LUND UNIVERSITY

Modeling and Control of Stiff Robots for Flexible Manufacturing

Dressler, Isolde

2012

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Dressler, I. (2012). *Modeling and Control of Stiff Robots for Flexible Manufacturing*. [Doctoral Thesis (monograph), Department of Automatic Control]. Department of Automatic Control, Lund Institute of Technology, Lund University.

Total number of authors:
1

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

Modeling and Control of Stiff Robots for Flexible Manufacturing

Modeling and Control of Stiff Robots for Flexible Manufacturing

Isolde Dressler

Department of Automatic Control
Lund University
Lund, September 2012

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

ISSN 0280–5316
ISRN LUTFD2/TFRT--1093--SE

© 2012 by Isolde Dressler. All rights reserved.
Printed in Sweden by Media-Tryck.
Lund 2012

Abstract

To survive on a global market, small and medium size enterprises (SMEs) require affordable and competitive industrial automation for high quality flexible manufacturing. This thesis contributes to the development of robot concepts that fit the needs of SMEs. A major part of the thesis deals with the modeling of the three degree-of-freedom (DOF) Gantry-Tau parallel kinematic robot, which has the potential to fulfill the requirements on accuracy, mechanical stiffness and conceptual flexibility of a robot for SMEs. Additionally, concepts that aid the SMEs to achieve the required accuracy and a more intuitive robot operation were developed and evaluated.

The modeling of the Gantry-Tau robot includes both kinematic and dynamic modeling. Based on the nominal kinematic model, kinematic error models were developed, as well as kinematics for the F1-type Gantry-Tau, a Gantry-Tau architecture extended to 6 DOF. The modeling was evaluated in kinematic calibration experiments. A rigid body model was derived and identified, including friction in the actuators. As noticeable flexible behaviour was observed, the compliance dynamics were identified by black box modeling.

Kinematic calibration was not only considered for evaluation of the kinematic models developed, but it was also studied how to automize the kinematic calibration procedure, so that it can be executed by non-expert SME staff after a possible geometric reconfiguration of the robot. In the search of affordable, accurate and reusable measurement devices for kinematic calibration in SMEs, the usage of camera vision for kinematic calibration was evaluated.

To make the programming of a robot trajectory fast and intuitive, lead-through programming was recently introduced. A new concept for lead-through programming in contact situations is proposed in this thesis, where two force sensors are used. While the first sensor is used for guiding the robot, the second force sensor measures the tool force, which

can prevent damage of the tool or workpiece and can help to keep a steady contact between tool and surface. The concept was demonstrated in two example applications.

A possibility to improve the performance for a repeatedly executed motion is iterative learning control (ILC). An ILC algorithm is evaluated on the Gantry-Tau robot, which uses an estimate of the tool motion, based on measurements from an accelerometer mounted at the end-effector plate and in addition to measurements on the motor side. The performance of the tool motion was shown to be considerably improved compared to the case when only motor side measurements are used.

Acknowledgements

First of all, I would like to thank my supervisor Anders Robertsson for giving me the opportunity to work in the Robot Lab and for introducing me to the world of robotics. He is a great source of inspiration, has enormous robotics knowledge and is always ready to help out and solve problems on the spot.

The colleagues in the Robot Lab, Anders Blomdell, Mathias Haage, Rolf Johansson, Klas Nilsson, and all the others helped me with their feedback, discussions, ideas or with solving my practical problems.

I also want to thank Torgny Brogårdh, ABB Robotics, for his very useful feedback, ideas and discussions. And literally, without him, this thesis would not exist. He invented the Gantry-Tau robot.

Kalle Åström and Carl Olsson from the Mathematical Imaging Group wrote the vision script and helped me with the image analysis. Kalle Åström also implemented a polynomial solver for the Stewart platform kinematics so that I could investigate the F1 kinematics further.

I would also like to thank Nicolas Andreff for teaching me how to calibrate parallel robots during my visit in Clermont-Ferrand.

During the ILC experiments, I had the opportunity to work together with Johanna Wallén. I think that without her energy and drive, we would not have come that far. It was a pleasure to meet and to work with you!

During my PhD, I also met many people from different companies, who in various ways supported my work, in particular my calibration measurements. In this context I want to thank the colleagues from ABB Robotics, Güdel AG, Boeing, Hexagon Metrology and CTI Sheffield.

The financial support of my work is greatfully acknowledged. I would like to express my gratitude to the following funders: The European Union's Sixth Framework Programme (FP6) under grant no. 011838 as part of the Integrated Project SMErobot, from the Swedish Research Council through the project VR-2006-5522 and through the LCCC Linnaeus Center, from the ELLIIT Excellence Center and from the European Union's Seventh

Framework Programme (FP7) through the ECHORD/MONROE project.

Of course, I should not forget my present and former colleagues at Automatic Control. Leif Andersson, Anders Blomdell, Rolf Braun and Stefan Skoog solved all my software and hardware problems. Eva, Eva, Britt-Marie, Lizette, Ingrid and Monika provided help in any kind of administrative problems. Thanks also to Tore Hägglund for his support and the many discussions about my thesis. I also want to thank everybody else who has been reading and giving me feedback on my thesis.

Finally, I want to thank my friends and my family for all the support they have given me during my years in Lund. Especially during this last year, I would not have managed without the support of my husband Dragan, who took care of our little daughter while I was commuting between Lund and our family home in Switzerland.

Contents

1

Introduction

1.1 Motivation

To survive on a global market, small and medium size enterprises (SMEs) need affordable and competitive industrial automation that eases fast deployment and extends available task repertoire. SMEs depend on their ability to cost-efficiently produce customized products. This requires flexibility, which is often achieved by manual labor. To maintain profitability on a global market, there is a desire for robots that efficiently can assist human workers. The SMErobot project [SMErobot, 2009] aimed at developing new robot concepts for SMEs.

Today's industrial robot concepts are too expensive and not flexible enough for SMEs, that often perform constantly changing tasks due to small lot sizes. Typically, industrial robots carry out a repetitive task during long time ranges, as e.g., a welding robot in the production line of one specific automobile type. They are delivered from the robot manufacturer as an optimized entity and their geometry is not easily adjustable to new tasks and products. Programming a task is in most cases time-consuming and involves a robot expert who is familiar with the manufacturer-specific programming language. Some tasks, e.g., grinding or deburring, require a high structural stiffness, and are often problematic to execute with the desired precision with an industrial robot of reasonable size. This is particularly a problem for serial kinematic geometries, which are today forming the larger part of industrial robots. An SME might however profit more from a robot assisting those tasks than e.g., a packaging robot, which usually does not need increased stiffness nor high accuracy. Additionally, there are more often major security concerns at SMEs about sharing workspace between humans and robots.

The problem of insufficient structural stiffness may be solved by using

Figure 1.1 Gantry-Tau prototype developed within the SMErobot project.

parallel instead of serial kinematic robots. The usage of parallel kinematic manipulators (PKMs) is despite their high performance potential still relatively uncommon. PKMs are in general stiffer in relation to inertia and more accurate than serial robots. They can be built with less moving weight and therefore reach higher accelerations with less motor power [Merlet, 2000]. The high stiffness makes PKMs interesting candidates for a wide range of typical SME tasks such as grinding, deburring and cutting. An inconvenience is that most parallel robots have a smaller workspace than typical serial manipulators, meaning lower cost efficiency.

An increasing interest in parallel robot geometries was initiated already in the 1960s by the Stewart-Gough platform [Stewart, 1965]. However, it was not until the Delta structure was invented [Clavel, 1991] that PKMs were introduced in a larger scale in industrial applications. An example of the growing use of parallel robots in industry is the ABB Flexpicker [ABB, 2009], which is based on the Delta structure. The Delta structure has centralized placement of the actuators, which enables pick-and-place applications with accelerations up to 12 g. However, the Delta structure is only used in applications with small payload and a relatively small working range.

A remaining challenge regarding parallel robots is to find a mechanical structure useful for high performance applications [Brogårdh, 2002]. The Gantry-Tau configuration [Johannesson *et al.*, 2003], illustrated in Figure 1.1, is designed to have a large workspace-to-footprint ratio compared to other parallel robots, while still being stiff in relation to its inertia compared to a serial robot.

Besides the large accessible workspace in relation to its footprint, the

Gantry-Tau PKM has also the potential to meet the specific needs of SMEs. A modular robot which is reconfigurable according to varying tasks is needed. The Gantry-Tau PKM is promising in this regard. It is possible to be implemented in a modular way so that the framework can be adjusted to specific tasks [Dressler *et al.*, 2007]. Links can easily be replaced by longer or shorter ones, and the linear actuators can be moved in space by adjusting the framework, thus making reconfiguration possible.

An important step in developing new robot structures is accurate kinematic and dynamic modeling. The modularity of the robot and the possible robot assembly and reconfiguration by SME staff may lead to kinematic errors which decrease the positioning accuracy. Therefore, an accurate kinematic model able to cope with potential assembling errors is needed. With the cartesian repeatability of one of the prototypes developed measured to 13 µm [Crothers *et al.*, 2010], accurate kinematic models are also needed to fully exploit the Gantry-Tau's accuracy potential. Another possible application sector for the Gantry-Tau besides SMEs is aerospace industry, as the parallel structure and arbitrarily extendable linear guideways make it convenient for machining large components with very high accuracy. To meet the accuracy requirements in aerospace industry, kinematic error models need to be developed.

Kinematic and dynamic high speed control, to exploit the high achievable accelerations of the Gantry-Tau robot, requires also accurate dynamic models of the manipulator [Murray *et al.*, 1994]. Despite the robot's high stiffness, flexible behaviour is observable at high frequencies. A rigid body model is thus important for improving the control performance in general, but not sufficient at high speed.

Simulations are essential to investigate the benefits and limits of robot design and control. One simulation program used is Modelica [Modelica, 2011], which is an object-oriented modeling language. An advantage of Modelica is that models from one domain easily can be extended with models from other domains, e.g., extending a mechanical robot model with a controller or an actuator. Modelica is also advantageous during the design process as the obtained models can be used for various purposes such as simulation and controller design. Together with additional files provided by Dymola [Dynasim, 2009], a commercial Modelica implementation, C-files generated for simulation can be used for hardware-in-the-loop simulations, e.g., for actual control of the robot. Optimica [Åkesson, 2008] is a Modelica language extension for solving optimization problems, facilitating thus to use an existing Modelica model e.g., for optimal control.

With the Gantry-Tau as new, SME suitable type of robot, some problems still remain to be solved before robots can enter SMEs on a large scale. Problems can arise for SMEs because every reconfiguration of a robot requires a new kinematic calibration and such recalibrations tend

to be difficult for non-expert SME staff. It is thus a matter of cost reduction for an SME to have an easy-to-use calibration procedure, which does neither require specific skills nor subcontracting with an external company. Tools are needed to assist the non-expert SME staff to execute repeated kinematic calibrations.

Together with tools for supporting the calibration process, new ways of calibrating robots using low-cost sensors have to be found. SMEs cannot afford laser trackers or other expensive measurement equipment. Sensors that can be used both for calibration and feedback control are preferable.

Also new, more intuitive ways of robot programming are needed for SME staff. They don't have the time and knowledge to program robots using a programming language like ABB's RAPID. Lead-through programming has recently been introduced for this purpose. In the SMErobot project, it was examined how to combine lead-through programming with intuitive programming instructions. Unlike with manual labor, the operator usually has no feedback of the force between tool and workpiece. Consequences might be tool or workpiece damage and difficulties to keep a steady contact. An additional force sensor can solve this problem.

However, with only lead-through programming, the high accuracy required by certain applications can often not be achieved. One possible solution is to use the lead-through programming rather as a calibration between robot tool and workpiece and then to process the desired trajectory further based on the workpiece's CAD design. Another solution to improve accuracy may be iterative learning control (ILC). The desired trajectory is traversed repeatedly and each time, the control signal is changed slightly in order to successively improve the performance. Often, ILC is carried out based on the motor angle measurements available, whereas the tool trajectory is the principal control objective. At high speeds, where flexible behaviour is pronounced, this can lead to suboptimal tool performance. Measurements of the tool motion in an industrial environment are either too expensive for SMEs or mechanically interfering with the application. A way to benefit from tool motion feedback is to use a sensor measuring the tool motion indirectly, e.g., an accelerometer mounted on the end-effector plate, together with an observer to estimate the tool motion.

This thesis tries to provide a part of the research necessary to help SMEs becoming more competitive by using efficient, flexibly deployable robot technology. A large part of the thesis focuses on the Gantry-Tau robot as a promising new robot structure for this purpose. Additionally, problems related to a more flexible, easier-to-use robot concept are addressed.

1.2 Outline and Contribution

Chapter 2: The Gantry-Tau Robot and Prototypes

The Gantry-Tau robot is positioned among related parallel robot architectures and its development summarized in view of recent parallel robot trends. The Gantry-Tau prototypes used for experiments in this thesis are shortly described.

The Gantry-Tau structure was invented by Torgny Broghårdh. The prototypes were developed within the [SMErobot, 2009] and [MONROE, 2012].

Chapter 3: Software Support for Modeling and Control

An extended ABB IRC5 controller presented here allowed to implement customized controllers in the experiments in Chapters 6 – 8. Using Modelica robot models together with Modelica based tools for various stages of robot development and control is discussed and illustrated by examples.

The ABB IRC5 extension was developed and implemented by Anders Blomdell, Anders Robertsson and Klas Nilsson in close cooperation with ABB Robotics. A Simulink Gantry-Tau kinematics library for usage with the extended controller was implemented by Johan Friman based on the author's previous work and under her supervision. The Modelica models of the Gantry-Tau robot and examples involving these models are exclusively the author's work. The Simulink controller for ILC was implemented together with Johanna Wallén.

Chapter 4: Kinematic Modeling

Based on the nominal kinematics by [Johannesson *et al.*, 2003], kinematic error models and kinematic models for extended Gantry-Tau geometries were developed. The validity of the nominal model assumptions was experimentally examined and the influences of the imperfections on the robot's accuracy studied with the aid of a geometric method presented in Appendix A.

The kinematic models presented are exclusively the author's work as well as the method in Appendix A. All experiments and evaluations were carried out by the author. The algorithm for finding all forward kinematics solutions of a hexapod robot was implemented by Kalle Åström from the Centre for Mathematical Sciences at Lund University.

Chapter 5: Kinematic Calibration

The kinematic models presented in Chapter 4 were validated by kinematic calibration and the maximum achievable accuracy of the Gantry-

Tau prototypes was examined. In search of suitable methods for assisting non-experts to perform kinematic calibration, vision based calibration was studied and an automated calibration method is presented.

The stereo vision script was implemented by Carl Olsson and Karl Åström from the Centre for Mathematical Sciences at Lund University. The actual image processing using stereo vision was done by Carl Olsson, while the image processing for testing the automated calibration method using only mono vision was done by the author. The laser tracker measurements for the D1 prototype calibration were assisted by [Hexagon Metrology, 2012]. All other work including experiments planning, execution and evaluation was done by the author.

Chapter 6: Dynamic Modeling

Rigid body modeling and identification are presented as well as black box modeling of the Gantry-Tau compliance dynamics.

The rigid body model was developed and identified by the author. The first compliance dynamic models were identified by Marzia Cescon for her master's thesis [Cescon, 2008]. For the joint article [Cescon *et al.*, 2009], Marzia's findings were complemented by results from laser tracker measurements performed together with the author. Models for estimation and ILC tuning were identified by the author together with Johanna Wallén.

Chapter 7: Iterative Learning Control

Observer-based ILC was applied to the Gantry-Tau robot. Measurements from an accelerometer mounted on the end-effector plate were used to estimate the end-effector motion and to update the control signal based on the end-effector rather than the actuator motion. The method was successfully tested in experiments and compared to other approaches.

The observer-based ILC approach was developed by Johanna Wallén. She also derived the conditions for stability and convergence of the ILC algorithm. Identification as well as ILC experiments were prepared, carried out and evaluated jointly by the author and Johanna Wallén at the Robot Lab in Lund. Even though a sharp distinction is difficult between Johannas's and the author's contribution to the experiments, there was naturally a tendency that Johanna with her expertise in ILC contributed more to ILC related steps, like tuning of the ILC algorithm, and the author contributed more in parts that required Gantry-Tau or experimental setup knowledge, like implementation of sensor drivers or interpretation of the results.

