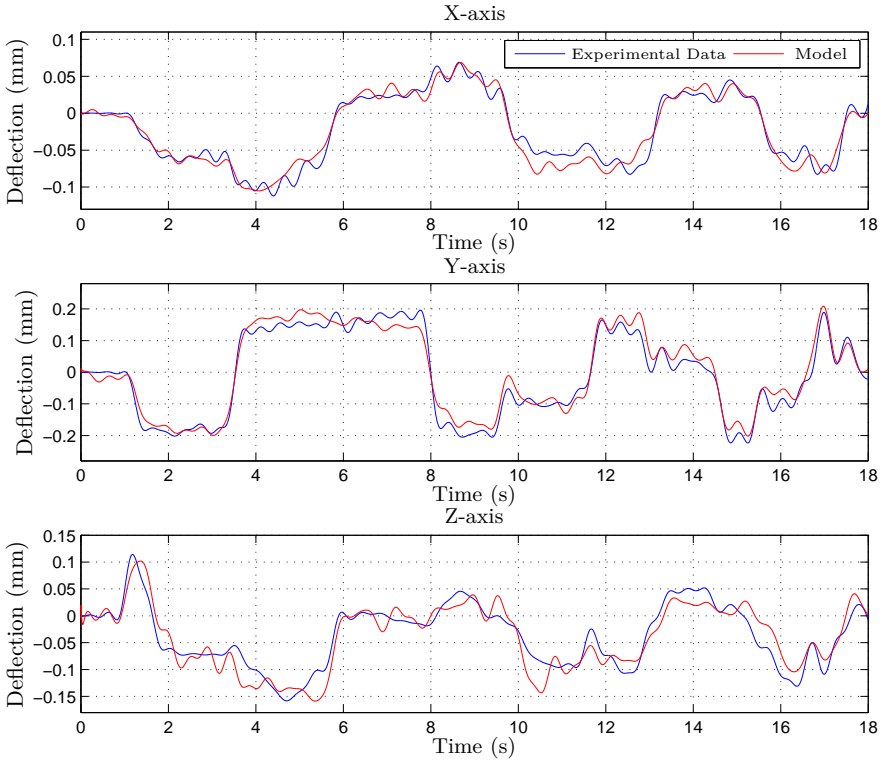


Figure 5.5 Measured deflections and model output (G_K) for all three Cartesian axes.

deflection in the z -direction is not negligible, indicating that the deflection is caused by the other force components. With the proposed MIMO dynamic modeling approach, higher performance can be achieved compared to the use of a SISO static models.

Force Estimation

As an alternative to using measurements from a force/torque sensor, a method for estimating the end-effector forces using the joint motor torques [Linderoth et al., 2013; Stolt, 2015] is considered here. Measurements of the joint motor torque for each joint cannot be directly accessed from the robot system, but the torque reference signal is available. Since the motor current loop is tightly controlled, the reference signal is assumed to sufficiently approximate measurements of the joint motor torque [Stolt, 2015].

The joint motor torque, denoted τ_m , is modeled as

$$\tau_m = \tau_g + \tau_{\text{dyn}} + \tau_{\text{ext}} + \tau_d, \quad (5.2)$$

where τ_g denotes torque caused by gravity, τ_{dyn} represents dynamic torque which appears when rapidly accelerating the robot, τ_{ext} is torque caused by external forces, and τ_d is torque originating from disturbances, such as friction, noise, and modeling errors. As mentioned previously in this section, because of the low accelerations that are used when performing milling, the dynamic torques are neglected. The external torques τ_{ext} , are assumed to originate from the external forces applied to the robot end effector, in this case the process forces that appear during milling, *i.e.*, the quantity that is to be estimated. The external torques and Cartesian forces are related as

$$\tau_{\text{ext}} = J^T(q)f \quad (5.3)$$

where $J(q)$ is the Jacobian matrix of the robot, and q the joint angles.

The disturbance torque τ_d is assumed to consist of two parts; friction torque τ_f and Gaussian measurement noise e :

$$\tau_d = \tau_f + e. \quad (5.4)$$

The friction torques mainly consist of Coulomb and viscous friction, and are velocity dependent. For high velocities, the Coulomb friction is modeled as a constant torque, whereas for low velocities the torque can vary significantly. It is therefore modeled as the outcome of a uniformly distributed random variable with a velocity-dependent range.

The gravity-compensated torque, denoted $\bar{\tau}$, is given as

$$\bar{\tau} = \tau_m - \tau_g = \tau_{\text{ext}} + \tau_d = J^T f + \tau_f + e. \quad (5.5)$$

Using this relation, the maximum likelihood estimate of the force is obtained by solving

$$\begin{aligned} & \underset{f, \tau_f}{\text{minimize}} && (\bar{\tau} - J^T f - \tau_f)^T R_e^{-1} (\bar{\tau} - J^T f - \tau_f) && (5.6) \\ & \text{subject to} && \tau_{f, \min} \leq \tau_f \leq \tau_{f, \max}, \end{aligned}$$

where R_e is the covariance matrix of e . The range of the uniform part of the disturbance torques is described by the inequality constraint. The optimization problem (5.6) is convex and of low order, and can therefore be solved in real-time. For further details on the force estimation, the reader is referred to [Stolt, 2015].

With this method, the estimated end-effector forces, denoted \hat{f} , can be used to identify the models G_M and G_K , without the need for a force sensor.

ILC Algorithms

In this section, the ILC algorithms are developed. First, in order to eliminate the kinematic, calibration, and load errors that occur when the robot is moving in free space, an initial free-space ILC method is employed. Then, three different ILC algorithms for milling are derived. The first algorithm uses external arm-side measurements in the workspace, the second uses measured end-effector force data, while the third relies on estimated force data.

Free-Space ILC As discussed in Section 5.2, a robot moving in free space is affected by kinematic errors, as well as errors caused by the end-effector load. Additionally, since the optical tracking system is used to evaluate the obtained accuracy for each of the methods, the quality of the relative calibration between the robot and tracking system is important. Since two of the proposed ILC methods are based on force data, which only provide relative data as opposed to the absolute position measurements that the tracking system provide, the position errors obtained in free space need to be eliminated in order to enable comparison of the methods. Thus, an ILC algorithm is developed in order to reduce the position errors in free space, using measurements from the optical tracking system.

The fundamental idea of the ILC algorithm in the context of robot position control is to improve the previous reference path sent to the robot system by adding a certain quantity to this path, based on the measured error in the previous execution. The update is performed such that the expected output in the next iteration is closer to the desired milling path. Here, this translates to updating the control input u_k from the current iteration, k , based on the position error as measured by the optical tracking system.

From the block diagram of the system describing the robot moving in free space, shown in Figure 5.2, the system is given by

$$y_{n,k} = G_R u_k + d_e, \quad (5.7)$$

where k denotes the iteration index, which is appended to all signals except the desired position r . With this relation, the model-based ILC algorithm, *e.g.*, described in [Norrlöf, 2000], is applied

$$u_{k+1} = u_k + QLe_k, \quad (5.8)$$

where the position error is defined as $e_k = r - y_{n,k}$, L is an approximate inverse of the transfer function from u_k to $y_{n,k}$, and Q is a low-pass or band-pass filter with appropriately chosen frequency properties. In particular, the choice of Q should be based on the desired frequency range for the ILC compensation.

The corrected control input, denoted u_c , as provided by the free-space ILC algorithm, is subsequently used as initial input u_0 for the milling ILC algorithms.

Arm-Side Measurement ILC In order to reduce the magnitude of the position errors during the milling task iteratively, the first approach proposed is to apply an ILC algorithm using the arm-side position measurements, provided by an optical tracking system.

Since the arm-side position of the robot is measured, the output $y_{n,k}$ is available in each iteration k . For this ILC structure, the complete milling model, as described by the block diagram in Figure 5.3, is given by

$$y_{n,k} = G_R u_k + G_K G_M y_{n,k}, \quad (5.9)$$

assuming $d_e = 0$, which may be reorganized into

$$y_{n,k} = (I - G_K G_M)^{-1} G_R u_k. \quad (5.10)$$

With this relation, the model-based ILC algorithm defined in (5.8) is applied, where $u_0 = u_c$ is used. In order to make the numerical computations more robust, the ILC update law is reformulated as follows

$$u_{k+1} = u_k + Q \tilde{L} (I - G_K G_M) e_k, \quad (5.11)$$

where \tilde{L} is chosen as an approximate inverse of G_R . The relation (5.11) is obtained by rewriting (5.8) and explicitly accounting for the structure of the system model (*cf.* relation in (5.9)), but avoiding the inversion of the term related to the milling process dynamics. If G_R contains non-minimum phase zeros, the inverse can, *e.g.*, be determined by mirroring these zeros in the unit circle prior to computing the inverse, see Figure 3.4 from Chapter 3.

In this chapter, the arm-side measurement ILC method is referred to as Method 1.

Force-Measurement ILC As shown in the previous paragraph, the assumption on explicit arm-side measurements led to a straightforward formulation of an ILC algorithm. Unfortunately, the sensors required for these kinds of measurements are expensive and not commonly available in the manufacturing industry today. Consequently, it is of interest to develop a method that is to some extent capable of estimating the arm-side measurements without such a sensor. As stated previously, the position errors that occur because of the milling process forces are not visible in the measurements of joint motor angles. Hence, to eliminate the need for arm-side measurements, an alternative is to use an additional sensor that (together with appropriate models) can estimate the arm-side position errors. A wrist-mounted force/torque sensor is a less expensive alternative to the optical tracking systems. Consequently, the proposed second approach is to construct a model-based estimation $\hat{y}_{n,k}$ of the position deviations using the joint-position resolver measurements and the force-sensor measurements according to

$$\hat{y}_{n,k} = y_{r,k} + G_K f_k, \quad (5.12)$$

to replace the arm-side position measurements y_n . Using this relation, the error signal used in the ILC algorithm is given by

$$e_k = r - \hat{y}_{n,k} = r - y_{r,k} - G_K f_k, \quad (5.13)$$

and the subsequent update law is then equivalent to (5.11).

In this chapter, the force-measurement ILC method is referred to as Method 2.

Force-Estimation ILC In addition to the two ILC methods for milling proposed in this chapter, which both require extra sensors, it is also desirable to develop a method that does not require additional sensors. Thus, an ILC algorithm that only uses measurements from sensors that already exist in the robot is developed, using the estimated end-effector forces obtained by the method stated earlier in this section. The algorithm is formed by substituting the measured forces f for the estimated forces \hat{f} in (5.13), and applying the ILC update law (5.11) with the reidentified models G_M and G_K .

In this chapter, the force-estimation ILC method is referred to as Method 3.

5.3 Experimental Setup

The experimental platform used to perform the milling experiments comprises an ABB IRB140 robot [ABB Robotics, 2015b] with an IRC5 controller, in combination with an open robot control extension called *ExtCtrl/ORCA* [Blomdell et al., 2010], running at 250 Hz. The proposed methods were implemented in MATLAB Simulink, where the simulation models were translated to C-code using *Real-Time Workshop* and compiled to be run on the extended robot system. In order to obtain task-space measurements of the tool position, a Nikon K600 optical tracking system [Nikon Metrology, 2010] was used. The tracking system had a specified absolute accuracy of approximately 50–75 μm for the used setup and measurement configuration. The robot was equipped with a wrist-mounted JR3 100M40A force/torque sensor [JR3, 2015], measuring forces and torques in the Cartesian directions. A Solectro UFM 1050 milling spindle [Solectro, 2015] with a maximum revolution speed of 22 000 rpm, equipped with a 6 mm diameter end mill with two teeth, was used.

The experimental setup is displayed in Figure 5.1, including the definition of the fixed world Cartesian coordinate system used throughout the experiments. As seen in Figure 5.1, the milling spindle was attached to the robot and the aluminum workpiece (type Al7075) was rigidly fixed in the robot workspace, such that the workpiece was aligned with the horizontal plane of the robot coordinate system. As seen in the figure, three LED units

were attached to the milling spindle in order to enable the tracking system to measure the tool position. The specific milling task considered in this chapter was to mill a 28×28 mm pocket with a 2 mm depth of cut; the desired milling path used for the experiments is displayed in Figure 5.6. A feed rate of 10 mm/s was used for all experiments. With the described experimental setup, the available measurement signals for each of the Cartesian axes were:

- y_r : estimated position of the robot tool, computed from the joint motor angle measurements using forward kinematics;
- y_n : arm-side position of the robot tool measured with the optical tracking system;
- f : force acting on the tool, measured by the wrist-mounted force/torque sensor.
- τ_m : joint motor torque reference for each of the six robot joints.

5.4 Experimental Results

The milling experiments were performed using the setup described in the previous section. In this section, the results obtained using the proposed ILC algorithms are described separately, followed by a comparison of the results.

Free-Space ILC

Prior to performing the milling experiments, the free-space ILC algorithm was run in order to reduce the influence of kinematic and calibration errors. After a single iteration, the MAE was found to be within the measurement accuracy of the optical tracking system. The MAE for iteration k is defined as

$$\text{MAE} = \frac{1}{N} \sum_{j=1}^N |e_{k,j}|, \quad (5.14)$$

where j is the sample index, $e_{k,j}$ is the tracking error, and N is the number of data points.

With the corrected control input obtained from the free-space ILC, an initial pocket milling was performed, serving as iteration 0 for all milling experiments using Methods 1–3.

Method 1

Using position measurements of the robot end effector, acquired by the optical tracking system described previously, four ILC iterations of milling were performed using Method 1. The desired path, as well as the milling

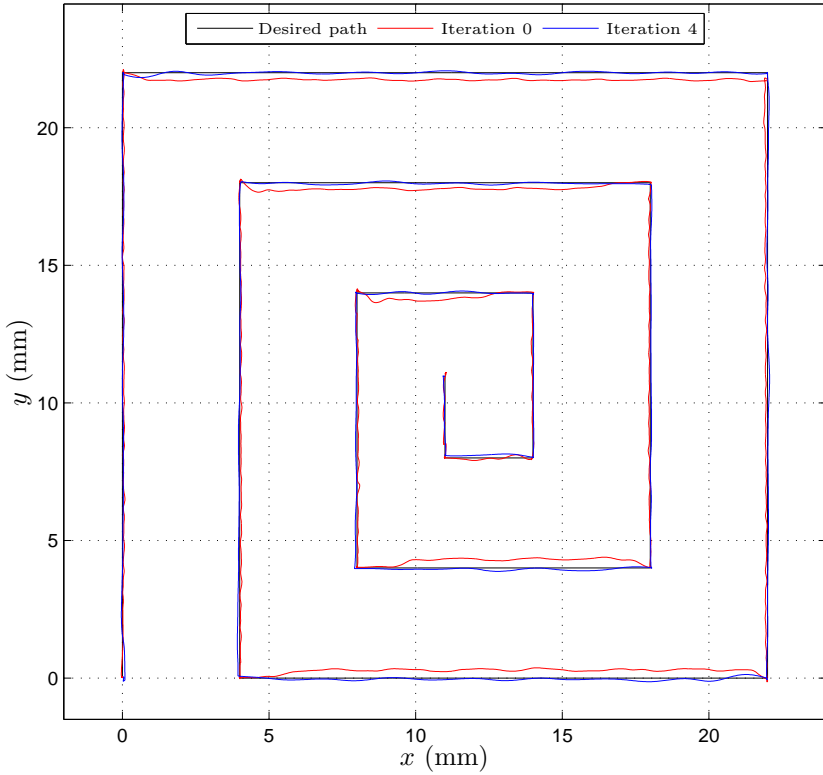


Figure 5.6 The desired and measured milling paths for iterations 0 and 4, using Method 1.

paths traversed in iteration 0 and 4, are displayed in Figure 5.6. It is noted that the stiffness is significantly higher in the x -direction as compared to the y -direction, since substantial deflections are mostly visible in the y -direction.

The evolution of the MAE of the obtained position errors along each Cartesian axis is displayed in Figure 5.7. It is observed from Figure 5.7 that the most significant error reduction is obtained already in the first iteration. This is expected with the use of a model-based ILC algorithm, given that the process model is accurate. An additional decrease in the MAE was obtained in the z -direction for iterations 2–4, possibly suggesting that the z -component of the process model could be improved. Nevertheless, the obtained position error is within the accuracy of the optical tracking system after four iterations.

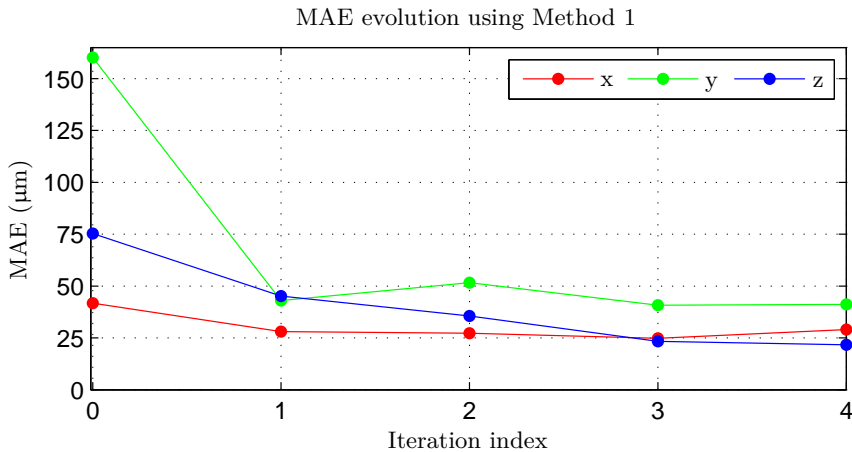


Figure 5.7 MAE for each ILC iteration and Cartesian direction, obtained using the arm-side measurement ILC algorithm (Method 1).

Method 2

With the force sensor introduced in Section 5.3, which provides measurements of the process forces affecting the robot end effector, four ILC iterations of milling were performed using Method 2. For evaluation purposes, the optical tracking system was used to measure the position errors.

The evolution of the MAE of the obtained position errors along each Cartesian axis is displayed in Figure 5.8. Similar to the results from using Method 1, the largest reduction in position error is achieved already in the first iteration. For the subsequent iterations, no significant additional decrease in the error is observed. However, the error appears to have converged, with an expected minor variation because of measurement noise. The results obtained with this approach are surprisingly close to that of Method 1, given the relevance of the available sensor data. The method clearly shows its capability of estimating and compensating the position deflections arising as a result of the process forces.

Method 3

Using the force-estimation method described in Section 5.2, which provides an estimate of the process forces affecting the robot end effector, four ILC iterations of milling were performed using Method 3. The experiments were evaluated using measurements from the optical tracking system.

Prior to performing the first ILC iteration, the performance of the force-estimation algorithm was evaluated. First, an experiment where the robot was moved throughout the major part of its workspace was performed,

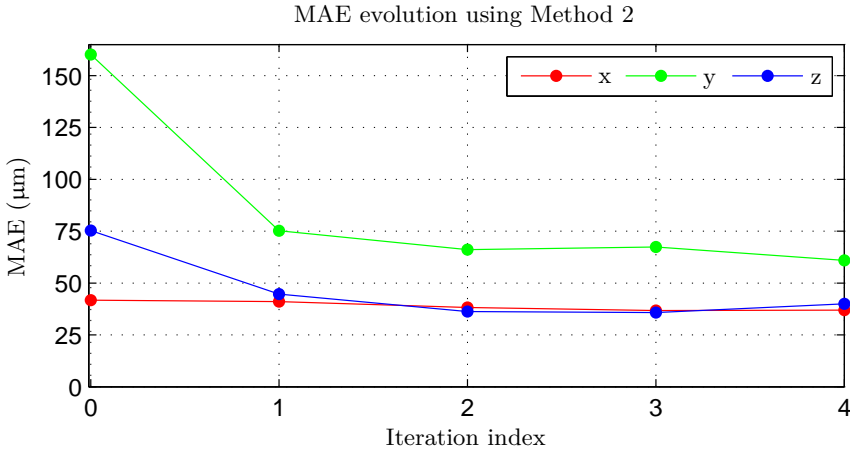


Figure 5.8 MAE for each ILC iteration and Cartesian direction, obtained using the force measurement ILC algorithm (Method 2).

in order to estimate the gravity torque τ_g and the friction for each joint. With these properties known, the end-effector forces could be estimated. Figure 5.9 displays the measured and the estimated forces in each Cartesian direction. It is noted that the estimated forces exhibit satisfactory correspondence to the measured forces in the x - and y -directions. In the z -direction, the estimate of the force is slightly less accurate. However, as the milling is performed in the xy -plane, only minor deflections are expected in the z -direction. Also, the force-estimation errors are to some extent assumed to be captured by the reidentified process models of the milling.

The evolution of the MAE of the obtained position errors along each Cartesian axis is displayed in Figure 5.10. As for Methods 1 and 2, the position error is reduced primarily in the first iteration. It is observed from Figure 5.10, that an increased fluctuation in the error in each iteration, compared to Methods 1 and 2, was obtained. This is not unexpected as the force is estimated from noisy motor torques. Nevertheless, a significant reduction of the position error was achieved using Method 3.

Comparison of Results

In order to quantify the results and compare the proposed methods, the evolution of the Euclidean norm of the MAE values for each method is displayed in Figure 5.11. Additionally, the MAE norm numbers for iterations 0 and 4 are presented in Table 5.1. Furthermore, the norm of the position error for the complete milling experiment using the proposed methods, is displayed in Figure 5.12. It is noted that the errors still exhibit

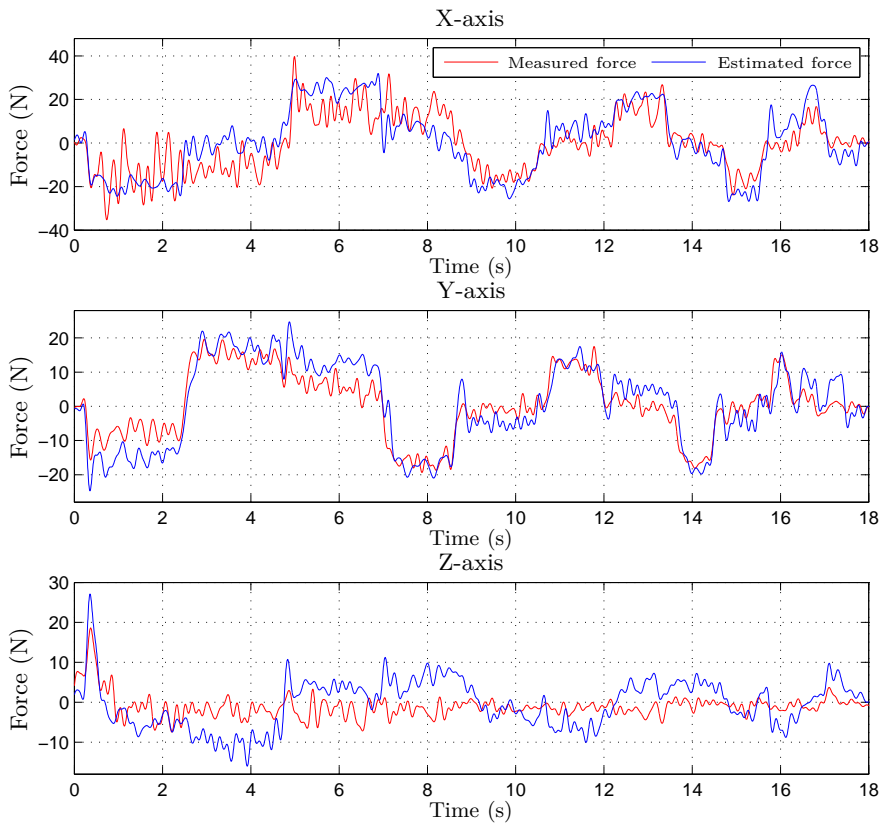


Figure 5.9 Force-estimation performance during a milling experiment.

Table 5.1 MAE norm of position errors, given in μm .

	Method 1	Method 2	Method 3
Iteration 0	180.1	180.1	180.1
Iteration 4	54.9	81.7	110.7
Ratio	3.3	2.2	1.6

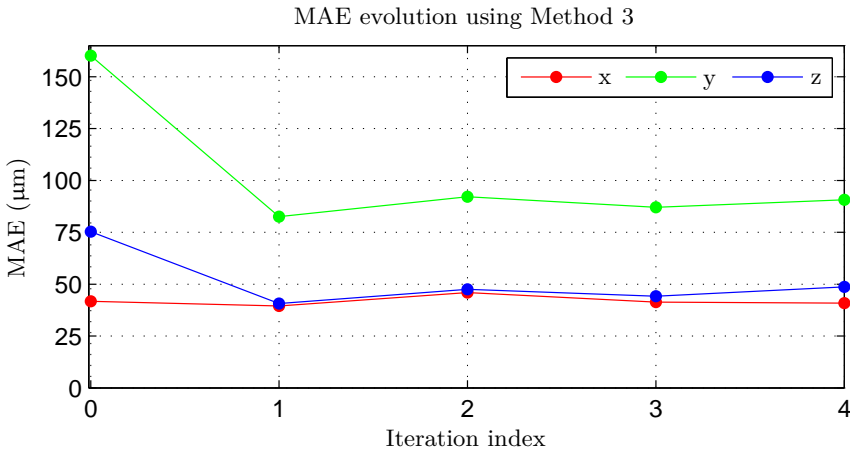


Figure 5.10 MAE for each ILC iteration and Cartesian direction, obtained using the force-estimation ILC algorithm (Method 3).

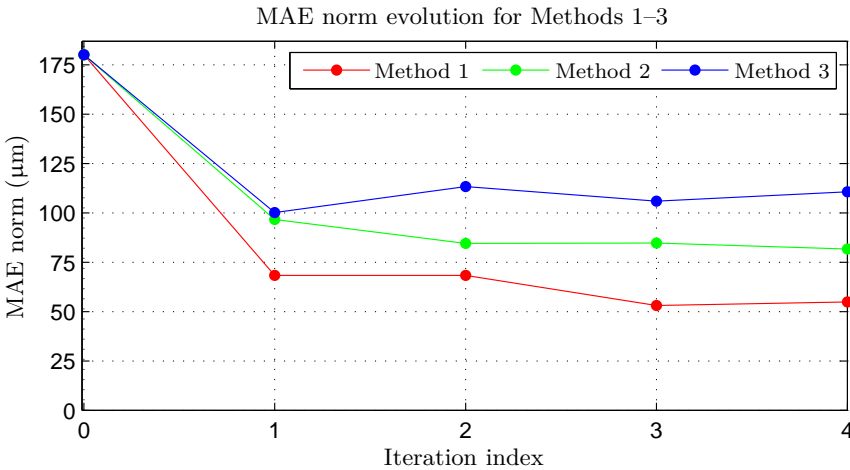


Figure 5.11 Summary of results using Methods 1–3, displaying the Euclidean norm of the MAE values for each ILC iteration k .

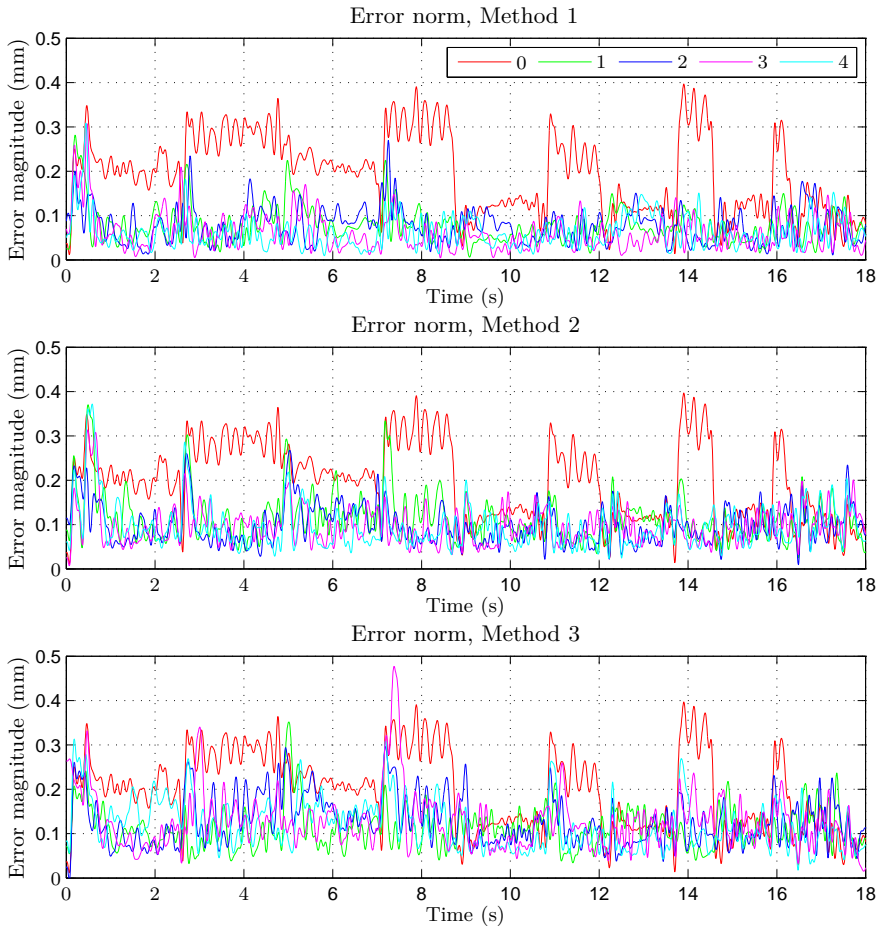


Figure 5.12 Error norm plots for each iteration k , for all three ILC methods.

some high-frequency content after the ILC iterations. This is most likely because the remaining frequencies are above the available compensation bandwidth, limited by the cut-off frequency of the low-pass filter Q , which is set to approximately 10 Hz in this chapter.