Chapter 8: Lead-through Programming with Tool Force Feedback

A novel concept for lead-through programming using two force sensors was developed and successfully tested in two example applications. While

the first sensor is used to lead the robot along a desired trajectory, the second sensor is used to control the contact force between tool and object in order to prevent damage and to keep a steady contact. The concept was tested in a surface tracking and a peg-in-hole experiment.

The author developed the new force control concept presented and carried out all experiments including the controller implementation.

Publications

This thesis is based on the publications below. The author's contributions to the articles written together with Marzia Cescon and Johanna Wallén, respectively, are described in detail in the outlines of Chapters 6 and 7 above. For the remaining articles, the author was the main contributor. The results of Chapter 8 are not published to date.

Cescon, M., I. Dressler, R. Johansson, and A. Robertsson (2009): "Subspace-based identification of compliance dynamics of parallel kinematic manipulator." In *Proc. 2009 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics. AIM 2009*, pp. 1028–1033. Singapore.

Dressler, I., T. Brogårdh, and A. Robertsson (2010): "A kinematic error model for a parallel Gantry-Tau manipulator." In *Proc. IEEE Int. Conf. on Robotics and Automation 2010. ICRA 2010*, pp. 3709–3714. Anchorage, AK, USA.

Dressler, I., A. Robertsson, and R. Johansson (2007): "Accuracy of kinematic and dynamic models of a Gantry-Tau parallel kinematic robot." In *Proc. IEEE Int. Conf. on Robotics and Automation. ICRA 2007*. Rome, Italy.

Dressler, I., A. Robertsson, and R. Johansson (2008): "Automatic kinematic calibration of a modular Gantry-Tau parallel robot from a kinematics point of view." In *Proc. 2008 IEEE Int. Conf.on Robotics and Automation. ICRA 2008*, pp. 1282–1287. Pasadena, CA, USA.

Dressler, I., J. Schiffer, and A. Robertsson (2009): "Modeling and control of a parallel robot using Modelica." In *Proc. 7th International Modelica Conference*. Como, Italy.

Wallén, J., I. Dressler, A. Robertsson, M. Norrlöf, and S. Gunnarsson (2011a): "Estimation-based ILC applied to a parallel robot." *submitted*.

Wallén, J., I. Dressler, A. Robertsson, M. Norrlöf, and S. Gunnarsson (2011b): "Observer-based ILC applied to the Gantry-Tau parallel kinematic robot." In *Proc. IFAC World Congress 2011*. Milano, Italy.

Related Publications

Andreff, N. and I. Dressler (2008): "Closed-form calibration of the Gantry-Tau parallel robot." In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2008. IROS 2008.* Nice, France.

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. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*. Anchorage, AK.

Crothers, P., P. Freeman, T. Brogårdh, I. Dressler, K. Nilsson, A. Robertsson, W. Zulauf, B. Felber, R. Loser, and K. Siercks (2009): "Characterisation of the Tau parallel kinematic machine for aerospace application." In *SAE 2009 AeroTech Congress & Exhibition*. Seattle, WA.

Dressler, I., M. Haage, K. Nilsson, R. Johansson, A. Robertsson, and T. Brogårdh (2007): "Configuration support and kinematics for a reconfigurable Gantry-Tau manipulator." In *Proc. Int. Conf. on Robotics and Automation. ICRA 2007*. Rome, Italy.

Haage, M., I. Dressler, A. Robertsson, K. Nilsson, T. Brogårdh, and R. Johansson (2009): "Reconfigurable parallel kinematic manipulator for flexible manufacturing." In *Proc. 13th IFAC Symposium on Information Control Problems in Manufacturing. INCOM 2009*. June 3 - 5, 2009, Moscow, Russia.

Other Publications

Dressler, I. (2004): "Code generation from JGrafchart to Modelica." Master's Thesis ISRN LUTFD2/TFRT--5726--SE. Department of Automatic Control, Lund University, Sweden.

Otter, M., K.-E. Årzén, and I. Dressler (2005): "Stategraph – A Modelica library for hierarchical state machines." In *Proc. 4th International Modelica Conference*. Hamburg-Harburg, Germany.

2

The Gantry-Tau Robot and Prototypes

This chapter describes the Gantry-Tau robot in the context of related PKM architectures. After an introduction of the Gantry-Tau in Section 2.1, an overview of the Gantry-Tau's development and similar manipulators is given in Section 2.2. The four different Gantry-Tau prototypes which were used for experiments are shortly presented in Section 2.3.

2.1 The Gantry-Tau Manipulator

The Gantry-Tau manipulator (Figures 2.1 and 2.2) is in its basic form a 3 degree of freedom (DOF) parallel kinematic robot. It has three kinematic chains actuated by prismatic joints. The prismatic joints are implemented as carts moving on linear guideways and are connected to the end-effector plate via link clusters and passive spherical joints, alternatively a spherical and a universal joint. The altogether six links are grouped in a 3-2-1 configuration. In the following, the word "arm" denotes a complete kinematic chain consisting of linear actuator, link cluster, passive spherical joints and end-effector plate. The robot arms and their components are named according to their number of links, e.g., arm 2 is the kinematic chain with two links, driven by cart 2. The positioning of the spherical joints on end-effector plate and carts according to the so-called Tau configuration is such that the links within one link cluster form parallelograms. This and the parallelism of the linear guideways assure a constant end-effector orientation, so that the 3 DOF are purely translational. The Gantry-Tau robot has, unlike most other parallel robots, a large workspace to footprint ratio. The workspace has no internal singularities.

The Gantry-Tau robot was first named in [Johannesson *et al.*, 2003],

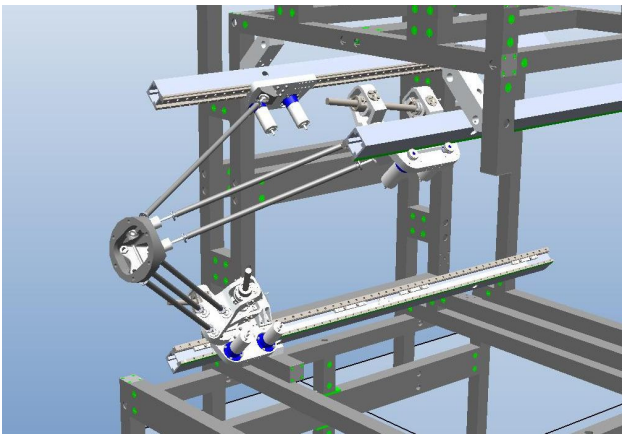


Figure 2.1 RobotStudio animation of the T2 Gantry-Tau prototype (see Section 2.3). Besides the basic 3 DOF structure illustrated also in Figure 2.2, the joint support plates on carts 2 and 3 can be reoriented giving the robot altogether 6 DOF. Additionally, the carts are driven by double motors to avoid backlash.

where the kinematics and an optimization of the workspace are described. A benchmark of the Gantry-Tau performance is presented in [Crothers *et al.*, 2010]. The mean omnidirectional repeatability of a Gantry-Tau prototype measured at the 3 DOF end-effector plate was found to be 13 µm. The corresponding actuator value is with 12 µm of the same order of magnitude and thus actuator errors are not adding up at the end-effector as is the case for serial robots. The stiffness of the same Gantry-Tau prototype was typically between $2\,\mathrm{N\,\mu m^{-1}}$ to $4\,\mathrm{N\,\mu m^{-1}}$. In [Cescon *et al.*, 2009], identification of the compliance dynamics of another Gantry-Tau prototype is presented. A lowest resonance frequency of 14 Hz was found in this work. Study results for related robots and the development of the Gantry-Tau robot are described in Section 2.2.

Several possibilities exist to extend the Gantry-Tau's 3 DOF for tasks requiring a higher DOF. Two of the prototypes constructed within the SMErobot project have been extended with a 2 DOF serial wrist. Even though the wrists were constructed in a lightweight way, a serial wrist risks to compromise the advantages of parallel kinematics like high accuracy, stiffness and acceleration. Another possibility is to reorient the joint support plates on the carts. This Gantry-Tau version, presented in Section 2.3 as the F1 prototype, enables a reorientation of 15° to 30° and has the full advantages of a parallel structure. Five purely parallel DOF can also be achieved by mounting two of the links of arm 2 and 3 on extra carts moving on guideways 2 and 3, respectively, as shown in Fig-

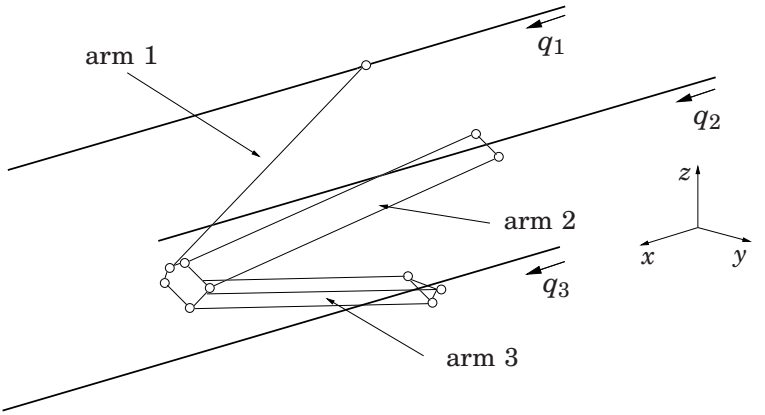


Figure 2.2 Schema of the Gantry-Tau robot with global coordinate system, actuator positions q_i, $i = \{1,2,3\}$, and kinematic chain notation.

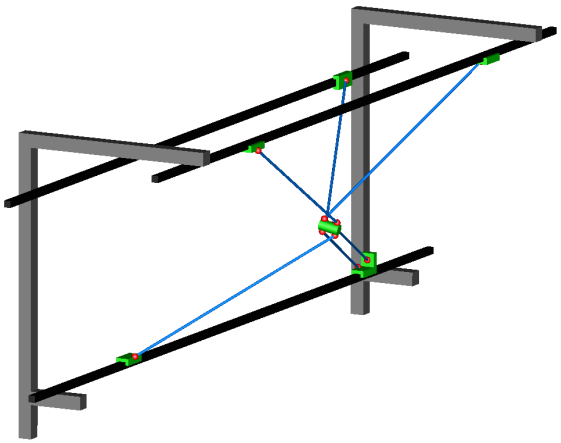


Figure 2.3 Proposal of extension to 5 DOF with two additional carts.

ure 2.3. In [Dressler *et al.*, 2007], the end-effector reorientation limit with the studied geometry was identified as approximately 5°.

Figure 2.4 shows the Gantry-Tau's workspace cross section for one of the prototypes presented in Section 2.3. The coordinate system is as shown in Figure 2.2. The robot has link lengths of 1.8 m to 2 m, giving a cross-section of the workspace in the yz-plane of approximately 1 m × 1 m. In the x-direction, the guideway length determines the workspace dimension.

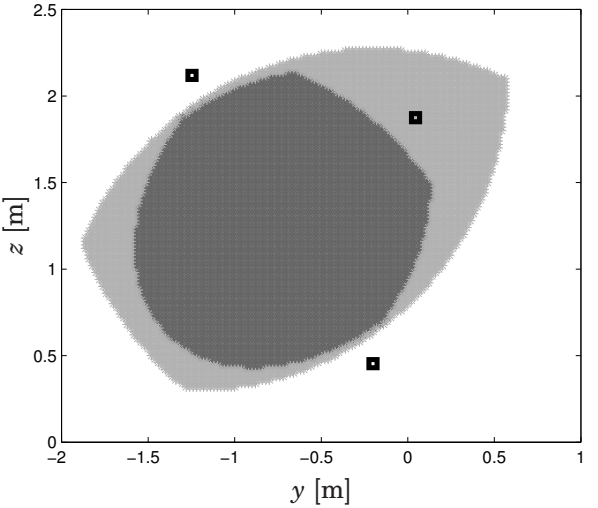


Figure 2.4 Workspace cross section for the L1 Gantry-Tau prototype presented in Section 2.3 with coordinate system as in Figure 2.2. The three small squares represent the guideways. In the light grey area the inverse kinematics has a solution, inside the dark grey area, also the ball joint angles are within a 40° limit.

2.2 Development and Related Architectures

Parallel Robots with Tau Configuration

A new family of parallel kinematic robots developed at ABB Robotics was first presented in [Brogårdh, 2000]. Instead of grouping six links in a 2-2-2 configuration as for the Delta parallel robot [Clavel, 1988], the links are clustered in a 3-2-1 configuration. The inspiration to the name Tau came from an early version of the Scara type of Tau robot described below, in which the motor shafts were arranged in such a way that they formed the letter τ, in the same way as the motor shafts in the Delta structure form the letter Δ.

The Tau configured arm system can be driven by prismatic or rotational actuators. Two different sorts of Tau robots have been realized: A Scara and a Gantry version. The Tau configuration robots are patented by ABB [Brogårdh, 1996].

The first parallel kinematic robot with the Tau configuration was a Scara type Tau (Figure 2.5). Kinematic and dynamic analysis of this robot was presented in [Cui *et al.*, 2005] and [Zhu *et al.*, 2005]. A performance

comparison with other robots can be found in [Crothers *et al.*, 2010]. A variant of the Scara Tau with a triangular three-link arm is studied in [Isaksson *et al.*, 2010]. It has an improved, i.e., reduced, dependence between end-effector position and orientation. In [Isaksson *et al.*, 2012], the Scara Tau concept is extended to the 6 DOF octahedral Hexarot robot.

The first Gantry-Tau prototype was built at the University of Queensland, Australia in 2004. This Gantry-Tau has, unlike the Gantry-Tau in this thesis, a triangular 3-link arm and consequently a variable end-effector plate orientation. To distinguish it from the Gantry-Tau version this thesis is based on, it will be called Queensland Gantry-Tau in the following, while Gantry-Tau without further specification will denote the version with constant end-effector orientation.

According to [Hovland *et al.*, 2007b], the triangular link has several advantages. It enables a reconfiguration of the robot, i.e., a change of the assembly mode by flipping the links to the other side of the carts. The spherical joint offsets on the actuator side are reduced and the robot has a larger reach at the workspace limits. The constant end-effector plate orientation can lead to collisions between links and plate. This is however not an issue for the prototypes in Section 2.3, as the spherical joints used reach their limits before a collision is at hand.

The first version of the Queensland Gantry-Tau had 3 DOF [Williams *et al.*, 2006]. Later, the links were exchanged with two of the links being prismatic joints, so that the new version had 5 DOF [Murray *et al.*, 2006]. The triangular arm and the end-effector plate design, which includes a sixth DOF, make a reconfiguration possible [Murray *et al.*, 2008], enlarging thus the workspace. In [Hovland *et al.*, 2007a], the benefits of exchanging the single link arm of the 3 DOF version with a telescopic actuator were evaluated. With the redundant actuation, higher stiffness can be achieved.

[Williams *et al.*, 2006] presented a 3 DOF kinematic error model and calibration; a 5 DOF inverse kinematic model was derived in [Murray *et al.*, 2006]. Collisions between links and end-effector plate reduce the workspace in both cases, but can partially be avoided by an improved end-effector plate design. The 5 DOF version of the Queensland Gantry-Tau allows reorientations of approximately 30° in the whole workspace.

In [Tyapin *et al.*, 2007] a geometrical method was presented to calculate the workspace volume and the volume of the unreachable workspace. The latter occurs in the center of the workspace for reconfigurable Gantry-Tau robots with sufficiently short linear actuators. Link lengths and linear guideway positioning were optimized for a maximal workspace. The workspace to footprint ratio for the Gantry-Tau prototype was found to be 2.7, which is large for a parallel robot. In [Tyapin *et al.*, 2008], the optimization was extended by considering the lowest resonance frequency.

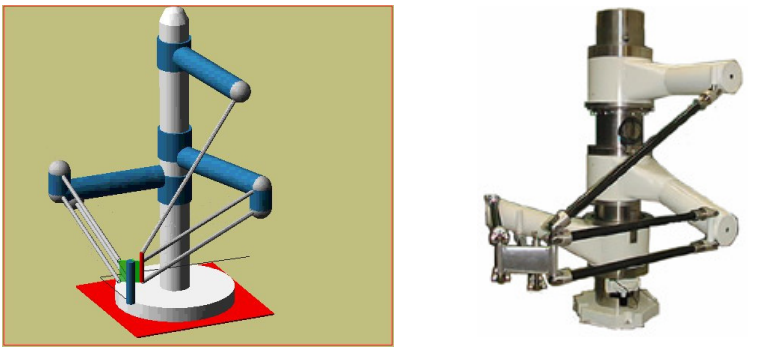


Figure 2.5 Animation (left) and prototype (right) of the Scara Tau robot developed by ABB Robotics.

A dynamic rigid-body model of the Gantry-Tau was verified by experiments in [Hovland *et al.*, 2007b]. This model includes also friction caused by the link forces. Static and dynamic stiffness was treated e.g., in [Tyapin *et al.*, 2008]. With optimized geometric design, the lowest calculated resonance frequency was larger than 45 Hz, while in [Hovland *et al.*, 2007b], the lowest measured resonance frequency was 13.5 Hz. The measurement was however done with the earlier, obviously less stiffer 3 DOF prototype. The stiffness modeling assumes flexible links and joints, while the framework and actuators are perfectly stiff. In [Brogårdh *et al.*, 2005], it was found that the stiffness of the framework did not match the high stiffness of the links and spherical joints, which were five times stiffer than those in [Tyapin *et al.*, 2008].

A way of reconfiguring the 5 DOF Gantry-Tau without passing through singularities was shown in [Murray *et al.*, 2008].

Related Parallel Robots

The most common parallel robot in scientific literature is undoubtedly the Stewart platform [Stewart, 1965], see Figure 2.6. An article search for "Stewart platform" in the Lund University Library database gives 25 149 results, whereas a search for "Delta robot" only 3665. In spite of intensive research, the Stewart platform has only found its way to industry in small number and very specific applications, e.g., as a positioning device [Symétrie, 2012]. The Delta robot dominates the industrial usage of parallel robots. At the 2012 Automatica fair in Munich [Automatica, 2012], a view in the online catalogue shows that out of 251 robotics exhibitors, ten companies offer parallel kinematic robots, of which five a Delta robot. In the 50 years since the first appearance of parallel robots, they have thus only reached limited application. The most successful parallel robot, the

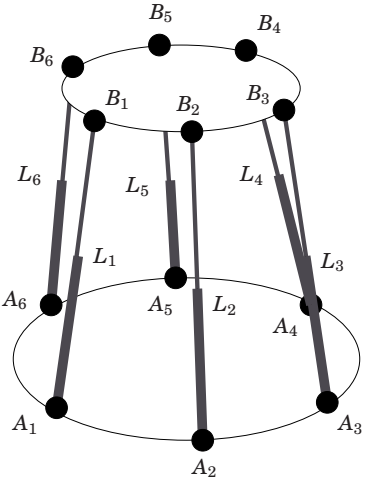


Figure 2.6 Stewart platform with link lengths L_i and spherical joint positions A_i on the base plate and B_i on the end-effector plate, $i = \{1,2,3,4,5,6\}$.

Delta, is mostly used for pick-and-place applications and thus only taking advantage of the PKM's high acceleration, but not of the in general high stiffness and accuracy. Other PKMs like the Tricept [Siciliano, 1999; S.L., 2012], a hybrid serial-parallel robot for machining applications, are used in limited numbers. This may have made it difficult to obtain funding and convince companies to invest in yet another new PKM structure, the Gantry-Tau robot.

The reason for the Stewart platform's dominant presence in literature is maybe its generality. Being a 6 DOF hexapod robot, i.e., a robot with six links, many results can be used for similar architectures, e.g., for the Gantry-Tau kinematics in Section 4.5. While the Stewart platform has actuated links, the 6 DOF Hexaglide [Honegger *et al.*, 1997] robot has six prismatic actuators moving the links' base points. The actuators are coplanar, and the workspace typically small. Clustering the six links, robots of lower than 6 DOF can be designed. Besides the Gantry-Tau robot with its 3-2-1 configuration, the Triglide [Budde *et al.*, 2008] is another 3 DOF robot, which has a 2-2-2 clustering of the links and prismatic actuators moving the links' base points.

2.3 Gantry-Tau Prototypes

The following prototypes, which were used in the research presented in this thesis, were constructed within the SMErobot and MONROE [MON-

ROE, 2012] projects. The links consist of carbon fibre tubes. High stiffness spherical joints ($115\,\mathrm{kN\,mm^{-1}}$ for the full size Gantry-Tau prototypes) were developed during the SMErobot project and further enhanced in the MONROE project. All prototypes are controlled by kinematics implemented by the author, or based on the author's work for the Java controlled T1.

The T1/T2 Gantry-Tau

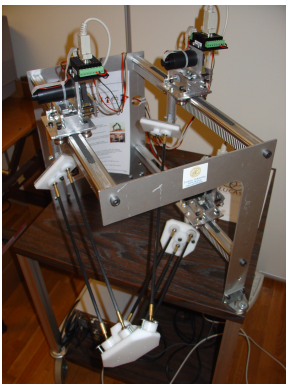
The T1, see Figure 2.7(a) for an early version, is a small tabletop prototype for teaching and research at Lund University. Various versions with continuously increasing size and improved performance were built. At first, it had the basic 3 DOF structure, but the newest version has 6 parallel DOF like the F1 below and is referred to as T2 (Figure 2.1). The control of the T1 prototype was implemented in real-time Java [Robertz *et al.*, 2007].

The D1 Gantry-Tau

The D1, see Figure 2.7(b), is a large size prototype developed by ABB Robotics, Güdel AG and Lund University. It was located at various sites, among others at CTI Sheffield, where calibration measurements and application tests were performed. Besides the basic 3 DOF Gantry-Tau structure, a lightweight 2 DOF serial wrist is mounted. It is controlled by a standard industrial ABB IRC5 controller with kinematics modules written by the author.

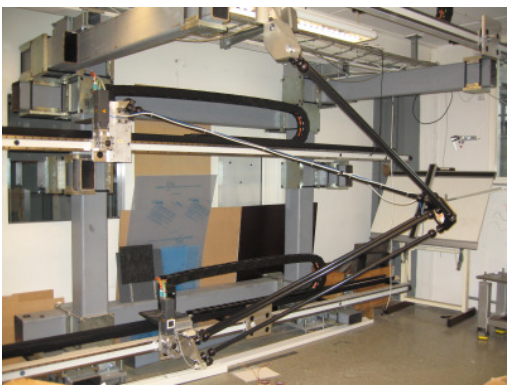
The L1/L2 Gantry-Tau

The L1, see Figure 2.7(c), is a somewhat smaller, pre-prototype version of the D1 robot located at Lund University. The first version had aluminum links which were later exchanged for stiffer carbon fibre links. By the end of 2008, a 2 DOF serial wrist was mounted. In 2010, the L1 Gantry-Tau was reconstructed with stiffer guideways and is referred to as L2 since then. Like the D1, it is controlled by ABB IRC5, but with the possibility to customize the controller [Blomdell *et al.*, 2010]. Most of the experiments presented in this thesis such as calibration, dynamic identification and ILC were performed on the L1 robot.



(a) T1 Gantry-Tau.

(b) D1 Gantry-Tau.



(c) L1 Gantry-Tau.

(d) F1 Gantry-Tau.

Figure 2.7 The Gantry-Tau prototypes.

The F1 Gantry-Tau

The F1, see Figure 2.7(d), has vertically mounted guideways. The joint attachment plates on the carts can be reoriented such that it has 6 parallel DOF. To increase positioning accuracy and remove backlash, the carts are actuated by double motors. It was mounted at Güdel AG in Switzerland and is now located at Lund University. The robot is controlled by the ISG-kernel software [ISG, 2012].

3

Software Support for Modeling and Control

3.1 Introduction

Flexible software tools are needed to support robotics research. At different stages of robot development, tools for modeling and simulation are required. For testing new motion control concepts, a high-bandwidth controller with an interface to external sensors is needed. Higher level concepts like trajectory optimization or model-based control might require interaction between different tools. Ideally, results, models and data can be interchanged between the different development stages. This means that either one tool comprises several purposes, or that several tools can interact and exchange data and/or models.

Different tools were used in the work presented in this thesis. This chapter intends to give some information on the nonstandard robot control system used and its possible usage together with other tools. In particular, Modelica based software and the Gantry-Tau modeling in the Modelica language done within the scope of this thesis will be addressed.

The remainder of the chapter is organized as follows: Section 3.2 describes the robot control system, Section 3.3 explains how Modelica models can be reused beyond pure simulation and e.g., interfaced with the extended IRC5 controller and Section 3.4 presents the Gantry-Tau modeling done in the Modelica language. Section 3.5 gives examples on how to use the software tools presented for iterative learning control, kinematic calibration and trajectory optimization.

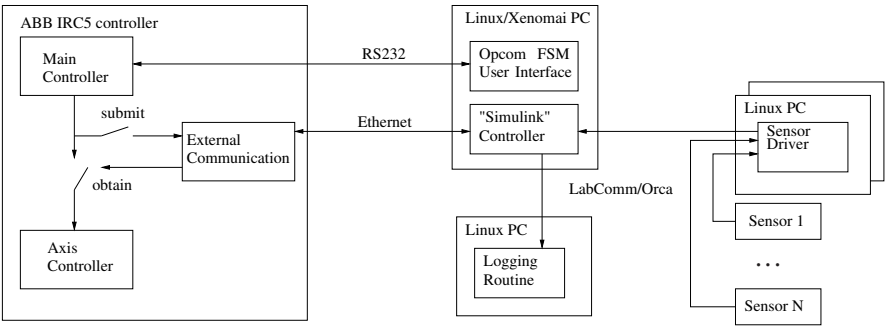


Figure 3.1 Structure of the extended ABB IRC5 control system.

3.2 An Extended ABB IRC5 Controller

To develop and test new robot control concepts, an open control interface with the possibility to connect computers for low level sensors control is needed. Today's industrial robot control systems are mostly highly optimized, closed systems without the possibility for high-bandwidth external sensing and interfaces for customized controllers. Below, an extended standard industrial ABB IRC5 controller [Blomdell *et al.*, 2005; Blomdell *et al.*, 2010] developed at Lund University and used for experiments presented in this thesis is described.

Figure 3.1 gives an overview of the extended controller structure. The ABB IRC5 controller is shown in the left of the picture. The standard version consists roughly of the main controller and an axis controller for each robot. The main controller includes path planning and trajectory generation from RAPID programs or manual jogging commands and sends motor reference values to the axis controller. The specially built IRC5 version used here contains an extra module that can communicate both with the internal IRC5 and external controllers. The values sent from main to axis controller can be read (submit state) and modified (obtain state) by the controller extension with a sampling time of 4 ms. Based on measurement signals received from the main controller, the reference values for the axis control can be modified based on customized control laws.

The customized controller communicating with the IRC5 system runs on a Xenomai Linux PC [Xenomai, 2012]. It can be implemented using Simulink and translated to C-code using Simulink Coder [MathWorks, 2012]. Besides standard Simulink blocks, more complex relations like kinematics, can be modeled using S-functions. Thus a Gantry-Tau kinematics library for Simulink was implemented [Friman, 2010] based on the au-

thor's IRC5 kinematics implementation. In Section 3.5, an example of such a controller modeled in Simulink is presented.

The Opcom user interface allows communication with the IRC5 system. From here, the customized controller can be loaded, submit and obtain states controlled and controller parameters changed. Via the LabComm/ORCA protocol [LabComm, 2010], data can be exchanged with the customized Simulink controller. This is mainly used for reading external sensors synchronized with the Simulink controller, but can as well be used for two-way communication applications like the haptic interface in [Eriksson and Welander, 2009].

An important property of the controller extension presented is the preservation of IRC5 safety functions. As well for safety reasons, it is preferable wherever possible that the external Simulink controller modifies the motor position and velocity reference for the axis controller instead of skipping the axis controller and modifying the motor torque reference directly.

3.3 Modelica-Based Robot Modeling and Control

Modelica [Modelica, 2011] is an object-oriented modeling language efficient in modeling complex multi-domain systems. Modelica is equation based, i.e., instead of solving for the differential equations and describing a system in explicit form, the equations are stated in implicit form. Causality between different submodels is not required, as it is e.g., in Simulink.

Below, the example of a general robot model illustrates the structure of Modelica models, which roughly consist of two sections. In the first section, all model parameters and variables are defined, with type (e.g., `motortorque`), name (e.g., `tau`) and eventually additional identifiers like `input`. The second section, indicated by the key word `equation`, contains the differential-algebraic equations describing the system.

```
model robotmodel
  parameterSet p;
  toolpose tcp;
  toolvel vtcp;
  toolacc atcp;
  jointpos q;
  jointvel vq;
  jointacc aq;
  input motortorque tau;
```

```
equation
  // relations between tool and joint coordinates
  0 = f_kin(tcp,q,p);
  0 = f_vel(tcp,vtcp,q,vq,p);
  0 = f_acc(tcp,vtcp,atcp,q,vq,aq,p);
  // force/torque balance
  0 = f_t(tau,tcp,vtcp,atcp,q,vq,aq,p);
  // state equations
  der(tcp) = vtcp;
  der(vtcp) = atcp;
end robotmodel;
```

In this example, the motor torques were chosen as input signals to the robot model and the variables referring to the tool center point (TCP) as states, but this can easily be changed. Apart from writing the model equations by hand, various libraries exist, e.g., the MultiBody library [Otter *et al.*, 2003] for rigid body systems. Tools like Dymola [Dynasim, 2009] offer a graphical user interface, so that a robot can intuitively be constructed by connecting links and joints. An advantage for PKM modeling is that with the MultiBody library, no special attention needs to be taken to kinematic redundancy or closed kinematic chains [Otter *et al.*, 2003].

Modelica robot models are conveniently used for simulation. Models can easily be extended, e.g., a model of the rigid body structure of the robot with motors or a controller. If needed, models can be migrated to other tools, as described in e.g., [Frenkel *et al.*, 2009]. The Gantry-Tau rigid body dynamics presented in Chapter 6 were extracted from a Modelica model using a tool described in [Dressler *et al.*, 2009]. In [Thümmel *et al.*, 2001], a feedforward signal for robot control was obtained by Modelica simulations of the inverse robot model. The usage of nonlinear inverse models in general for control by hardware-in-the-loop simulations of Modelica models was presented in [Thümmel *et al.*, 2005]. In [Pedreira Carabel and Zambrano García, 2011], a two-wheeled self-balancing vehicle is modeled using Modelica and by the aid of Dymola, a controller is designed and code for real-time control generated. How the inverse dynamic Modelica model of a parallel robot can be used for hardware-in-the-loop simulation control is described in [Krabbes and Meißner, 2006]. [Dressler *et al.*, 2009] describes modeling and control of the Gantry-Tau robot using the Modelica language.

Optimica [Åkesson, 2008] is a recently developed Modelica extension for describing optimization problems. A new class `optimization` is introduced together with an additional section `constraint`. The code example below illustrates the basic features of Optimica. For a better overview, many constraints and boundary conditions that need to be specified in a real application, such as start and end pose of the robot, were omitted.

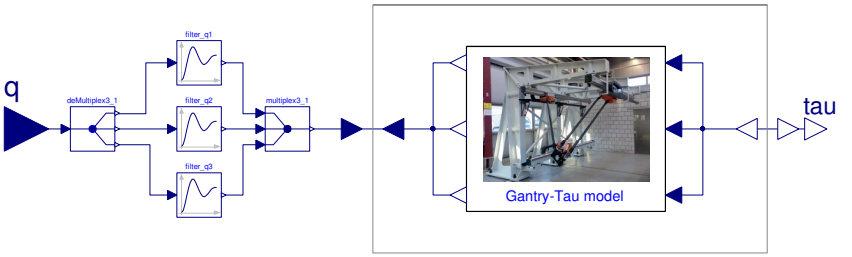


Figure 3.2 Example model used for S-function generation: An inverse dynamical model of the Gantry-Tau robot is constructed from the forward dynamic model ("Gantry-Tau model"). The 3 filters to the left are necessary to make the inverse model causal. Note how the robot model is connected to an additional block to invert the signal direction.

```
optimization opt_robot (
    objective = cost(finalTime),
    startTime = 0,
    finalTime = 1)
  robotmodel robot;
  Real cost;
equation
  der(cost) = f(robot.tau);
constraint
  robot.tau <= 1;
  robot.tau >= -1;
  robot.vtcp(finalTime) = 0;
end opt_robot;
```

Optimica also enables the formulation of minimum time problems by setting `finalTime(free=true)` and including it in the cost. JModelica.org [JModelica, 2010; Åkesson *et al.*, 2010] is an open source tool for simulation and optimization based on the Modelica and Optimica languages. For user interaction, the Python language [Python, 2011] is utilized. In robotics, Optimica can be used e.g., for optimal control or calibration as shown in Section 3.5.