In order to briefly investigate the error reduction capabilities of the proposed methods in more demanding experiments, additional pocket milling experiments were performed, with a 4 mm depth of cut. As seen in Table 5.1, Method 1 resulted in the largest error reduction, and was therefore chosen for these experiments. The desired path, as well as the paths obtained in the

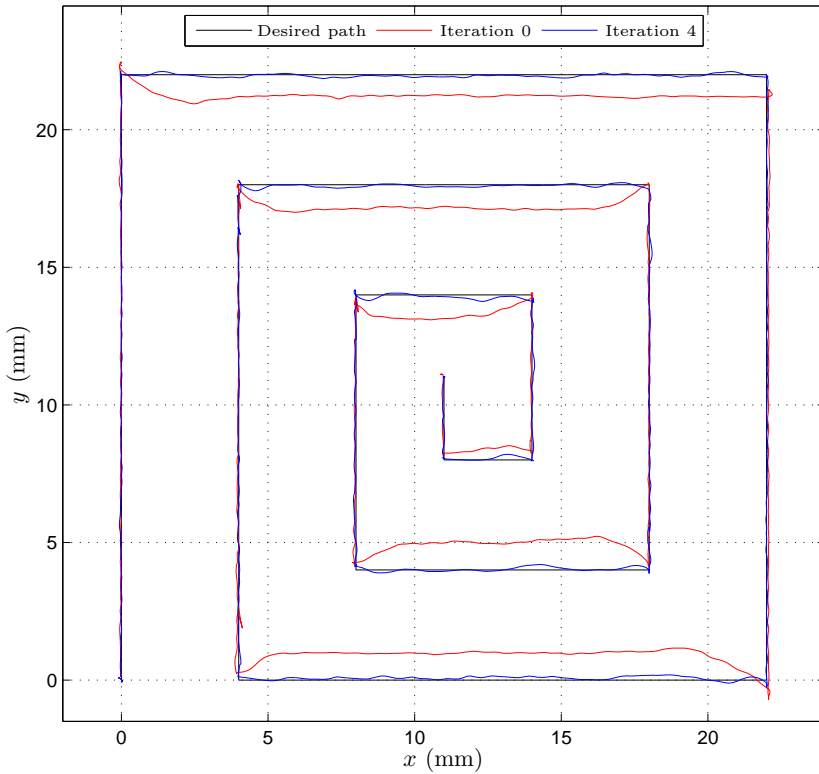


Figure 5.13 The desired and measured milling paths for iterations 0 and 4, using Method 1.

initial iteration and iteration 4, are displayed in Figure 5.13. As expected, the position deflections are significantly larger, *cf.* Figure 5.6. The evolution of the MAE over the ILC iterations is shown in Figure 5.14. Calculating the MAE norm for iterations 0 and 4, an error reduction by a factor of approximately six was achieved for the deep-cut experiments.

5.5 Discussion

As an alternative to the approaches presented in previous chapters, three different ILC methods for increasing the absolute accuracy of robotic machining tasks were proposed and experimentally verified. As mentioned in Section 5.1, the batch-oriented nature of modern production induces significant repetitiveness in the tasks to be performed, thus enabling the

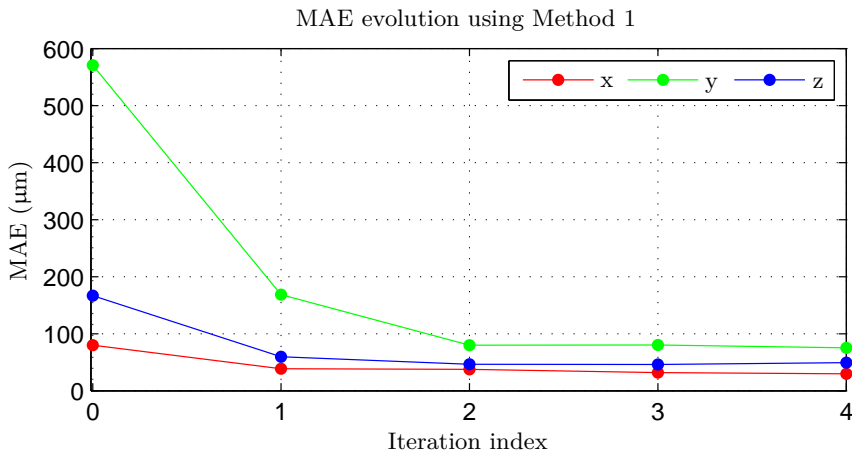


Figure 5.14 MAE for each ILC iteration and Cartesian direction, obtained using the arm-side measurement ILC algorithm (Method 1).

possibility of improving the performance of the task from workpiece to workpiece. In addition, the use of model-based ILC enables fast convergence rates, thus minimizing the amount of sub-standard pieces produced during the learning process. All of the proposed ILC methods resulted in a significantly increased accuracy. However, the accuracy has not reached the performance offered by CNC machines or that achieved with approaches using additional hardware, such as the macro/micro configuration presented in Chapter 2. Still, all proposed methods do improve the accuracy; decreases of approximately 70%, 55%, and 38% of the uncompensated error were achieved using Methods 1–3, respectively. Additionally, in a more demanding milling experiment using Method 1, the error was reduced by 84%. Hence, the results obtained by milling with industrial robots can be improved without the addition of extra hardware, as shown by the use of force estimation in Method 3. At an extra cost, a force sensor could be added to the robot system, making it possible to reduce the position error further.

With the arm-side measurement ILC algorithm, the measurement accuracy of the optical tracking system provides a lower limit on the accuracy that can be achieved in the milling. Thus, since an MAE norm of approximately 55 μm was reached, the algorithm can be considered successful.

A limitation of the force-based approaches is the inability to eliminate kinematic and calibration errors, that are also present in the free-space motion. This was, however, solved by employing a free-space ILC algorithm prior to performing the milling. Kinematic calibration methods might also be

used to correct these errors, see for instance [Chen et al., 2008]. Furthermore, the force-based approaches are limited by the accuracy of the deflection model, sensor noise, and force-estimation errors for Method 3. Also, to some extent, individual variations in the workpieces used in the experiment and the effect of the wear of the tool also introduced certain non-repetitive errors and disturbances that the ILC algorithm cannot compensate. This is however a problem for all ILC methods, even with arm-side measurements of the robot position. Another limitation of the force-based ILC methods is that the optical tracking system is needed in order to identify the dynamic models, as well as performing the free-space ILC iterations. However, this procedure only needs to be performed once for each new part, making it possible to rent a tracking system for this procedure, instead of purchasing a permanent system.

A natural question for the future applicability of the method is the generality of the models required for the ILC algorithms. The deflection model G_K obviously depends on the robot in use and the specific configuration, but given that the same limited workspace is used for all milling operations, it can be considered independent of the path traversed in the milling. Hence, a modeling effort should be carried out in order to characterize the robot in the configuration used for the milling operations, but the identified model could be used for multiple machining tasks in a limited workspace. For machining tasks that require larger workspaces, it could be of interest to incorporate joint-based stiffness models for the compensation, *e.g.*, utilizing the models developed in Chapter 4.

Contrary to the deflection model, the process-force model G_M does depend on the milling conditions, so different models are required for different tasks. In this chapter, a data-driven modeling approach was used. However, different models available in the machining literature relating the process forces with the milling conditions could be used as well, see, *e.g.*, [Grote and Antonsson, 2009]. Furthermore, it would be beneficial not only to compensate the Cartesian position errors but the possible orientation errors as well.

5.6 Conclusions

Three different model-based ILC methods were presented, with the aim of reducing the position errors in machining tasks with robots. The first method was based on position measurements of the arm-side of the robot in task space, the second method used process-force measurements, and the third used joint motor torques to estimate the process forces. The experimental results obtained from milling experiments in aluminum showed a significant improvement in position accuracy using the algorithm based on arm-side position measurements. An absolute error decrease of up to

84% was obtained with this algorithm. In turn, the force-based and the force-estimation approaches were capable of decreasing the error by 55% and 38%, respectively. Although the optical tracking system is needed for the initial calibration, significant savings can be achieved with the use of the second and third method, because of the lower investment cost in sensors.

6

Frequency-Domain Iterative Learning Control

6.1 Introduction

This chapter is based on the manuscript [Sörnmo et al., 2015a].

In this chapter, further investigation and application of ILC methods are provided, although a different kind of manipulator is considered. The considered manipulator is a *marine vibrator*, which is used to manipulate the air or water pressure around its shell, by precise position control of the shell. In turn, this can be used for marine seismic acquisition, *i.e.*, mapping of the ocean floor. To this purpose, the use of impulsive sources, such as airguns or explosives, have been dominating for several decades. However, recent developments of marine vibrators offer numerous advantages, if the device is controlled properly. Marine vibrators are environmentally friendly compared to impulsive sources, because of the reduced peak source strength in terms of pressure. The frequency content can be controlled such that there is less impact on marine mammals outside the seismic band. The frequency range of interest for the signals that are to be emitted from the marine vibrator is approximately 5–100 Hz. It is crucial that only these frequencies are emitted, as higher frequencies may interfere with marine animal life [LGL and MAI, 2011]. However, since marine vibrators are advanced mechanical systems, nonlinear effects and high-frequency harmonics are likely to be present. The marine vibrator considered in this chapter has been found to produce high-energy overtones, which need to be attenuated in order to fulfill the specifications. The contribution of this chapter is the modeling and control of such a marine vibrator, to the purpose of overtone attenuation and reference tracking. A model-based iterative solution is presented, based on frequency-domain ILC, considering both SISO and MIMO cases. Additionally, a method for adaptive reidentification of the vibrator model for diverging frequencies is presented.

Previous Research

The mechanical design of marine vibrators, including the driving means and measurement systems, has thoroughly been researched in, *e.g.*, [Graydon and Delbert, 1969; Tenganhn, 2006; Tenganhn, 2009]. The aspect of precise position control of marine vibrators is not as well represented in the literature. In [Tenganhn, 2011], a time-domain ILC scheme for vibrator position control is proposed. However, no results are presented.

For the application considered in this chapter, frequency-domain ILC was found to be immensely superior to time-domain ILC, see Section 6.7. Convergence properties of ILC algorithms in the frequency domain are analyzed in [Goh, 1994; Norrlöf and Gunnarsson, 2002]. A set of different applications of the frequency-domain inversion-based ILC approaches have been considered previously; nanopositioning applications are developed in [Tien et al., 2005; Yan et al., 2012] and acoustic noise reduction is investigated in [Waite et al., 2008].

A modeling-free inversion-based ILC algorithm is presented in [Kim and Zou, 2008; Kim and Zou, 2013], where the frequency-domain input/output relation is used in each iteration to eliminate model dependence. The algorithm was applied to a nanofabrication scenario in [Yan et al., 2009]. This method is, however, not applicable to the scenario considered in this chapter, as sufficient excitation is not provided in the complete frequency compensation range. Instead, an adaptive reidentification algorithm is developed in this chapter, in order to cope with process variations.

Disposition

This chapter is organized as follows. Section 6.2 describes the objectives of marine seismic acquisition, followed by a description of the hardware in Section 6.3. Control-system design principles are developed in Section 6.4, and modeling and control design are presented in Section 6.5. Results from experiments performed in air are presented in Section 6.6, followed by a comparison to time-domain ILC in Section 6.7. Finally, the method and experimental results are discussed, and conclusions are drawn in Section 6.8.

6.2 Marine Seismic Acquisition

Reflection seismology is used by petroleum geologists and geophysicists to map and interpret potential petroleum reservoirs. Oil and gas explorers use seismic surveys to produce detailed images of the various rock types and their location beneath the sea floor. This information is then used to determine the location and size of oil and gas reservoirs. The size and scale of seismic surveys have increased alongside the significant increases in computer power during the last 30 years. This has led the seismic industry from

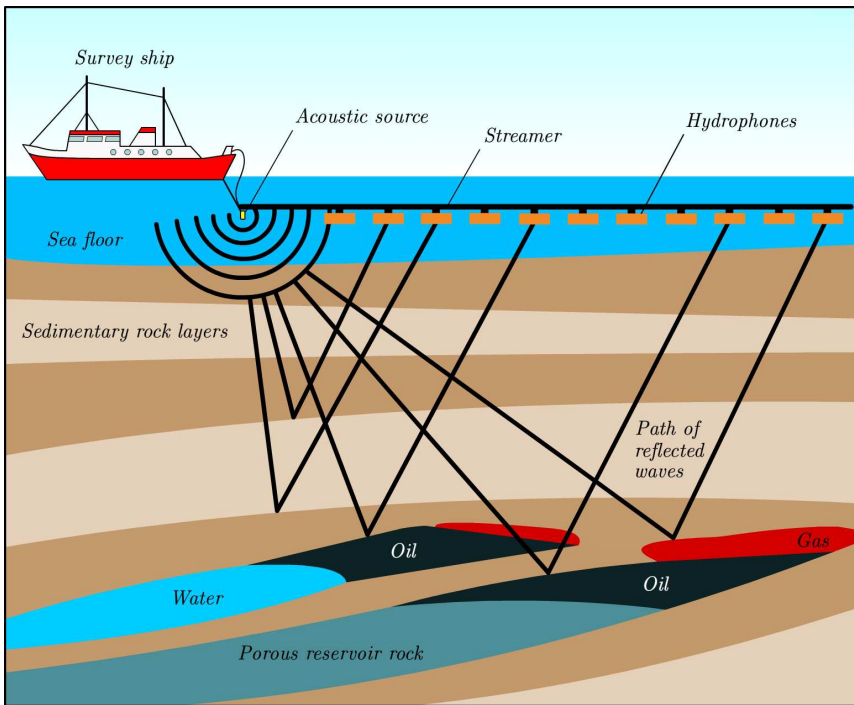


Figure 6.1 Illustration of the marine seismic acquisition principle, adapted from [KrisEnergy, 2015].

laboriously, and therefore rarely, acquiring small 3D surveys in the 1980s, to now routinely acquiring large-scale high resolution 3D surveys. The goals and basic principles have remained the same, but the methods have slightly changed over the years. Traditional marine seismic surveys are conducted using specially-equipped vessels that tow one or more cables containing a series of hydrophones located at constant intervals, see Figure 6.1. The cables are known as streamers, with 2D surveys using only one streamer and 3D surveys employing up to twelve or more. The streamers are deployed just beneath the water surface, at a fixed distance from the vessel. The seismic source, traditionally an impulsive source or an array of impulsive sources, is also deployed beneath the water surface and is located between the vessel and the first receiver. Marine seismic surveys generate a significant quantity of data, since each streamer can be more than 8 km long, containing hundreds of channels, and the seismic source is typically fired every 10 or 20 seconds.

6.3 Marine Vibrators

The considered marine vibrator, in this chapter also referred to as a *transducer*, uses a *flexensional shell* to facilitate the low-frequency application, with an intended frequency range of 20–100 Hz. The transducer drivers are based on electrical coils operating in a magnetic field and spring elements that transfer the force from the electrical driver to the flexensional shell, see Figure 6.2. The input to the marine vibrator is thus the driving current to the electrical coils, which displace the shell so that acoustic signals, *i.e.*, the output of the system, are generated. Methods on how to measure the output are discussed later in this section.

Low-frequency sources face the problem of poor efficiency if a good impedance match with the surrounding water cannot be achieved. Straightforward calculations on the radiation from a vibrating piston with a radius of 0.3 m and a source level of 195 dB (relative to 1 μPa), yields 0.074% efficiency at 10 Hz. The same piston has an efficiency of 99.9% at 10 kHz [Kinsler et al., 1999]. This leads to differentiator-type characteristics for the low-frequency dynamics of the system.

The mechanical construction of the considered marine vibrator exhibits two resonances; the lowest resonance originates from the shell interacting with the equivalent fluid mass and the second resonance from the spring elements with a resonance frequency in the upper frequency band. Having two resonances separated in the frequency band of interest makes it possible to achieve high efficiency. Additional details on the mechanical design of the vibrator can be found in [Tenghamn, 2006].

A marine vibrator offers environmental advantages over impulsive sources. Given the high-profile environmental discussions on output peak power of seismic sources, vibrator technology offers a superior solution by spreading the energy in time; thus reducing the acoustic peak power. Additionally, the fact that a marine vibrator can generate arbitrary signals makes it useful for various types of spread-spectrum signals that can reduce the environmental impact even further.

Sensors

A variety of sensors may be used to acquire information about the signal emitted by the vibrator, such as

- Accelerometers mounted on different locations on the vibrator.
- Sensors measuring displacement of different vibrator locations relative to a reference position.
- Microphones mounted in the air-filled volume inside the vibrator shell.

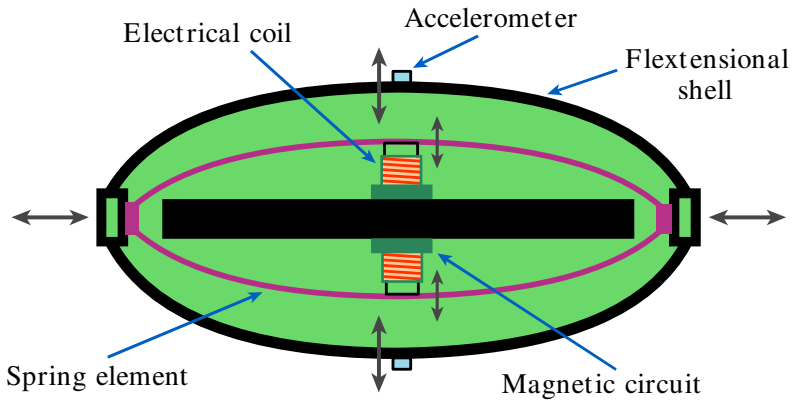


Figure 6.2 Schematic drawing of the transducer actuation principle. The two flextensional shells are moved by the electrical coils via a construction of springs. This spring construction creates resonances at suitable frequencies, in order to improve the impedance match to the water, thus increasing the vibrator efficiency for low frequencies.

- Hydrophones mounted outside the vibrator, either close to the vibrator (near-field measurement) or at a distance from the vibrator (far-field measurement).

All mentioned sensors have advantages and disadvantages. Accelerometers give the possibility to measure acceleration at discrete points on the vibrator with good signal-to-noise ratio (SNR). The bandwidth of these sensors is high enough to provide input to the ILC algorithm over the complete frequency region of interest. Accelerometers measure movements of the vibrator itself, while the quantity of primary interest is the acoustic sound pressure level generated by the vibrator. Hydrophones may be used to measure the sound pressure in the water, but the measurement task is complicated by the fact that the vibrator is often operated in a fairly reverberant environment where multi-path contributions make the received signal more complex. Microphones may also be placed inside the air-filled vibrator cavity, but as in the hydrophone case internal reflections and different propagation paths disturb the sound pressure signal.

For proof of concept, all experiments described in this chapter were performed in air. In the experiments, two accelerometers were used to monitor the vibrator motion, mounted centered on the vibrator shell, one on each side, as seen in Figures 6.2 and 6.3. These positions were chosen after extensive experimentation. Tests were also performed with microphones mounted inside and outside the vibrator. Multi-path contributions complicated the

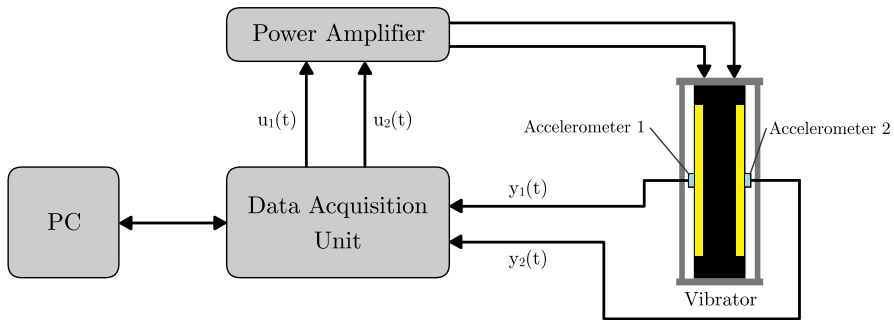


Figure 6.3 An overview of the marine vibrator measurement system.

measurements and data extraction, therefore no successful ILC results were obtained with microphones. The main focus of the experiments has therefore been to obtain a stable ILC, using accelerometers mounted on the vibrator shells.

Measurement System

Figure 6.3 shows an overview of the utilized measurement system. The system consists of:

- Two accelerometers (with pre-amplifiers).
- A data acquisition unit, used to sample the accelerometer signals and to generate control signals.
- A power amplifier to drive the coils in the vibrator.
- A personal computer for data retrieval and processing.
- Cabling for distribution of control and accelerometer signals to/from the vibrator.

The accelerometers were mounted in aluminum housings, together with pre-amplifiers, which were attached to the vibrator shell. High drive currents were fed into the driver coils during operation. These may disturb the inputs of the data acquisition system, especially since long cables were used between the acquisition system and the vibrator. In order to obtain the desired SNR in the accelerometer signals, dedicated pre-amplifiers with balanced outputs were mounted in close vicinity of the accelerometers to minimize common-mode interference from the driver side of the system. The harmonic distortion of the sensor should preferably be an order of magnitude lower than the maximum acceptable harmonic distortion in the vibrator output. The sensor spectral noise floor should also preferably be an order of magnitude

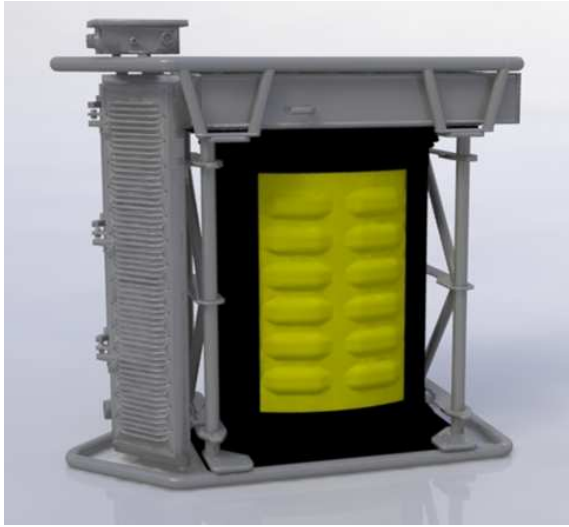


Figure 6.4 3D rendering of the marine vibrator. The device is approximately two meters tall.

lower than the corresponding maximum acceptable overtone output from the vibrator.

An audio data acquisition system of type Fireface UCX [RME-Audio, 2015] was used to sample the accelerometer signals and to generate the control signals. Both inputs and outputs on the Fireface were balanced. A sampling frequency of 44.1 kHz was used to sample the two accelerometer outputs. Two analog outputs of the Fireface system were used to generate control signals to the vibrator. The control signals were amplified in a LAB4000 power amplifier which drove the vibrator coils. A standard personal computer was used as interface to the Fireface.

Specifications

The objective of the control of the marine vibrator is to emit a well-defined acoustic signal, *i.e.*, to produce an acoustic output that matches a desired reference signal with high accuracy. The reference signal is typically of low frequency. The design specification considered in this chapter states that tracking errors above 100 Hz should be attenuated to a spectral density level 40 dB below that of the desired reference signal. In this chapter, measures of dB are given by

$$A_{dB} = 20 \log_{10} \left(\frac{Y}{Y_0} \right) \text{ dB}, \quad (6.1)$$

where A_{dB} is the amplitude in dB of the ratio of the signals Y and Y_0 .

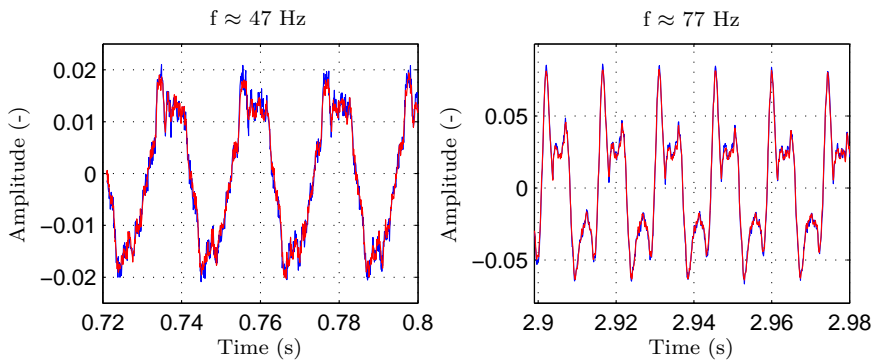


Figure 6.5 Two segments of the transducer output for an experiment performed twice, with identical linear chirp signals as input. Experiment 1 and 2 are shown in blue and red, respectively. Note the high repeatability between the experiments, motivating the choice of ILC as control strategy.

6.4 Control System Design Principles

An initial test of one side of the transducer was conducted, by subsequently performing the same experiment twice using a linear chirp signal as input, *i.e.*, a sinusoid with frequency linearly increasing with respect to time. Two different segments of the system response from both experiments are displayed in Figure 6.5. It is evident that significant nonlinear properties are present in the system, since the output does not closely resemble a sine wave, especially around 77 Hz. Despite this, the responses show a highly repeatable behavior, as the outputs from both experiments match closely. Furthermore, an estimated frequency spectrum of the data, shown in Figure 6.6, reveals complex dynamics with numerous resonance peaks and notches.

Because of these defining properties, ILC is considered to be an appropriate control approach for this application, as identical reference signals are to be used repeatedly and the dynamics was found to be repeatable. Specifically, frequency-domain ILC was found to be highly advantageous over traditional time-domain ILC. For further details on this, the reader is referred to Section 6.7.

6.5 Modeling and Control Design

In this section, the procedure of characterizing and obtaining a model of the marine vibrator is considered. The subsequent ILC design for the SISO and MIMO cases is then described. Furthermore, a method for adaptive reidentification of the vibrator model is developed.

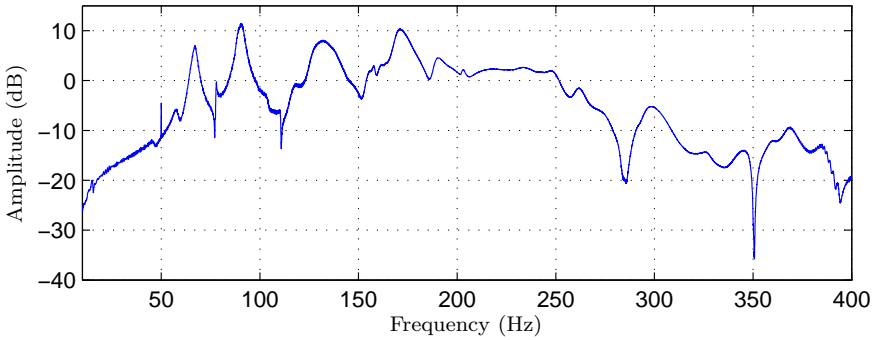


Figure 6.6 Estimated frequency spectrum of the transducer response to a linear chirp signal. The spectrum indicates a high-dimensional system, making frequency-domain ILC an attractive alternative.

System Characterization

In order to ensure convergence of the ILC algorithm, it is necessary to have an accurate estimate of the system transfer function [Norrlöf and Gunnarsson, 2002]. An initial estimate of the dynamics can be achieved by using a linear chirp in the frequency range 5–1000 Hz as input. However, since the accuracy of the estimated transfer function is critical, single frequency sine-waves were instead used for a discrete set of frequencies, as a higher number of periods are used in the estimation for each frequency to obtain improved accuracy.

It was found that experiments performed on an unloaded transducer exhibited considerably different dynamics, especially close to resonances and notches, than for a preloaded transducer. Since the control will be performed on a preloaded transducer, the identification must also be performed on the same. Thus, in order to identify one of the sides of the transducer, the input signal

$$u_1(n) = A \sin(2\pi f_i^0 t_s n) + B \sin(2\pi f_i t_s n) \quad (6.2)$$

is used, where n is the sample index, t_s is the sample period, and the exciting frequency of interest is denoted f_i . The preloading frequency, denoted f_i^0 , is chosen such that its harmonics do not coincide with the frequency of interest, f_i . The non-parametric estimate of the transfer function for the linear dynamics of one side of the transducer is then calculated as

$$G(f_i) = \frac{Y(f_i)}{U(f_i)}, \quad (6.3)$$

with a tapered cosine window (also known as a Tukey window) [Harris, 1978] applied to the measured data, and the discrete Fourier transform (DFT) to obtain $U(f)$ and $Y(f)$ from the sampled signals $u(n)$ and $y(n)$. The DFT for

calculating $U(f)$ is defined as [Mitra and Kuo, 2006]

$$U(f) = \sum_{n=0}^{N-1} u(n)e^{-i2\pi f \frac{n}{N}} \quad (6.4)$$

where N is the number of available samples of $u(n)$. The DFT of $y(n)$ is performed analogously.

In order to be able to perform ILC on both sides of the transducer, an estimate of the MIMO transfer matrix is required. This can be acquired by performing two separate identifications; one for each input of the transducer. For identification of the first transducer side, the input signals are given by

$$\begin{bmatrix} u_1(n) \\ u_2(n) \end{bmatrix} = \begin{bmatrix} A \sin 2\pi f_i^0 t_s n + B \sin 2\pi f_i t_s n \\ A \sin 2\pi f_i^0 t_s n \end{bmatrix}. \quad (6.5)$$

Identification of the second side is performed analogously, with permuted input signals. The obtained responses from the two identification experiments are then used to form the complete transfer function matrix, denoted $G(f)$:

$$G(f_i) = \begin{bmatrix} \frac{Y_1(f_i)}{U_1(f_i)} & \frac{Y_1(f_i)}{U_2(f_i)} \\ \frac{Y_2(f_i)}{U_1(f_i)} & \frac{Y_2(f_i)}{U_2(f_i)} \end{bmatrix} = \begin{bmatrix} G_{11}(f_i) & G_{12}(f_i) \\ G_{21}(f_i) & G_{22}(f_i) \end{bmatrix}. \quad (6.6)$$

SISO Control

For systems that are used to track the same reference repeatedly, ILC is an appealing control strategy, given that the disturbances affecting the system are repeatable and that the system exhibits time-invariant dynamics. The algorithm makes use of the tracking error obtained in the current iteration to correct the control input for the next iteration.

Consider a discrete-time system on the form

$$y(n) = G(q)u(n) + f(u(n)) + v(n), \quad (6.7)$$

where $u(n)$ is the input, $y(n)$ is the output, $v(n)$ is white noise, $f(\cdot)$ is the nonlinear part of the dynamics, and $G(q)$ is a filter for the linear part of the dynamics, expressed in the difference operator q . Given a reference $r(n)$ that is to be tracked, the model-based discrete-time ILC update law is stated as [Arimoto et al., 1984; Norrlöf, 2000]

$$u_{k+1}(n) = Q_1(q)u_k(n) + Q_2(q)\widehat{G}^{-1}(q)e_k(n), \quad (6.8)$$

where k is the iteration index, $e_k(n) = r(n) - y_k(n)$ is the tracking error in the current iteration k , and \widehat{G}^{-1} is an approximation of the inverse process

dynamics. Furthermore, $Q_1(q)$ and $Q_2(q)$ are user-defined filters, commonly chosen as low-pass or band-pass filters, used to define the frequency region where the ILC compensation should be active. Also, the magnitude of $Q_2(q)$ determines the learning rate. The frequency-domain counterpart of the ILC update law is consequently written as

$$U_{k+1}(f) = Q_1(f)U_k(f) + Q_2(f)\widehat{G}^{-1}(f)E_k(f) \quad (6.9)$$

where the filters $Q_1(f)$ and $Q_2(f)$ need not necessarily be transformed from continuous-time, but can advantageously be designed directly in the frequency domain. In this chapter, filter coefficients were chosen as

$$Q_1(f) = 1, \quad \forall f \quad (6.10)$$

$$Q_2(f, k) = \begin{cases} K(k), & f \in f_{ILC} \\ 0, & f \notin f_{ILC} \end{cases} \quad (6.11)$$

where f_{ILC} is the desired set of frequencies for which the ILC algorithm should be active, and $K(k) \in [0, 1]$ is an iteration-index dependent gain; thus the added argument k of Q_2 . With this choice of Q_1 , the control input generated by the ILC algorithm is not limited in frequency. Also, with the proposed iteration-index dependent gain, fast convergence can be achieved with a high value of $K(k)$ for small k , followed by increased accuracy and robustness to process drift by lowering $K(k)$ for higher k . For the SISO case, the choice of $Q_2(f)$ must satisfy the criterion

$$\sup_f |I - G(f)\widehat{G}^{-1}(f)Q_2(f)| < 1 \quad (6.12)$$

in order for the ILC algorithm to converge [Norrlöf and Gunnarsson, 2002].

Special Case — Single-Frequency Reference The special case in which the system is only required to track a single-frequency reference, *i.e.*, a sine-wave, is considered in this paragraph. It is assumed that, given a sinusoidal input with root frequency f_0 , the system only produces significant signal energy at this frequency and its overtones, which are defined as

$$f_n = (n + 1)f_0, \quad n = 1, 2, 3, \dots \quad (6.13)$$

Additionally, it is assumed that the frequency f_0 is a multiple of the fundamental discrete frequency, in order to avoid spectral leakage. The fundamental discrete frequency f_γ is defined as

$$f_\gamma = \frac{1}{2T}, \quad (6.14)$$

where T is the duration of the experiment. Then, it is sufficient that the ILC update law only corrects the input at the root and overtone frequencies. The update law (6.9) thus becomes

$$U_{k+1}(f_n) = Q_1(f_n)U_k(f_n) + Q_2(f_n, k)\widehat{G}^{-1}(f_n)E_k(f_n). \quad (6.15)$$

Consequently, the estimate of the system transfer function $G(f)$ only needs to be known for the frequencies f_n .

MIMO Control

In order to extend the ILC approach presented for the SISO case such that it works for MIMO systems, the update law (6.9) is modified to

$$\bar{U}_{k+1}(f) = Q_1(f)\bar{U}_k(f) + Q_2(f, k)\widehat{G}^{-1}(f)\bar{E}_k(f) \quad (6.16)$$

where the vectors

$$\bar{U}_k = \begin{bmatrix} U_k^1 \\ U_k^2 \\ \vdots \\ U_k^m \end{bmatrix}, \quad \bar{E}_k = \begin{bmatrix} E_k^1 \\ E_k^2 \\ \vdots \\ E_k^l \end{bmatrix} \quad (6.17)$$

are introduced, given that $G(f)$ is an $l \times m$ matrix. In order for the MIMO ILC algorithm to converge, the choice of $Q_2(f)$ must satisfy the criterion [Doh and Moon, 2004]

$$\sup_f \|I - G(f)\widehat{G}^{-1}(f)Q_2(f)\| < 1, \quad (6.18)$$

where $\|\cdot\|$ denotes the operator norm.

Since the transfer matrix $G(f)$ may become close to or even singular for certain frequencies, the Tikhonov regularized inverse [Tikhonov and Arsenin, 1977] of $G(f)$ is introduced, which is given by

$$\widehat{G}^{-1*} = (G^T G + \varepsilon I)^{-1} G^T \quad (6.19)$$

where $\varepsilon > 0$. It was found that the regularization improved the robustness of the ILC algorithm significantly.

Adaptive Reidentification

As a result of the mechanical construction of the transducer, drift in the process dynamics occurs occasionally. These process variations may cause the ILC algorithm to diverge, especially around peak and notch frequencies where a small discrepancy between the model and the process is sufficient to cause divergence. Therefore, it is desirable to update the transfer function

to account for the process variations, *i.e.*, using an adaptive ILC algorithm. However, as a full frequency characterization of the transducer does not fulfill the specification, and takes more than 30 minutes to perform, this is not viable to use during operation. Furthermore, using linear chirp signals ranging over the entire operating frequency range does not yield sufficient resolution, as discussed earlier in this section. Thus, only a limited frequency range around a diverging frequency is considered at a time. Diverging frequencies are detected by checking for which frequencies the condition

$$\eta_1|Y(f)| + \eta_2|U(f)| > \delta, \quad f \in f_\xi \quad (6.20)$$

is satisfied, where η_1 and η_2 are weighting parameters, δ a threshold parameter, and f_ξ is a set of frequencies where divergence could be expected. The condition (6.20) takes both the magnitude of the input and the output into account. This was found to be an efficient way of detecting divergence in the ILC algorithm, especially for frequencies where the process has small gain, since divergence is hard to detect early if only the system output is considered.

From the obtained set of diverging frequencies, the frequency that gives the maximum value of (6.20) is selected as the reidentification frequency f_{ri} . In order to obtain sufficient excitation at and around the frequency f_{ri} , a linear chirp signal is formed. In contrast to the initial frequency characterization of the transducer, a chirp signal is sufficient in this case, since only a limited frequency range is considered for the reidentification. The chirp signal, denoted $u_e(n)$, is given as

$$u_e(n) = c_a(f) \sin(2\pi f(n)t_s n), \quad f(n) \in [f_{ri} - f_\Delta, f_{ri} + f_\Delta] \quad (6.21)$$

where c_a , is the amplitude of the signal and f_Δ defines the interval in which the identification is to be performed in. The chirp signal $u_e(n)$ is then added to the input signals one at a time. This way, the operation of the transducer does not have to be interrupted to perform the reidentification process, and the desired reference can be tracked simultaneously as the added chirp signal is applied. However, the specification defined in Section 6.3 should still be fulfilled, *i.e.*, the amplitude of the response of the superimposed chirp signal must not exceed the specified limit. Therefore, the choice of the chirp signal amplitude $c_a(f)$ is important. Since the transfer matrix of the system is inaccurate around the reidentification frequency, it can only be used to form a conservative estimate of the amplitude. The amplitude of the chirp signal for reidentification of channel 1 is chosen as

$$c_a(f) = \kappa \frac{a_d}{\max(|G_{11}(f)|, |G_{12}(f)|)} \quad (6.22)$$

where κ is a parameter chosen based on the expected uncertainty in the transfer matrix $G(f)$, and a_d is the desired amplitude of the response. The amplitude for reidentification of channel 2 is determined analogously.

Furthermore, the ILC update is deactivated in order not to act on the extra signal that is added to the input signals. Thus, the previous inputs obtained from the ILC updates are used and modified such that the input power at the reidentification frequencies is reset to zero. The modified inputs in combination with the added chirp signal are then used for the reidentification.

Once the required reidentification experiments have been performed, *i.e.*, one per input, the obtained input/output data are used to update the transfer matrix $G(f)$ at the frequencies in the desired interval. The elements of the transfer matrix for these frequencies are updated using

$$G_{i,j}(f) = \frac{Y_{i,k}(f) - Y_{i,k-1}(f)}{U_{e,j}(f)}, \quad f \in [f_{ri} - f_{\Delta}, f_{ri} + f_{\Delta}] \quad (6.23)$$

where k denotes iteration index, U_e is the chirp signal u_e transformed using DFT, j and i are the input and output channel indices, respectively. Once all elements of $G(f)$ have been updated, the ILC operation can be resumed. The reidentification procedure is summarized in Algorithm 1.

```

while  $k$  do
    if (6.20) true then
        for channel = 1:2 do
            form chirp signal  $u_e$  using (6.21) and (6.22);
            run  $U_{k+1} = U_k + U_e$ ;
        end
        update  $G_{i,j}(f)$  using (6.23);
    else
        run (6.16);
         $k = k + 1$ ;
    end
end

```

Algorithm 1: Summary of the reidentification algorithm.

6.6 Experimental Results

This section presents results from identification experiments, as well as results from a variety of ILC experiments. Furthermore, results of using the adaptive reidentification algorithm are also provided. All data were collected at 44.1 kHz, and subsequently downsampled by a factor of 4. Because of the fast sampling rate and the expected frequency content below 1000 Hz, aliasing effects are assumed nonexistent.

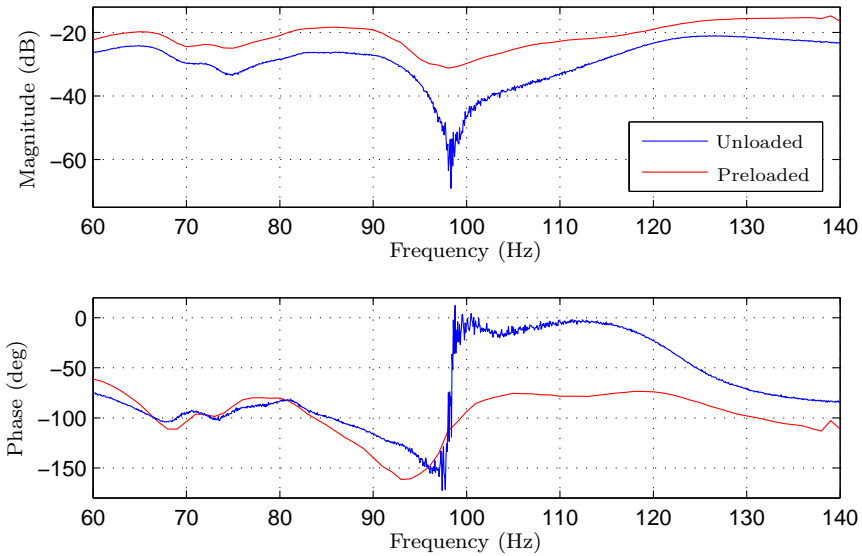


Figure 6.7 Magnitude and phase of the estimated $G_{11}(f)$ for the unloaded and preloaded transducer, shown in blue and red, respectively. The large difference indicates the need for performing system identification on a preloaded system which is closer to the later operational mode.

Estimation of Transfer Matrix

To the purpose of frequency characterization of the considered transducer, the method described in Section 6.5 was carried out, both on the unloaded and the preloaded system. An initial sweep of evenly spaced single-frequency sine waves from 15–650 Hz was performed, where single-Hertz increments were used. Furthermore, each sine wave lasted 0.5 s for each channel, *i.e.*, $N = 5512$. Based on the result, the grid of frequencies was modified to yield higher accuracy around resonance peaks and notches, *i.e.*, by adding more frequencies in these areas. With the new grid, the identification was performed once again, resulting in a more accurate estimate of $G(f)$.

In Section 6.5, it was stated that a significant difference in the transfer functions between the unloaded and preloaded system was to be expected. This was found to be true, see Figure 6.7, where the estimated frequency spectrum of one side of the transducer is displayed, for both the unloaded and preloaded cases. It was confirmed through ILC experiments that convergence could only be achieved with the use of the preloaded estimate of $G(f)$. The frequency spectra of the estimated $G(f)$, obtained from the preloaded experiments on the transducer, are displayed in Figure 6.8.

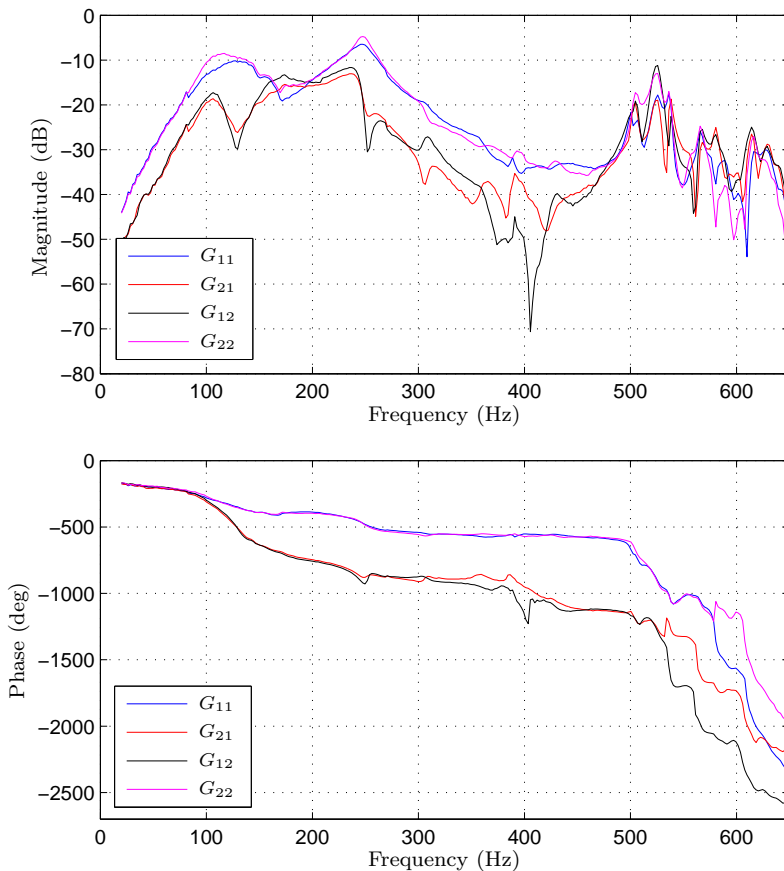


Figure 6.8 Spectrum of the estimated transfer matrix $G(f)$. For frequencies up to 150 Hz the matrix is diagonally dominant but for higher frequencies the spring system of the vibrator gives rise to large cross couplings between the two sides. This fact makes separate control of the two sides infeasible and indicates the need for MIMO control.

Single-Frequency Tracking

The special case described in Section 6.5, where a single-frequency reference is to be tracked, is considered in this section. ILC was performed for several frequencies in the range [20,100] Hz. However, since similar results were obtained, only one frequency is presented here. In this experiment, a reference signal with frequency $f_0 = 24$ Hz with a duration of 5 s ($N = 55125$) was considered, and the objective was to suppress the four first overtones to a level of 40 dB below the amplitude of the root frequency. The design parameter

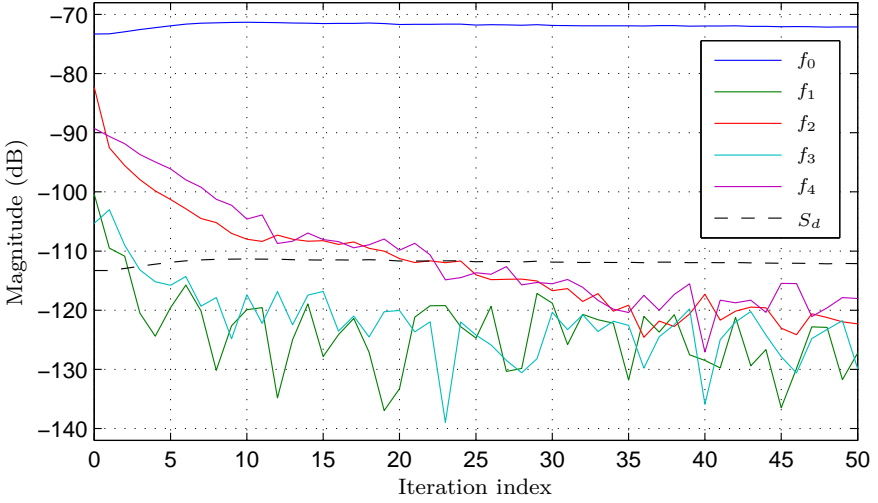


Figure 6.9 Magnitude plot of the root frequency $f_0 = 24$ Hz and the four first overtones, shown as function of the iteration index. The dashed black line represents the desired objective of 40 dB suppression (denoted S_d) from the root frequency.

$Q_2(f, k)$ defined in (6.11) was chosen as

$$Q_2(f, k) = \begin{cases} 1, & f \in f_{\text{ILC}}, k = 1, 2, \dots, 12 \\ 0.25, & f \in f_{\text{ILC}}, k = 13, 14, \dots \\ 0, & f \notin f_{\text{ILC}}, \forall k \end{cases} \quad (6.24)$$

where $f_{\text{ILC}} = \{f_1, f_2, f_3, f_4\}$ in this case. For this case, only one side of the transducer was considered for control. The first element G_{11} of G was therefore used for the ILC updates at the f_{ILC} frequencies.

The results of the experiments are found in Figure 6.9, where it is shown how the magnitudes of the overtones evolve over the ILC iterations. The objective is fulfilled after 24 iterations and an additional 10–15 dB suppression of the overtones is achieved. Looking at the estimated spectrum of the response, see Figure 6.10, it is once again confirmed that the overtones have been successfully suppressed. However, it is further noted that several other frequencies, not obviously connected to the root frequency f_0 , exhibit significant signal power. The power-grid frequency of 50 Hz is especially visible, which is to be expected. In order to suppress these frequencies, and to be able track any reference signal, focus is shifted to performing ILC for all frequencies.

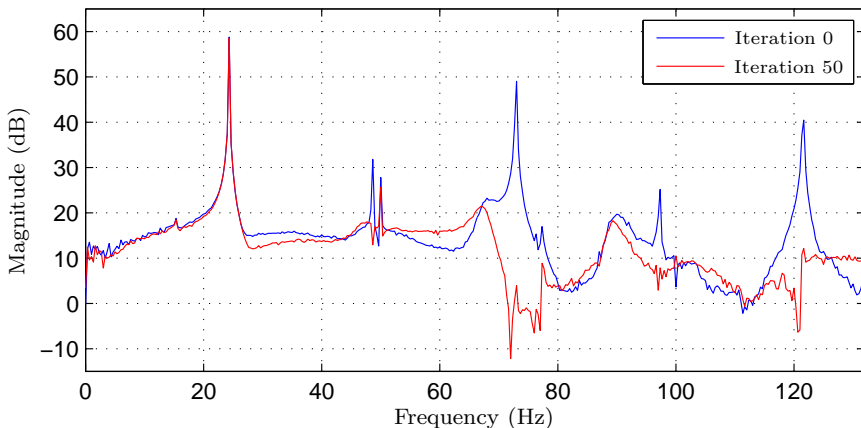


Figure 6.10 Spectrum of the transducer response before and after 50 ILC iterations, with a 24 Hz sine-wave reference.

Full-Frequency Tracking

For the experiments presented from this point, linear chirp signals in the frequency range of $f = [40, 90]$ Hz with a duration of 5 s ($N = 55125$) were used as reference. The objective of the experiments was to suppress frequencies above 100 Hz to a level of 40 dB below the reference magnitude. It was observed in experiments that the objective was fulfilled for frequencies above 650 Hz without active use of ILC for these frequencies, and therefore only results below this frequency are presented.

Initially, only one side of the transducer was considered for control. The first element G_{11} of G was therefore used for the ILC updates. The parameter Q_2 was chosen as

$$Q_2(f, k) = \begin{cases} 0.3, & f \in f_{\text{ILC}}, k = 1, 2, \dots, 15 \\ 0.15, & f \in f_{\text{ILC}}, k = 16, 17, \dots \\ 0, & f \notin f_{\text{ILC}}, \forall k \end{cases} \quad (6.25)$$

where $f_{\text{ILC}} = [0, 650]$ Hz. The choice of having lower gains in Q_2 compared to the single-frequency case is motivated by the fact that convergence is required for all frequencies, not just a small set of frequencies. A low gain will give slower convergence, but the algorithm will be more robust against process drift and model uncertainty.

The results of performing ILC for the SISO case are displayed in Figure 6.11, in the form of spectrograms for iterations 0 and 30. The scaling in the spectrograms is such that the darkest blue color represents the objective, *i.e.*, the level 40 dB below the reference-signal magnitude. Thus, a fulfilled

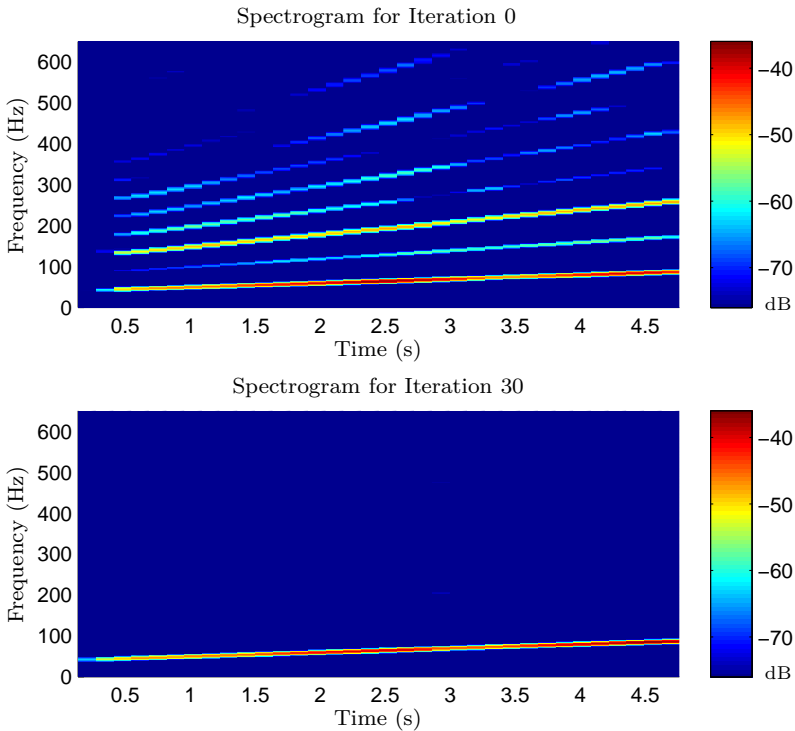


Figure 6.11 Spectrogram of the transducer response for ILC iterations 0 and 30. The SISO ILC algorithm suppresses all harmonics by 40 dB at the measurement point of the single accelerometer.

objective is easily spotted as a completely dark blue spectrogram for all frequencies except for the reference frequencies. The chirp reference can be seen in the bottom of both spectrograms in Figure 6.11, as a line from 40 to 90 Hz. The top panel in Figure 6.11 shows that the initial run, with input equal to the desired reference, exhibits significant energy in the undesired frequency range. As seen in the lower panel, the objective is achieved after 30 ILC iterations.

With this ILC structure, satisfactory control was achieved at the point on the shell measured by the accelerometer. However, only one side of the transducer is currently controlled, and both sides are required to operate simultaneously. In an experiment attempting to use the same input to both sides, poor tracking performance was obtained. This is most likely because of the strong cross couplings that are present between the two sides of the transducer, *cf.* Figure 6.8. Because of these cross couplings, the option of performing decentralized ILC was ruled out. Thus, MIMO ILC is considered

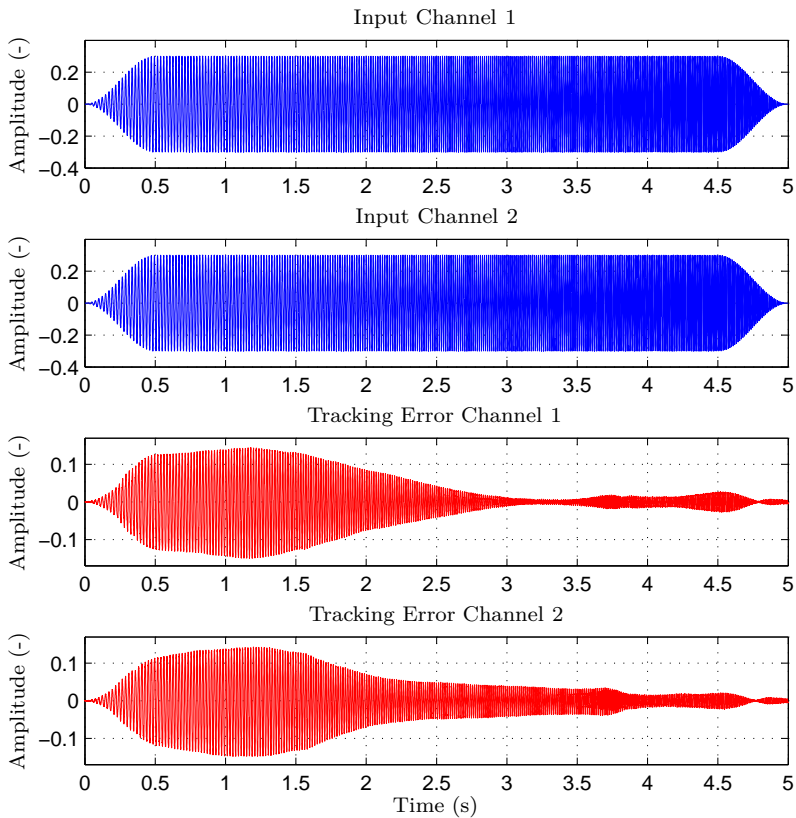


Figure 6.12 Time plot of the initial chirp input (iteration 0) and the transducer response compared to the desired response. The top two panels show the input for the respective channel, the two lower plots show the obtained tracking error.

next. The objective is the same as for previous experiments, as well as the desired reference which is used for both sides of the transducer. Furthermore, the Q_2 filter defined in (6.25) is used, as well as the estimate of the transfer matrix G obtained earlier in this section.

The initial chirp input and the resulting tracking errors are displayed in Figure 6.12. Spectrograms of the responses are shown in Figure 6.13. After performing 30 ILC iterations, the corresponding plots are shown in Figures 6.14 and 6.15. Also, zoomed time plots for the inputs and responses for iterations 0 and 30 are provided in Figure 6.16.

As illustrated by the figures, the objective is reached for both sides of the transducer, and the chirp reference is tracked as desired. In order to quantify

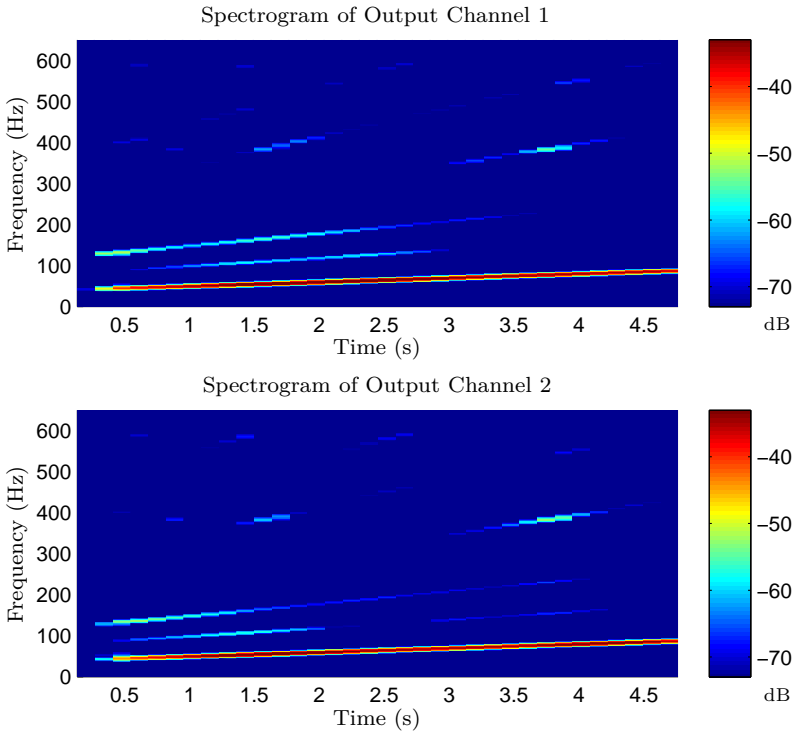


Figure 6.13 Spectrogram of the transducer response to the initial MIMO chirp input (iteration 0).

the results, the integrated absolute error (IAE) measure was considered, defined as

$$IAE = \sum_{n=0}^{N-1} |e_k(n)|. \quad (6.26)$$

In Figure 6.17, the evolution of the IAE for each channel is plotted as function of the iteration index. It is noted that convergence is achieved since the IAE is kept around its minimum value after 20 iterations, and does not increase. Furthermore, the IAE was decreased by a factor of approximately 50 for both channels.

Reidentification Results

In the majority of the performed experiments, the desired behavior was obtained. However, in a few experiments, divergence of the ILC algorithm was experienced for one or more frequencies. This was concluded to be a result

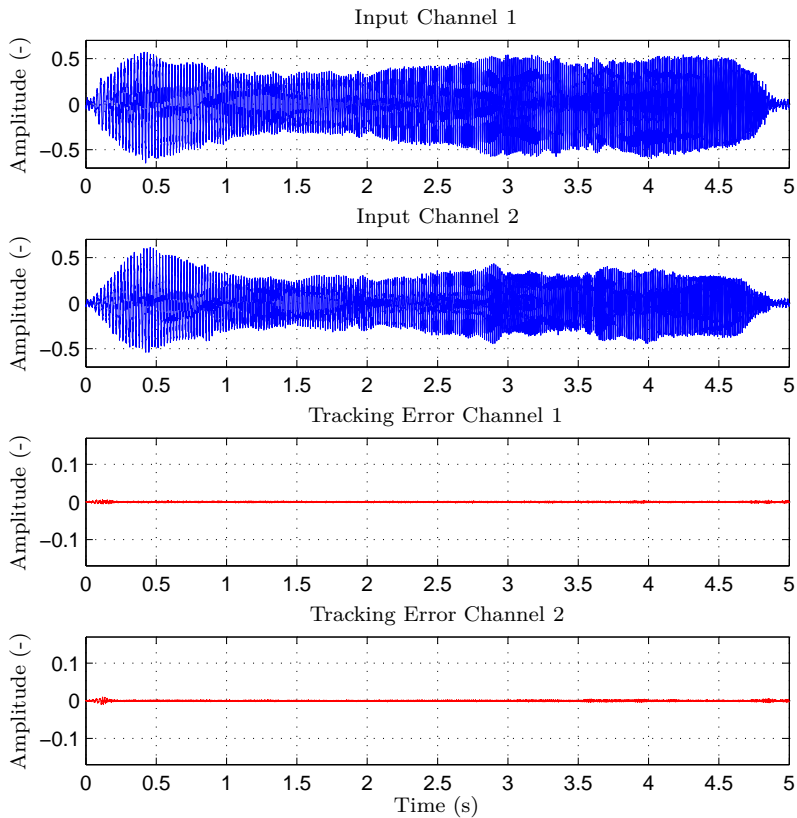


Figure 6.14 Time plot of the control input and the transducer response compared to the desired response, after 30 MIMO ILC iterations. The top two panels show the input for the respective channel, and the two lower plots show the obtained tracking error.

of drift in the dynamics of the transducer. Figure 6.18 shows the spectra of the two channels for a set of iterations, where it is noted that the peak at approximately 545 Hz is increasing for each iteration. With proper tuning of the parameters, the condition (6.20) was fulfilled for iteration 16 of the experiments displayed in Figure 6.18, and the reidentification procedure was triggered, as defined in Algorithm 1 in Section 6.5. The parameter f_{Δ} was chosen to 10 Hz, such that the reidentification was performed in the frequency interval 535–555 Hz. The magnitude of the initial estimate of $G(f)$, and the magnitude of the updated $G(f)$ are displayed in Figure 6.19. It is noted that the notch that was present in two of the elements of the initial $G(f)$, at around 545 Hz, does not exist in the updated estimate of the transfer matrix. Looking

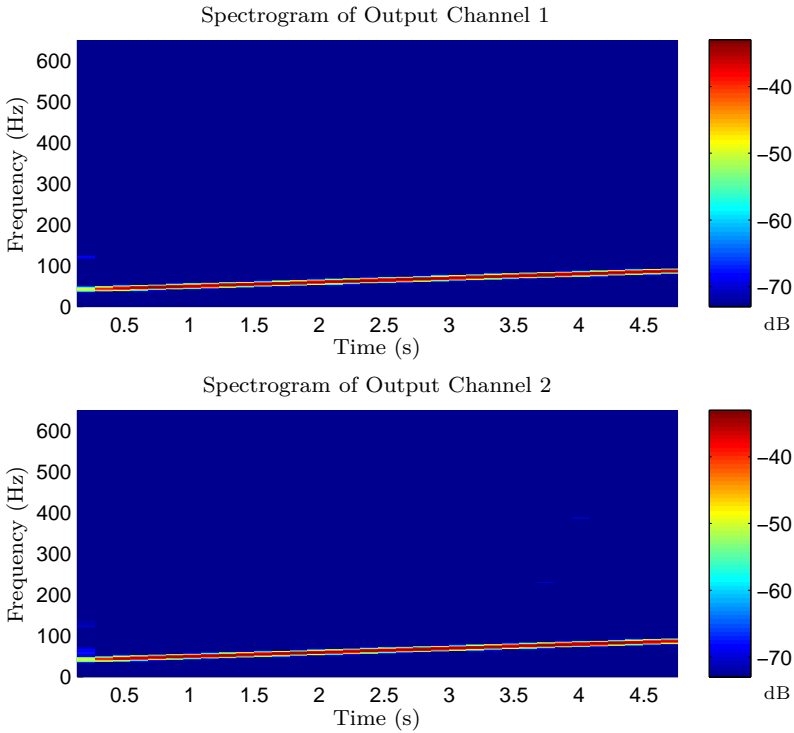


Figure 6.15 Spectrogram of the transducer response after 30 MIMO ILC iterations.

at the response of the reidentification experiments, see Figure 6.20, the 40 dB suppression objective is fulfilled with the proposed choice of amplitude, given in (6.22). With the updated $G(f)$, the ILC iterations were resumed and the resulting spectra of the output channels are shown in Figure 6.21, where convergence was restored for the reidentified frequencies.