In combination with the extended robot control system presented in the previous section, Modelica models can be used for real-time control. An S-function in the Matlab/Simulink controller is then associated to the Modelica model. The C-code required by the S-function can be generated using e.g., Dymola. Figure 3.2 shows an example model used for C-code generation for the inverse robot dynamics feedforward control, see Chapter 6.

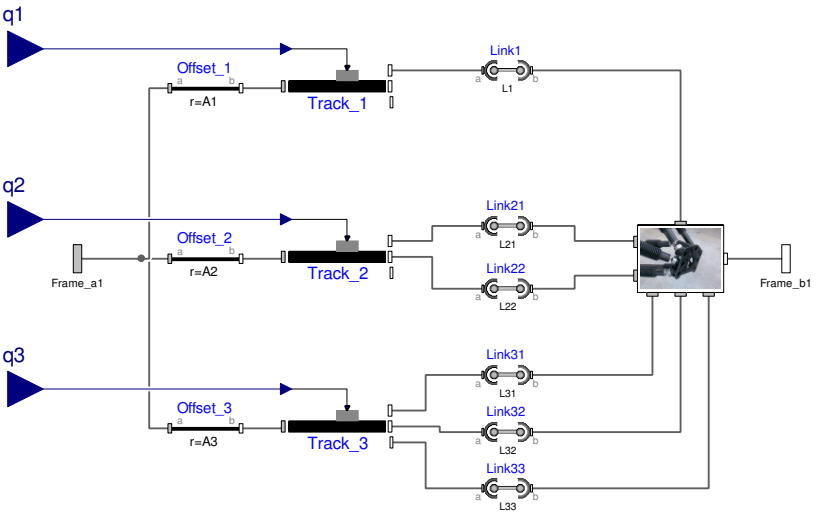


Figure 3.3 Modelica model of full Gantry-Tau forward kinematics with six links.

3.4 Gantry-Tau Modeling in Modelica

A library including the different parts of the Gantry-Tau robot and Gantry-Tau models for different purposes was implemented in Modelica and the MultiBody library. The library contains the mechanical parts like models for the actuators, the end-effector plate or the 2 DOF wrist, parameter sets for the different prototypes and robot models.

It is difficult to have one single model for all purposes, but simple to adapt the basic robot structure accordingly. Separate models are needed e.g., for inverse and forward kinematics. The prismatic joints in the forward kinematics model need to be active, i.e., include an input signal for the reference position. For the inverse kinematics, the end-effector position is given and the joint positions need to be determined. Therefore, the prismatic joints need to be passive, i.e., their position is not determined by a reference input but in relation to other robot parts connected in the same kinematic chain.

As the end-effector orientation is constant for all cart positions, it is sufficient to consider one link per kinematic chain, see Chapter 4. In addition to a full model with six links (Figure 3.3), a simplified Gantry-Tau model has been implemented (Figure 3.4). Here the end-effector orientation is kept constant by an additional block which contains three passive, serially connected prismatic joints aligned with the three principal coordi-

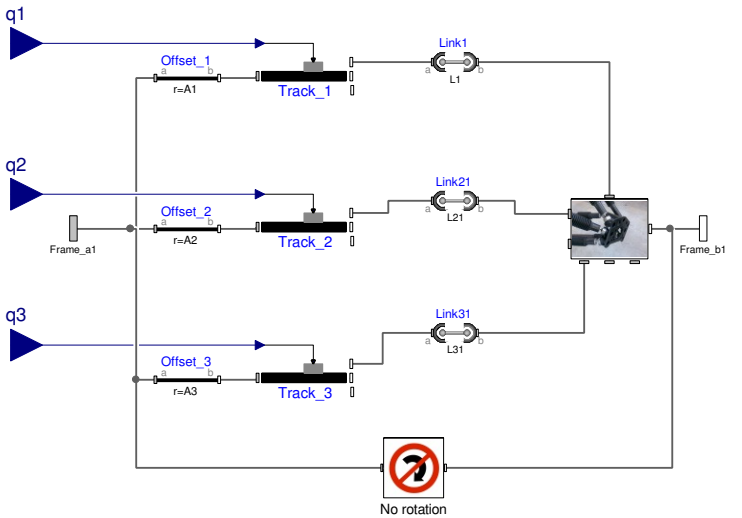


Figure 3.4 Modelica model of a simplified Gantry-Tau kinematics. The lowermost block in the model prevents the end-effector from changing orientation, and thus the model requires only three links instead of six.

nate axes. Each of the three kinematic chains visible in Figure 3.4 consists of a model for guideway and cart positioned in the base coordinate system by a `FixedTranslation` block and a link connected to the end-effector plate. The cart positions are the model's input signals.

The full Gantry-Tau model in Figure 3.3 can be used for modeling the kinematic error model presented in Section 4.4. It is possible to appoint different lengths to the links within one cluster or to modify the spherical joint placement so that it is not according to the Tau configuration any more, thus causing a variable end-effector orientation. With simply adding three rotational joints between guideways and links, a F1-type Gantry-Tau can be modeled. The model shown here is a forward kinematics model. It is more difficult to construct a working, full six-link inverse kinematic model. As the kinematic error model still has only three translational DOF, the end-effector position has to be specified without locking the orientation. This can in principle be done putting a spherical joint between the end-effector plate and a block specifying the position, but the resulting equation system could not be solved. To obtain a working model, customized equation systems should be utilized rather than standard parts.

The model used for extraction of the dynamic model equations is shown in Figure 3.5. The nominal, simplified kinematics with only three links

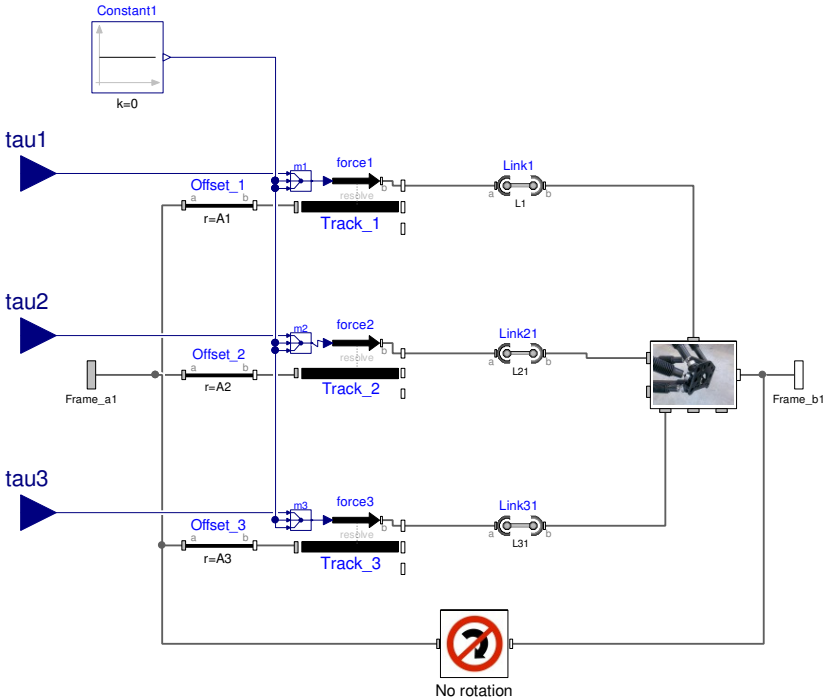


Figure 3.5 Modelica model of Gantry-Tau with actuator torques used for extraction of rigid body dynamic equations.

is used here. Instead of appointing the robot's actuator positions as in Figure 3.3, a force in direction of the actuator axis is applied to each cart.

Friction has to be added manually to the robot models. The commonly adopted combination of Coulomb and viscous friction might pose problems with some tools as JModelica.org, as it is discontinuous around zero velocity. In that case, it can be approximated by a piecewise linear function, see Figure 3.6. Another possibility, if possible, is to constrain the joint movement in one direction and use a linear friction model. In [Krabbes and Meißner, 2006], the friction force was filtered to avoid this problem. The same limitations and approaches are applicable for more advanced friction models, e.g., containing the Stribeck effect.

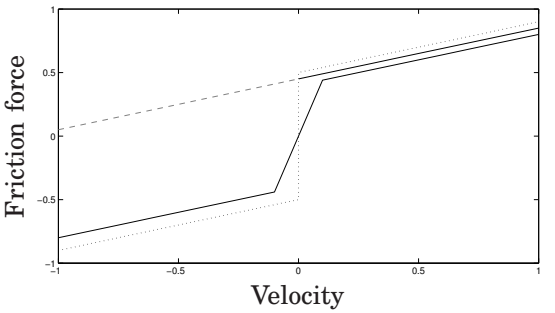


Figure 3.6 Illustration of the friction force approximation for the Modelica Gantry-Tau model. The original discontinuous model with Coulomb and viscous friction (dotted) together with the continuous, piecewise linear model (solid) and the linear approximation for positive velocities (dashed/solid). For better visibility, the Coulomb friction coefficients were slightly modified between the models.

3.5 Application Examples

Iterative Learning Control

Below, the controller extension presented is illustrated by an ILC application. Here, only the usage of the extended robot controller is addressed, whereas the experimental results are discussed in detail in Chapter 7. Even if this example does not reflect the full potential in adding complex controllers including e.g., stateflow diagrams for mode changes, it captures the main communication flow and relates to the described modularity of the system.

When applying ILC, the robot is to “learn” a specific motion. This is done by modifying the control signal with small corrective terms calculated from the control errors every time a motion is performed. Alternatively to the control signal, the ILC can act on the reference signal to an inner controller. Thus, the motion should converge iteratively to the desired reference trajectory.

Of the signals which can be modified by the external controller, the control signal which is acting most directly on each motor is the motor torque reference. For the ILC experiments however, the corrective terms are added to the motor-angle reference. The derivative of the ILC input signal is added to the motor angular velocity reference. Without limiting the ILC performance, this is a considerably safer choice than modifying the torque references.

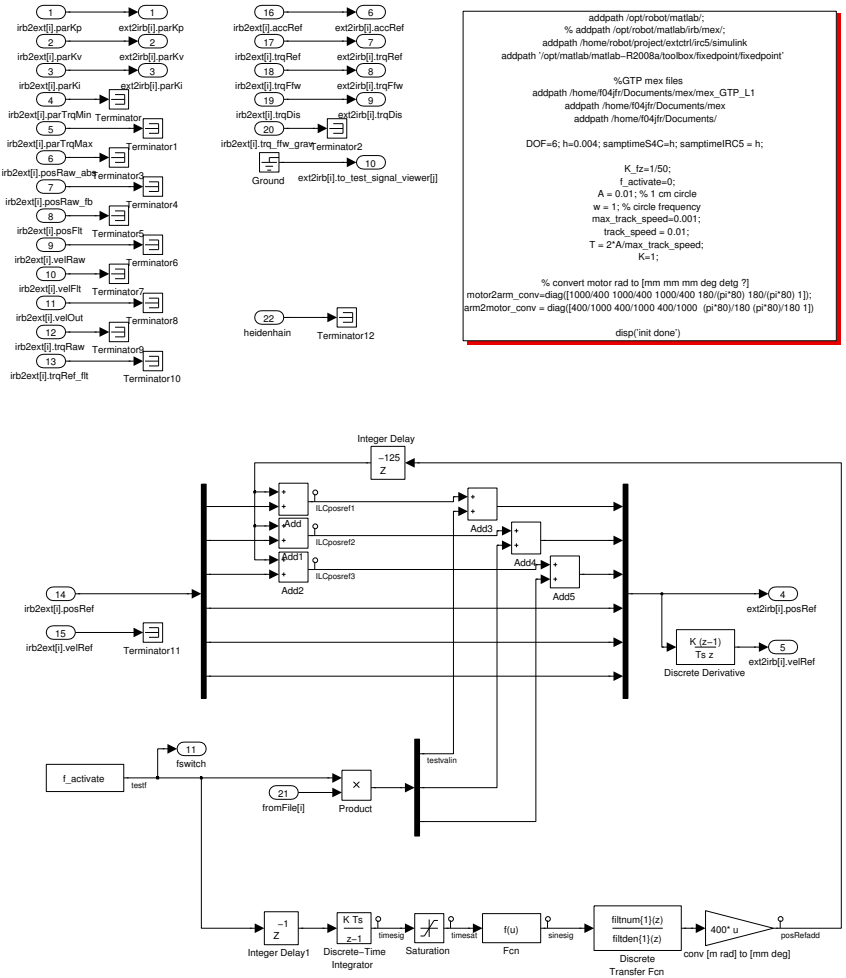


Figure 3.7 Example: Simulink model for ILC: The model contains the standard input and output signals from and to the IRC5 system and an input from an external linear encoder (`heidenhain`). The reference signal is generated from the user interface input `f_activate`, and the correction terms are read from a file, generated by the ILC-algorithm.

Figure 3.7 shows the Matlab/Simulink ILC implementation. The ILC references and corrective terms are added to the constant, internal IRC5 values for motor position and velocity references. The external controller can in principle also be used together with a reference trajectory from the IRC5 main controller, e.g., given by a RAPID program. However, it is more convenient for the synchronization between the reference trajectory and the corrective terms to control the complete motion in the external controller. The only external sensor in this experiment is a linear encoder to measure the arm side position. The ILC experiment can be started manually by the user in the Opcom user interface via a switch parameter.

The switch parameter (`f_activate`) is implemented as a modifiable parameter and the experiment thus started from the Opcom user interface. The reading of reference trajectory and corrective terms from a file are synchronized with this parameter.

For the reference trajectory, here a filtered sine signal was chosen. The synchronization with the `f_activate` switch is here done by integrating `f_activate`, which is set to 1 to start the experiment. This creates a time signals `t`, which is fed into a `sin(t)` block. A saturation block on the time signal terminates the sine after an appropriate number of cycles. A more flexible way would be to read the reference from a file, as described below for the corrective terms.

The corrective terms are calculated offline and written to a file. A C-program was written that uses the LabComm protocol to connect to the ORCA client. It receives the `f_activate` signal, and when it switches to 1, it opens the specified file and starts to send the corrective values to the ORCA client with the same sampling interval it receives `f_activate` from the ORCA client.

Any additional sensor, such as the high-resolution linear encoder in this example, can communicate with the ORCA client in a similar way. A C-program sets up the connection and sends the measurement data to the ORCA client. To ensure exact synchronization of the sampling, the sending can be adjusted to data reception from the ORCA client.

Kinematic Calibration with Optimica

The procedure of kinematic calibration is explained in detail in Chapter 5. Here, the aspects of using Optimica for kinematic calibration are considered.

Calibration of the nominal 3 DOF Gantry-Tau kinematics determines altogether 21 kinematic parameters. For that, the end-effector position $T = (X, Y, Z)$ is measured for a number N of actuator positions (q_1, q_2, q_3). The optimized parameters minimize a cost function based on the equations describing the kinematics.

As the MultiBody Library is not yet compatible with Optimica, a so-

called flat Modelica model for optimization had to be generated, i.e., a model with only one hierarchy level. For that, the model equations can be extracted automatically from the MultiBody model as described above in Section 3.3. With the measured data and a subset of these equations describing the kinematics, a model for optimization is then generated as follows. Of course, the model below can easily be written manually, but an automated method simplifies modifications as e.g., for a new prototype with new nominal parameters or for a modified kinematic structure.

```
model GTPKinCalib

  parameter Real q1[N] = {data}; // Reference
  parameter Real q2[N] = {data}; // actuator
  parameter Real q3[N] = {data}; // positions

  parameter Real X[N] = {data}; // Measured
  parameter Real Y[N] = {data}; // end-effector
  parameter Real Z[N] = {data}; // positions

  // Kinematic parameters
  parameter Real L1; // Length link 1
  parameter Real sA1[3]; // Offset actuator 1
  parameter Real v1[3]; // Axis actuator 1
  parameter Real L2; // Length link 2

  ...

  Real f1[N];
  Real f2[N];
  Real f3[N];
  Real cost;

equation
  for i in 1:N loop
    f1[i] = Residual kinematic constraint link 1, measurement i;
    f2[i] = Residual kinematic constraint link 2, measurement i;
    f3[i] = Residual kinematic constraint link 3, measurement i;
  end for;

  cost = f1[1]^2+f2[1]^2+f3[1]^2+ ...;

end GTPKinCalib;
```

The variable `cost` is then minimized using Optimica. The L1 prototype's end-effector position $T = (X, Y, Z)$ was recorded for 176 robot poses with known actuator positions (q_1, q_2, q_3) using a laser tracker. Figure 3.8 shows the calibration results. The calibrated model has a mean absolute tool positioning error of about 140 µm.