6.7 Comparison to Time-Domain ILC

As motivation for choosing frequency-domain ILC instead of traditional time-domain ILC, an investigation and comparison of the differences are provided here. As indicated in the time-domain ILC update law in (6.8), the inverse of a parametric model describing the system dynamics is required. This presents two potential difficulties: identifying a model that captures the MIMO system dynamics sufficiently well and obtaining a stable inverse of the model.

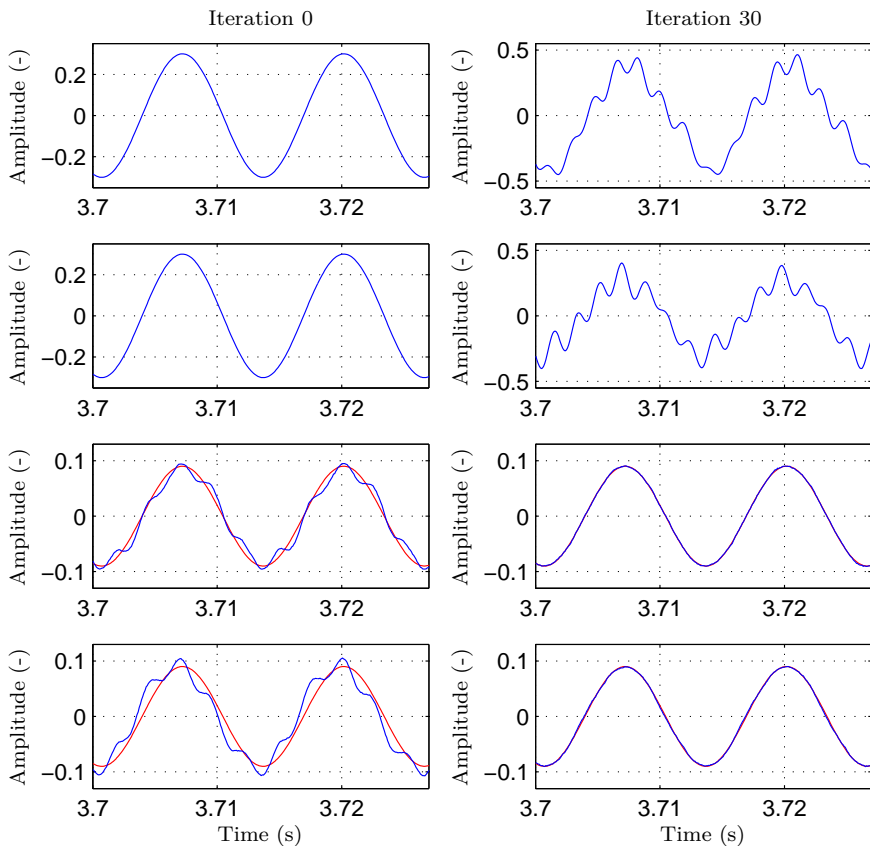


Figure 6.16 Zoomed time plot of MIMO ILC iterations 0 and 30. The top two rows of plots show the input for the respective channel, and the two bottom rows of plots show the measured responses in blue and the desired response in red.

In order to identify a parametric time-domain model, system identification methods such as N_4SID [Overschee and De Moor, 1994] was used on previously obtained identification data. Initially, SISO data were used for identification, where excitation was provided in the frequency range of 10–400 Hz. It was found that a model order of at least 18 was needed to achieve acceptable fit to the data. In terms of *variance accounted for* (VAF), the fit for an identified state-space model of order 18 was 75.9%. Even though satisfactory VAF values can be achieved with high model orders, small discrepancies in amplitude or phase for certain frequencies, such as resonance or notch frequencies, may result in divergence of the ILC

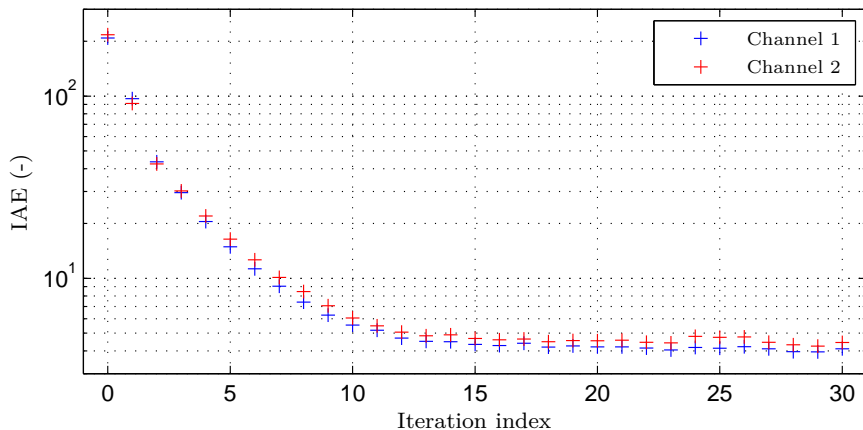


Figure 6.17 Error evolution of the MIMO ILC experiments, showing IAE values for both channels.

algorithm. Additionally, in order to capture the MIMO dynamics of the full frequency range up to 650 Hz, even higher model orders are required.

Furthermore, inverting the model is not straightforward, as inverting non-minimum phase zeros gives unstable models. The identified 18th order model contains four non-minimum phase zeros, which makes model inversion complicated. To obtain a stable ILC iteration, the technique of using non-causal system inverses must be used, see [Markusson et al., 2002]. Moreover, in order to cope with the drift in the dynamics, no other choice than to reidentify the entire model is given, as it is not possible to update parts of a parametric model.

6.8 Discussion and Conclusions

The proposed frequency-domain ILC strategy proved effective in suppressing undesired harmonics, as well as achieving accurate reference tracking. The desired level of 40 dB suppression of the harmonics was achieved for all experiments, after approximately 20 iterations. It was shown that, in the case of a sinusoidal reference, both the modeling and the ILC algorithm could be simplified while still yielding an overtone suppression of up to 55 dB. For the MIMO case, the absolute tracking error was reduced by a factor of approximately 50 for each channel. Faster convergence can be achieved by increasing the magnitude of the filter Q_2 . This, however, requires higher accuracy of the frequency-response estimate.

A procedure for detecting drift in the dynamics and performing a

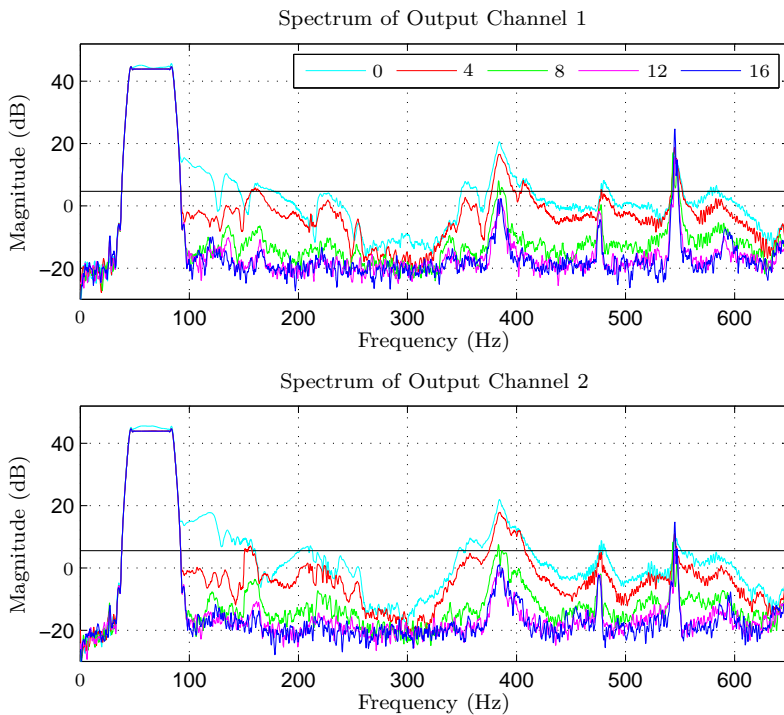


Figure 6.18 ILC experiments performed using an initial estimate of $G(f)$. Spectra of the measured response from iterations $\{0, 4, 8, 12, 16\}$ are displayed in cyan, red, green, magenta, and blue, respectively. The suppression level of 40 dB is shown in black. The algorithm gives sufficient suppression at most frequencies, but problems occurred around 545 Hz. This was solved by applying the reidentification algorithm, shown in the following figures.

subsequent reidentification of the estimated transfer matrix was developed. It was shown that the reidentification experiments with excitation around a problem frequency, could ensure convergence with the updated transfer matrix, while still fulfilling the objective of 40 dB suppression of harmonics.

The proposed method was compared to traditional time-domain ILC, and it was found that frequency-domain ILC is highly advantageous for this application. This was primarily because of the fact that time-domain ILC requires high-order models as well as complicated methods for obtaining stable model inverses. Also, the possibility to update the model for certain frequencies was shown to be advantageous, as divergence could be avoided with the proposed method.

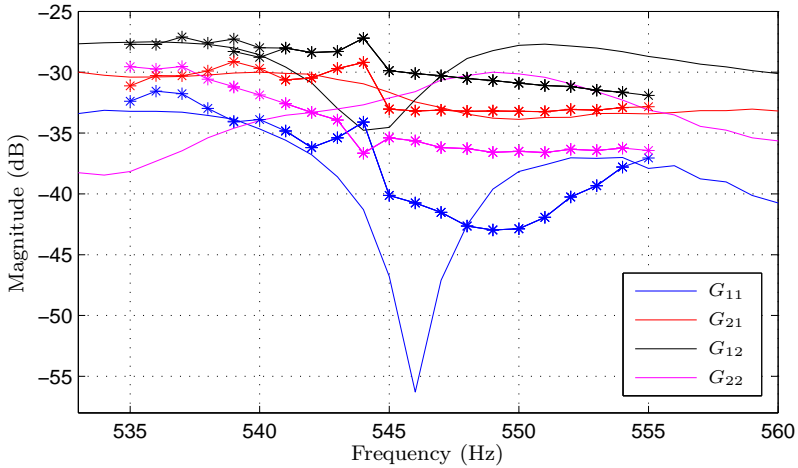


Figure 6.19 Magnitude plot of the initial estimate of $G(f)$ and the obtained magnitude from the reidentification, shown in thin lines and starred lines, respectively. The measurements indicate that a change in dynamics has occurred around 545 Hz.

Although the experiments presented in this chapter were performed in air, the only major expected difference to underwater operation is that the process dynamics will be compressed towards low frequencies. Therefore, the system characterization and ILC iterations must be performed underwater in order for the ILC to converge and to obtain the desired output. However, as at least 20 ILC iterations are needed to suppress harmful harmonics, it could be possible to perform the iterations in a controlled underwater environment and subsequently warm start the transducer when performing the seismic acquisition in the sea. Since the proposed method is meant to be run continuously during seismic acquisition, possible changes in the dynamics could then be handled by the adaptive reidentification algorithm.

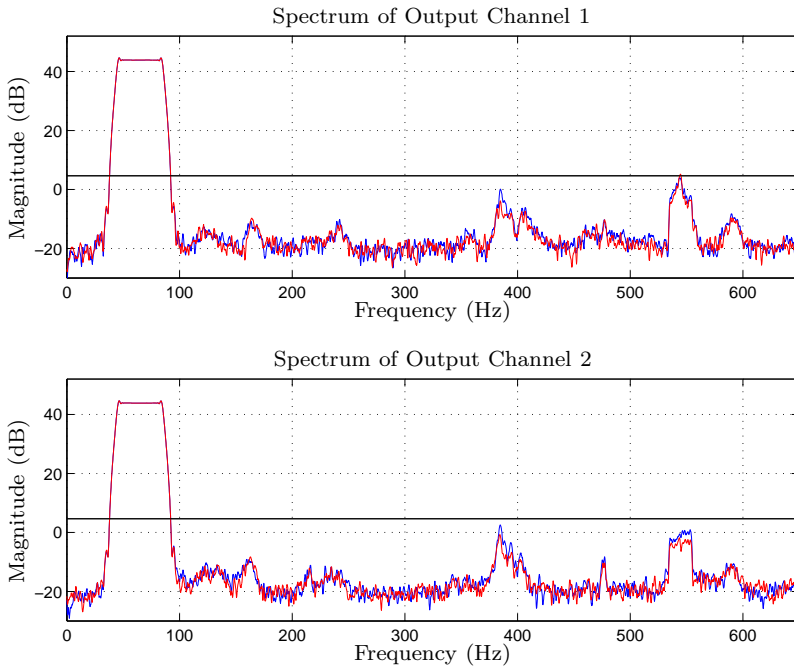


Figure 6.20 Spectra of the measured response from the reidentification experiments, with a superimposed chirp signal in the frequency range 535–555 Hz on the channel 1 and 2 inputs, shown in red and blue, respectively. The suppression level of 40 dB is shown in black. Note that reidentification is done without violating the specification.

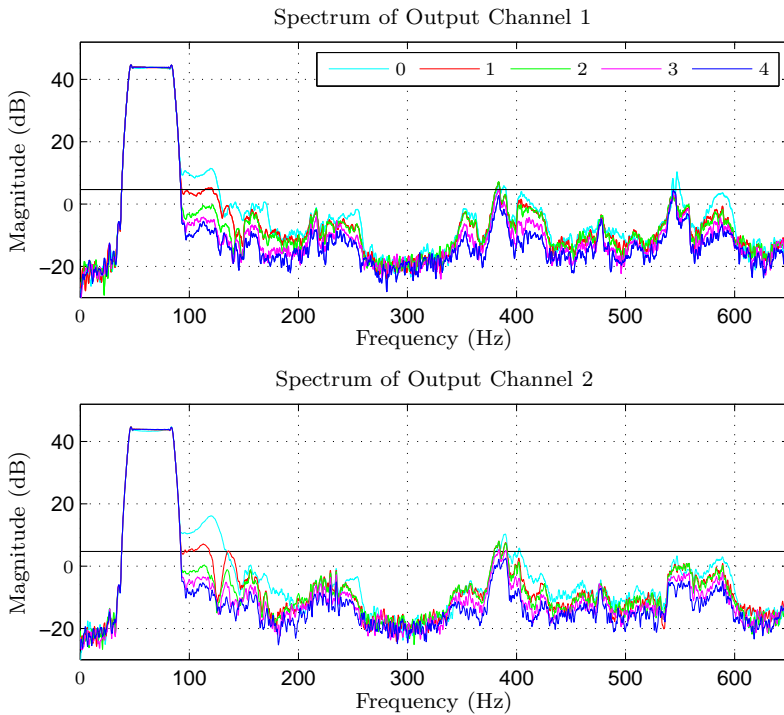


Figure 6.21 ILC experiments performed with updated $G(f)$, obtained by reidentification. Spectra of the measured response from iterations $\{0, 1, 2, 3, 4\}$ displayed in cyan, red, green, magenta, and blue, respectively. The suppression level of 40 dB is shown in black. After the reidentification, the ILC algorithm is stable also around 545 Hz.

7

Cycle-Time Minimization of Wood Milling

7.1 Introduction

This chapter is based on the publications [Sörnmo et al., 2012a] and [Sörnmo et al., 2015b].

Machining processes in the industry of today, such as milling, grinding, and deburring, are traditionally performed using position control. In order to avoid excessive process forces, which may result in tool breakage or scorched material, a conservative feed rate has to be used. By instead controlling the process forces, the feed rate (and consequently the material removal rate), can be maximized. This will in turn lead to decreased cycle times. Because of the batch-oriented nature of modern production, even a small reduction in cycle time can amount to significant savings in production expenses in the long run.

The control problem in this chapter is formulated as continuously adjusting the feed rate of the workpiece/tool in order to achieve and maintain the maximum allowed force. The machining process forces depend nonlinearly on several parameters, such as spindle speed, machining tool, depth of cut, and material stiffness. Since some of these parameters are likely to change during the process, it is desirable to continuously adapt the parameters of the force controller. The use of a controller with fixed parameters may result in loss of time efficiency or stability problems.

Additional cycle-time reductions can be achieved by intelligent path planning of the machining task. However, this may not be a straightforward task, especially in the case when machining is to be performed in non-isotropic materials such as wood. Different machining settings, such as depth of cut and machining direction, may result in varying material-removal efficiency. Without profound comprehension of the specific machining process, it is difficult to take advantage of this. By employing a learning

algorithm, no prior knowledge is required in order to obtain an efficient milling path.

This chapter considers the problem of minimizing cycle times for rough-cut machining processes in non-isotropic materials. A model-based adaptive force-control scheme is developed to ensure that the fastest possible feed rate is used at all time instances. The machining-force dynamics is modeled as a linear parameter-varying (LPV) system, where the parameters are estimated such that the controller is adapted to different machining conditions. A learning algorithm that minimizes a process-related cost function is also developed. The proposed method is verified both in simulation and in pocket-milling experiments using an oak-milling setup.

In this research, learning is considered to be performed on two levels. Firstly, the adaptive controller learns the parameters associated with the milling process, in order to adapt to the current conditions and control the force such that the feed rate is maximized. Secondly, on a higher level, the characteristics of each performed milling segment are learned and subsequently utilized to replan the milling path such that the cycle time for each milling task is minimized.

Previous Research

This chapter extends and refines the previous research initially published in [Sörnmo et al., 2012a], where an adaptive feed-rate force controller was presented and evaluated in milling experiments with varying vertical depth of cut. The force controller is here redesigned to account for transversal forces, and an improved controller structure is presented and utilized. Furthermore, this chapter introduces a learning algorithm for path planning in combination with the force control.

Control design for industrial manipulators performing contact tasks is discussed in, *e.g.*, [Hogan and Buerger, 2005]. It was shown that the environment, *i.e.*, the work object, can be modeled as an admittance, whereby it follows that the robot should act as an impedance in the closed kinematic chain. Hence, the aim of impedance control for robots is to control the dynamic relation between the force and the position.

A self-tuning PI controller for controlling machining forces was presented in [He et al., 2007], where the machining force was modeled as a static nonlinear relation between the feed rate and the depth of cut. In [Liu et al., 2001], an adaptive control constraint was considered, based on several control structures such as PID control, neural-network control, and fuzzy control. Feed-rate force controllers based on model reference adaptive control (MRAC) were presented in [Daneshmend and Pak, 1986; Lauderbaugh and Ulsoy, 1989; Rober and Shin, 1996]. An overview of force-control technologies in machining is provided in [Wang et al., 2008].

An established method is the maximum material removal controller, which switches between discrete feed-rate levels in order to maintain an approximate force reference. In [Lauderbaugh and Ulsoy, 1988], the importance of using an adaptive controller for force-controlled machining processes is demonstrated through simulations and machining experiments.

Previous research in learning for machining processes is mostly focused on monitoring and anomaly detection. In [Burke and Rangwala, 1991], neural-network based pattern-recognition techniques were employed in order to monitor a metal-cutting process, specifically the condition of the cutting tool. A learning algorithm based on support-vector machines for recognizing process abnormalities was proposed in [Cho et al., 2005]. An adaptive network-based fuzzy inference system was used in [Ho et al., 2009], in combination with the genetic learning algorithm, in order to predict the workpiece surface roughness for a milling process. In [Rangwala and Dornfeld, 1989], a neural-network approach to learning how the input variables of the machining operation affect the outputs was presented. The network was then used in simulations to predict input conditions to maximize metal removal.

Efficient path planning, in terms of minimizing the path length, for machining operations has thoroughly been researched, see, *e.g.*, [Wang et al., 1987; Yang and Lee, 2002; Park and Choi, 2000; Lin and Koren, 1996; Lo, 1999]. Finding the shortest path in a complex rough-cut milling scenario was considered in [Suh and Shin, 1996], where a self-organizing map approach was taken. However, none of these papers take the machining process dynamics into account. Path-planning complex rough-cut pockets with islands in a plunge-milling scenario was considered in [Chen and Abdelkhalek, 2014], where the path length and the number of plunges were minimized.

In [Meng Lim and Menq, 1997], a problem formulation similar to the one in this chapter was considered, where the aim was to optimize cutting path and feed rates for machining operations. Geometric maximum feed-rate maps were established based on force models, such as the flexible force model [Feng and Menq, 1996], and are subsequently utilized for optimizing the path-planning. The method in [Meng Lim and Menq, 1997] does not, in contrast to the method proposed in this chapter, take into account the varying cutting conditions, *i.e.*, neither force control nor learning were performed.

Disposition

This chapter is organized as follows: Modeling and design of the force controller, and the proposed learning algorithm, are presented in Section 7.2. In Section 7.3, the experimental setup is described, and results from both simulations and experiments using the proposed method are presented. A discussion of the method and the results, as well as aspects on future research, are provided in Section 7.4. Finally, conclusions are drawn in Section 7.5.

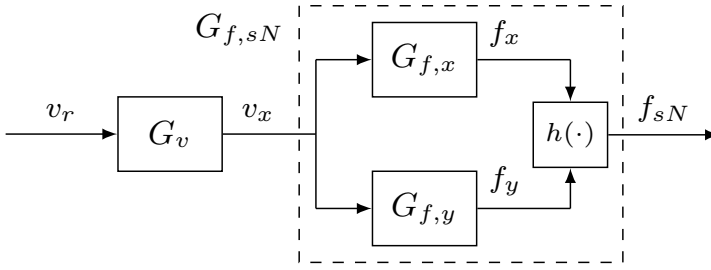


Figure 7.1 Block diagram of the proposed modeling approach, where x is the feed direction of the milling and $h(f_x, f_y) = \text{sign}(f_x) \sqrt{f_x^2 + f_y^2}$.

7.2 Modeling and Control Design

This section describes the proposed modeling approach of the milling process, followed by the design of the feed-rate force controller. Furthermore, a description and analysis of the proposed learning algorithm are provided.

Force-Control Design

Along the feed direction of the machining, the aim is to control the process forces by adjusting the feed rate, which in robotic machining corresponds to the velocity of the robot end effector. In this chapter, a model-based solution is pursued and it is therefore necessary to develop models of both the robot dynamics and the milling-process dynamics. For the robot dynamics, a linear system-identification approach was adopted. This choice is motivated by the fact that only Cartesian control is to be performed, and the linear approximation is valid in the limited workspace of the machining task.

The proposed modeling approach is presented in Figure 7.1, where x denotes the feed direction, and y is the transversal direction in the horizontal plane. The details and variables of the proposed modeling approach will be discussed in the following.

Similar to the identification of the robot model in Chapter 5, a model with velocity reference v_r as input, and actual velocity v_x in the Cartesian space as output was identified from experimental data of the robot moving in free space, using the N4SID subspace algorithm [Overschee and De Moor, 1994]. The robot velocity dynamics $G_v(s)$ are obtained as a state-space model of the innovations form [Ljung, 1987; Johansson, 1993]

$$\dot{x}(t) = Ax(t) + Bv_r(t) + Ke(t) \quad (7.1)$$

$$v_x(t) = Cx(t) + e(t), \quad (7.2)$$

where $e(t)$ is white noise and K is the corresponding Kalman gain. The system matrices A, B, C , and K are determined by the system identification

algorithm. In this case, a third-order model was obtained, where the order was determined by investigating the singular values of the identification data.

As mentioned earlier, the machining process forces depend on several parameters. These parameters exhibit a nonlinear relationship with the process forces and may change over time, thus making the process difficult to model. Here, a first-order dynamic model is derived, with time-varying parameters to the purpose of capturing the nonlinear properties and *a priori* unknown changes in the machining process. The model is derived using Hooke's law [Ugural and Fenster, 2003] which can be stated as

$$f_x(t) = K_f p_x(t), \quad (7.3)$$

where $f_x(t)$ is the force component in the feed direction, K_f the material stiffness, and $p_x(t)$ the depth of the deformation into the material. By assuming that material is removed at a rate proportional to the integral of the applied force, (7.3) is modified to the differential equation

$$f_x(t) = K_f \left(p_x(t) - \int_0^t \kappa^{-1} f_x(\tau) d\tau \right), \quad (7.4)$$

where the parameter κ can be interpreted as the inverse material removal rate. It is obvious from (7.4) that a large value of κ will result in a slow material removal rate and thus a large machining force. Transforming (7.4) to the frequency-domain and substituting position for velocity gives

$$F_x(s) = \frac{s}{sK_f^{-1} + \kappa^{-1}} P_x(s) = \underbrace{\frac{1}{sK_f^{-1} + \kappa^{-1}}}_{G_{f,x}(s)} V_x(s). \quad (7.5)$$

The process forces acting on the machining tool are, however, present in the orthogonal directions as well, not only in the feed direction. In two-dimensional machining in the horizontal plane, the vertical component of the force is disregarded. In the horizontal plane, the resulting force must be considered for the force control, since it limits the feed rate according to the discussion in Section 7.1. The resulting force $f_N(t)$ in the horizontal plane is given by the Euclidean norm of the force components

$$f_N(t) = \sqrt{f_x(t)^2 + f_y(t)^2}, \quad (7.6)$$

where $f_y(t)$ is the force component orthogonal to the feed direction, in the plane. By assuming that the x - and y -components of the force are proportional with a parameter β

$$f_y(t) = \beta f_x(t), \quad (7.7)$$

(7.6) can be rewritten to

$$f_N(t) = \sqrt{(1 + \beta^2)f_x(t)^2} = \sqrt{(1 + \beta^2)}|f_x(t)|. \quad (7.8)$$

The signed resulting force $f_{sN}(t)$ is then defined as

$$f_{sN}(t) = f_N(t)\text{sign}(f_x(t)) = \sqrt{(1 + \beta^2)}f_x(t). \quad (7.9)$$

By applying the Laplace transform on (7.9), and combining the result with (7.5), the transfer function $G_{f,sN}(s)$ from $v_x(t)$ to $f_{sN}(t)$ is determined as

$$F_{sN}(s) = \underbrace{\frac{\sqrt{1 + \beta^2}}{sK_f^{-1} + \kappa^{-1}}}_{G_{f,sN}(s)} V_x(s). \quad (7.10)$$

The complete transfer function for the system $G_{f,sN}(s)G_v(s)$, that is, from velocity reference to force, is now given by a fourth-order system. By combining the velocity and force dynamics, an adaptive linear-quadratic integral (LQI) controller [Zhou and Doyle, 1998] can be designed for the complete system [Sörnmo et al., 2012a]. However, as it is of interest to control not only the force but the velocity as well, a cascaded LQI control structure is proposed. This choice is also well suited because of the fact that the velocity dynamics are not time varying. Thus, only the outer force controller needs to be adapted.

Since the states of the robot velocity dynamics are not measurable, a Kalman filter (KF) is introduced in order to estimate the states, based on the measured velocity and the identified model. The Kalman filter is given by [Åström and Wittenmark, 1997] as

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bv_r(t) + K(v_x(t) - C\hat{x}(t)) \quad (7.11)$$

$$\hat{v}_x(t) = C\hat{x}(t). \quad (7.12)$$

The filter is implemented in discrete-time, using the Luenberger zero-lag observer [Åström and Wittenmark, 1997], which is given by

$$\hat{x}_{k|k} = (I - KC)(\Phi\hat{x}_{k-1|k-1} + \Gamma v_{r,k-1}) + K v_{x,k}, \quad (7.13)$$

where Φ and Γ are the discrete-time counterparts of the system matrices A and B in (7.1), and k is the sample index. Since the model is identified with experimental data with subtracted mean, the filter is extended with a disturbance state [Åström and Wittenmark, 1997] in order to achieve the correct static gain. With all state estimates available, the inner loop can be closed using LQI control. The integral action is achieved by introducing an

integral state $x_i(t)$, which augments the system in (7.1)–(7.2). The integral state is defined as

$$x_i(t) = \int_0^t (v_d(\tau) - v_x(\tau)) \, d\tau, \quad (7.14)$$

where $v_d(t)$ is the desired velocity. The augmented state vector is denoted by $x_e(t)$ and is given by

$$x_e(t) = \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix}. \quad (7.15)$$

The state-feedback control law is written as

$$v_r(t) = -L_v \hat{x}_e(t) + v_d(t) + v_{\text{init}}, \quad (7.16)$$

where v_{init} is a constant initial velocity that is added to the input of inner controller in order to provide excitation for the estimation. The LQ cost function is stated as

$$J(v_r) = \int_0^\infty x_e(t)^T Q_v x_e(t) + v_r(t)^T R_v v_r(t) \, dt, \quad (7.17)$$

where Q_v and R_v are user-defined weights. The cost function (7.17) is minimized by solving the *algebraic Riccati equation* [Zhou and Doyle, 1998], which gives the optimal value of the gain vector L_v .

In order to simplify the design of the outer-loop force controller, the inner closed-loop velocity dynamics is approximated by a first-order model. Given that the inner-loop controller provides sufficient damping of potential oscillatory modes, the approximation only needs to capture the bandwidth of the inner closed loop, and the proposed first-order approximation is thus valid. The approximated system is given on the form (7.1)–(7.2) with system matrices and Kalman gain denoted \bar{A} , \bar{B} , \bar{C} , and \bar{K} respectively. The complete model for the system is obtained by augmenting the state-space model with the force dynamics given in (7.10). Since the force and velocity can be measured, they are favorably chosen as states, such that the measurements can be used directly for feedback. By application of the inverse Laplace transform on (7.10), and the introduction of $x_f(t) = f_{sN}(t)$, $x_v(t) = v_x(t)$, and $\bar{\beta} = \sqrt{1 + \beta^2}$, the following relation is obtained

$$\dot{x}_f(t) = -K_f \kappa^{-1} x_f(t) + K_f \bar{\beta} x_v(t). \quad (7.18)$$

Further, substituting the closed-loop velocity dynamics into (7.18) gives

$$\dot{x}_f(t) = -K_f \kappa^{-1} x_f(t) + K_f \bar{\beta} \bar{C} x_v(t) + K_f \bar{\beta} e(t). \quad (7.19)$$

By defining the system output as the force, the augmented state-space model can be written as

$$\begin{bmatrix} \dot{x}_v(t) \\ \dot{x}_f(t) \end{bmatrix} = \begin{bmatrix} \bar{A} & 0 \\ K_f \bar{\beta} \bar{C} & -K_f \kappa^{-1} \end{bmatrix} \begin{bmatrix} x_v(t) \\ x_f(t) \end{bmatrix} + \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} v_d(t) + \begin{bmatrix} \bar{K} \\ K_f \bar{\beta} \end{bmatrix} e(t) \quad (7.20)$$

$$f_{sN}(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_v(t) \\ x_f(t) \end{bmatrix} + w(t), \quad (7.21)$$

where $w(t)$ is white noise. An integral state is introduced

$$x_i^f(t) = \int_0^t (f_d(\tau) - f_{sN}(\tau)) \, d\tau, \quad (7.22)$$

where f_d is the desired force. By extending (7.20) with (7.22), the obtained model is used to design the LQI force controller with gain vector L_f , following the same procedure as described for the velocity controller. Finally, this gives the complete closed-loop system with force reference $f_d(t)$ as input. A block diagram of the system is displayed in Figure 7.2.

The controller has so far been derived assuming constant parameters. As discussed earlier, the material removal rate κ is likely to be time varying and should thus be estimated continuously. This also applies to the parameter β , and since f_x , f_y , and f_{sN} are measured, the desired parameters can be estimated by the Kalman filter. This is accomplished by extending the discrete-time state vector from (7.13) with the estimates of the parameters κ and β

$$\hat{x}_{x,k} = \begin{bmatrix} \hat{x}_k \\ \hat{\kappa}_k \\ \hat{\beta}_k \end{bmatrix}, \quad (7.23)$$

where the parameters are modeled as random walks [Spitzer, 1964], and thus the extended Φ and Γ -matrices are written as

$$\tilde{\Phi} = \begin{bmatrix} \Phi & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \tilde{\Gamma} = \begin{bmatrix} \Gamma \\ 0 \\ 0 \end{bmatrix}. \quad (7.24)$$

Furthermore, with the new state vector, the extended C -matrix becomes time varying since it contains force measurements:

$$\tilde{C}_k = \begin{bmatrix} C & 0 & 0 \\ 0 & f_{sN,k} & 0 \\ 0 & 0 & f_{x,k} \end{bmatrix}. \quad (7.25)$$

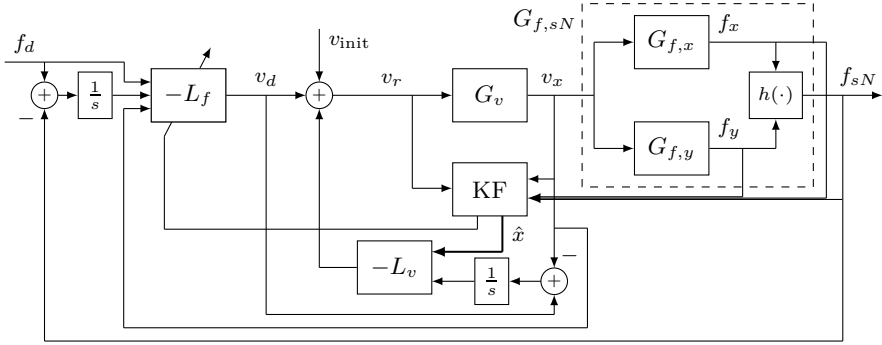


Figure 7.2 Schematic block diagram of the proposed force-control structure.

The Kalman gain vector K is extended to a matrix \tilde{K} , as well as the output vector $v_{x,k}$ in (7.12) which is extended using (7.7) and (7.10) to incorporate the new output signals needed for the estimation

$$\tilde{v}_{x,k} = \begin{bmatrix} v_{x,k} \\ C\hat{x}_{k-1}\sqrt{1 + \hat{\beta}^2} - \frac{f_{sN,k} - f_{sN,k-1}}{K_f h} \\ f_{y,k} \end{bmatrix}, \quad (7.26)$$

where h is the sample period of the system. With these extensions, the parameters can be estimated by modifying (7.13).

As mentioned previously, an initial velocity reference v_{init} is provided to the inner controller, in order to provide excitation for the parameter estimation. Once the covariances of the estimates of κ and β pass below predefined thresholds, the outer controller is activated and the estimates are used to continuously adapt the outer controller. The adaptation of the controller is achieved by updating the model in (7.20) using the current estimates, and subsequently solving the LQI problem described earlier.

Learning Algorithm Design

In this section, an approach to sequential learning of the time-optimal milling path is proposed. The assumptions on the milling are that an initial rough cut is to be performed (where time is the primary concern, not the absolute accuracy of the machined workpiece) and that milling is performed in a two-dimensional plane, such as in pocket milling. Moreover, it is assumed that the proposed adaptive force controller is employed, which means that the maximum possible velocity is achieved, given the desired machining force, under the prevailing cutting conditions. Considering that wood is an organic

material, it naturally exhibits non-isotropic behavior. It is here assumed that the material that milling is to be performed in is directionally isotropic and spatially invariant. This means that for a milling in an arbitrary direction and in its opposite direction, both will exhibit repeatable behavior over the length of the workpiece, although possibly different from each other. Also, the usage of different coverages and cutting directions of the milling tool are expected to give varying behavior because of the non-symmetry of the teeth of the tool. This implies that the milling parameters that need to be determined regarding the path planning are the coverage $\xi \in (0, 1]$ of the tool (measured as a fraction of the milling-tool diameter), the direction of the feed rate in each point along the path, as well as the cutting direction of the tool in relation to the feed rate.

Path-Planning Problem Initially, it is assumed that the milling path is segmented into N parts (where the parameter N is unknown *a priori*) and that the area to be machined is spatially discretized¹ in a grid. In order to describe the points in the milling area, a time-varying binary matrix P is introduced, where 0 indicates that milling has not been performed and 1 indicates that milling has been performed in that particular point. The notation P_n is used to denote this matrix after $n - 1$ milling segments have been performed. Furthermore, given the initial position $p_{n,r}$ of the robot end effector before the start of segment n , the possible milling approaches γ in terms of coverage ξ and milling direction for each point in the grid of P are assumed to be provided by the function $\Psi(p_{n,r}, P_n)$. The expected time for performing milling segment n with approach γ is denoted $\mu(t_{n,\gamma_n})$, where t_{n,γ_n} is the milling time for segment n . The time for reaching the starting point of each milling segment is denoted $t_{n,r}$. Consequently, the time-optimal path-planning problem can be stated as

$$\underset{\gamma_i, i \in \{1, \dots, N\}}{\text{minimize}} \quad \sum_{n=1}^N \frac{\mu(t_{n,\gamma_n}) + t_{n,r}}{l_n \xi_n d} \quad (7.27)$$

$$\text{subject to} \quad \gamma_i \in \Psi(p_{i,r}, P_i) \quad (7.28)$$

where l_n is the length of milling segment n , and d is the diameter of the tool. The duration of the milling is normalized with the product of the length of the milling segment and the coverage in order to weight the different milling approaches similarly in the cost function. It is clear that the optimization problem is NP-hard, since it is a combinatorial problem with an infinite number of possible milling approaches for each segment. This is similar to the asymmetric traveling salesman problem (ATSP) [Frieze et al., 1982; Reinelt,

¹The required density of the discretized path grid is determined by the desired granularity of the coverage ξ in the path planning.

1994; Gouveia and Pires, 1999]. Because of the extensive complexity of this problem, a near time-optimal solution is here pursued that is shown to work well in many practical scenarios in Section 7.3.

Sequential Near Time-Optimal Path Planning For this method, two assumptions are made. The first assumption is that the decision space regarding the milling directions consists of two different possibilities; forward along the x -axis and backwards along the x -axis, *i.e.*, two-directional path planning. The second assumption is that the coverage ξ is chosen as a discrete set; here this set is chosen as $\{1/3, 2/3, 1\}$. Considering that the milling segments with $\xi < 1$ can be performed both with the teeth of the milling tool directed in the feed rate direction (known as *up milling*) and the opposite (known as *down milling*), there are a total of ten possible milling approaches γ . Hence, the function $\Psi(p_{n,r}, P_n)$ provides a maximum of ten different milling approaches (given the state of P_n and the current robot configuration $p_{n,r}$). A set of milling type examples are displayed in Figure 7.3.

A sequential approach for computing an approximate solution to (7.27)–(7.28) is proposed. This strategy has two advantages; first, it reduces the complexity of the complete path-planning problem and second, it enables online learning of the milling characteristics. To the purpose of learning, the time t_γ for performing a milling segment with approach γ is stored together with the corresponding standard deviation $\sigma_{f,\gamma}$ of the measured norm of the process force in stationarity. Gaussian distributions are assumed for both of these quantities, where data points caused by anomalies in the material, such as knots and twigs, are removed using a simple outlier detection algorithm.

The milling strategy γ is determined such that it minimizes the time for performing the milling of the segment and returning to the starting side of the segment. For the latter, two options exist; either performing a milling, or lifting the tool on the return trip and not performing a milling (at a significantly faster velocity than possible during milling). The average time for performing milling in both directions (segments i and $i + 1$) is given by

$$t_A(\gamma_i) = \frac{t_{i,\gamma_i} + t_{i+1}}{2} \quad (7.29)$$

where

$$t_{i,\gamma_i} = \frac{\mu(t_{i,\gamma_i}) + t_{i,r}}{l_i \xi_i d}, \quad \gamma_i \in \Psi(p_{i,r}, P_i), \quad (7.30)$$

$$t_{i+1} = \arg \min_{\gamma_{i+1} \in \Psi(p_{i+1,r}, P_{i+1})} \left(\frac{\mu(t_{i+1,\gamma_{i+1}}) + t_{i+1,r}}{l_{i+1} \xi_{i+1} d} \right), \quad (7.31)$$

whereas the time t_B for performing milling in only one direction (segment i)

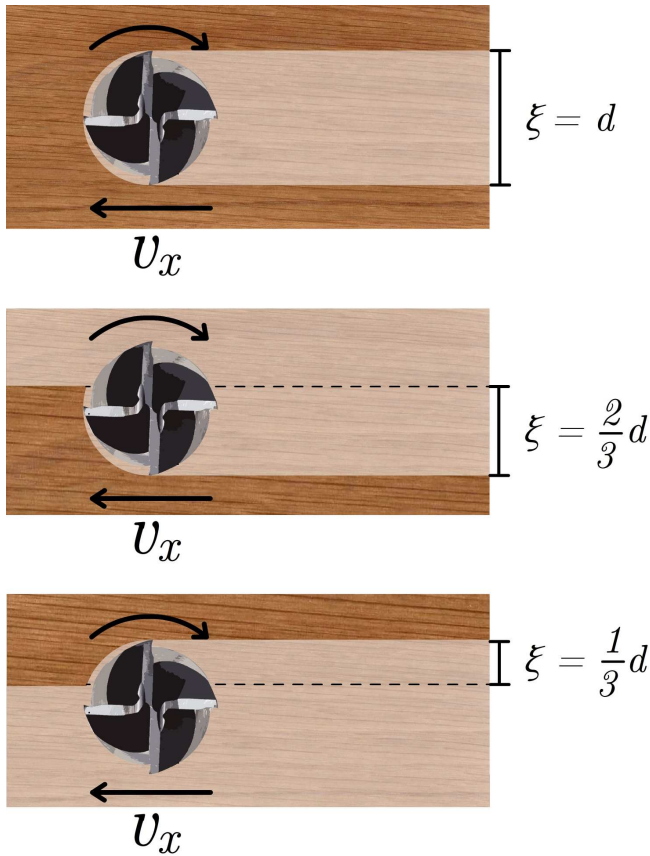


Figure 7.3 Example of three different milling types, with different coverages and cutting directions of the tool. The milling types from top to bottom are; slot, up, and down milling, respectively [Grote and Antonsson, 2009].

and then returning to the starting side of the workpiece is given by

$$t_B(\gamma_i) = \frac{\mu(t_i, \gamma_i) + t_{i,r} + t_{i,c}}{l_i \xi_i d}, \quad \gamma_i \in \Psi(p_{i,r}, P_i), \quad (7.32)$$

where $t_{i,c}$ is the time required for returning to the initial side, *i.e.*, moving a distance of

$$\sqrt{l_i^2 + (\xi_i d)^2}. \quad (7.33)$$

The optimal milling strategy for segment n is given by

$$\gamma_n = \arg \min_{\gamma_i \in \Psi(p_{n,r}, P_n)} (\min(t_A(\gamma_i), t_B(\gamma_i))) \quad (7.34)$$

Prior to performing the first segment, the optimal starting point must be determined. This is done by calculating the total cost for the complete milling task using (7.29)–(7.34), for each of the possible starting corners. The corner point that exhibits the lowest total cost is selected as the starting point, and the first milling segment determined by (7.34) is performed. If $t_A < t_B$, milling is performed in both directions and if $t_B < t_A$, milling is only performed in one direction with a subsequent free-space movement to the initial side. After each segment, a new data point is obtained, which means that the expected value $\mu(t_i, \gamma_i)$ may change, and consequently also t_A and t_B . Therefore, with the newly acquired data, the previously calculated solution to the complete path-planning problem may not be optimal anymore. The calculations are thus redone after each milling segment is performed, using the updated sets and the new corner points obtained from P_n . Should a corner point different from the current point exhibit a lower total cost, with the cost of the robot moving to the new starting point included, the switch is performed before restarting the milling.

Because of the reduced complexity provided by the proposed solution to the minimization problem for the complete planning problem, the implementation becomes time efficient. For reasonably sized pockets, the optimization problem is possible to solve within less than 1 ms on a standard desktop computer. Thus, no time has to be wasted during the milling task by waiting for the path planning to be performed.

Acquisition of Learning Data In order for the learning algorithm to work, an initial set of training data for each milling type is required. This can be obtained by simply performing each milling approach a given number of times. Since Gaussian distributions are assumed for the time for performing the milling and the standard deviation of the process force, at least two points per type are needed. However, since the training experiments are performed in material that would be used to mill the pockets, both material and time are wasted during the training procedure. It is therefore desirable to develop an alternative auto-training algorithm, that acquires training data automatically while milling the desired pockets. The proposed strategy to performing auto-training is defined as follows:

For each milling type γ , a training set S_γ is stored. If the training set S_γ is empty, the mean of the milling durations is assumed to be a small number close to zero. This way, once a certain milling approach has been performed, a significantly longer duration will be obtained, and the algorithm will consequently switch to a different type until all types have been performed once. However, as this only gives an initial training set of one data point per milling type, the sensitivity to outliers becomes high. The auto-training is therefore further extended, such that if the standard deviation of the set is zero, *i.e.*, when there is only one data point, the mean is divided by

a factor η . This will result in the same behavior as before; the algorithm will try all milling types once again, unless the duration for one type is actually η times shorter than another type. The parameter η can be tuned based on the expected frequency and range of outliers. An outlier-detection algorithm is used for the data stored in S_γ , which disregards data points that fall outside an interval defined as the standard deviation multiplied by a predefined factor, conditioned that the number of data points is larger than a given value. However, any outlier detection algorithm can be used.

Also, in order to account for changes in the milling conditions, *e.g.*, tool wear, forgetting action is introduced in the algorithm. The data points in the sets S_γ are multiplied by a tapered cosine-window of appropriate length, such that older points are gradually weighted less. The calculations of mean and standard deviation for each of the sets S_γ are modified to account for the weights. Thus, should an abrupt change occur, the algorithm will gradually adapt to the new conditions. The adaptation rate depends on the length of the window, where a short length will give a fast rate but increased sensitivity to outliers.

Extensions of the Path-Planning Problem As extensions of the pure time-minimization problem, there are several other parameters that could be of interest to consider when determining the optimal path. The expected value $\mu(\sigma_{f,\gamma})$ of the standard deviations of the milling forces for milling strategy γ is of interest, since it could be beneficial to optimize the path planning so as to avoid milling approaches that exhibit process forces with high variance. Also, the standard deviation $\sigma(t_\gamma)$ of the milling durations for the different milling types could be added in the cost function, in order to be able to either avoid or choose high-variance milling types. The extended cost function considered for minimization at segment n is therefore

$$\frac{\mu(t_{n,\gamma_n}) + t_{n,r}}{l_n \xi_n d} + w_1 \mu(\sigma_{f,\gamma_n}) + w_2 \sigma(t_{n,\gamma_n}), \quad \gamma_n \in \Psi(p_{n,r}, P_n), \quad (7.35)$$

where w_1 and w_2 are weighting factors. In order to minimize this cost function, it is straightforward to modify the algorithm proposed in Section 7.2 in order to incorporate the new terms.

7.3 Results

In this section, the proposed methods are evaluated in both simulations and experiments. First, the setup used for performing the milling experiments is described, followed by performance results of the proposed force-control architecture. Second, results from applying the learning algorithm in combination with force control are presented, for a large set of different scenarios.

Experimental Setup

Experiments were performed using an ABB IRB140 robot [ABB Robotics, 2015b] with an IRC5 controller, in combination with an open robot control extension called *ExtCtrl/ORCA* [Blomdell et al., 2010], running at 250 Hz. The proposed method was implemented in MATLAB Simulink, where the simulation models were translated to C-code using *Real-Time Workshop* and compiled in order to run them on the extended robot system. The robot was equipped with a wrist-mounted JR3 100M40A force/torque sensor [JR3, 2015], measuring forces and torques in the Cartesian directions. A Solectro UFM 1050 milling spindle [Solectro, 2015] with a maximum revolution speed of 11 000 rpm, was equipped with a 6 mm diameter end mill with two teeth. The workpiece was rigidly fixed in the robot workspace, such that the workpiece was aligned with the horizontal plane of the robot coordinate system. The full experimental setup is displayed in Figure 7.4.

Controller Performance – Simulations

Simulations were performed using MATLAB Simulink, by discretizing and implementing the models and the control architecture described in Section 7.2. The inner controller was designed by choosing the weight matrices Q_v and R_v such that the control bandwidth is not significantly decreased and that damping is introduced, in order to avoid oscillations or overshoots in the velocity response. The outer force controller was designed to give as fast response as possible, without any oscillations, *i.e.*, ideally purely real poles of the closed-loop system.

For the simulations, a material stiffness $K_f = 75$ N/mm was assumed, and the initial values for κ and β were set to 0.4 and 0.5, respectively. In order to obtain a feasible estimate of κ and β prior to activating the outer LQI controller, the velocity reference was initially set to a constant speed, $v_{\text{init}} = 15$ mm/s, so as to provide excitation for the Kalman filter. Once the covariances of the estimates passed below the predefined thresholds, the outer controller was activated and its control signal was added to the initial velocity reference.

A simulation with a desired force $f_d = 10$ N and time-varying κ and β was performed; the results are displayed in Figure 7.5. The figure shows a fast and well-damped force response without stationary error and accurate estimation of κ and β . Also, the influence of the time-varying parameters is barely visible in the force response.

Controller Performance – Experiments

For the experiments, the material stiffness K_f was interpreted as the interaction stiffness, because of the fact that the machining forces not only depend on the material properties, but on the combined stiffness of the

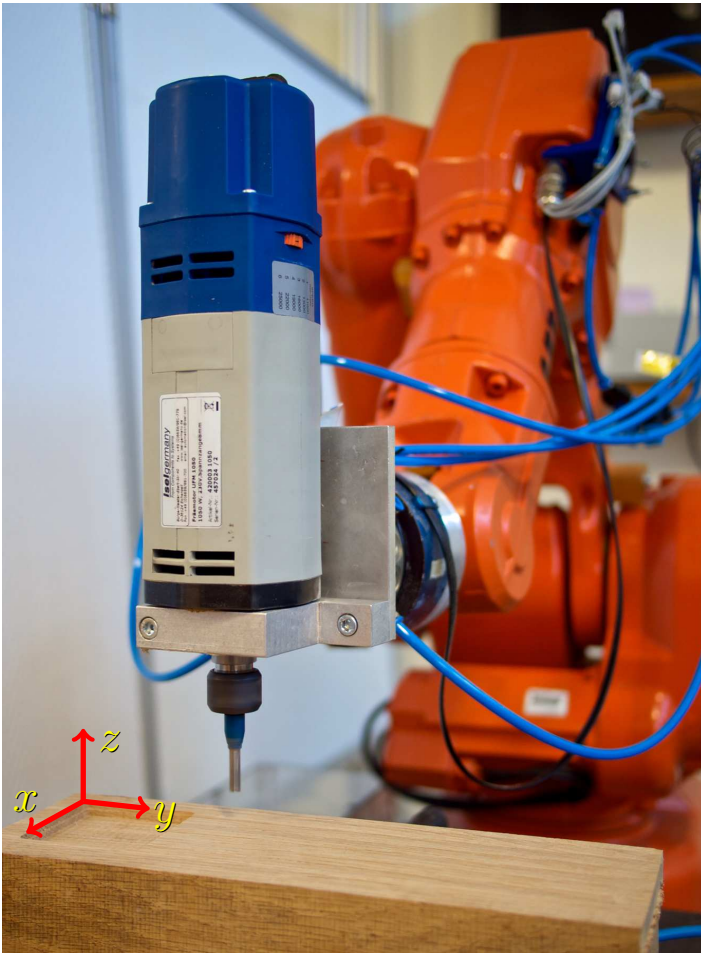


Figure 7.4 Experimental setup for performing force-controlled milling with an ABB IRB140 robot, where the milling coordinate system is shown in red.

workpiece, the tool, and the robot. It is to be noted that the stiffness of the robot in this context refers to the perceived robot stiffness, as determined from the measurements of joint motor angle. It is further assumed that the interaction stiffness is assumed constant within the limited workspace of the milling process. By measuring force and position of the robot during a simple contact experiment, the interaction stiffness was determined as 77 N/mm. A milling experiment with $f_d = 10$ N, which involves several segments with different milling types was performed, and the resulting control performance

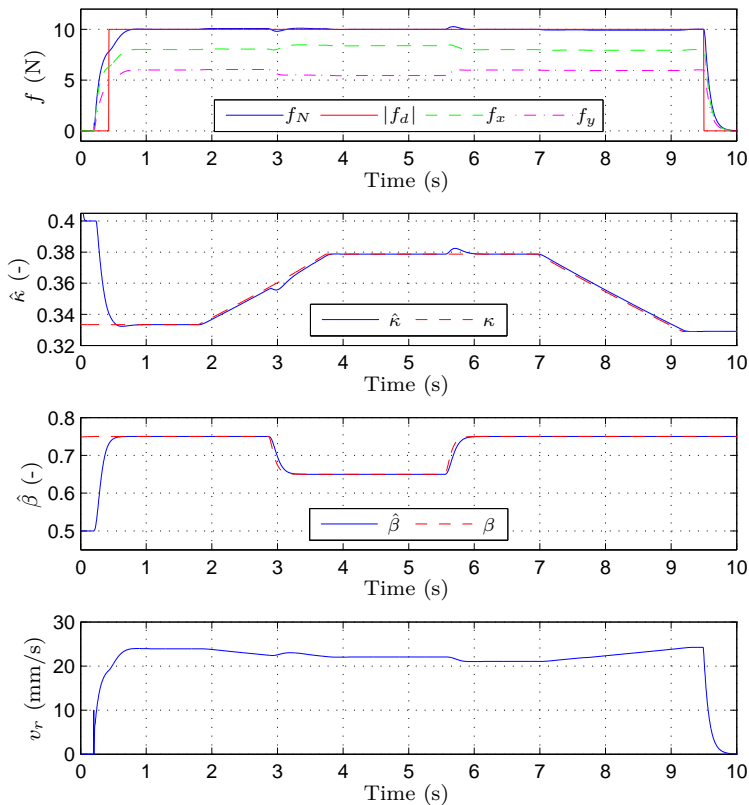


Figure 7.5 Force-controller performance simulation. The top panel shows $|f_d|$, f_x , f_y , and f_N in red, green, magenta, and blue, respectively. The second panel shows the actual κ in red and $\hat{\kappa}$ in blue, the third panel shows the actual β in red and $\hat{\beta}$ in blue. The bottom panel shows the feed rate v_r .

is displayed in Figure 7.6. Similar to the simulation, the experiment shows a fast force response with a measured force close to the desired value. It is to be noted that position-controlled milling is performed during the transitions between the segments, which explains why the force does not go to zero in between the force controlled milling segments.

Learning-Algorithm Simulations

A simulation model was constructed in MATLAB Simulink, where data from initial milling training tests were used for the milling durations t_{n,γ_n} for each segment. Five different scenarios were considered in simulation:

1. Perform training and mill three pockets, time minimization.

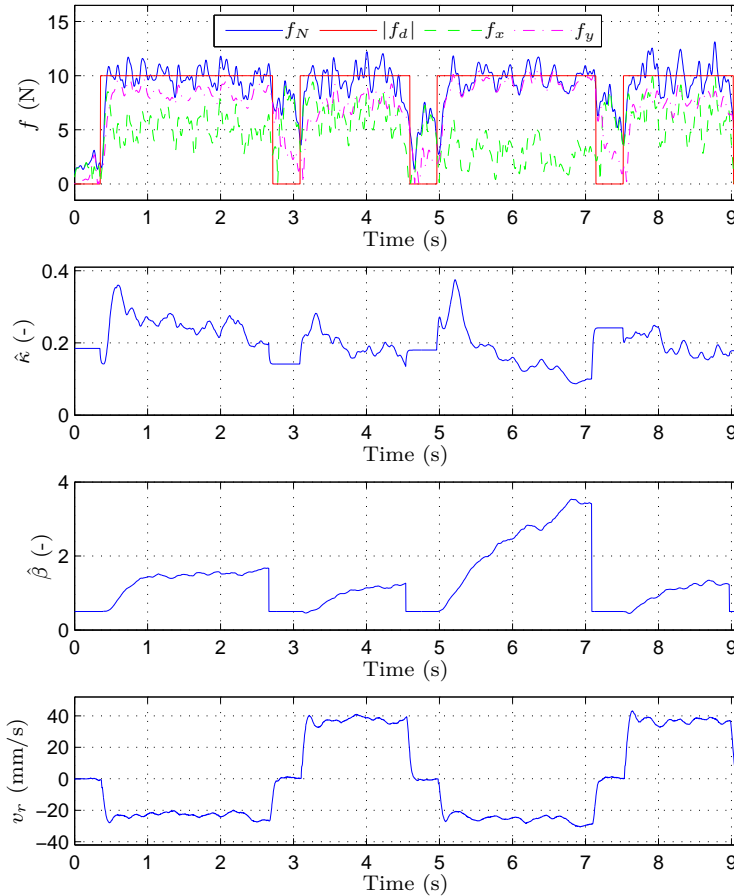


Figure 7.6 Force-controller performance during several segments of milling in oak. The top panel shows $|f_d|$, f_x , f_y , and f_N in red, green, magenta, and blue, respectively.

2. Mill three pockets using initial training data obtained in Simulation 1, minimize time and variance of milling durations.
3. Mill three pockets using auto-training, $\eta = 2$, time minimization.
4. Mill three pockets using auto-training, $\eta = 1.25$, time minimization.
5. Mill three pockets using auto-training, $\eta = 2$, time minimization, where the milling conditions are modified such that it is five times more time consuming to mill in the positive directions.

Table 7.1 Simulated milling durations, in seconds.

	Sim. 1	Sim. 2	Sim. 3	Sim. 4	Sim. 5
Training	69.2	-	-	-	
Pocket 1	28.3	34.3	35.0	35.0	76.5
Pocket 2	28.3	30.0	34.2	29.6	39.0
Pocket 3	28.3	28.3	28.3	28.3	38.6
Pocket mean	51.4	30.9	32.5	31.0	51.4

For Simulation 2, Gaussian noise with standard deviation $\sigma_s = 0.1$ was added to the milling durations, in order to achieve a more realistic result. For the other simulations, no noise was added such that repeatable and comparable results could be obtained. The resulting durations for the simulations are presented in Table 7.1. Figure 7.7 shows the milling path obtained from Simulation 1, where it can be seen that the optimal path is a combination of full and two-thirds coverage. The optimality is confirmed by performing a brute-force search over the possible milling paths for a given data set, which in this case corresponds to over 10^8 different paths. The obtained path and minimum milling time are identical to what the proposed algorithm provided. The search takes several hours on a standard desktop computer, which makes it infeasible to use in a practical setup, and thus it is only used for verification of the proposed algorithm.

The aim of Simulation 2 is not only to minimize time, but the variance of the milling durations. If it is assumed that the higher value of the two initial data points is an outlier, then the milling type in question should theoretically not be used, if the cost of the lower point would not be considered to begin with. The weight w_2 could be tuned based on this reasoning, where the corresponding weight would be given as

$$w_2 = -\frac{1}{\sqrt{2}}. \quad (7.36)$$

The negative sign of w_2 is chosen such that the cost is reduced for milling types with high variance. This will make the algorithm choose milling strategies that may not be time optimal, in an attempt to lower the variance. However, in order to make the effect of minimizing the variance more apparent, a higher weight was used for the experiments. Also, as some milling types may naturally exhibit high variances, a variable weight that increases the cost for each new data point was introduced in the cost function:

$$w_2 = -\frac{5}{k_\gamma} \quad (7.37)$$

where k_γ is the number of data points in S_γ for the current milling type γ .