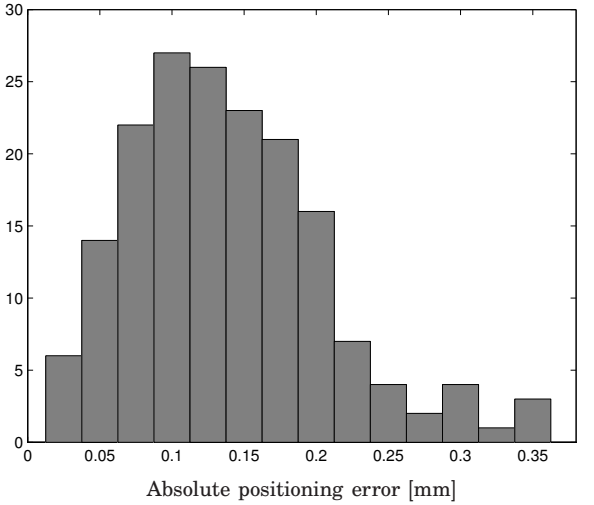


Figure 3.8 Positioning accuracy of the L1 Gantry-Tau after calibration: Histogram of absolute positioning error measured at TCP.

Optimal Trajectory Generation with Optimica

The usage of Optimica for dynamic optimization applications like motion planning is illustrated by a comparison of direct trajectory planning [Choset *et al.*, 2005] and optimal path tracking [Pfeiffer and Johanni, 1987; Verscheure *et al.*, 2009]. For direct trajectory planning, the complete motion is optimized in one step, whereas for optimal path tracking, the motion along a given path is optimized. Optimica was first used for optimal robot control in [Hast *et al.*, 2009]. The speed along a path recorded by lead-through programming was optimized, subject to torque constraints, along the concept of [Dahl, 1992].

With a simple change in the Modelica model, the robot motion can be restricted to a given path. Thus, no penalty for deviating from the path has to be added to the cost function. The code example below from the `equation` section illustrates the modification for a robot with three translational DOF.

```
// state equations
der(tcp.x) = dtcp.x;
der(tcp.y) = dtcp.y;
der(tcp.z) = dtcp.z;
der(tcp.dx) = ddtcp.x;
der(tcp.dy) = ddtcp.y;
der(tcp.dz) = ddtcp.z;
```

The original code above is replaced with the lines below. The number of states is reduced from six to two. Instead of declaring all three motor torques as `input` signals, only one of the torques is an input, and the other two torque signals are treated as dependent variables.

```
// state equations
der(tcp.x) = dtcp.x;
der(tcp.dx) = ddtcp.x;

// new algebraic equations
tcp.y = f_y(tcp.x);
tcp.z = f_z(tcp.x);

tcp.dy = f_dy(tcp.dx);
tcp.dz = f_dz(tcp.dx);

tcp.ddy = f_ddy(tcp.ddx);
tcp.ddz = f_ddz(tcp.ddx);
```

The motion between two given poses is optimized for the Gantry-Tau robot. A free optimization of the complete trajectory is compared to optimizing along a linear Cartesian path between the poses. In both cases, constraints on the motor torques are considered. In both start and end pose, velocities and accelerations are zero. The cost function C is a sum of the total time duration T_{Final} and a weighted term proportional to the energy used.

$$C = T_{\text{Final}} + w \sum_{i=1}^{3} \int \tau_i^2 dt$$

Results with different values for w are presented below, including the minimum time problem for $w = 0$. The path tracking is optimized for $w = 0$. As initial guess, the same linear motion between start and end pose with a trapezoidal velocity profile was used for all cases.

Figures 3.9 – 3.10 and Table 3.1 show the results. For the minimum time problem and a free path, all three motor torques are at their maximal values for the largest part of the motion. For the path constrained optimization, motor 3 is at its maximum value. This is consistent with motor 3 traveling the largest distance. With increasing energy term in the cost function, the torques decrease towards the path constrained case. The time optimal path saves around 0.07 s, that is 7 %, compared to the linear path, but to the cost of nearly twice the energy. The increasing number of iterations with decreasing w for the direct trajectory optimization can be explained by a comparison with the optimal paths in Figure 3.9. The differences in number of iterations in Table 3.1 correlates with how close the optimized paths were to the initial guess trajectory.

Table 3.1 Direct versus path constrained optimization.

	Final Time [s]	Energy $\sum \int \tau_i^2 dt$	Number of Iterations
Direct trajectory optimization with weight w			
$w = 0$	0.9074	103.8448	199
$w = 0.001$	0.9191	77.0024	53
$w = 0.01$	0.9968	50.1301	26
Path constrained optimization			
	0.97612	57.32659	70

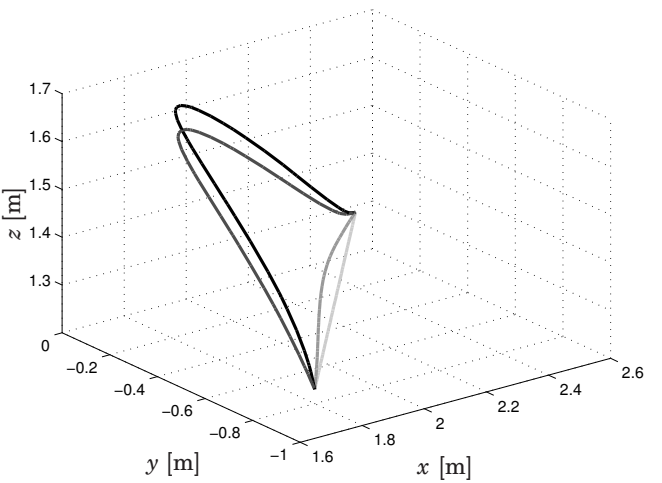


Figure 3.9 Optimized TCP path. Path constrained optimization (lightest gray shade) and direct optimization with $w = 0.001$, $w = 0.01$ and $w = 0$ (increasing darkness of gray shade with decreasing w).

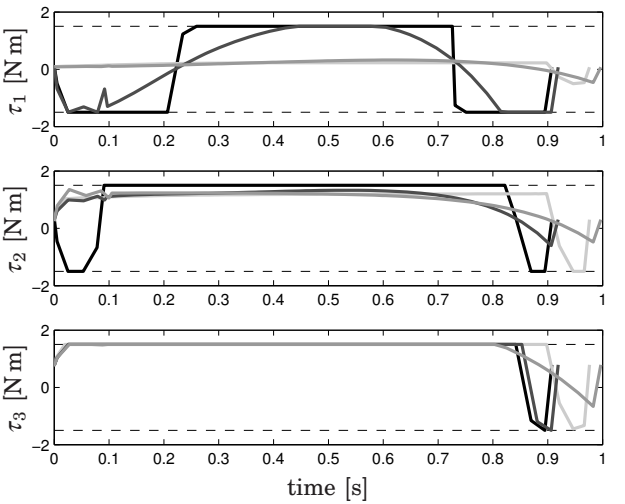


Figure 3.10 Optimized motor torques for the TCP paths in Figure 3.9.

4

Kinematic Modeling

The kinematic modeling presented in this chapter is an important basis for Chapter 5 on kinematic calibration. First, a brief summary of parallel robot kinematics and the nominal kinematic model of the basic 3 DOF Gantry-Tau robot is given. Then, the validity of the nominal model is discussed based on measurements and the influence of modeling errors on the Cartesian positioning accuracy studied. Based on the results, error kinematic modeling of the Gantry-Tau robot is studied. The chapter concludes with kinematic modeling of the extended 6 DOF Gantry-Tau structure.

4.1 Preliminaries

The kinematics of a robot describes the relation between the actuator positions q and the end-effector (tool) pose (T, R_T) with position $T = (T_x, T_y, T_z)^T$ and orientation R_T. Further, the kinematics is a function of the kinematic parameter set s and possibly a configuration state c to distinguish between several solutions if more than one exist. Unlike for serial robots, for parallel robots the forward kinematics

$$(T, R_T) = f_{\mathrm{fk}}(q, s, c) \tag{4.1}$$

is in general more difficult to solve than the inverse kinematics

$$q = f_{\mathrm{ik}}(T, R_T, s, c) \tag{4.2}$$

The configuration state c comprehends both forward and inverse kinematics configuration, which are possibly independent from each other. General solution methods exist [Merlet, 2000]; analytic or geometric methods for the inverse kinematics and iterative methods for the forward kinematics. The kinematic constraint for each kinematic chain i connecting the

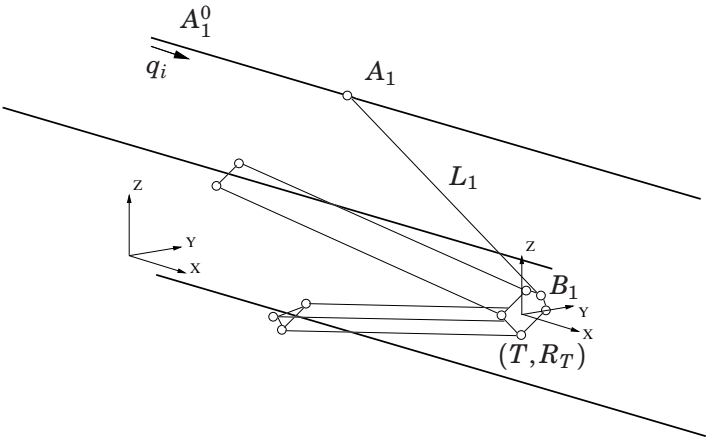


Figure 4.1 Schematic Gantry-Tau PKM with parameter and variable notation exemplified on arm 1.

basis with actuators (attachment point A_i) and the end-effector (attachment point B_i) often plays a central roll. For many PKMs including the Gantry-Tau it can be written in the form

$$L_i \cdot u_i = A_i(q_i) - B_i(T,R) \tag{4.3}$$

where u_i is the unitary vector along link i with length L_i. In many cases the kinematic constraint is used as a scalar equation

$$L_i^2 = \|A_i(q_i) - B_i(T,R)\|^2 \tag{4.4}$$

For some PKMs, e.g., the Stewart platform, the attachment point A_i on the base side of kinematic chain i is fix and it is rather the vector along the link, $L_i \cdot u_i$, that depends on q_i.

4.2 Nominal Kinematic Model and Validity

Nominal Kinematic Model

The nominal kinematics of the Gantry-Tau PKM was first described by [Johannesson *et al.*, 2003].

Figure 4.1 illustrates the notation used for geometric parameters and variables. Linear guideways and link clusters are numbered according to the number of links in the kinematic chain. Two different coordinate frames are used: The global frame and the end-effector frame.

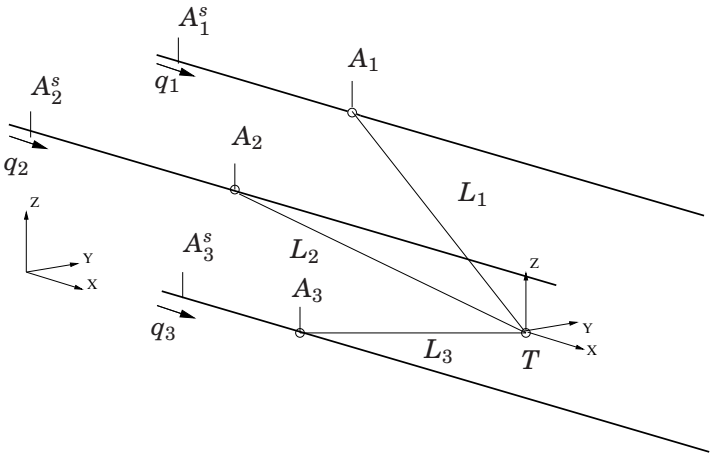


Figure 4.2 Schema of simplified Gantry-Tau.

For each link j of arm i, the spherical joint positions A_{ij} and B_{ij} in the kinematic constraint (4.3) or (4.4) can be expressed as

$$A_{ij} = A_{ij}^0 + q_i \cdot v \tag{4.5}$$

$$B_{ij} = T + R_T \cdot B_{ij}^0 \tag{4.6}$$

where A_{ij}^0 is the spherical joint position associated with $q_i = 0$, v the direction of the guideways, and B_{ij}^0 the (constant) spherical joint position given in the end-effector frame.

The kinematic chain $i = 1$ has only one link, link 1. Chain 2 has two links, $2a$ and $2b$, and chain 3 three links, $3a$, $3b$ and $3c$.

Thanks to the Tau-configuration, the orientation of the end-effector plate is constant and the 3 DOF of the robot are completely translational. It is therefore sufficient to consider the simplified geometry in Figure 4.2 with one link representing each cluster. The spherical joint positions of the simplified robot are then

$$A_i = (A_{ij}^0 - R_T \cdot B_{ij}^0) + q_i \cdot v = A_i^s + q_i \cdot v \tag{4.7}$$

$$B_i = T \tag{4.8}$$

with A_i^s as the spherical joint position associated with $q_i = 0$ of the simplified robot. As the links within a cluster form parallelograms, (4.7) and (4.8) are valid for any link of that cluster. Now the model has only three links, $i = \{1,2,3\}$. The kinematic constraint equation for link i

$$L_i^2 = \|A_i^s + q_i \cdot v - T\|^2 \tag{4.9}$$

can then easily and independently for each kinematic chain be solved for q_i to solve the inverse kinematics.

The forward kinematics problem consists in solving the three equations (4.9) for the end-effector position T, its orientation R_T being constant and known. This problem, which is equivalent to intersecting 3 spheres, is called trilateration and is well-known in many other domains [Coope, 2000]. In [Johannesson *et al.*, 2003], a stepwise geometric solution for the Gantry-Tau forward kinematics is suggested. Another possibility is to solve the equations with the aid of a symbolic computation tool like Maple.

For the nominal forward kinematics, two solutions exist. The end-effector can be on either side of the actuators. For the inverse kinematic problem, for each link cluster two solutions exist, altogether $2^3 = 8$ solutions. When only one particular solution is desired, e.g., when using the kinematic model for control, a configuration state c has to decide which one of the solutions is wanted. Note, however, that the configuration state resulting from forward kinematics is not related to the inverse kinematics configuration.

The workspace of the simplified robot, i.e., all robot poses where a mathematical solution of the kinematics exists, is reduced for the real six link robot, not only by limits of the passive and active joints, but also by possible collisions between the links and the framework or with the end-effector plate. For reconfiguration, i.e., moving from one kinematic solution to the other, it is as well often a problem to find a trajectory that does not pass through singularities. The Gantry-Tau prototypes treated in this thesis cannot move from one configuration to the other. However, the Queensland Gantry-Tau can reconfigure [Murray *et al.*, 2008] as well as other PKMs with a similar structure, e.g., the Triglide [Budde *et al.*, 2008]. While the Queensland Gantry-Tau can reconfigure kinematically, i.e., without passing through a singularity, the Triglide has to use its inertia to traverse a singularity when reconfiguring. A reconfiguration without passing singularities is much preferred as the robot is controllable along the complete path.

Modeling Assumptions

The nominal kinematics relies on the following assumptions:

- The spherical joints are positioned on the carts and on the end-effector plate according to the Tau configuration.
- The guideways are linear and parallel to each other.
- All links within one cluster have equal length.

Table 4.1 Actuator accuracy: Residuals $\|e\|$ in (4.10).

Prototype	D1			L1		
Actuator	1	2	3	1	2	3
mean [µm]	32	77	54	163	76	116
std [µm]	14	36	18	87	47	62

These assumptions assure the links within one cluster building up parallelograms and the end-effector thus keeping a constant orientation. In the following, measurements are presented in order to assess how well these assumptions hold in practice. The consequences of the results on the kinematics and Cartesian positioning accuracy are then discussed. The analysis focuses on the linearity of the tracks and the deviation of the end-effector orientation from its constant nominal orientation. Section 4.3 will then study in more detail how modeling errors influence the Cartesian positioning accuracy.

Linearity of guideways Based on the linear actuator model (4.7), the position of a point P attached to one of the carts as the cart moves along the guideway can be expressed as

$$P(q) = P(q_0 = 0) + q \cdot v + e \tag{4.10}$$

where e is the modeling error.

Measurements to evaluate the actuator linearity were performed on the L1 and D1 Gantry-Tau prototypes. Using a laser tracker [Leica Geosystems, 2009], with a corner cube reflector for D1 and a cateye reflector for L1, the three-dimensional cart position was measured along the guideways keeping the yz-position of the end-effector constant, with the coordinate system according to Figure 4.2. For the L1 prototype, the guideway measurements were repeated for a different end-effector yz-position and thus different link angles to investigate the influence of small load changes on the cart position. The reflectors were mounted as close as possible to the spherical joint centers.