By dividing by k_γ , the cost reduction is decreased for each new point, so that milling types that still exhibit a high variance after many data points do not get a cost reduction. In order to emphasize the effect, outliers were arbitrarily added to the initial training data on a subset of the milling types, denoted $\bar{\gamma}$. The initial standard deviations for these milling types are given by

$$\sigma(t_{\bar{\gamma}}) = [1.18 \quad 1.36 \quad 2.18 \quad 1.26 \quad 1.49] \text{ s}, \quad (7.38)$$

and the obtained standard deviations after the simulation were

$$\sigma(t_{\bar{\gamma}}) = [0.06 \quad 0.05 \quad 0.07 \quad 0.11 \quad 0.15] \text{ s}. \quad (7.39)$$

It is clear from (7.38)–(7.39) that the standard deviations have been significantly reduced, on average 94%. As a result, the milling durations have increased, as can be seen in Table 7.1. However, the same duration as in Simulation 1 could still be obtained after three pockets, since none of the types that initially exhibited a high variance turned out to lower the total cost.

In Simulation 3, with the use of auto-training, the optimal path was achieved after performing only three pockets, see Figure 7.8. It can be seen in the figure, that for the first two pockets the starting point is switched several times in order to gather training data for all milling types. Looking at the mean duration per pocket, it is clear that the auto-training in Simulation 3 was more time efficient than the one in Simulation 1, since it achieved a 40% decrease in duration. In Simulation 4, the value of the parameter η was decreased from 2 to 1.25, which gave an even lower mean duration per pocket, since slow milling types are rejected faster.

In Simulation 5, it is significantly more time consuming to mill in the positive direction. It is therefore to be expected that only negative direction milling segments are to be performed, *i.e.*, $t_B < t_A$ will be true, as discussed in Section 7.2. Because of the fact that the positive direction is set to have milling times more than η times longer than the negative direction, the algorithm converges after only one pocket. The obtained milling path has, as expected, converged to excluding all positive direction segments and lifting the tool instead, see Figure 7.9. This will naturally increase the duration compared to Simulation 1, 3, and 4, since the milling-duration parameters have been modified.

Evaluation of Learning in Experiments

Five different scenarios were considered for milling experiments:

1. Perform training and mill three pockets, time minimization.
2. Mill three pockets using the initial training data obtained in Experiment 1, minimize time and variance of milling durations.

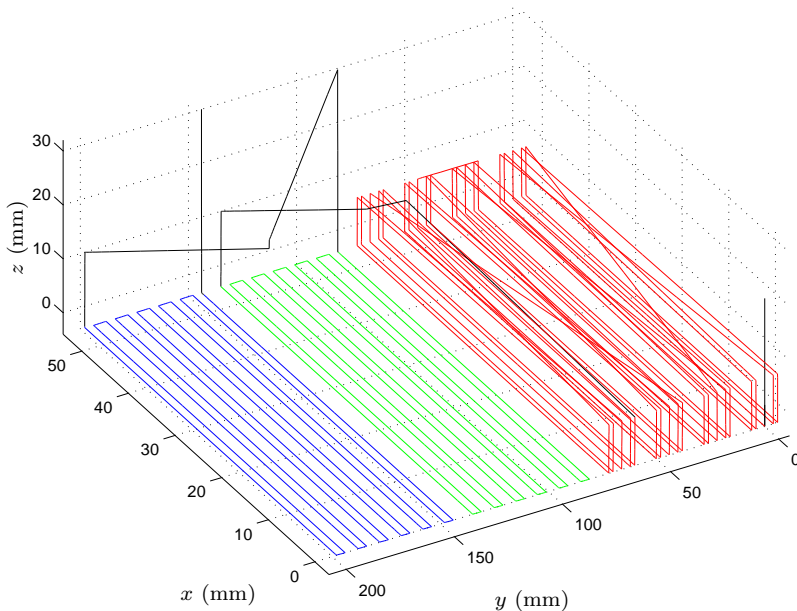


Figure 7.7 Milling path for Simulation 1, where the black lines represent the transitions between training and/or pockets. The training phase is shown in red, pockets 1 and 2 in green and blue, respectively.

3. Mill three pockets using auto-training, $\eta = 2$, time minimization.
4. Mill three pockets using auto-training, $\eta = 2$, minimize time and process-force variance.
5. Mill three pockets using manual path planning, without any knowledge of the milling process. No learning is performed.

Experiments were performed by milling 60×60 mm pockets in a block of oak, with a 3 mm depth of cut and a force reference of 10 N. The resulting durations for the millings are presented in Table 7.2.

The overall results presented in Table 7.2 show that the algorithm is successful, as milling durations consistently decrease for each pocket in the respective experiment. Also, a minimum in milling duration appears to have been achieved, since all time-minimization experiments result in approximately the same value. The results of Experiments 1 and 3 exhibit close agreement with the corresponding simulations, and again confirm that the auto-training is capable of reducing the mean duration per pocket (35%) while still achieving the minimum duration obtained for the pre-trained pockets. For Experiment 2, the same weight w_2 as in the corresponding

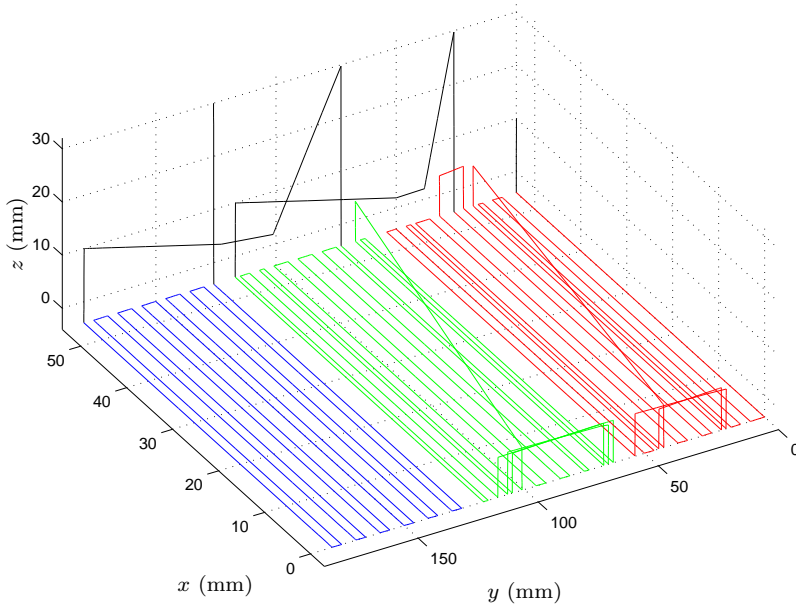


Figure 7.8 Milling path for Simulation 3, where the black lines represent the transitions between pockets. Pockets 1–3 are shown in red, green, and blue, respectively.

Table 7.2 Experimental milling durations, in seconds. Note that no learning was performed in Exp. 5.

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Training	59.0	-	-	-	-
Pocket 1	24.9	24.5	32.9	30.8	42.8
Pocket 2	23.8	23.9	28.6	29.1	27.5
Pocket 3	23.6	23.7	23.7	28.0	31.0
Pocket mean	43.8	24.0	28.4	29.3	33.8

simulation was used, see (7.37). The initial training data obtained in Experiment 1 was used, which is given by

$$t_\gamma = \begin{bmatrix} 6.6 & 6.3 & 11.7 & 7.0 & 9.8 & 7.1 & 8.0 & 9.9 & 7.2 & 9.5 \\ 5.9 & 5.8 & 10.4 & 6.5 & 8.1 & 7.0 & 8.3 & 8.6 & 7.6 & 8.9 \end{bmatrix} \text{ s.}$$

The standard deviations of the milling durations for each γ in the sets S_γ are given by

$$\sigma(t_\gamma) = 10^{-1} \begin{bmatrix} 5.2 & 3.4 & 9.2 & 3.4 & 12.0 & 1.3 & 2.4 & 8.9 & 2.9 & 4.5 \end{bmatrix} \text{ s.}$$

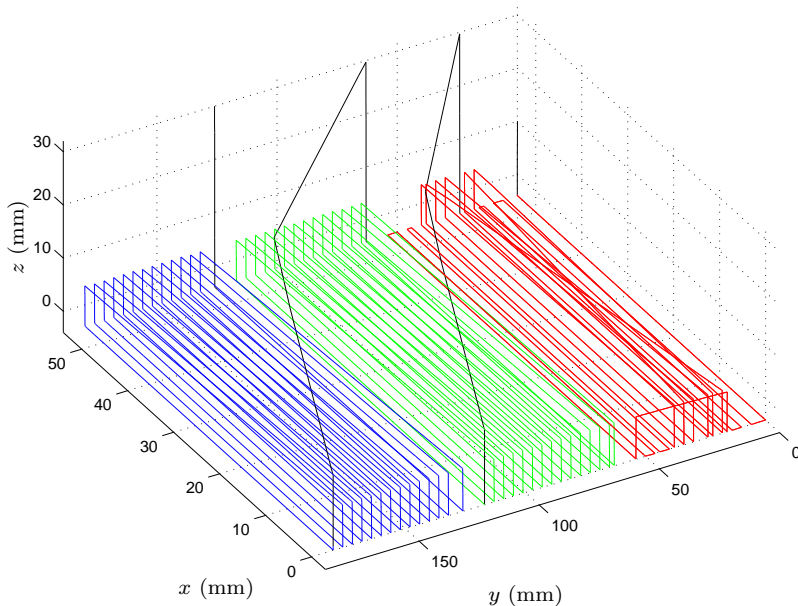


Figure 7.9 Milling path for Simulation 5, where the black lines represent the transitions between pockets. Pockets 1–3 are shown in red, green, and blue, respectively.

As no artificial outliers were introduced, only a few of the standard deviations exhibited significant values. However, even with the cost reduction achieved by using the weight (7.37), the cost was still not low enough for the algorithm to try the milling types with the highest standard deviations. Thus, no significant reduction in standard deviation was obtained, and the result of Experiment 2 turned out similar to Experiment 1.

For Experiment 4, a weight $w_1 = 5$ was used for the cost function (7.35). The standard deviations of the milling process forces vary between approximately 0.6–8 N, over the different milling types. For a pocket using only time optimization, the mean of the force standard deviations was 1.7 N. Comparing this to the result of Experiment 4, where a mean of 1.2 N was achieved, a decrease of 29% is noted. As can be observed in Table 7.2, the duration for milling the pocket has increased by a few seconds. This is expected since the algorithm is no longer solely striving for time optimality.

For Experiment 5, three manually planned milling paths were produced and carried out using only force control, without any learning. The paths are visualized in Figure 7.10. As seen in Table 7.2, the milling durations obtained from Experiment 5 vary over a large range. Comparing these durations to

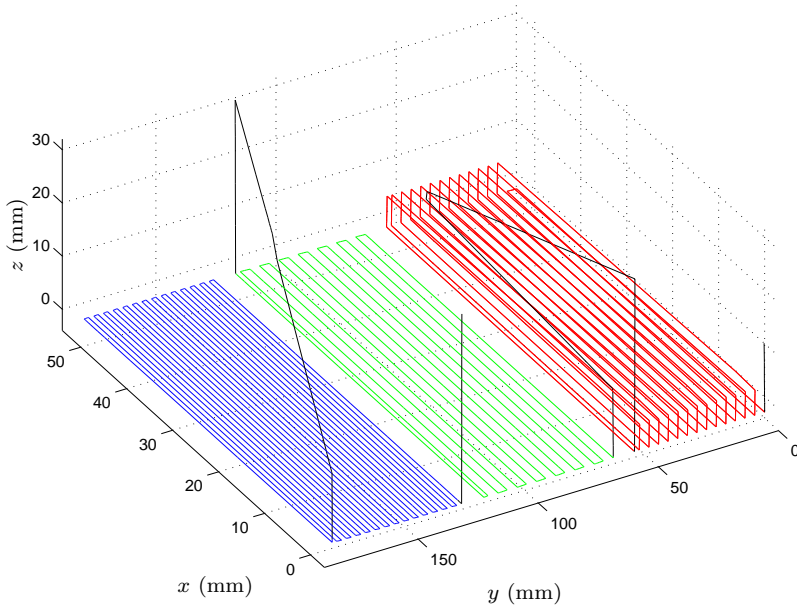


Figure 7.10 Milling path for Experiment 5, where the black lines represent the transitions between pockets. Pockets 1–3 are shown in red, green and blue, respectively.

the achieved minimum duration of 23.7 s, cycle-time reductions from 14 % up to 45 % were achieved, and on average 28 %.

For further comparison, position-controlled millings were performed using conservative feed rates, calculated as the minimum velocity for each milling type, obtained during a force-controlled pocket milling. Using the same path-planning as obtained with the learning algorithm, an average duration of 27.6 s was achieved for milling pockets using position control. This corresponds to a cycle-time reduction of 14 % with the use of force control.

An example of how a workpiece looks after milling is displayed in Figure 7.11. In this particular workpiece, training was performed, as can be seen in the top left, as well as three pockets in three layers.

7.4 Discussion

The proposed force-control strategy proved to be efficient in maintaining the desired force reference, and the cycle time could be reduced by approximately 14%. As also shown in [Lauderbaugh and Ulsoy, 1988; Sörnmo et al., 2012a], the use of an adaptive controller was beneficial, since both κ and β

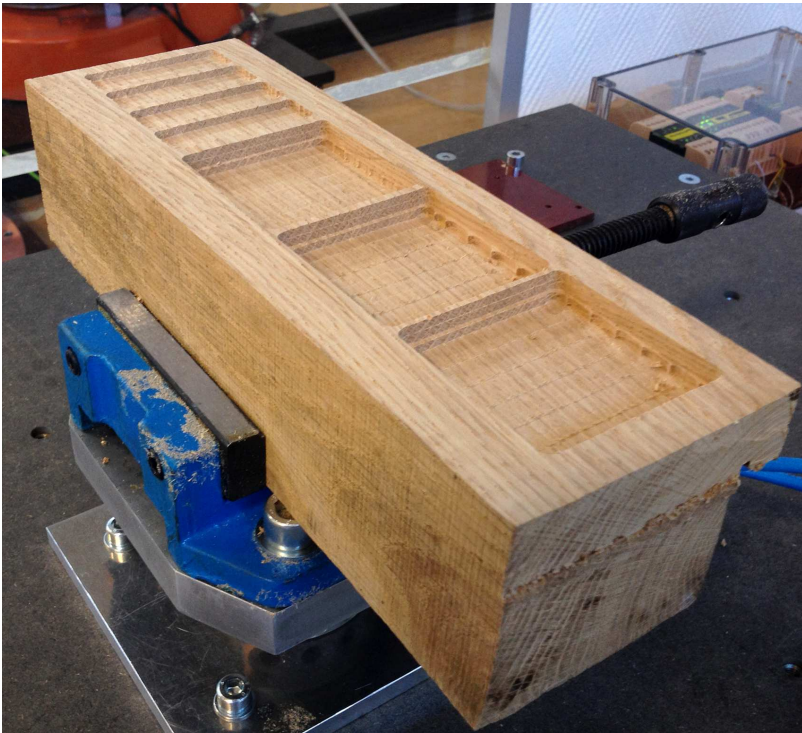


Figure 7.11 The resulting workpiece after performing several milling experiments.

vary significantly between experiments, as seen in Figure 7.6. Furthermore, for milling with a varying vertical depth of cut, the cycle-time reductions obtained by using force control would be greater, because of the need for an even more conservative feed rate when using only position control. However, this would require modification of the learning algorithm to account for the varying vertical depth of cut.

Force control for material removal maximization has thoroughly been researched in the past [He et al., 2007; Liu et al., 2001; Daneshmend and Pak, 1986; Lauderbaugh and Ulsoy, 1989; Rober and Shin, 1996; Wang et al., 2008; Lauderbaugh and Ulsoy, 1988], as mentioned in the introduction. The force-control structure proposed in this chapter is systematically derived and model based, without the need for complicated nonlinear modeling of the cutting process with a large set of parameters. It was shown in this chapter that a simple model such as the first-order LPV model is capable of achieving high performance. While the utilized models are simple, the proposed force-control structure is of rather high complexity. Less complex

controllers, such as PID or impedance controllers, may be used as an alternative. However, depending on the properties of the robot dynamics, sufficient damping or bandwidth may not be possible to achieve with less complex controllers.

For the learning algorithm, both the results of Simulation 1 and Experiment 1 showed that it is successful in learning the path planning that minimizes time, as the durations decrease for each pocket. In simulations, the optimality was confirmed by performing a brute-force search. However, in experiments it is naturally difficult to know if a minimum really has been achieved, since the optimal path is not available nor possible to derive. From Table 7.2 it can be seen that the minimum time for several different experiments was obtained between 23.6 and 23.8 s, which indicates that a minimum has been achieved. Some variation in the results is to be expected, because of the organic nature of the material.

In contrast to Simulation 2, where artificial outliers were introduced, Experiment 2 did not show any significant reduction of the standard deviations. This can be explained by the fact that the milling strategies that exhibited a high variance also exhibited long milling durations. In this case, using the weight defined in (7.37), which initially corresponds to a reduction of 2.5 standard deviations in cost, was still not low enough to be considered by the algorithm. This is a desirable effect, since it means that even if the milling type would be tried, it is highly unlikely that the result will be 2.5 standard deviations below the previous mean. However, in the unfortunate case of obtaining two outliers in the initial data, the algorithm will be unable to identify this and will consistently reject that milling type. Further, a more conservative weight w_2 could probably have been used, such as the weight proposed in (7.36), if minimizing the durations for pockets 1 and 2 also was of interest.

Experiment 3 utilized the proposed auto-training algorithm, which was proven to be effective since the training duration was decreased significantly, and no material was wasted during the training. It was shown in simulations that a faster convergence to the optimal path could be achieved by using a lower value of the parameter η . This is a result of the algorithm rejecting milling types that take a long time to perform, based on the first and only data point obtained for that type. This renders the algorithm vulnerable to outliers, since the algorithm is acting on a single data point, which could possibly be an outlier. Thus, choosing η becomes a trade-off between convergence rate of the auto-training and the sensitivity to outliers in the training data. Consequently, with a highly homogeneous material, $\eta \approx 1$ can be used to increase the convergence rate. Conversely, for workpieces with a large amount of knots and twigs, $\eta \geq 2$ should be used to avoid acting on non-representative training data.

In Experiment 4, a 29% reduction of the process-force standard deviation

was achieved. This reduction could be desirable since it may prolong the lifespan of the milling tool, by preserving the sharpness of the tool. This could in turn lead to reduced operating costs because of fewer tool changes. However, as presented in Table 7.2, the duration for each milling has increased in comparison to the pockets milled using the minimum-time approach. Thus, it becomes a trade-off between duration per pocket and tool wear, determined by the choice of the weight w_1 .

Intuitively, it could be expected that the minimum-time milling path should be a combination of the milling types with full coverage, since it represents the geometrically shortest path. This was indeed the case for a few experiments. However, for most experiments the optimal path was found to be combinations of milling types with coverage $1/3$ and $2/3$. The obtained variation can be explained by the natural variation of the wood workpiece, *e.g.*, the direction of the wood fibers. This confirms the claim that path planning for milling in non-isotropic materials is not straightforward and shows that the proposed learning algorithm is beneficial for such scenarios.

A possible limitation of the proposed learning algorithm, is the assumption that the workpiece properties are spatially invariant. For the experiments presented in this chapter, the assumption appears to be valid, and any deviations are, as intended, handled by the outlier detection algorithm. However, in scenarios where considerably larger pockets are to be milled, there may be local areas of the workpiece exhibiting different properties. Thus, if the training is performed in a non-representative area of the workpiece, the results may become sub-optimal. A possible remedy for this could be to increase the forgetting rate, such that the algorithm adapts to the current conditions more rapidly. Another possible limitation of the proposed method is that only the pocket milling geometry is considered. However, since rough-cut milling is considered, pockets are sufficient for most scenarios. If not, it is possible to extend the method to consider additional geometries.

For performing milling in other materials than wood, *e.g.*, aluminum or steel, the proposed method could also be used, where the force control would be just as effective as shown in this chapter. The learning algorithm would likely also be effective, since cutting conditions vary with different coverages. Because of the isotropic properties of metal, the algorithm could probably be simplified, since milling types for one direction could theoretically be merged with the corresponding types of different directions. Even though the evaluation was performed using an industrial robot, the method itself is not robot specific and can be applied to, *e.g.*, machine tools as well.

Since this chapter considered a rough-cut milling process, the milling accuracy was not the primary objective. However, should the milling accuracy be of interest, it would be appropriate to introduce $\hat{\beta}$ in the cost function, since strong transversal forces give rise to robot deflections and in turn low

path accuracy [Zhang et al., 2005b]. Adding a new term that penalizes high values of $\hat{\beta}$ will thus cause the algorithm to avoid milling types that exhibit strong transversal forces. However, since the resulting force is controlled to a constant reference, it is not unlikely that the fastest milling types exhibit small transversal process forces, such that the majority of the process forces is found in the feed direction. Thus, milling types with strong transversal forces are inherently avoided even in the pure minimum-time planning.

Previous research in the area of optimizing path planning for machining operations has mostly focused on minimizing the path length [Wang et al., 1987; Yang and Lee, 2002; Park and Choi, 2000; Lin and Koren, 1996; Lo, 1999], *i.e.*, finding the shortest geometrical path that completes the operation. In contrast to the present chapter, these papers do not consider the machining process dynamics. In [Meng Lim and Menq, 1997], geometric maximum feed-rate maps are established based on force models, in order to optimize the path-planning. While that method considers the machining process dynamics, it does not learn nor take into account possible model variations that would likely occur when milling in non-isotropic material. The method proposed in this chapter is independent of *a priori* information about the machining process that is needed to establish process-force models.

Future Research

As future research, it could be of interest to complement the forgetting action by introducing a tool-wear monitoring method, *e.g.*, as described in [Burke and Rangwala, 1991]. As briefly mentioned previously, it could also be interesting to consider the accuracy of the milling in combination with the time minimization. Another aspect for the future, could be to consider more advanced milling geometries, which may require additional milling directions, both vertical and horizontal. A four-directional path-planning approach is considered in Chapter 8.

7.5 Conclusions

This chapter considered the problem of minimizing cycle times for rough-cut machining processes in non-isotropic materials. A model-based adaptive force-control architecture combined with a learning algorithm to find the optimal machining path, was developed. The proposed method was verified in simulations and in pocket-milling experiments performed in oak.

The duration for performing a pocket milling was decreased by 14% with the use of force control, compared to position control. Furthermore, the milling durations were further decreased by employing the learning algorithm, on average an additional 28%.

An auto-training method was developed, which resulted in less waste of material and approximately a 35 % decrease in mean duration per pocket in experiments. The learning algorithm was extended to minimize process-force variance and the milling-duration variance, where reductions of 29 % and 94 % were achieved, respectively.

The proposed learning algorithm is independent of *a priori* information of the machining process, and can be initialized arbitrarily. Also, the method can be applied to machining in other materials, without the need for modifying any parameters.

8

Reinforcement-Learning Approach to Path Planning

8.1 Introduction

In this chapter, the path-planning problem from Chapter 7, where the objective was to minimize cycle times for machining operations in wood, is revisited. Further investigation of the solution proposed in Chapter 7 showed that sub-optimal paths were produced for a handful of scenarios. In this chapter, a reinforcement-learning approach based on Q-learning is taken, in order to obtain optimal paths for all scenarios.

Because of the non-isotropic properties of wood, additional cycle-time reductions may be achieved by considering a higher number of milling directions in the path planning. Therefore, the path-planning problem is extended to consider two additional directions. As the standard formulation of Q-learning is not feasible to use for this problem, a feature-based Q-learning approach is proposed.

Previous Research

An overview of reinforcement learning methods is provided in [Kaelbling et al., 1996; Barto, 1998]. Q-learning was first introduced in [Watkins, 1989], and the subsequent convergence proof is found in [Watkins and Dayan, 1992]. Feature-based Q-learning, also known as function approximation, is investigated in [Irodova and Sloan, 2005], where it was shown that similar results to that obtained using Q-learning can be achieved at considerably increased efficiency. Additional Q-function approximation methods are discussed in [Wiering and Van Otterlo, 2012].

As mentioned in Chapter 7, the considered two-directional path-planning problem is similar to the asymmetric traveling salesman problem (ATSP). A modified version of Q-learning, called ant-Q, for solving the ATSP is presented in [Dorigo and Gambardella, 2014]. However, for the specific milling

path-planning problem considered in this chapter, no previous publications exist to the author's knowledge.

Disposition

This chapter is organized as follows: In Section 8.2, both the two- and four-directional path-planning problems are considered. Simulation results from using both methods are presented in Section 8.3, and compared to results from the method proposed in Chapter 7. The results are discussed and conclusions are drawn in Section 8.4.

8.2 Method

In this section, both the two- and four-directional path-planning problems for cycle-time minimization are considered. Different learning algorithms are proposed for each problem, in order to obtain optimal paths for all possible scenarios.

Two-Directional Path Planning

The path-planning method presented in Chapter 7 was capable of finding the optimal milling path in the considered scenarios, which was verified by performing brute-force searches. However, in certain scenarios, it is found that sub-optimal paths are chosen because of endpoint constraints that cannot be taken into account by the algorithm. This will be further discussed in Section 8.3. In this section, Q-learning [Watkins, 1989] is employed to solve the two-directional path-planning problem to obtain optimal paths for all scenarios.

Q-Learning Q-learning is a model-free reinforcement-learning technique that is used to find the optimal policy, given a set of states and actions. Each state s has a set of actions a associated with it, which in this section corresponds to transitions between the states. In turn, a reward r is associated with each action. The optimal policy thus defines the path from the starting state to the ending state, which maximizes the obtained reward. In order to learn the optimal policy, a large amount of random transitions are performed through the state space, and based on the obtained data, a policy matrix $Q(s, a)$ is learned according to [Barto, 1998]

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a^*} Q(s^*, a^*)), \quad (8.1)$$

where $\alpha \in (0, 1]$ is the step size, *i.e.*, the learning rate, γ is a discount factor, and the *-notation denotes the subsequent future event. Once the Q -matrix

is learned, it is used to determine the optimal action a^* for each state. The future action from state s is commonly chosen as

$$a^* = \arg \max_a Q(s, a). \quad (8.2)$$

In order to represent the two-directional path-planning problem for pocket milling, a state space is formed. The milling direction, *i.e.*, the feed direction, is here denoted x and the transversal direction in the milling plane is denoted y . The workpiece is divided into a discrete set of y -positions, where each position is respresented by two states, corresponding to performing a milling segment in positive or negative x -direction, respectively. Since Q-learning strives to maximize the total obtained reward, the minimum-time optimization problem, which is considered in this chapter, is reformulated as maximizing the negative duration of a complete milling. Thus, the reward for performing a state transition is chosen as the negative milling duration for that segment. Consequently, all illegal state transitions are set to give a reward of $-\infty$.

In order for the Q-learning to work, the reward matrix must be static during the learning procedure, *i.e.*, independent of the order of the previously performed state transitions. To achieve this, the state space is divided into two parts, one for each direction of y . Since the durations of performing milling segments may change throughout the milling, the action rewards will change as well. Therefore, it must be possible to perform transitions between milling in the positive and negative y -direction, in order to maximize the reward. Because of this, an updated reward matrix with the current rewards has to be constructed after each state transition.

However, given the current rewards, transitions to a different milling direction are only allowed from the current state. Intuitively, this is reasonable since if the other milling direction is optimal with the new rewards, it is not possible that another switch in direction is to be performed with the current rewards. This is also required for the reward matrix to become independent of previously visited states. Because of the changes in the reward matrix after each state transition, the reinforcement-learning training procedure must also be performed after each transition. In order to enable the learning algorithm to find the optimal starting point, an additional state is introduced, which only allows transitions to the four corners of the state space.

As shown in Chapter 7, the use of different milling tool coverages in combination with different milling directions resulted in varying material removal rates. In this chapter, three different coverages are considered for each direction; full coverage, and half coverage for both up and down milling. In order to account for the different coverages, the state space is modified such that each state is split in several parts in the y -direction, depending

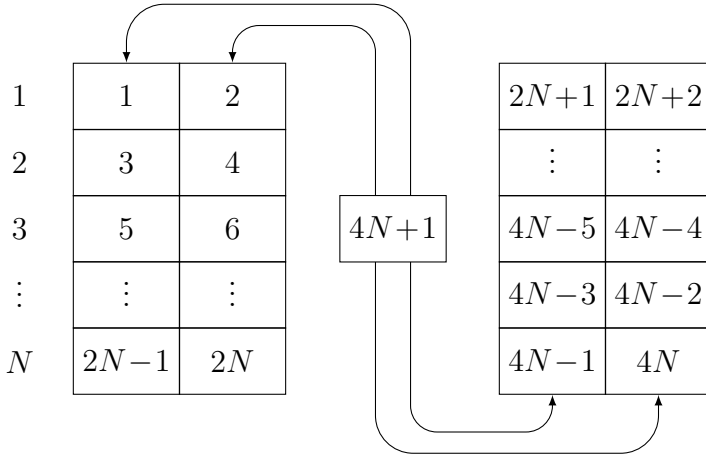


Figure 8.1 Illustration of the state space defined for the Q-learning algorithm. The numbers on the left specify the geometric position of the states in the y -direction. The state denoted $4N+1$ represents the starting state. Note that only the transitions from the starting state are illustrated in the figure.

on the desired coverage resolution. Here, the states are split in two as a result of the use of half coverage. With this modification, the tool occupies an additional adjacent state in y -direction for each state. Consequently, to perform a milling with full coverage, one row of states in the y -direction is only visited by occupying the adjacent states. However, milling is still performed in those positions. A vector P with binary elements is used to keep track of where milling has been performed, which is needed to update the reward matrix after each state transition. The complete state space for learning the time-optimal milling path for an $N \times N$ -sized pocket is illustrated in Figure 8.1. In the figure, performing milling in positive x -direction is represented by the left states in the two main state clusters, and vice versa. Also, the left state cluster represents milling in positive y -direction, and vice versa. Transitions between states within each column correspond to lifting the tool and moving to perform a milling in the same direction as the previous milling. The obtained reward for these state transitions includes both the time for lifting and moving the tool, and the milling duration.

With the proposed state space, the optimal policy for the current state is learned using (8.1), and the subsequent action is chosen and carried out according to (8.2). Once the action is carried out, a new reward matrix is formed, training is performed, and the procedure is repeated until P does not contain any zeros.

In this chapter, the two-directional Q-learning is referred to as Method 1.

Four-Directional Path Planning

While the two-directional path planning of the milling process was shown to be successful in reducing cycle-times significantly in Chapter 7, additional reductions may be achieved by considering a higher number of milling directions. However, the complexity of the path-planning problem increases dramatically with the addition of more directions. Thus, in this section, a different reinforcement-learning method for the four-directional path-planning problem is considered. In this setting, milling can be performed in both directions along the x - and y -axes of the coordinate system defined for the two-directional case. Additionally, the propagation direction of the milling, which is transversal to the direction of the milling segments, is denoted θ .

Simply dividing the state space defined in Figure 8.1 once more to account for the added directions is not possible, since forming an unambiguous reward matrix for this scenario is impossible. This is caused by the fact that the rewards change depending on the previous state transitions, in contrast to the two-directional problem. Thus, performing training to obtain a policy matrix Q will give ambiguous results. A different approach is to divide the workpiece into an $N \times M$ grid of positions, where each position is represented by a set of states, describing all unique combinations of the previously milled positions. With this state-space structure, the number of required states will grow rapidly as the grid size increases. In order to represent all possible states for a grid of size $N \times M$, the number of required states is given by 2^{NM} . This exponential increase in states renders the standard formulation of Q-learning impractical and computationally infeasible for larger grids. Thus, for the four-directional path-planning problem, a different approach is taken, which is described next.

Feature-Based Q-Learning As mentioned in the previous paragraph, the number of states required to represent all possible scenarios for a grid of positions quickly becomes significantly too large. In order to reduce the state space, feature-based Q-learning [Irodova and Sloan, 2005], also known as function approximation, is employed in this section. With this method, the states are replaced by positions, represented by a matrix P , where each matrix element is binary. In the matrix, a zero indicates that milling has not been performed and one indicates that milling has been performed in that particular point. In order to account for different tool coverages, similar to the two-directional case, the tool occupies additional adjacent positions. With the coverages previously defined in this section, the tool occupies a total of four positions in P for every position.

In order to transition between the different positions, a set of actions a are defined, *e.g.*, move one position north or two positions east in the matrix. Furthermore, a set of features F is defined, which based on the

current position, action, and P -matrix return a reward. The features can be chosen arbitrarily, but the choices are important in order to obtain the desired behavior of the learning algorithm. The idea is then to approximate the Q -function, *e.g.*, by forming a linear combination of the features. The resulting value for each action, denoted Q_w , is used to determine which action should be chosen. The linear approximation is written as

$$Q_w(a) = w_1 F_1(a) + w_2 F_2(a) + \dots + w_n F_n(a), \quad (8.3)$$

where w_1, w_2, \dots, w_n are scalar weights, initialized as random numbers in the interval $[-1, 0]$. Similar to (8.2), the optimal action is determined as the action that returns the highest value of Q_w with

$$a^* = \arg \max_a Q_w(a). \quad (8.4)$$

The algorithm is trained by finding the optimal values of the weights, by means of a gradient descent search, according to [Irodova and Sloan, 2005]

$$\delta = r + \gamma Q_w(a^*) - Q_w(a) \quad (8.5)$$

$$w_i^k = w_i^{k-1} + \alpha \delta F_i(a) \quad (8.6)$$

where α and γ are defined in the same way as for (8.1), and k is the iteration index. Training data are acquired by starting at one of the starting points at random, and subsequently performing actions based on the current $Q_w(a)$ until the complete task is done. This procedure is repeated a large number of times, until the values of the weights w have converged. As for the Q-learning method described for the two-directional problem, the rewards are chosen as the negative value of the milling duration for that segment. However, in contrast to the two-directional case, milling segments can vary in length. Thus, each reward is calculated based on the length of the segment, with acceleration and deceleration durations added.

While the feature-based Q-learning algorithm is fairly simple to implement, the procedure of feature engineering, *i.e.*, how to define the features, is non-trivial. The function Q_w , which is linear in F_i , must convey the best action to carry out, not just for the current position, but for the subsequent future positions as well. For example, in some scenarios it may be beneficial to choose an action that gives a low immediate reward, but which will ultimately result in a higher total reward. Since a unique weight is learned for each feature, irrelevant features will obtain small weights and thus not affect the choice of action.

The set of actions can be defined arbitrarily, but the choice of actions may affect the feature-engineering procedure. The natural choice of actions for moving in a four-directional grid of positions, would be to represent a

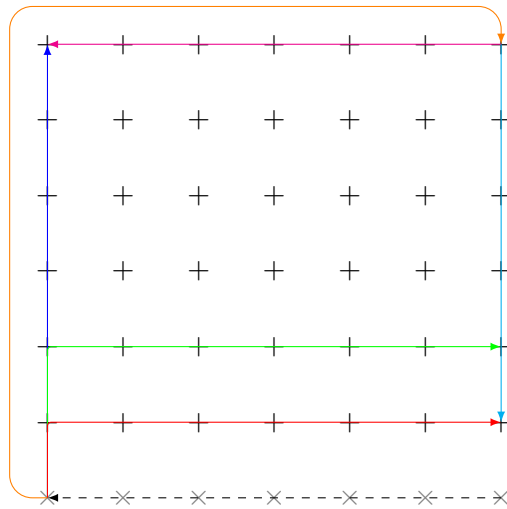


Figure 8.2 Illustration of a subset of the 32 actions for the feature-based Q-learning algorithm. The gray crosses and black plus signs represent visited and not yet visited states, respectively. The dashed line shows the previous action, and the solid lines represent possible actions from the current starting point. For clarity, only a subset of the possible actions are shown in the figure. The orange line represents an action that lifts the tool and moves to a new starting point. For each state, the adjacent three states in the south-east direction are occupied.

step in each of the four directions by separate actions. However, with this choice, it was found that the feature-engineering procedure turned out to be extremely complicated. Thus, the actions were redefined such that movement in each milling direction is performed until no more material remains in that direction, defined by the values of P . This is reasonable, as performing longer milling segments intuitively gives increased efficiency, because of the required acceleration time for each segment. Additional actions are formed as all possible combinations of moving one or two positions in each spatial direction of the matrix, then turning left or right and milling until no more material remains. This makes for a total number of 20 actions, for which milling is performed. In order to be able to switch starting point, actions that lift the tool and transition to the other possible starting points defined by P , are formed. For all combinations of starting points and coverages, the total number of actions is increased to 32. An illustration of a subset of these actions is displayed in Figure 8.2.

The described actions are defined such that it is not possible to divide the matrix P into separate submatrices of zero elements, *i.e.*, the zeros in P

will always form a rectangular matrix. The only exception to this is when an action that lifts the tool has been performed. The subsequent action is then constrained such that the zeros in P will form a rectangular matrix when it has been carried out.

Initially, two features are considered. The first feature is chosen as the normalized inverse duration for performing an action

$$F_1(a) = \frac{\varepsilon_1 Z_P}{t(a)} \quad (8.7)$$

where ε_1 is a normalizing constant, Z_P is the number zeros in the matrix P , and $t(a)$ is the duration of performing action a . If an action is not possible to perform, because of geometrical constraints or that milling has already been performed in the targeted position, the function $t(a)$ returns an infinite time. The inverse of $t(a)$ is chosen such that a short time gives a high reward, and a long time gives a low reward.

The second feature is chosen as

$$F_2(a) = \frac{\varepsilon_2 Z_P^2}{T_\Sigma(a)} \quad (8.8)$$

where ε_2 is a normalizing constant, and $T_\Sigma(a)$ is the estimated total duration for completing the milling task, given the action a .

For actions that do not change the propagation direction θ of the milling, the estimate of the complete milling task duration in that direction, denoted $t_{\text{me}}(\theta)$, is formed as

$$t_{\text{me}}(\theta) = \min_{a_1, a_2} \left(N_e(\theta) \frac{t(a_1) + t(a_2)}{\xi(a_1) + \xi(a_2)} \right) \quad (8.9)$$

where $N_e(\theta)$ is the number of columns/rows of zeros in the direction θ that remains in P . The set of actions that minimize (8.9) for each direction θ are denoted a_o . Since all directions can be reached by performing at most two actions, two additional estimates are formed for the actions that change the direction according to

$$t_{d1}(a) = t(a) + t_{\text{me}}(\tilde{\theta}) \quad (8.10)$$

$$t_{d2}(a) = t(a) + \min_{a^* \notin a_o} (t(a^*) + t_{\text{me}}(\tilde{\theta})) \quad (8.11)$$

where $\tilde{\theta}$ denotes the new direction, t_{d1} and t_{d2} denote the durations of using one or two actions to switch direction and complete the milling task. The function $T_\Sigma(a)$ is defined as

$$T_\Sigma(a) = \begin{cases} t_{\text{me}}(\theta), & a \in \theta, \quad a \in a_o \\ \infty, & a \in \theta, \quad a \notin a_o \\ \min(t_{d1}(a), t_{d2}(a)) & a \notin \theta \end{cases} \quad (8.12)$$

where an infinite time is given for actions that do not belong to a_o , such that the feature F_2 in (8.8) returns zero reward for these actions.

Additional features were not considered, as simulations showed that the proposed features were sufficient in approximating the Q -function. In order for the learning algorithm to be able to adapt to changes in milling durations, a short training phase is initiated subsequent to performing a milling segment, if a different duration $t(a)$ than expected was obtained. The previously learned values of the weights are used as initial values of w for the training.

In this chapter, the four-directional feature-based Q-learning is referred to as Method 2.

8.3 Results

In this section, simulation results obtained using the proposed methods are presented, in a set of different scenarios. Furthermore, a comparison to the method presented in Chapter 7 is provided. The same values of the milling durations for each of the segments are used for all simulations, and the durations are assumed to be constant throughout the simulation, unless stated otherwise. Also, all pocket dimensions are given in normalized units.

Two-Directional Path Planning

In order to evaluate the proposed method for two-directional path planning, a pocket with the dimensions 12×12 was initially considered. With this pocket size, the method in this chapter and the method presented in Chapter 7 produced identical paths, verified to be optimal by a brute-force search. However, changing the size to 13×13 , different paths were obtained using the two methods, see Figure 8.3. As a result of the uneven grid size, the two actions that proved most efficient, do not fit evenly in the grid. Consequently, a different action is needed to complete the task. This is where the two methods differ; Method 1 is able to identify the problem and learn what path to take in advance, in contrast to the method from Chapter 7, where the decision is made once the problem has been encountered. With the proposed method, an additional reduction in cycle time was achieved, as compared to the previous method, *cf.* the captions of Figure 8.3.

Four-Directional Path Planning

Training of the feature-based Q-learning algorithm was carried out as described in Section 8.2, in order to learn the optimal values of the weights w . The result of the training procedure is shown in Figure 8.4, where the values of the weights throughout the training are displayed. It is noted that both weights converge to a steady level. Considering the same 13×13 pocket as in the previous subsection, an identical path to Figure 8.3(b) was obtained

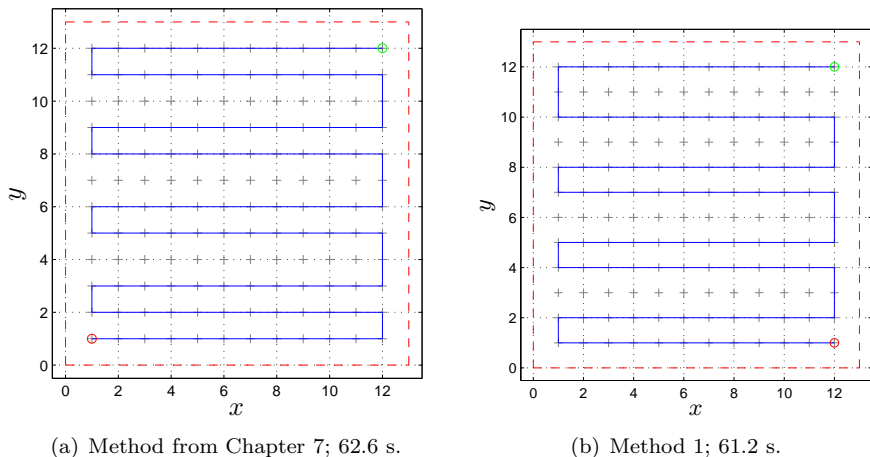


Figure 8.3 Visualization of path-planning simulations in a 13×13 grid, comparing the method from Chapter 7 to Method 1. The green and red circles indicate the starting and stopping coordinates, respectively.

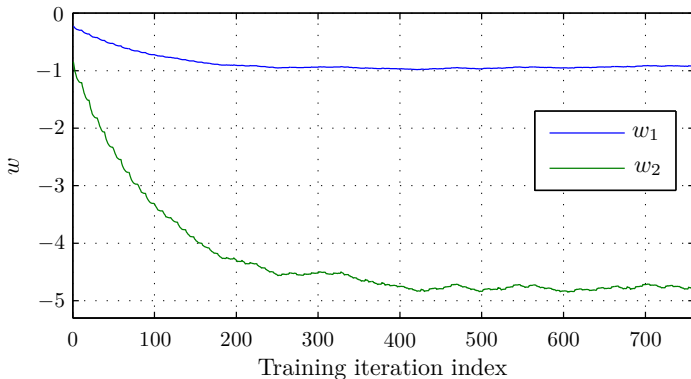


Figure 8.4 Training phase of Method 2, where the weights for each corresponding feature are learned iteratively.

for a simulation using Method 2. This is desirable as the milling durations in the other propagation directions are assumed longer in the simulation.

For further evaluation of the proposed methods, a larger pocket with the dimensions 18×18 , was considered. In this scenario, the same milling durations as in the previous simulations were used initially. However, as the path was executed, the durations of milling in the initially optimal direction were set

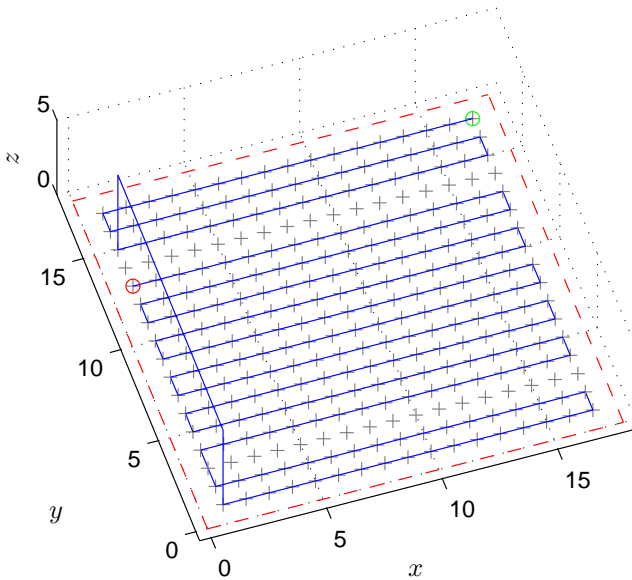


Figure 8.5 Visualization of a simulation in an 18×18 grid using Method 1, where the assumed milling durations in the initial θ -direction are increased by 40%. The total duration needed to perform the complete milling was 162.9 s. The green and red circles indicate the starting and finishing coordinates, respectively.

to increase by 40%. Consequently, the optimal direction is changed, and the algorithms must plan accordingly. It is assumed that the new duration of a milling segment is known after the segment has been performed once, *i.e.*, old durations for the segment are not taken into account. This is suitable for simulations, since no outliers are present.

The resulting paths from using Methods 1 and 2 are displayed in Figures 8.5 and 8.6, respectively. It is noted from Figure 8.5 that as the milling durations are updated, the actions that were initially found optimal are substituted because of the 40% increase in duration. Once three actions in the initially optimal direction have been performed, the algorithm determined it to be beneficial to lift the tool in order to switch θ -direction. For the four-directional planning case, the algorithm deemed it to be advantageous to switch direction already after carrying out the first action. This is explained by the fact that the θ -direction can be changed without lifting the tool, making it possible to achieve a higher efficiency by reducing the threshold for it to be beneficial to switch directions.

An additional simulation of a pocket was performed for each of the two

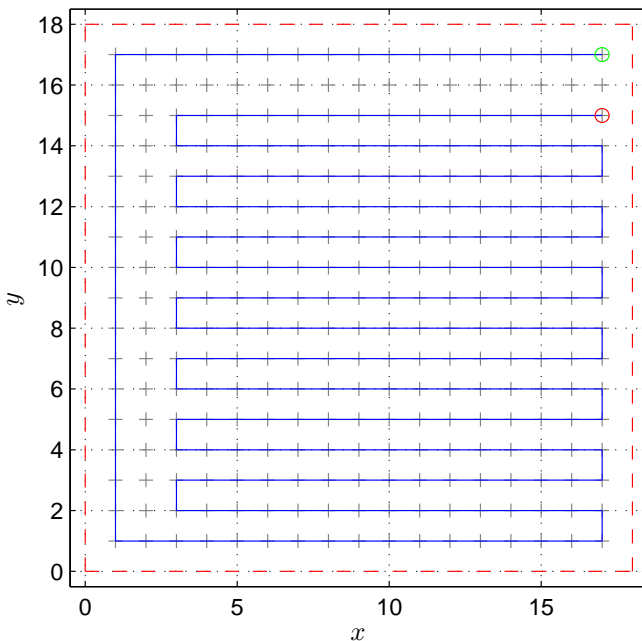


Figure 8.6 Visualization of a simulation in an 18×18 grid using Method 2, where the assumed milling durations in the initial θ -direction are increased by 40%. The total duration needed to perform the complete milling was 154.2 s. The green and red circles indicate the starting and finishing coordinates, respectively.

methods, in order to find the optimal paths with prior knowledge of the 40% increased durations in the previously optimal θ -direction. The paths obtained using the two methods are displayed in Figure 8.7, where it is noted that different directions of the paths were chosen in Figure 8.7(a) and Figure 8.7(b). This is simply because Method 1 only considers two different directions, whereas Method 2 can choose between additional two θ -directions. This naturally gives an advantage to Method 2, which is also reflected in the total durations of the two pockets.

Non-square pocket-milling simulations were performed using Method 2 to investigate the influence of the acceleration time of the robot for each segment. For this simulation, it was defined that it should be slightly more efficient to mill in both positive and negative x -direction, as compared to the y -direction. However, as the pocket is not square, the number of milling segments affect the total time, as a result of the acceleration time for each segment. Thus, two simulations were performed, one with fast acceleration and one with slow acceleration. The results of the simulations are displayed

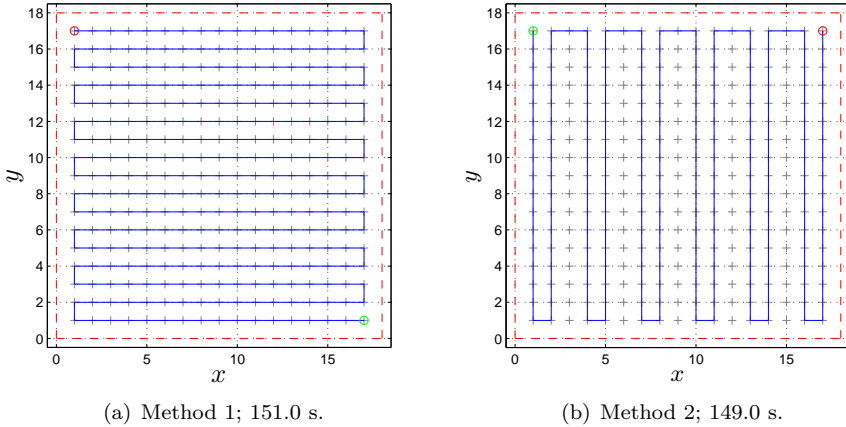


Figure 8.7 Visualization of simulations in an 18×18 grid, with knowledge of the increased milling durations in the initial θ -direction. The green and red circles indicate the starting and finishing coordinates, respectively.

in Figure 8.8. For the first simulation, it is noted that even though a significantly higher number of segments are performed compared to milling in the y -direction, it is still more efficient to mill in the x -direction. A total time of 69.4 s was obtained for the pocket. For the second simulation, the acceleration durations were increased by a factor of four for each segment, and therefore the algorithm chooses milling in the y -direction instead. The total time for the pocket is increased to 74.9 s, which is expected since the acceleration durations have increased. For comparison, the previously optimal path with the new acceleration times yields a total time of 76.7 s.

8.4 Discussion and Conclusions

The proposed Q-learning algorithm for two-directional path planning of machining tasks was found to improve the results obtained with the method from Chapter 7, as the optimal path was found for each considered scenario (verified by a brute-force search). Additionally, the problem was extended to incorporate two additional milling directions, rendering the problem significantly more complex. A feature-based Q-learning approach was taken, and the method was shown to achieve an optimal path for the same scenario used to evaluate the Q-learning method. For this scenario, it was concluded that the path was optimal since it was identical to the path produced by the Q-learning algorithm. However, since the number of possible paths for the four-directional problem becomes extremely high, optimality was investigated

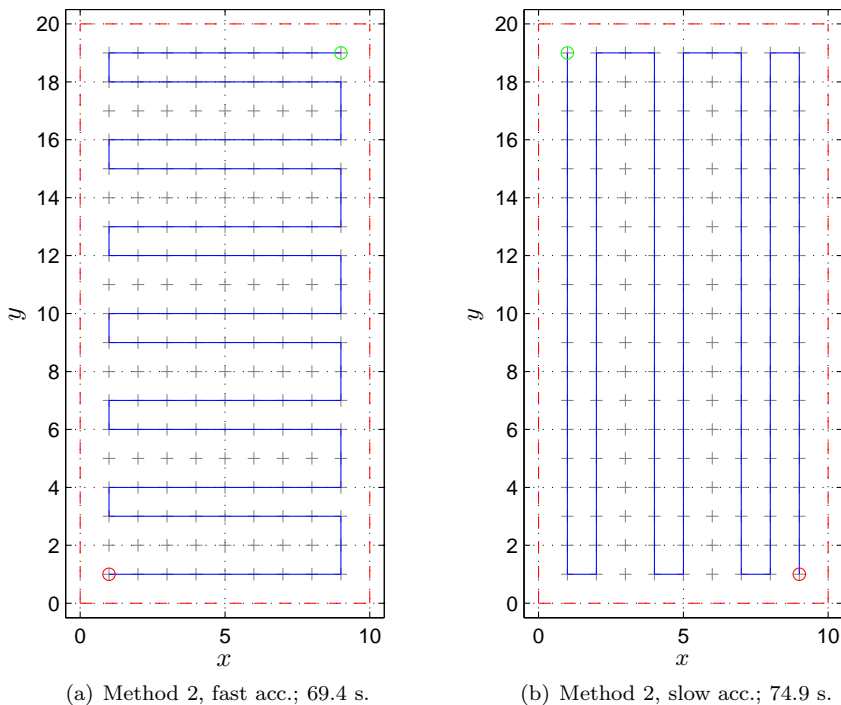


Figure 8.8 Visualization of path-planning simulations in a 10×20 grid, with fast and slow acceleration shown in the left and right plot, respectively. The green and red circles indicate the starting and finishing coordinates, respectively.

by performing selective brute-force searches for the added directions, where the obviously non-optimal milling directions and coverages were disregarded.

It was further shown that for experiments where the initial training data was partially incorrect, both proposed methods quickly adapted to find a new path. The four-directional method was, not unexpectedly, found to reduce the cycle-time even further than the two-directional method.

A limitation to the proposed method for four-directional path planning is that it does not consider irregular paths patterns or, *e.g.*, spiraling motions. This could be incorporated in the method by redefining the actions, and also adding the possibility to start in the middle of the pocket. However, it is unlikely that a spiraling motion could be optimal, because of the large number of short milling segments, where the duration of the acceleration reduces the efficiency.

A drawback of the methods proposed in this chapter, compared to the

method presented in Chapter 7, is the required computation time of the training procedures. The method from Chapter 7 was possible to run within each sample period of the robot controller, while the methods in this chapter require one to two additional magnitudes, depending on the size of the grid. However, in the context of milling segment durations, this can still be considered negligible.

An inherent limitation of feature-based Q-learning is that it relies on an approximation of the Q-function—optimality is not guaranteed. Nevertheless, the approximation is necessary, since the original problem would be infeasible to solve, as was shown in Section 8.2. Additional features can be included in order to improve the approximation. However, the proposed features were found to be sufficient for the considered application, as all presented simulations were found to produce optimal paths. Still, in scenarios that are not considered here, sub-optimal paths may be produced as a result of the approximation.

For future research, it would be interesting to perform milling experiments using the proposed four-directional path-planning method, and investigate how much the cycle time could be reduced as compared to that of the two-directional path.

9

Conclusions

This thesis has presented methods for adaptation and learning, applied to a set of manipulator tasks, mainly focused on machining operations. To the purpose of increasing the accuracy of machining tasks, three different approaches were taken.

First, a macro/micro-manipulator approach was considered, where two different robot cells were developed; a prototype cell for one-dimensional milling in aluminum, and a complete robot cell with an integrated approach to three-dimensional robotic machining. Modeling and control of the micro manipulator, as well as the design of an MVPC mid-ranging control structure was presented. For milling in aluminum, the absolute accuracy was significantly increased, the MAE was reduced by a factor of up to approximately 70, to a level lower than $2\ \mu\text{m}$ along each of the three axes. Additionally, the surface accuracy of the millings was increased by up to a factor of three. For milling in steel, a significant increase of the workpiece surface accuracy was also achieved; up to eight times measured by MAE values. For all milling experiments with compensation, the desired accuracy of $\pm 50\ \mu\text{m}$ was achieved. It was further shown that the mid-ranging control structure was essential in order to maintain active compensation throughout the duration of the milling.

As an alternative to the MVPC mid-ranging control, an adaptive IMC control structure was proposed for the macro/micro-manipulator setup, where the manipulators were treated as closed-loop systems containing internal velocity saturations. The proposed controller was evaluated using a mock-up setup, where the simulation and experimental results exhibited close correspondence. It was shown that the proposed control structure performs well in the presence of process variations, internal saturations, and position disturbances, with a performance increase of up to 56% with respect to previously established methods.

Second, the non-ideal properties of the robot joints that lower the position accuracy of the robot were modeled. Specifically, joint compliance and backlash were considered. A method for identifying the unknown joint

parameters was presented, which relies on clamping the robot end effector, such that all DoF of the robot are locked. The method was evaluated in extensive experiments in different setups, and verified using measurements from an optical tracking system.

Third, in order to reduce the position errors in machining tasks performed with robots, three different model-based ILC methods were presented. The first method was based on position measurements of the arm-side of the robot in task space, the second method used process-force measurements, and the third used joint motor torques to estimate the process forces. The experimental results obtained from milling experiments in aluminum showed that the algorithm based on arm-side position measurements was capable of decreasing the position error by up to 84%. In turn, the force-based and the force-estimation approaches decreased the error by 55% and 38%, respectively. Significant savings can be achieved with the use of the second and third method, because of the lower or no investment cost in sensors.

The three methods for increasing machining accuracy presented in this thesis, are considered as a contribution towards the goal of industrial robots becoming a viable alternative to machine tools. However, the achieved machining accuracy is still not as high as what machine tools can offer, but is sufficient for many applications in today's industry.

Further investigation and application of ILC methods were considered for a different kind of manipulator, a marine vibrator, for the application of marine seismic acquisition. A frequency-domain ILC strategy was proposed, which proved effective in suppressing undesired harmonics as well as providing accurate reference tracking. The desired suppression level of the harmonics was achieved for all experiments, within approximately 20 iterations of the ILC algorithm. The absolute tracking error was reduced by a factor of approximately 50 for each of the two channels. Additionally, a procedure for detecting drift in the dynamics and a subsequent reidentification procedure of the estimated transfer matrix were developed. It was shown that convergence could be ensured with the updated transfer matrix, while still suppressing the harmonics. Furthermore, the proposed frequency-domain ILC algorithm was briefly compared to traditional time-domain ILC, and it was found that frequency-domain ILC is highly advantageous for the considered application.

The final problem considered in this thesis concerned minimizing cycle times for rough-cut machining processes in non-isotropic materials. A model-based adaptive force-control architecture, combined with a learning algorithm to find the optimal machining path, was developed. The proposed method was verified in simulations and in pocket-milling experiments performed in oak. With the use of force control, the duration for performing a pocket milling was decreased by 14% as compared to using only position control. The milling durations were further decreased by employing the

learning algorithm; on average an additional 28 % reduction. Furthermore, an auto-training method was developed, which resulted in less waste of material and approximately 35 % decreased mean duration per pocket in experiments. The learning algorithm was extended to minimize process-force variance and the milling-duration variance, where reductions of 29 % and 94 % were achieved, respectively.

The learning algorithm for path planning was refined to incorporate a reinforcement learning algorithm, in order to find the optimal machining path for all scenarios. A Q-learning algorithm for two-directional path planning of machining tasks was proposed and was found to provide optimal paths. Additionally, the problem was extended to incorporate two additional machining directions, rendering the problem significantly more complex. A feature-based Q-learning approach was taken and was shown to achieve optimal paths. It was further shown that for experiments where the initial training data were partially incorrect, both proposed methods quickly adapted to find a new path. The four-directional method was found to reduce the cycle-time even more than the two-directional.

It is to be noted that the learning algorithms for path planning can be used both for robots and machine tools, given that the required sensor measurements are available.

Bibliography

- ABB Robotics (2010). *Technical reference manual*. Data sheet nr. 3HAC16581-1 Rev J.
- ABB Robotics (2015a). *ABB IRB120 Industrial Robot Data sheet*. Data sheet nr. ROB0149EN_D.
- ABB Robotics (2015b). *ABB IRB140 Industrial Robot Data sheet*. Data sheet nr. PR10031EN_R15.
- ABB Robotics (2015). *ABB IRB2400 Industrial Robot Data sheet*. Data sheet nr. PR10034 EN_R7.
- Abele, E., J. Bauer, T. Hemker, R. Laurischkat, H. Meier, S. Reese, and O. von Stryk (2011). “Comparison and validation of implementations of a flexible joint multibody dynamics system model for an industrial robot”. *CIRP J. Manufacturing Science and Technology* **4**:1, pp. 38–43.
- Abele, E., M. Weigold, and S. Rothenbücher (2007). “Modeling and identification of an industrial robot for machining applications”. *CIRP Annals—Manufacturing Technology* **56**:1, pp. 387–390.
- Abele, E., S. Rothenbücher, and M. Weigold (2008). “Cartesian compliance model for industrial robots using virtual joints”. *Production Engineering* **2**:3, pp. 339–343.
- Ahn, H.-S., Y. Chen, and K. L. Moore (2007). “Iterative learning control: brief survey and categorization”. *IEEE Trans. Systems Man and Cybernetics—part C: Applications and Reviews* **37**:6, pp. 1099–1149.
- Al Janaideh, M., Y. Feng, S. Rakheja, C.-Y. Su, and C. Alain Rabbath (2009). “Hysteresis compensation for smart actuators using inverse generalized Prandtl-Ishlinskii model”. In: *Proc. IEEE Am. Control Conf.* St. Louis, MO, pp. 307–312.
- Alfred Jäger GmbH (2013). *Chopper 3300 H S5 Spindle*. Ober-Mörlen, Germany.
- Allison, B. and A. Isaksson (1998). “Design and performance of mid-ranging controllers”. *J. Process Control* **8**:5, pp. 469–474.