Table 4.1 shows the residuals $\|e\|$ from (4.10) for the measurement data for both prototypes. The mean norm of the residuals differs considerable between the actuators and prototypes and lies between 32 µm to 77 µm for the D1 prototype and between 76 µm to 163 µm for the L1 prototype. The standard deviation is approximately half of the mean value.

In Figure 4.4, the modeling error of the D1 prototype is divided into two parts:

- A part projected on the actuator axis

$$e_{\parallel} = e^T \cdot v \tag{4.11}$$

  This part may be caused by bad positioning or errors in the transmission.

- A part orthogonal to it

$$e_{\perp} = e - e_{\parallel} \tag{4.12}$$

  This part of the error may be caused by the guideway lacking stiffness or straightness.

It can be seen that actuator 2, and to a lower degree actuator 3, exhibit a cyclic error along the actuator axis. A plausible reason for this might be mechanical imperfection of the transmission. As the cycle length between the teeth of the rack- and pinion transmission was much shorter than the measurement resolution of 50 mm, aliasing effects can be presumed. Another source of error could be the motor angle sensor for the robot servo. Even though resolvers are considered to give very accurate angle measurements, amplitude deviations and imperfect quadrature of the resolver may cause the effect of a small additive sinusoidal error [Hanselman, 1991]. However, the D1 prototype was unfortunately not available to repeat the measurements on with a higher resolution. To investigate the qualitative behavior and risk of aliasing, measurements were instead made on the L2-prototype, which has a similar motor and transmission system as the D1-prototype.

The cart position for a linear actuator of the L2-robot was measured with a length gauge (Heidenhain ST 3078 [Heidenhain, 2010]) and a linear actuator model with respect to the motor angles was estimated by the least squares method to (a sequence) of motions along the rail. Figure 4.3 shows the error e between the externals measurement and the linear actuator model for uni-directional and bi-directional motions, respectively. The measurements supports the occurrence of periodic patterns with shorter period(s) than the measurement grid used for the calibration. One can also get a good estimate of the width of the backlash in the transmission ($\approx 40\mu m$) from the measurements of the bi-directional motion, see Figure 4.3 (lower plot).

Another influence on P's position can be forces acting on a cart. That might be load changes, which are not present during the measurements, or even if the other two carts are moved to another position, the gravity force transmitted in the link changes and might affect the cart. To study

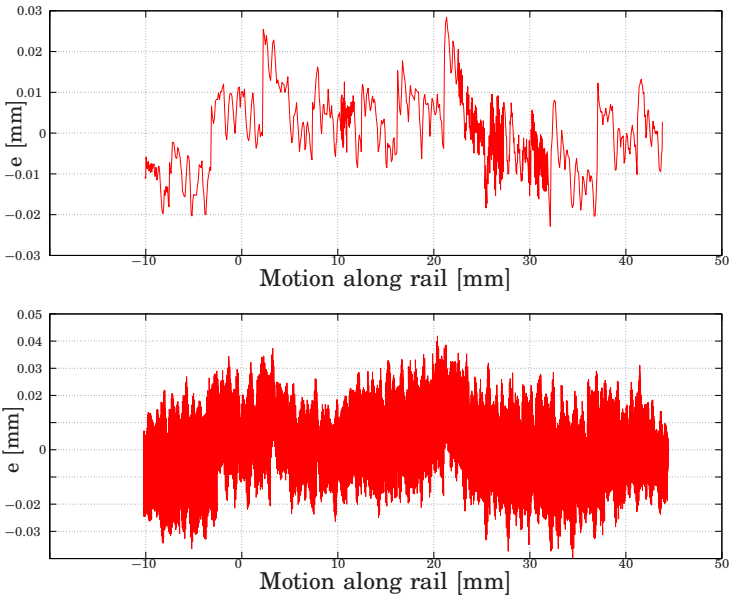


Figure 4.3 Error measurements for the L2-prototype. *(Upper)* Actuator model error e, uni-directional motion. *(Lower)* Actuator model error e, bi-directional motion.

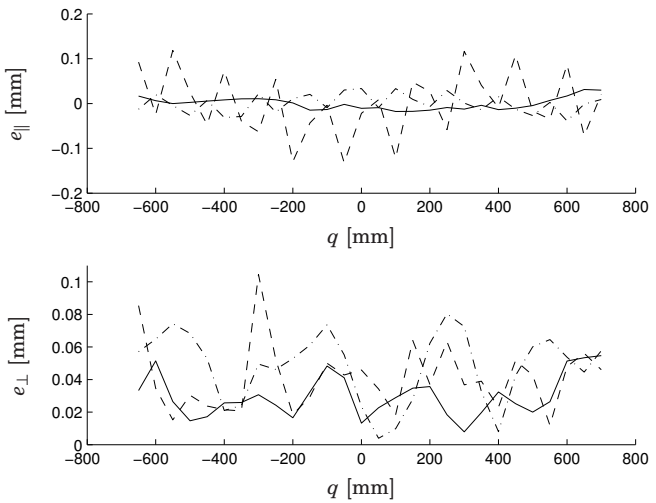


Figure 4.4 Actuator accuracy for D1 prototype: Residuals e in (4.10), $e_{\parallel}$ projected on (upper diagram) and $e_{\perp}$ orthogonal to (lower diagram) actuator axis for actuators 1 (solid), 2 (dashed) and 3 (dashed-dotted).

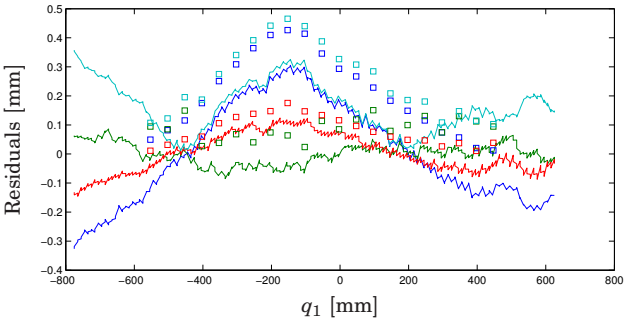


Figure 4.5 Actuator accuracy for L1 prototype: Residuals e in (4.10) for actuator 1, x (blue), y (green), z (red) and norm (turquoise). The squares show the residuals of data with the end-effector having a different position in the plane orthogonal to the actuators. The values are given in the measurement frame with the y-axis approximately along the actuators and the z-axis upwards.

this influence, L1 measurements with different yz-position of the end-effector were performed. Figure 4.5 shows the results. The solid lines show the residuals $\|e\|$ of the data used for modeling, i.e., calculated with the optimized P_0 and v, from data with one constant yz-position. The squares are the model residuals of data from a second experiment where the end-effector's position in the yz-plane was changed, but the same P_0 and v as with the solid lines are used. The measured cart position varied up to 100 µm. The experiment results suggest that even small load changes caused by a redistribution of the gravity force may give rise to substantial deviations from the actuator model.

These experiments give an indication about how large errors can be expected in the modeling of A_i in (4.7). However, there are other possible errors that cannot be identified with the measurements performed.

As the measurements were only three-dimensional, they cannot detect a shifting orientation of a cart along the guideway. Therefore, the reflectors were placed as near as possible to the spherical joints.

Other problems that might affect the actuators are e.g., backlash, wear, temperature or deformations. To eliminate the influence of backlash in the measurements, the carts were moved only in one direction. Other prototypes, namely F1 and T2, have linear actuators driven by double motors to eliminate backlash altogether.

End-effector orientation For each of 176 TCP poses lying on a grid filling the D1 robot's workspace, the position and orientation of the 3 DOF end-effector were recorded with a laser tracker and T-Mac [Leica Geosystems, 2009]. Figure 4.6 illustrates the measurement setup and poses.

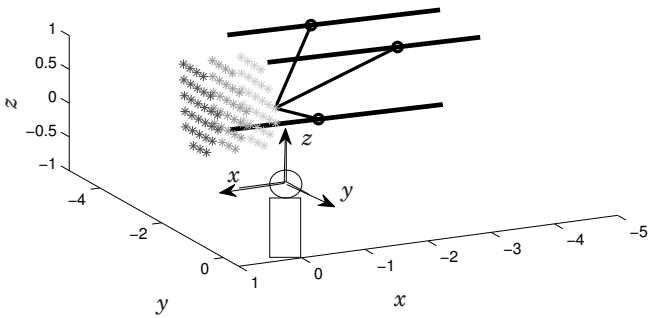


Figure 4.6 Schema of simplified Gantry-Tau with measurement points and corresponding laser tracker positioning. Here, only every second measurement point layer is shown for better visibility (first layer to the left in the figure). The coordinate system is scaled in [m].

Errors in the end-effector orientation can have different causes. If the carts on the guideways have a varying orientation, the end-effector orientation will vary as the end-effector moves in parallel to the actuator axes. If the links of the robot are such that they cause orientation errors, e.g., having different lengths of incorrect joint placements, the orientation error will be the same as the end-effector moves in parallel to the actuator axes, as the links of the robot do not move relatively to each other.

Figure 4.7 shows the end-effector orientation represented as ZYZ Euler angles (α, β, γ) along the grid. The maximal Euler angle variations lie between 0.1° (α) and 0.5° (β). The repeating pattern exhibited can be associated with the six grid layers orthogonal to the actuator axes that the end-effector is traversing. The robot is alternatively moving forward in one layer and then traversing backwards along the same path for the next layer, see Figure 4.6. The pattern in Figure 4.7 indicates that the TCP orientation errors are mainly caused by the links. The end-effector orientation varies only little for positions with the same *yz*-coordinates, i.e., for a movement parallel to the actuator axes. but more within one layer, i.e., for positions with different *yz*-coordinates. Together with the manner in which the grid is traversed, this creates the regular pattern. If the cart orientations varied along the guideways, the end-effector would change its orientation when moving in parallel to the actuators.

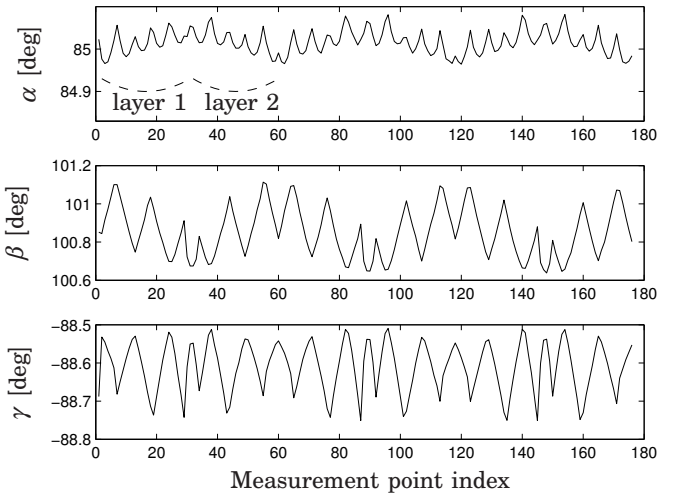


Figure 4.7 Orientation error: ZYZ Euler angles of end-effector plate orientation measured along a grid (Figure 4.6). The pattern corresponding to the grid point layers is marked out for layers 1 and 2.

4.3 Influence of Modeling Errors on Positioning Accuracy

This section relates the modeling errors found in the previous section to the Cartesian positioning accuracy of the Gantry-Tau robot using the method described in Appendix A.

Related Work

In [Kim and Choi, 2000], a distinction is made between forward error bound analysis, which consists in finding the end-effector errors given a bound on the joint errors, and inverse error bound analysis, where the joint tolerance limits given a maximal possible end-effector error are determined. The accuracy of hexapod parallel robots, in particular the Stewart platform, has been studied extensively. An early example is [Wang and Masory, 1993], where an error model of the Stewart platform using the Denavit-Hartenberg convention is developed and the end-effector pose deviation from the desired trajectory is studied, given random errors which lie within the manufacturing tolerances by simulating the nominal inverse and error forward kinematics. In [Wang and Ehmann, 2002], an automated error analysis system is developed for computation and graphical illustration of the influences of geometrical errors on the end-effector pose accuracy of the Stewart platform. In [Merlet and Daney, 2005], it is

Table 4.2 Assumed actuator errors for simulation study. All values in [µm].

Actuator i	Case 1	Case 2	
	$e_{i,x} = e_{i,y} = e_{i,z}$	$e_{i,x}$	$e_{i,y} = e_{i,z}$
1	32	11	28
2	77	58	38
3	54	19	47

described how to find robot dimensions that guarantee a given end-effector accuracy with bounded manufacturing errors and actuator accuracy using interval analysis of the kinematic Jacobian matrix. Recent work includes accuracy analysis using a force-based method based on the kinetostatic dualism [Pott *et al.*, 2007], particle swarm optimization [Hui and Feng, 2009] or Edgeworth series and information entropy [Du *et al.*, 2010]. Even though most studies are performed on purely geometrical errors, [Merlet, 2000] gives examples of other work on thermal, dynamic or gravity induced errors.

Accuracy Analysis

In the following, two different scenarios are simulated, based on the D1 prototype geometry and accuracies found previously. The Cartesian positioning error in a workspace slice orthogonal to the actuator axes is examined. First, it is assumed that the actuator positioning error is equal in all direction, i.e., the real spherical joint position A_i lies inside a sphere around the modeled position. This actuator modeling error has the same influence on the Cartesian accuracy as a length error in the corresponding link. Second, it is assumed that the error is not equal in all directions, but that the error along the actuator axis differs from the error orthogonal to it. The actuator has not the same mechanical properties in all directions, and the error along the actuator axis depends on additional factors like the transmission and the position control, which motivates this choice. Measurements of these two error components are shown in Figure 4.4. The real spherical joint position A_i lies then inside an ellipsoid with axes $(e_{i,x}, e_{i,y}, e_{i,z})$ aligned with the base coordinate directions. Table 4.2 summarizes the assumed actuator modeling errors. The analysis method considers only errors in the actuator modeling and is based on the nominal kinematics with only three links, i.e., it is not examined how other modeling errors, e.g., in the link lengths within one cluster, affect the end-effector position and orientation. Such an analysis can be carried out using the kinematic error model in Section 4.4.

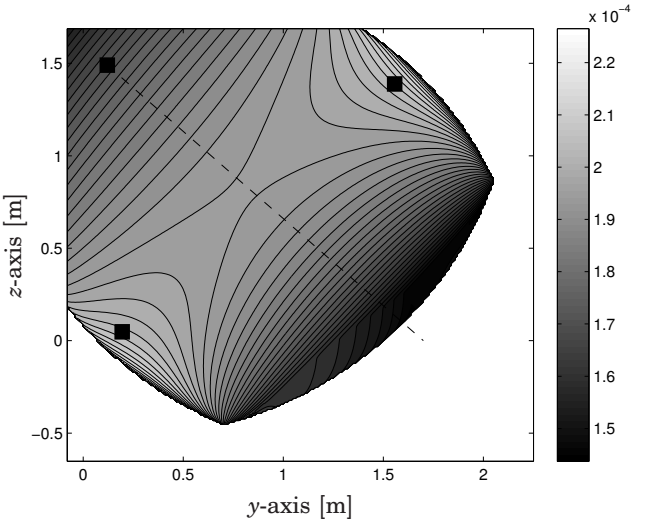


Figure 4.8 End-effector accuracy in a *yz*-workspace section: Level plot of maximum possible positioning error [m] for case 1. The black squares indicate the actuator positions. The area examined is very roughly limited by the robot framework (left and upper edge of the figure).

Figure 4.8 shows the maximum possible Cartesian error for case 1. The values differ between 140 µm to 230 µm, which is considerable larger than the actuator modeling error. The smallest values are reached at the workspace edge in the lower right corner of the figure. In the workspace center, there is a saddle point with an error of approximately 0.2 mm. The maximum error is largest close to the actuators in the lower left and upper right corner of the figure. The shape of the Cartesian error bounds was also studied. As shown in Appendix A, it can be approximated by a parallelepiped. By comparing the area of the three pairs of parallelogram faces, which are orthogonal to one of the links each, one can see if the error is dominant in a certain direction or within a certain plane. Figure 4.9 shows that the error is most dominant in the plane orthogonal to link 1, which is expected since actuator 1 exhibits the smallest modeling error.

Figure 4.10 shows the maximum possible Cartesian positioning error for case 2. The values vary between 90 µm and 170 µm. The largest actuator modeling error is in the actuator axis direction of actuator 2. This matches with the tool error being largest in proximity of guideway 2, where link 2 is close to being parallel to guideway 2.