- Allison, B. and S. Ogawa (2003). “Design and tuning of valve position controllers with industrial applications”. *Trans. Institute of Measurement and Control* **25**:1, pp. 3–16.
- Arimoto, S., S. Kawamura, and F. Miyazaki (1984). “Bettering operation of robots by learning”. *J. Robotic Systems* **1**:2, pp. 123–140.
- Åström, K. J. and T. Hägglund (2005). *Advanced PID Control*. ISA—The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC.
- Åström, K. J. and B. Wittenmark (1997). *Computer-Controlled Systems*. Prentice Hall, Englewood Cliffs, NJ.
- Bagge Carlson, F., A. Robertsson, and R. Johansson (2015). “Modeling and identification of position and temperature dependent friction phenomena without temperature sensing”. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Hamburg, Germany.
- Barto, A. G. (1998). *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA.
- Beckhoff Automation GmbH (2013). *Beckhoff information system, TwinCAT 2*. Verl, Germany.
- Bemporad, A. (1998). “Reference governor for constrained nonlinear systems”. *IEEE Trans. Automatic Control* **43**:3, pp. 415–419.
- Bennett, D., J. Hollerbach, and P. Henri (1992). “Kinematic calibration by direct estimation of the Jacobian matrix”. In: *Proc. IEEE Int. Conf. Robotics and Automation*. Nice, France, pp. 351–357.
- Bittencourt, A. C., E. Wernholt, S. Sander-Tavallaey, and T. Brogårdh (2010). “An extended friction model to capture load and temperature effects in robot joints”. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Taipei, Taiwan, pp. 6161–6167.
- Blomdell, A., I. Dressler, K. Nilsson, and A. Robertsson (2010). “Flexible application development and high-performance motion control based on external sensing and reconfiguration of ABB industrial robot controllers”. In: *Proc. Workshop of “Innovative Robot Control Architectures for Demanding (Research) Applications—How to Modify and Enhance Commercial Controllers”*, *IEEE Int. Conf. Robotics and Automation*. Anchorage, Alaska, USA, pp. 62–66.
- Bristow, D. A., M. Tharayil, and A. G. Alleyne (2006). “A survey of iterative learning control”. *Control Systems Magazine, IEEE* **26**:3, pp. 96–114.
- Burke, L. I. and S. Rangwala (1991). “Tool condition monitoring in metal cutting: a neural network approach”. *J. Intelligent Manufacturing* **2**:5, pp. 269–280.

- Cano Marchal, P., O. Sörnmo, B. Olofsson, A. Robertsson, J. Gómez Ortega, and R. Johansson (2014). “Iterative learning control for machining with industrial robots”. In: *IFAC World Congress*. Cape Town, South Africa, pp. 9327–9333.
- Casalino, G. and G. Bartolini (1984). “A learning procedure for the control of movements of robotic manipulators”. In: *IASTED Symp. Robotics and Automation*. Amsterdam, Netherlands, pp. 108–111.
- Chen, H., T. Fuhlbrügge, S. Choi, J. Wang, and X. Li (2008). “Practical industrial robot zero offset calibration”. In: *Proc. IEEE Int. Conf. Automation Science and Engineering*. Washington, D.C., pp. 516–521.
- Chen, Z. C. and S. Abdelkhalik (2014). “A new approach to planning plungers paths for efficient 2.5-axis computer numerically controlled plunge milling of complex pockets with islands”. *ASME J. Manufacturing Science and Engineering* **136**:4, pp. 1087–1357.
- Cho, S., S. Asfour, A. Onar, and N. Kaundinya (2005). “Tool breakage detection using support vector machine learning in a milling process”. *Int. J. Machine Tools and Manufacture* **45**:3, pp. 241–249.
- COMET (2013). *Plug-and-produce COmponents and METHods for adaptive control of industrial robots enabling cost effective, high precision manufacturing in factories of the future*. EU/FP7-project (Ref. #258769). URL: <http://www.cometproject.eu>.
- Craig, J. J. (1984). “Adaptive control of manipulators through repeated trials”. In: *Proc. American Control Conf.* San Diego, CA, pp. 1566–1573.
- Cuypers, W., N. Van Gestel, A. Voet, J. Kruth, J. Mingneau, and P. Bleys (2009). “Optical measurement techniques for mobile and large-scale dimensional metrology”. *Optics and Lasers in Engineering* **47**:3, pp. 292–300.
- Daneshmend, L. and H. Pak (1986). “Model reference adaptive control of feed force in turning”. *ASME J. Dynamic Systems, Measurement, and Control* **108**:3, pp. 215–222.
- Datta, A. and J. Ochoa (1996). “Adaptive internal model control: design and stability analysis”. *Automatica* **32**:2, pp. 261–266.
- Denavit, J. and R. S. Hartenberg (1955). “A kinematic notation for lower-pair mechanisms based on matrices”. *Trans. ASME J. Appl. Mech.* **23**, pp. 215–221.
- Doh, T.-Y. and J.-H. Moon (2004). “Feedback-based iterative learning control for uncertain linear mimo systems”. In: *Asian Control Conference*. Vol. 1. IEEE, pp. 198–203.
- Dorigo, M. and L. Gambardella (2014). “Ant-Q: a reinforcement learning approach to the traveling salesman problem”. In: *Proc. Int. Conf. Machine Learning*, pp. 252–260.

- dSPACE GmbH (2007). *DS1103 PPC Controller Board—Hardware Installation and Configuration*. dSPACE GmbH, Paderborn, Germany.
- Dumas, C., S. Caro, S. Garnier, and B. Furet (2011). “Joint stiffness identification of six-revolute industrial serial robots”. *J. Robotics and Computer-Integrated Manufacturing* **27**:4, pp. 881–888.
- Economou, C., M. Morari, and B. Palsson (1986). “Internal model control: extension to nonlinear system”. *Industrial & Engineering Chemistry Process Design and Development* **25**:2, pp. 403–411.
- Egeland, O. and J. R. Saggi (1990). “Kinematics and control of a space manipulator using the macro-micro manipulator concept”. In: *IEEE Conf. Decision and Control*. IEEE, Honolulu, HI, pp. 3096–3101.
- Eielsen, A., J. T. Gravdahl, and K. Leang (2015). “Low-order continuous-time robust repetitive control: Application in nanopositioning”. *Mechatronics*. In press. DOI: 10 . 1016 / j . mechatronics.2015.07.006.
- Fasse, E. and N. Hogan (1995). “Control of physical contact and dynamic interaction”. In: *Proc. Int. Symp. Robotics Research*. Munich, Germany, pp. 28–38.
- Feng, H.-Y. and C.-H. Menq (1996). “A flexible ball-end milling system model for cutting force and machining error prediction”. *ASME J. Manufacturing Science and Engineering* **118**:4, pp. 461–469.
- Frieze, A. M., G. Galbiati, and F. Maffioli (1982). “On the worst-case performance of some algorithms for the asymmetric traveling salesman problem”. *Networks* **12**:1, pp. 23–39.
- Garcia, C. and M. Morari (1982). “Internal model control. a unifying review and some new results”. *Industrial & Engineering Chemistry Process Design and Development* **21**:2, pp. 308–323.
- Gayadeen, S. and W. Heath (2009). “An internal model control approach to mid-ranging control”. *IFAC Adv. Control of Chemical Processes* **7**:1.
- Gilbert, E., I. Kolmanovsky, and K. Tan (1995). “Discrete-time reference governors and the nonlinear control of systems with state and control constraints”. *Int. J. Robust and Nonlinear Control* **5**:5, pp. 487–504.
- Goh, C. (1994). “A frequency domain analysis of learning control”. *ASME J. Dynamic Systems, Measurement, and Control* **116**:4, pp. 781–786.
- Gouveia, L. and J. M. Pires (1999). “The asymmetric travelling salesman problem and a reformulation of the Miller–Tucker–Zemlin constraints”. *European J. Operational Research* **112**:1, pp. 134–146.
- Graydon, B. and F. Delbert (1969). *Marine vibrator devices*. US Patent 3,482,646.

- Grote, K. and E. Antonsson (2009). *Springer Handbook of Mechanical Engineering: Manufacturing Engineering—Machining Processes*. No. 10. Springer Verlag, Berlin Heidelberg, Germany.
- Hakvoort, W., R. Aarts, J. Van Dijk, and J. Jonker (2007). “Model-based iterative learning control applied to an industrial robot with elasticity”. In: *Proc. IEEE Conf. Decision and Control*. New Orleans, LA, pp. 4185–4190.
- Harris, F. J. (1978). “On the use of windows for harmonic analysis with the discrete Fourier transform”. *Proc. IEEE* **66**:1, pp. 51–83.
- Haugwitz, S., M. Karlsson, S. Velut, and P. Hagander (2005). “Anti-windup in mid-ranging control”. In: *Proc. IEEE Conf. Decision and Control and European Control Conf.* Seville, Spain.
- Haverkamp, B. and M. Verhaegen (1997). *SMI Toolbox: State Space Model Identification Software for Multivariable Dynamical Systems*. Delft University of Technology, Delft, The Netherlands.
- He, J., Z. Pan, and H. Zhang (2007). “Adaptive force control for robotic machining process”. In: *Proc. IEEE Am. Control Conf.* New York City, NY.
- Heidenhain (2013). URL: <http://www.heidenhain.com>.
- Ho, W.-H., J.-T. Tsai, B.-T. Lin, and J.-H. Chou (2009). “Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid Taguchi-genetic learning algorithm”. *Expert Systems with Applications* **36**:2, pp. 3216–3222.
- Hogan, N. (1985). “Impedance control: An approach to manipulation: parts I-III”. *ASME J. Dynamic Systems, Measurement, and Control* **107**, pp. 1–24.
- Hogan, N. and S. P. Buerger (2005). *Impedance and Interaction Control, Robotics and Automation Handbook*. CRC Press, Boca Raton, FL, pp. 19.1–19.24.
- Hovland, G., S. Hanssen, S. Moberg, T. Brogårdh, S. Gunnarson, and M. Isaksson (2002). “Nonlinear identification of backlash in robot transmissions”. In: *Int. Symp. Robotics*. Stockholm, Sweden.
- Hu, Q. and G. Rangaiah (1999). “Adaptive internal model control of nonlinear processes”. *Chemical Engineering Science* **54**:9, pp. 1205–1220.
- Hunt, K. and D. Sbarbaro (1991). “Neural networks for nonlinear internal model control”. In: *Control Theory and Applications, IEE Proc. D*. Vol. 138. 5. IET, pp. 431–438.
- Irodova, M. and R. H. Sloan (2005). “Reinforcement learning and function approximation.” In: *FLAIRS Conf.* Clearwater Beach, FL, pp. 455–460.

- Johansson, R. (1993). *System Modeling and Identification*. Prentice Hall, Englewood Cliffs, NJ.
- Jonsson, M., A. Stolt, A. Robertsson, S. von Gegerfelt, and K. Nilsson (2013). “On force control for assembly and deburring of castings”. *Production Engineering* **7**:4, pp. 351–360.
- Joubair, A., A. Nubiola, and I. Bonev (2013). “Calibration efficiency analysis based on five observability indices and two calibration models for a six-axis industrial robot”. *Int. J. Aerospace* **6**:1, pp. 161–168.
- JR3 (2015). *JR3 100M40A Data sheet*.
- Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). “Reinforcement learning: a survey”. *J. Artificial Intelligence Research*, pp. 237–285.
- Kalman, R. E. (1960). “A new approach to linear filtering and prediction problems”. *ASME Trans. J. Basic Engineering* **82**:D, pp. 35–45.
- Keyence Corp. (2006). *LK-G Series User Manual*. Keyence Corp., Osaka, Japan.
- Kim, K.-S. and Q. Zou (2008). “Model-less inversion-based iterative control for output tracking: piezo actuator example”. In: *Proc. IEEE Am. Control Conf.* Pp. 2710–2715.
- Kim, K.-S. and Q. Zou (2013). “A modeling-free inversion-based iterative feedforward control for precision output tracking of linear time-invariant systems”. *IEEE/ASME Trans. Mechatronics* **18**:6, pp. 1767–1777.
- Kinsler, L. E., A. R. Frey, A. B. Coppens, and J. V. Sanders (1999). *Fundamentals of acoustics*. 4th ed. Vol. 1. Wiley-VCH, Weinheim, Germany.
- KrisEnergy (2015). <http://www.krisenergy.com/>.
- KUKA Roboter GmbH (2013). *KR125 Serie 2000 Data Sheet*. RoMeDBKR125-04.99.02.
- Lauderbaugh, L. and A. Ulsoy (1988). “Dynamic modeling for control of the milling process”. *ASME J. Manufacturing Science and Engineering* **110**:4, pp. 367–375.
- Lauderbaugh, L. and A. Ulsoy (1989). “Model reference adaptive force control in milling”. *ASME J. Manufacturing Science and Engineering* **111**:1, pp. 13–21.
- Lehmann, C., B. Olofsson, K. Nilsson, M. Halbauer, M. Haage, A. Robertsson, O. Sörnmo, and U. Berger (2013). “Robot joint modeling and parameter identification using the clamping method”. In: *Proc. IFAC Conf. Manufacturing Modeling, Management, and Control*. St. Petersburg, Russia, pp. 843–848.

- LGL and MAI (2011). “Environmental assessment of marine vibroseis”. In: *Joint Industry Programme, E&P Sound and Marine Life, Int. Assoc. Oil & Gas Producers*. London, U.K.
- Li, Y. and Q. Xu (2011). “A totally decoupled piezo-driven XYZ flexure parallel micropositioning stage for micro/nanomanipulation”. *IEEE Trans. Automation Science and Engineering* **8**:20, pp. 265–279.
- Liaw, H. C. and B. Shirinzadeh (2010). “Constrained motion tracking control of piezo-actuated flexure-based four-bar mechanisms for micro/nano manipulation”. *IEEE Trans. Automation Science and Engineering* **7**:3, pp. 699–705.
- Lin, R.-S. and Y. Koren (1996). “Efficient tool-path planning for machining free-form surfaces”. *ASME J. Manufacturing Science and Engineering* **118**:1, pp. 20–28.
- Linderoth, M., A. Stolt, A. Robertsson, and R. Johansson (2013). “Robotic force estimation using motor torques and modeling of low velocity friction disturbances”. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Tokyo, Japan.
- Liu, Y., T. Cheng, and L. Zuo (2001). “Adaptive control constraint of machining process”. *Int. J. Adv. Manufacturing Technology* **17**, pp. 720–726.
- Ljung, L. (1987). *System identification: Theory for the user*. Prentice Hall, Upper Saddle River, NJ.
- Ljung, L. (2010). *System Identification Toolbox 7: Users’s Guide*. The MathWorks, Inc., Natick, Massachusetts.
- Lo, C.-C. (1999). “Efficient cutter-path planning for five-axis surface machining with a flat-end cutter”. *Computer-Aided Design* **31**:9, pp. 557–566.
- Mahr GmbH (2011). *MarSurf M400 Flyer*. Mahr GmbH, Göttingen, Germany.
- Markusson, O., H. Hjalmarsson, and M. Norrlöf (2002). “A general framework for iterative learning control”. In: *Proc. IFAC World Congress*. Barcelona, Spain.
- MathWorks Inc. (2010). *Real-Time Workshop 7: Users’s Guide*. The MathWorks, Inc., Natick, Massachusetts.
- Meng Lim, E. and C.-H. Menq (1997). “Integrated planning for precision machining of complex surfaces. Part 1: Cutting-path and feedrate optimization”. *Int. J. Machine Tools and Manufacture* **37**:1, pp. 61–75.
- Mitra, S. K. and Y. Kuo (2006). *Digital signal processing: A computer-based approach*. Vol. 2. McGraw-Hill New York.

- Miyazaki, F., S. Kawamura, M. Matsumori, and S. Arimoto (1986). “Learning control scheme for a class of robot systems with elasticity”. In: *Proc. IEEE Conf. Decision and Control*. Vol. 25. Athens, Greece, pp. 74–79.
- Nikon Metrology (2010). *K-series optical CMM solutions*. Data sheet Optical_CMM_EN_0311.
- NorrLöf, M. (2002). “An adaptive iterative learning control algorithm with experiments on an industrial robot”. *IEEE Trans. Robot. Autom.* **18**:2, pp. 245–251.
- NorrLöf, M. (2000). *Iterative Learning Control: Analysis, Design, and Experiments*. PhD thesis. Linköping University, Sweden.
- NorrLöf, M. and S. Gunnarsson (2002). “Time and frequency domain convergence properties in iterative learning control”. *Int. J. Control* **75**:14, pp. 1114–1126.
- Olofsson, B., O. Sörnmo, U. Schneider, A. Robertsson, A. Puzik, and R. Johansson (2011a). “Modeling and control of a piezo-actuated high-dynamic compensation mechanism for industrial robots”. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Francisco, CA, pp. 4704–4709.
- Olofsson, B., O. Sörnmo, U. Schneider, M. Barho, A. Robertsson, and R. Johansson (2012). “Increasing the accuracy for a piezo-actuated micro manipulator for industrial robots using model-based nonlinear control”. In: *Int. IFAC Symp. Robot Control*. Dubrovnik, Croatia, pp. 277–282.
- Olofsson, B., O. Sörnmo, U. Schneider, A. Robertsson, A. Puzik, and R. Johansson (2011b). “Modeling and control of a piezo-actuated high-dynamic compensation mechanism for industrial robots”. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. San Francisco, CA, pp. 4704–4709.
- Olsson, T., M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson, M. Björkman, R. Isaksson, G. Ossbahr, and T. Brogårdh (2010). “Cost-efficient drilling using industrial robots with high-bandwidth force feedback”. *J. Robotics and Computer-Integrated Manufacturing* **26**, pp. 24–38.
- Overschee, P. van and B. De Moor (1994). “N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems”. *Automatica* **30**:1, pp. 75–93.
- Pan, Z. and H. Zhang (2009). “Improving robotic machining accuracy by real-time compensation”. In: *Proc. ICROS-SICE Int. Joint Conf.* Fukuoka, Japan, pp. 4289–4294.
- Park, S. C. and B. K. Choi (2000). “Tool-path planning for direction-parallel area milling”. *Computer-Aided Design* **32**:1, pp. 17–25.

- Puzik, A., C. Meyer, and A. Verl (2010). “Results of robot machining with additional 3D-piezo-actuation-mechanism for error compensation”. In: *CIRP Int. Conf. Intelligent Computation in Manufacturing Engineering: Innovative and Cognitive Production Technology and Systems*. Capri, Italy.
- Puzik, A., A. Pott, C. Meyer, and A. Verl (2009). “Industrial robots for machining processes in combination with an additional actuation mechanism for error compensation”. In: *Int. Conf. Manufacturing Research*. University of Warwick, United Kingdom.
- Puzik, A. (2011). *Genauigkeitssteigerung bei der spanenden Bearbeitung mit Industrierobotern durch Fehlerkompensation mit 3D-Ausgleichsaktuatorik*. Ph.D. Dissertation. University of Stuttgart, Stuttgart, Germany.
- Rangwala, S. S. and D. A. Dornfeld (1989). “Learning and optimization of machining operations using computing abilities of neural networks”. *IEEE Trans. Systems, Man and Cybernetics* **19**:2, pp. 299–314.
- Reinelt, G. (1994). *The Traveling Salesman: Computational solutions for TSP applications*. Springer-Verlag, New York.
- Reinl, C., M. Friedmann, J. Bauer, M. Pischian, E. Abele, and O. von Stryk (2011). “Model-based off-line compensation of path deviation for industrial robots in milling applications”. In: *Proc. IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics*. Budapest, Hungary, pp. 367–372.
- Reis GmbH (2011). *Reis RV40 Fact Sheet*. Reis GmbH, Obernburg, Germany.
- RME-Audio (2015). *Fireface UCX product information*. URL: http://www.rme-audio.de/en_products_fireface_ucx.php.
- Rober, S. and Y. Shin (1996). “Control of cutting force for end milling processes using an extended model reference adaptive control scheme”. *ASME J. Manufacturing Science and Engineering* **118**:3, pp. 339–347.
- Roth, Z., B. Mooring, and B. Ravani (1987). “An overview of robot calibration”. *IEEE J. Robotics and Automation* **5**:3, pp. 377–385.
- Ruderman, M., F. Hoffmann, and T. Bertram (2009). “Modeling and identification of elastic robot joints with hysteresis and backlash”. *IEEE Trans. Industrial Electronics* **56**:10, pp. 3840–3847.
- Schneider, U., M. Ansaloni, M. Drust, F. Leali, and A. Verl (2013a). “Experimental investigation of error sources in robot machining”. In: *Proc. Int. Conf. Flexible Automation and Intelligent Manufacturing*. Porto, Portugal, pp. 14–26.
- Schneider, U., J. R. Diaz Posada, and A. Verl (2015). “Automatic pose optimization for robotic processes”. In: *IEEE Int. Conf. Robotics and Automation*. Seattle, WA, pp. 2054–2059.

- Schneider, U., J. R. Diaz Posada, M. Drust, and A. Verl (2013b). “Position control of an industrial robot using an optical measurement system for machining purposes”. In: *Proc. Int. Conf. Manufacturing Research*. Cranfield University, UK, pp. 307–312.
- Schneider, U., M. Drust, A. Puzik, and A. Verl (2013c). “Compensation of errors in robot machining with a parallel 3D-piezo compensation mechanism”. In: *Proc. CIRP Conf. Manufacturing Systems*. Sesimbra, Portugal, pp. 305–310.
- Schneider, U., B. Olofsson, O. Sörnmo, M. Drust, A. Robertsson, M. Hägele, and R. Johansson (2014). “Integrated approach to robotic machining with macro/micro-actuation”. *J. Robotics and Computer-Integrated Manufacturing* **30**:6, pp. 636–647.
- Schröer, K., S. Albright, and M. Grethlein (1997). “Complete, minimal and model-continuous kinematic models for robot calibration”. *J. Robotics and Computer-Integrated Manufacturing* **13**:1, pp. 73–85.
- Sharon, A., N. Hogan, and D. E. Hardt (1993). “The macro/micro manipulator: an improved architecture for robot control”. *J. Robotics and Computer-Integrated Manufacturing* **10**:3, pp. 209–222.
- Siciliano, B., L. Sciavicco, L. Villani, and G. Oriolo (2009). *Robotics: Modelling, Planning and Control*. Springer-Verlag, London.
- Silva, G. and A. Datta (1999). “Adaptive internal model control: the discrete-time case”. In: *Proc. IEEE Am. Control Conf.* Vol. 1. San Diego, CA, pp. 547–555.
- Solectro (2015). *Solectro UFM 1050 Data sheet*. Data sheet nr. isel_data_UFM500_e-170812.
- Sörnmo, O., B. Bernhardsson, O. Kröling, P. Gunnarsson, and R. Tenghamn (2015a). “Frequency-domain iterative learning control of a marine vibrator”. *J. Control Engineering Practice*. Submitted.
- Sörnmo, O., B. Olofsson, A. Robertsson, and R. Johansson (2013). “Adaptive internal model control for mid-ranging of closed-loop systems with internal saturation”. In: *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. Tokyo, Japan, pp. 4893–4899.
- Sörnmo, O., B. Olofsson, A. Robertsson, and R. Johansson (2012a). “Increasing time-efficiency and accuracy of robotic machining processes using model-based adaptive force control”. In: *Int. IFAC Symp. Robot Control*. Dubrovnik, Croatia, pp. 543–548.
- Sörnmo, O., B. Olofsson, A. Robertsson, and R. Johansson (2015b). “Learning approach to cycle-time minimization of wood milling using adaptive force control”. *ASME J. Manufacturing Science and Engineering* **138**:1, pp. 011013–011013-11. DOI: 10.1115/1.4030751.

- Sörnmo, O., B. Olofsson, U. Schneider, A. Robertsson, and R. Johansson (2012b). “Increasing the milling accuracy for industrial robots using a piezo-actuated high-dynamic micro manipulator”. In: *IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics*. Kaohsiung, Taiwan, pp. 104–110.
- Spitzer, F. (1964). *Principles of random walk*. Springer-Verlag, New York.
- Spong, M. W., S. Hutchinson, and M. Vidyasagar (2006). *Robot Modeling and Control*. John Wiley & Sons, Hoboken, NJ.
- Stolt, A. (2015). *On Robotic Assembly using Contact Force Control and Estimation*. PhD thesis. Dept. Automatic Control, LTH, Lund University, Sweden.
- Suh, S.-H. and Y.-S. Shin (1996). “Neural network modeling for tool path planning of the rough cut in complex pocket milling”. *J. Manufacturing Systems* **15**:5, pp. 295–304.
- Sun, Y. and B. Yang (2009). “Compensation of hysteresis nonlinearity for the piezoelectric actuators”. In: *Proc. IEEE Int. Conf. Computer Science and Information Technology*. St. Louis, MO, pp. 307–312.
- Tenghamn, R. (2006). “An electrical marine vibrator with a flextensional shell”. *Exploration Geophysics* **37**:4, pp. 286–291.
- Tenghamn, R. (2009). *Driving means for acoustic marine vibrator*. US Patent 7,551,518.
- Tenghamn, R. (2011). *Control system for marine vibrators and seismic acquisition system using such control system*. US Patent 7,974,152.
- Tien, S., Q. Zou, and S. Devasia (2005). “Iterative control of dynamics-coupling-caused errors in piezoscanners during high-speed afm operation”. *IEEE Trans. Control Systems Technology* **13**:6, pp. 921–931.
- Tikhonov, A. N. and V. Y. Arsenin (1977). *Solutions of ill-posed problems*. V. H. Winston and Sons, Washington, DC.
- Tyapin, I., G. Hovland, and T. Brogårdh (2014). “Method for estimating combined controller, joint and link stiffnesses of an industrial robot”. In: *IEEE Int. Symp. Robotic and Sensors Environments*. Timisoara, Romania.
- Uchiyama, M. (1978). “Formulation of high-speed motion pattern of a mechanical arm by trial”. *Trans. Society of Instrument and Control Engineers* **14**:6, 706–712 (in Japanese).
- Ugural, A. C. and S. K. Fenster (2003). *Advanced strength and applied elasticity*. Prentice Hall, Upper Saddle River, NJ.
- Van Dijk, N., E. Doppenberg, R. Faassen, N. van de Wouw, J. Oosterling, and H. Nijmeijer (2010). “Automatic in-process chatter avoidance in the high-speed milling process”. *ASME J. Dynamic Systems, Measurement, and Control* **132**:3, pp. 031006–031006-14. DOI: 10.1115/1.400082.

- Verhaegen, M. and P. Dewilde (1992). “Subspace model identification—The output-error state-space model identification class of algorithms”. *Int. J. Control* **56**, pp. 1187–1210.
- Vuong, N. D., M. H. Ang Jr., T. M. Lim, and S. Y. Lim (2009). “Multi-rate operational space control of compliant motion in robotic manipulators”. In: *Proc. of 2009 IEEE Int. Conf. Systems, Man and Cybernetics*. San Antonio, TX, pp. 3175–3180.
- Waite, T. C., Q. Zou, and A. Kelkar (2008). “Inversion-based feedforward approach to broadband acoustic noise reduction”. *ASME J. Vibration and Acoustics* **130**:5, pp. 051010–051010-7. DOI: 10.1115/1.2948411.
- Wang, H., H. Chang, R. Wysk, and A. Chandawarkar (1987). “On the efficiency of NC tool path planning for face milling operations”. *ASME J. Manufacturing Science and Engineering* **109**:4, pp. 370–376.
- Wang, J., G. Zhang, H. Zhang, and T. Fuhlbrigge (2008). “Force control technologies for new robotic applications”. In: *IEEE Int. Conf. Technology for Practical Robot Applications*. Woburn, MA, pp. 143–149.
- Wang, J., H. Zhang, and T. Fuhlbrigge (2009). “Improving machining accuracy with robot deformation compensation”. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*. St. Louis, MO, pp. 3826–3831.
- Watkins, C. J. and P. Dayan (1992). “Q-learning”. *Machine learning* **8**:3-4, pp. 279–292.
- Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis. University of Cambridge, Cambridge, England.
- Werth GmbH (2013). *CMM VideoCheck HA400*. Gießen, Germany.
- Wiener, N. (1949). *The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*. Originally issued as a classified MIT Rad. Lab. Report in February, 1942. Wiley, New York.
- Wiering, M. and M. Van Otterlo (2012). “Reinforcement learning”. In: *Adaptation, Learning, and Optimization*. Vol. 12. Springer Verlag, London.
- Xu, J.-X., S. K. Panda, and T. H. Lee (2010). *Real-time Iterative Learning Control: Design and Applications*. Springer Verlag, London.
- Yan, Y., H. Wang, and Q. Zou (2012). “A decoupled inversion-based iterative control approach to multi-axis precision positioning: 3D nanopositioning example”. *Automatica* **48**:1, pp. 167–176.
- Yan, Y., Q. Zou, and Z. Lin (2009). “A control approach to high-speed probe-based nanofabrication”. *Nanotechnology* **20**:17. DOI: <http://dx.doi.org/10.1088/0957-4484/20/17/175301>.

- Yang, S.-H. and S.-G. Lee (2002). “CNC tool-path planning for high-speed high-resolution machining using a new tool-path calculation algorithm”. *Int. J. Adv. Manufacturing Technology* **20**:5, pp. 326–333.
- Zhang, H., J. Wang, G. Zhang, Z. Gan, Z. Pan, H. Cui, and Z. Zhu (2005a). “Machining with flexible manipulator: toward improving robotic machining performance”. In: *Proc. IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics*. Monterey, California, USA, pp. 1127–1132.
- Zhang, H., J. Wang, G. Zhang, Z. Gan, Z. Pan, H. Cui, and Z. Zhu (2005b). “Machining with flexible manipulator: Toward improving robotic machining performance”. In: *Proc. IEEE/ASME Int. Conf. Adv. Intelligent Mechatronics*. Monterey, CA, pp. 1127–1132.
- Zheng, A., V. Mayuresh, and M. Morari (1994). “Anti-windup design for internal model control”. *Int. J. Control* **60**:5, pp. 1015–1024.
- Zhou, K. and J. Doyle (1998). *Essentials of robust control*. Prentice Hall, Upper Saddle River, NJ.