Parallel robots are commonly considered as more accurate than serial robots, and it is sometimes said that their geometric errors compensate for each other rather than adding up as for serial robots. The results

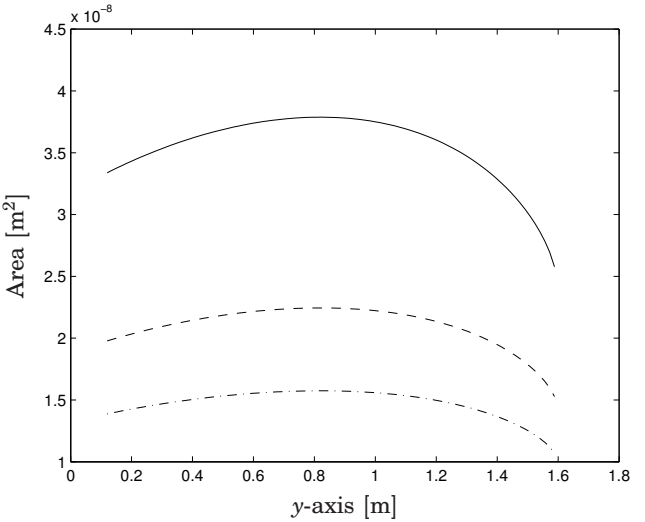


Figure 4.9 Geometry of Cartesian error bounds: Area of the parallelogram faces of the parallelepiped approximation along the dashed line in Figure 4.8. Faces orthogonal to link 1 (solid), link 2 (dashed-dotted) and link 3 (dashed).

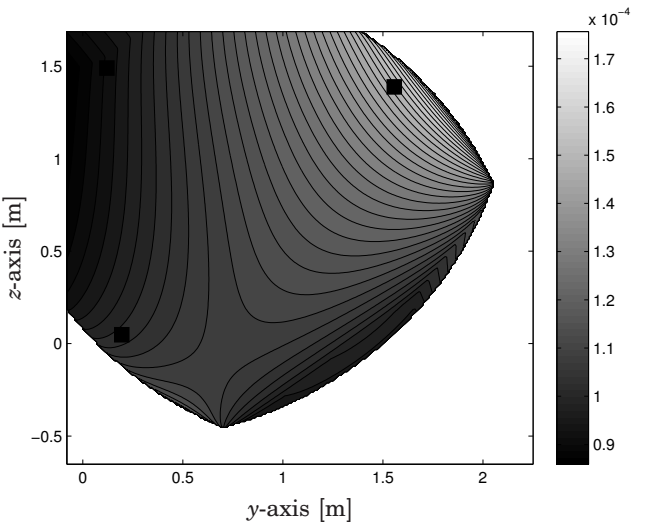


Figure 4.10 End-effector accuracy on a *yz*-workspace section: Level plot of maximum possible positioning error [m] for case 2. The black squares indicate the actuator positions. The area examined is very roughly limited by the robot framework (left and upper edge of the figure).

show that in the worst case, the Cartesian error can be a multiple of the error of each arm. The results here should be seen as a motivation for the kinematic error modeling in the following section. In order to assess the Gantry-Tau's sensitivity to modeling errors properly, a more thorough study has to be performed, including more aspects than only the actuators and a comparison with other methods.

4.4 Kinematic Error Model

Kinematic error models for numerous parallel manipulator architectures were presented in the past. An early example is [Wang and Masory, 1993], which presents a kinematic error model for a Stewart platform. In [Cui *et al.*, 2005] and [Williams *et al.*, 2006] error kinematic models for the Scara Tau and the Queensland Gantry-Tau robot, respectively, are presented. To the author's knowledge, the models presented were often verified by simulations and few results on the kinematic accuracy from experimental measurements exist.

In the following, kinematic error models are discussed with a focus on the linear guideways and the link clusters. The error kinematic model for the guideways concentrates on arbitrary actuator axes as in [Williams *et al.*, 2006]. A more prototype-specific model for non-linear actuators is presented in Chapter 5 in connection with the calibration of the kinematic error model.

Actuators

The nominal kinematic model assumes that the actuators are perfectly linear and parallel to each other. Assuming an arbitrary orientation of guideway i instead of all guideways being perfectly parallel, the position A_i of cart i can be expressed as

$$A_i = A_i^0 + q_i \cdot v_i \tag{4.13}$$

where v_i is the unit vector along the guideway in positive q_i direction, which is now different for each guideway.

This change does not affect the kinematics solution method, except for exchanging v with v_i. The stepwise geometric method [Johannesson *et al.*, 2003] can be applied to solve the forward kinematic problem if the cart positions are modified according to (4.13) and the inverse kinematics can still be solved independently for each actuator. The inverse and forward kinematic problems have the same number of solutions as in the case of parallel articulator axes.

However, the simplified kinematic model is based on the Tau configuration and constant end-effector orientation. Errors in the guideway orientation are most likely caused by imprecise mounting of the guideways, inherent in the basic concept of a reconfigurable robot structure for SMEs. Especially the usage of box joints [BoxJoint, 2011] for the frameworks of the L1 and T1 prototypes, which eases reconfiguration of the robot, makes small orientation errors around the z- or y-axes, see Figure 4.1, more probable than for an elaborated CAD design with accurately drilled holes for mounting the framework. With errors in the guideway orientation, the orientation of the carts and link geometry will not be the intended one and it is probable that the links do not form exact parallelograms any more. That might cause the end-effector to have a varying orientation across the workspace, and the simplified kinematic model is not valid any more. Even if the resulting end-effector orientation is constant, it might not be the intended one. Not all guideways affect the end-effector orientation. Link cluster 1 has no influence on it, while link clusters 2 and 3, which form a parallelogram and a prism with triangular base, respectively, determine the end-effector orientation.

If only the commanded actuator positions q_i and the 3-dimensional end-effector position data are used for calibration, it is not possible to identify the orientation of the end-effector.

The kinematic error model of the Queensland Gantry-Tau presented in [Williams *et al.*, 2006] adopts two-dimensional orientation and positioning errors for each guideway, while it does not discuss the end-effector orientation errors that may likely be caused by imprecise mounting of the guideways.

In this work, positioning errors of the guideways were not explicitely taken into account because the mathematical description of the robot kinematics does not change and a kinematic calibration identifies the changed parameters directly without the need of an additional parameter describing the deviation from the nominal parameter. The kinematic error model of the links presented below takes cart and end-effector orientation errors into account. The guideway model presented above assumes still perfectly linear actuators. In Chapter 5 a nonlinear guideway model will be presented.

Link clusters

This section presents a kinematic model describing errors in the link clusters. For this, all six links have been taken into account instead of only three. Varying end-effector orientation errors can arise if the links in one cluster have slightly different lengths or if the joint placement on the carts and on the end-effector is not according to the Tau configuration.

The kinematic constraint equation for link j of arm i and for any

actuator model $A_{ij}(q_i)$ is then

$$L_{ij}^2 - \| A_{ij}(q_i) - (T + R_T \cdot B_{ij}^0) \|^2 = 0 \tag{4.14}$$

Assuming a linear actuator with $A_{ij}(q_i)$ according to equation (4.5), the inverse kinematics is as easy to solve as the nominal kinematics. More advanced actuator models might require iterative solutions.

As the value of A_{ij} is known, the forward kinematics can be solved with any actuator model using an iterative Newton-Raphson scheme [Merlet, 2000]. A minimal representation X of the end-effector pose (T, R_T), in this case with *ZYZ* Euler angles instead of the rotation matrix R_T, is updated every iteration k according to

$$X_{k+1} = X_k + (\frac{\partial f_{\text{ik}}}{\partial X})_k^{-1} (q - f_{\text{ik}}(X_k, s, c)) \tag{4.15}$$

until the absolute joint error $|q - f_{\text{ik}}(X_k, s, c)|$ is sufficiently small. This solution was implemented and used for the work presented in Section 5.3.

In principle, the inverse kinematics f_{ik} can be exchanged for that of any other hexapod robot with the same link lengths and end-effector plate geometry. Consequently, some results on the Stewart platform kinematics are also valid for the Gantry-Tau kinematics. These results are discussed in Section 4.5 together with the forward kinematics of the F1-type Gantry-Tau, which is solved in a similar manner. As the F1-type kinematics were used for real-time control, a more thorough examination of the iterative algorithm was required compared to the kinematic error model presented here.

4.5 Kinematics of the F1-type Gantry-Tau

The F1-type Gantry-Tau robot has an extended kinematic structure developed by Adam Nilsson in cooperation with Güdel AG [Nilsson, 2011]. The spherical joints on carts 2 and 3 are mounted on a rotating plate. For cart 3, this plate can also tilt. Figures 4.11 and 4.12 illustrate the extended architecture and the notation for the joint variables. The rotation axes for q_4 and q_5 are parallel to each other. Unlike for a fully parallel robot having only one actuated joint per kinematic chain, arm 2 is now a serial combination of a prismatic and a rotational actuator and arm 3 a combination of a prismatic and two rotational actuators.

The T2 prototype has the same structure for the parallel kinematic wrist, but its linear actuators are arranged horizontally instead of vertically. An often utilized convention is to arrange the z-axis of the base

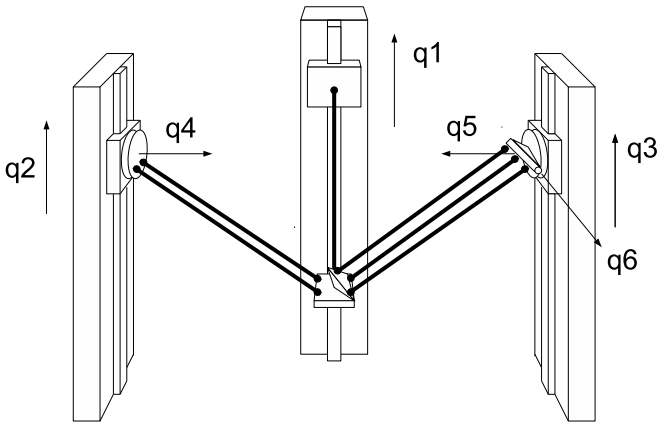


Figure 4.11 Schematic overview of the F1-type Gantry-Tau robot. The prismatic actuators move along the indicated axes q_1 to q_3, the rotational actuators rotate around the axes q_4 to q_6.

coordinate sytem vertically. This was applied here for the F1 kinematics, rather than using a coordinate system with its x-axis aligned with the linear actuators as for the horizontally arranged Gantry-Tau robots. The coordinate systems for the F1 robot differ therefore from the coordinate systems used in the rest of the thesis. The z-axis is vertical and parallel to the linear actuator axes, and the xy-plane is arranged horizontally and orthogonal to the linear actuator axes.

Other hybrid robot structures with serial and parallel parts are often a serial combination of parallel or serial structures as presented in e.g., [Tanev, 2000]. The D1 and L1 prototypes with the serial wrist are such serial combinations, while the idea of extending the Gantry-Tau's DOF using extra carts on the guideways, which is shortly presented in Chapter 2 in Figure 2.3, is still a purely parallel robot. A version of the Queensland Gantry-Tau was extended to 5 DOF by exchanging one link in each of clusters 2 and 3 by a prismatic joint [Murray *et al.*, 2006], resulting thus like the F1-type Gantry-Tau in a parallel combination of serial chains.

The F1's combination of serial and parallel kinematics makes it difficult to find a completely analytic solution for both forward and inverse kinematics. If the plate rotations q_4 and q_5 are synchronized such that $q_4 = -q_5$, end-effector orientation and position can be considered separately and an analytic kinematic solution can be found. In this case, the

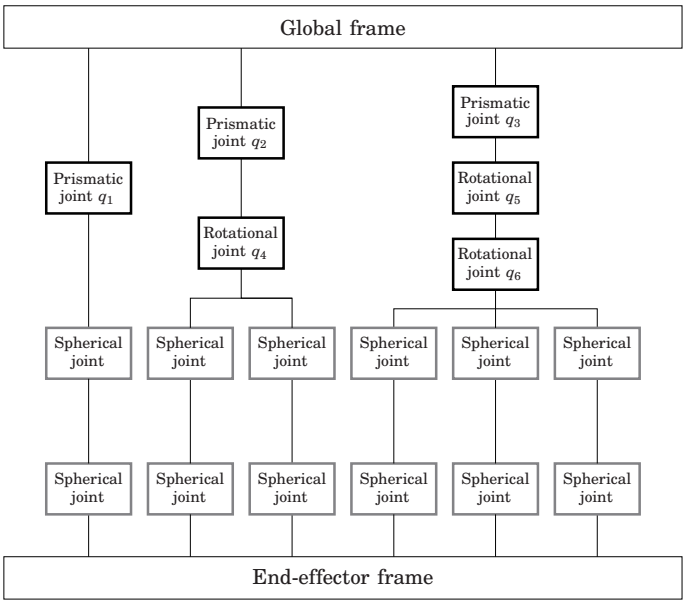


Figure 4.12 Illustration of the kinematic chains of the F1-type Gantry-Tau robot. Actuated joints are marked with black boxes, passive joints with grey boxes.

F1 robot has only 5 instead of 6 DOF. For the general 6 DOF case, iterative solutions have to be used for both forward and inverse kinematics. The remainder of this section discusses the analytic 5 DOF kinematics and the iterative 6 DOF kinematics.

All kinematics presented below require the calculation of the spherical joint positions on the end-effector plate and on the carts. For the end-effector plate, the relation does not change with now six actuators, whereas for the joints on the carts, more complex kinematic chains need to be considered for arms 2 and 3. As an alternative to the customized coordinate systems used here, standard parametrizations like the Denavit-Hartenberg convention [Hartenberg and Denavit, 1955] can be used.

$$\begin{bmatrix} A_1 & 1 \end{bmatrix}^T = H_1(q_1) \cdot \begin{bmatrix} A_1^{\mathrm{bj}} & 1 \end{bmatrix}^T \tag{4.16}$$

$$\begin{bmatrix} A_{2j} & 1 \end{bmatrix}^T = H_2(q_2, q_4) \cdot \begin{bmatrix} A_{2j}^{\mathrm{bj}} & 1 \end{bmatrix}^T, \qquad j = \{a, b\} \tag{4.17}$$

$$\begin{bmatrix} A_{3j} & 1 \end{bmatrix}^T = H_3(q_3, q_5, q_6) \cdot \begin{bmatrix} A_{3j}^{\mathrm{bj}} & 1 \end{bmatrix}^T, \qquad j = \{a, b, c\} \tag{4.18}$$

$$B_{ij} = T + R_T \cdot B_{ij}^0 \tag{4.19}$$

where A_{ij}^{bj} is the ball joint position on cart i and H_i is the homogeneous

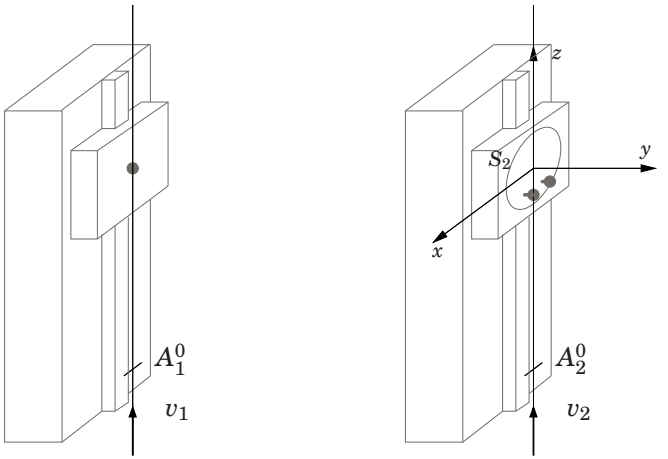


Figure 4.13 Illustration of the local coordinate systems for arm 1 (left) and arm 2 (right). Observe that for arm 2, the coordinate system S_2 is attached to the cart, but not to the rotating plate, i.e., the ball joint positions are constant with respect to a coordinate system that rotates around the y-axis of S_2.

coordinate transform between the moving coordinate frame attached to the corresponding cart and the global frame. In contrast to A_{ij}^0 in (4.5), A_{ij}^{bj} is not expressed in the global frame, but in a coordinate frame fixed to the moving joint plate, and thus has a constant value. The parameters and coordinate systems are illustrated in Figures 4.13 and 4.14.

Arm 1 As no rotational joint is involved, (4.5) is valid and

$$H_1(q_1) = \begin{bmatrix} \mathbf{I} & A_1^0 + q_1 \cdot v_1 \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.20}$$

Without loss of generality, it is assumed that $A_1^{\text{bj}} = \mathbf{0}$. The guideway base point A_1^0 corresponds in that case to the ball joint center position for $q_1 = 0$.

Arm 2 Instead of the ball joint center as in (4.5) and (4.20), A_2^0 refers here to the origin of the coordinate system S_2 for $q_2 = 0$. The orientation of S_2 in the global frame is defined by $R_{1_{u_2}}$, see Figure 4.13.

$$H_2(q_2, q_4) = \begin{bmatrix} R_{1_{u_2}} \cdot R_{q_4} & A_2^0 + q_2 \cdot v_2 \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.21}$$

Without loss of generality, the coordinate frame S_2 is chosen such that q_4

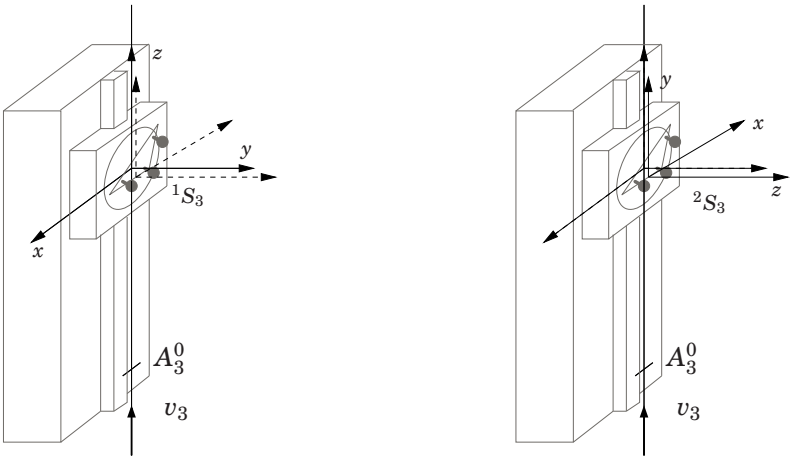


Figure 4.14 Illustration of the local coordinate systems for arm 3. The y-axis of the frame $^1S_3$ defines the rotation of joint q_5 (left) and the x-axis of the frame $^2S_3$ the rotation of joint q_6. To better be able to distinguish between the two coordinate systems, in the left picture, the coordinate axes of $^1S_3$ are drawn in solid lines and in the right picture the axes of $^2S_3$. The axes of the second coordinate system are drawn in dashed lines.

rotates around its y-axis, so that

$$R_{q_4} = \begin{bmatrix} \cos q_4 & 0 & \sin q_4 \\ 0 & 1 & 0 \\ -\sin q_4 & 0 & \cos q_4 \end{bmatrix} \tag{4.22}$$

Arm 3 Similar to arm 2, A_3^0 and $R_{1_{u_3}}$ refer for $q_3 = 0$ to the origin and orientation, respectively, of the coordinate system $^1S_3$, which defines the q_5 rotation by its y-axis. The location of the coordinate system $^2S_3$ with respect to $^1S_3$ is determined by the offset r_3 and orientation $R_{2_{u_3}}$.

$$H_3(q_3, q_5, q_6) = \begin{bmatrix} R_{1_{u_3}} \cdot R_{q_5} \cdot R_{2_{u_3}} \cdot R_{q_6} & A_3^0 + q_3 \cdot v_3 + R_{1_{u_3}} \cdot R_{q_5} \cdot r_3 \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.23}$$

Unlike q_4 and q_5, which are rotations around the y-axis, the convention for q_6 was defined such that it rotates around the x-axis of $^2S_3$, so that

$$R_{q_6} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos q_6 & -\sin q_6 \\ 0 & \sin q_6 & \cos q_6 \end{bmatrix} \tag{4.24}$$

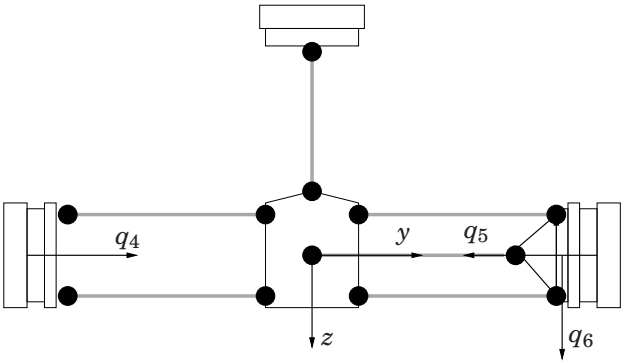


Figure 4.15 Relation of end-effector orientation and rotational joint positions for 5 DOF kinematics: If $q_4 = -q_5$, the end-effector rotation around its y-axis corresponds exactly to the q_4 rotation. The end-effector's rotation around its z-axis corresponds to the q_6 rotation. For the 5 DOF case, the end-effector cannot rotate around its x-axis, which is orthogonal to the image plane.

Implicit 5 DOF Kinematics

If the rotational joints q_4 and q_5 move equally such that $q_4 = -q_5$, end-effector position and orientation can be considered separately. The end-effector orientation is then directly related to the joint values (q_4, q_5, q_6) as illustrated in Figure 4.15. Knowing the end-effector orientation and (q_4, q_5, q_6), structural parameters $s_{3\text{DOF}}$ for a corresponding 3 DOF Gantry-Tau robot can be calculated and the 3 DOF kinematics can be solved.

The inverse kinematics is then

$$\begin{aligned}(q_4, q_5, q_6) &= f_{\text{ik, rot}}(R_T, s_{5\text{DOF}}) \\ s_{3\text{DOF}} &= f_s(s_{5\text{DOF}}, R_T, (q_4, q_5, q_6)) \\ (q_1, q_2, q_3) &= f_{\text{ik, 3DOF}}(T, s_{3\text{DOF}}, c)\end{aligned} \tag{4.25}$$

and the forward kinematics

$$\begin{aligned}R_T &= f_{\text{fk, rot}}((q_4, q_5, q_6), s_{5\text{DOF}}) \\ s_{3\text{DOF}} &= f_s(s_{5\text{DOF}}, R_T, (q_4, q_5, q_6)) \\ T &= f_{\text{fk, 3DOF}}((q_1, q_2, q_3), s_{3\text{DOF}}, c)\end{aligned} \tag{4.26}$$

For the rotational part of the kinematics, only one solution exists, as the rotational joint values correspond directly to the end-effector rotation around the x and y-axes. As before, the 3 DOF Gantry Tau kinematics has several solutions, two for the forward kinematics and $2^3 = 8$ for the inverse kinematics, determined by the configuration state c. These numbers describe however the mathematical solutions to the 3 DOF kinematics. The

full six-link robot may not be capable of reaching all 3 DOF solutions, but then, more solutions may be found with the full 6 DOF kinematics.

The constraint $q_4 = -q_5$ prevents the end-effector to rotate around it x-axis, so that it has only two rotational DOF. Another consequence is that the link clusters do not twist, i.e., the links always form parallelograms.

Iterative Inverse 6-DOF Kinematics

For solving the inverse kinematics, the actuated joint positions for a given end-effector position and orientation are calculated. As before, this can be done separately for each kinematic chain. For arm 1, an analytical solution analog to the 3 DOF kinematics exists.

The inverse kinematics of arms 2 and 3 is solved using a standard Newton-Raphson iterative algorithm. The procedure is exemplified for arm 3. The joint variables $Q = (q_3, q_5, q_6)$ are updated according to

$$Q_{k+1} = Q_k + \frac{\partial Q}{\partial e} \cdot e_k \tag{4.27}$$

until a sufficiently small error residual e_k is obtained, with

$$e_k = \begin{pmatrix} L_{3a}^2 - \|B_{3a} - A_{3a}(Q_k)\|^2 \\ L_{3b}^2 - \|B_{3b} - A_{3b}(Q_k)\|^2 \\ L_{3c}^2 - \|B_{3c} - A_{3c}(Q_k)\|^2 \end{pmatrix} \tag{4.28}$$

The term $\Delta Q_k = \dfrac{\partial Q}{\partial e} \cdot e_k$ for updating Q is calculated by solving the linear equation system

$$\frac{\partial e}{\partial Q} \Delta Q_k = e_k \tag{4.29}$$

or by using the pseudoinverse of the matrix $\dfrac{\partial e}{\partial Q}$.

In particular for a stable real-time usage, it is important to know that the algorithm always converges to the desired solution. Although theoretical results on the convergence of the Newton-Raphson algorithm exist, it is in practice very difficult to determine analytically how many real solutions, or assembly modes, the F1-type inverse kinematics has and how large the domain of convergence around these solutions are. An examination of the convergence thus relies on numerical simulations. The inverse kinematics presented were implemented and successfully used for real-time control. The iterative inverse kinematics was, as often implemented, running at a high sampling rate, and using the solution from the previous sample as initial condition for the next sample to preserve the configuration.

Iterative Forward 6-DOF Kinematics

The F1 forward kinematics can be solved in a similar way as the six-links error kinematics from Section 4.4. As mentioned in Section 4.4, once the ball joint positions A_{ij} have been calculated with (4.16) – (4.18), any hexapod forward kinematics can be used to calculate the end-effector pose. Instead of using the actual F1-type kinematics or the available six-link error kinematics, the Stewart platform kinematics was chosen. Besides the numerous available research results, it has the advantage that the inverse kinematics, which has to be calculated in every iteration, always has a solution.

Similar to (4.15), the algorithm is formulated as

$$X_{k+1} = X_k + \mathbf{J}_{\mathrm{SP}} \, (\mathbf{L} - \mathbf{AB}_k) \tag{4.30}$$

where $X = (T, \alpha, \beta, \gamma)^T$ is a minimal representation of the end-effector pose using ZYZ Euler angles. $\mathbf{L} = (L_1, L_{2a}, L_{2b}, L_{3a}, L_{3b}, L_{3c})^T$ is the constant vector of all six link lengths and $\mathbf{AB}_k$ the vector of the distances $(B_{ij}(X_k) - A_{ij})$ between the six ball joint pairs for a given end-effector pose X_k at iteration k calculated with (4.16) – (4.19). The row of the Stewart platform's inverse Jacobian matrix $\mathbf{J}_{\mathrm{SP}}^{-1}$ corresponding to link ij is calculated as in [Merlet, 2000]

$$\left[\frac{AB_{ij}^T}{\|AB_{ij}\|} \quad \frac{AT_{ij}^T \cdot \frac{\partial R_T}{\partial \alpha} \cdot B_{ij}}{\|AB_{ij}\|} \quad \frac{AT_{ij}^T \cdot \frac{\partial R_T}{\partial \beta} \cdot B_{ij}}{\|AB_{ij}\|} \quad \frac{AT_{ij}^T \cdot \frac{\partial R_T}{\partial \gamma} \cdot B_{ij}}{\|AB_{ij}\|} \right] \tag{4.31}$$

where $AB_{ij} = B_{ij} - A_{ij}$ is the vector along link ij and $AT_{ij} = T - A_{ij}$ is the vector between the end-effector frame's origin and ball joint ij. As for the inverse kinematics algorithm for arm 2 and 3, the term $\mathbf{J}_{\mathrm{SP}} \, (\mathbf{L} - \mathbf{AB}_k)$ is calculated either using the pseudoinverse of $\mathbf{J}_{\mathrm{SP}}^{-1}$ or by solving a linear equation system.

To obtain fast convergence with a good initial value X_0, which is close to the desired solution, the implicit 5 DOF kinematics is used.

$$X_0 = f_{\mathrm{fk,\,5DOF}}(q_{\mathrm{5DOF}}) \tag{4.32}$$

The assumption $q_4 = -q_5$ is assured by

$$q_{\mathrm{5DOF}} = [q_1, q_2, q_3, \frac{q_4 - q_5}{2}, \frac{q_5 - q_4}{2}, q_6] \tag{4.33}$$

There exist as well other schemes to solve the forward kinematics, e.g., interval analysis [Merlet, 2000], which were not considered here.

Assembly Modes

For a stable real-time control, it is important that the algorithms converge fast, and that they converge to the desired solution. The inverse kinematics is calculated at the control system's sampling rate, and the control system architecture is often such that the previous value is available, which gives a close enough start value for each iteration. Therefore, only the forward kinematics is examined further here.

Numerous publications treat the Stewart platform forward kinematics. Several studies, as e.g., [Raghavan, 1993], found that there exist 40 solutions, or assembly modes, for a general Stewart platform, but that some solutions may be complex valued. In [Dietmaier, 1998], a method to change the geometry of a given Stewart platform such that it has 40 real solutions is presented. In [Husty, 1996], it is shown how algebraic manipulations of the forward kinematic problem lead to a 40th order univariate polynomial. In [Innocenti, 2001] an improved method is presented, which is faster and more accurate.

It is difficult to derive general, analytical results regarding the convergence of the forward kinematics presented above. Such studies often rely, at least partially, on numerical simulations. In [Merlet, 2000], the domain of convergence of a parallel robot is examined using Kantorovitch's theorem [Miel, 1981]. This was done for only one specific pose, and additional simulations showed that the convergence domain was much larger than the bound the theorem gave.

The forward kinematics presented above was successfully tested using a trajectory through 200 poses chosen for the calibration experiment in Section 5.4. To get additional insight in the F1-type kinematics, the number and location of assembly modes were studied for a few poses. With none of the previous forward kinematics implementations readily available to obtain all 40 solutions, a new algorithm was implemented by Karl Åström from the Mathematic Imaging Group at Lund University and is described below.

Problem formulation The forward kinematics of the Stewart platform, see Figure 2.6, can be phrased as solving a system of polynomial equations in several unknowns. Assume that the coordinates of the six points on the platform are given with respect to the unknown end-effector coordinate system as $(B_1, B_2, B_3, B_4, B_5, B_6)$, and that the six points attached to the base are known to be $(A_1, A_2, A_3, A_4, A_5, A_6)$, expressed in the base coordinate system. The end-effector plate is connected to the base with six links of known lengths $(L_1, L_2, L_3, L_4, L_5, L_6)$. The forward kinematics are given by the rotation matrix R_T and the translation vector $T = (T_x, T_y, T_z)^T$ such that

$$\|A_i - (T + R_T \cdot B_i)\| = L_i \tag{4.34}$$

This can be converted into a polynomial equation by squaring the equations and by parametrizing the rotation matrix. We chose to parametrize R_T using a quaternion vector (a,b,c,d), i.e.,

$$R_T = \begin{bmatrix} a^2+b^2-c^2-d^2 & 2bc-2ad & 2ac+2bd \\ 2ad+2bc & a^2-b^2+c^2-d^2 & 2cd-2ab \\ 2bd-2ac & 2ab+2cd & a^2-b^2-c^2+d^2 \end{bmatrix}$$

which is a rotation matrix for $a^2+b^2+c^2+d^2=1$. The four elements of a quaternion can be parameterized as $(\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2})\cdot\mathbf{n})$, corresponding to a rotation of angle θ around the unit vector $\mathbf{n}=(n_x,n_y,n_z)^T$. The six squared equations become

$$A_i^T A_i + T^T T + B_i^T B_i - 2A_i^T T + 2B_i^T R_T T - 2A_i^T R_T B_i - L_i^2 = 0 \quad (4.35)$$

Without loss of generality, one may assume that $A_1 = B_1 = 0$. This makes the first equation particularly simple.

$$T^T T - L_1^2 = 0 \quad (4.36)$$

By subtracting this equation from the remaining five equations, the $T^T T$ term can be removed. The resulting system of seven equations is

$$\begin{aligned} a^2+b^2+c^2+d^2-1 &= 0 \\ T^T T - L_1^2 &= 0 \\ K_i - 2A_i^T T + 2B_i^T R_T T - 2A_i^T R_T^T B_i &= 0, \quad i=2,3,4,5,6 \end{aligned} \quad (4.37)$$

with the constant $K_i = A_i^T A_i + B_i^T B_i + L_1^2 - L_i^2$. The equation system depends on the seven unknowns (a,b,c,d,T_x,T_y,T_z).

The actual implementation uses a slightly different choice of variables. However, for better understanding, a notation in line with the kinematic description presented in this thesis was used here.

Solving polynomial systems A solution technique in Matlab was implemented, along with [Stewénius, 2005; Byröd *et al.*, 2009]. The technique is based on forming an expanded set of equations, by multiplying the original seven equations with a number of monomials, i.e., one-term polynomials. Typically, low order monomials up to a certain degree are used. All expanded equations are then expressed as $Cm=0$ with the sparse coefficient matrix C and the monomial vector m. Using numerical linear algebra, it is possible to express higher order monomials in terms of certain low-order basis monomials. In this particular solver, the number

of base monomials are 80. This number corresponds to the 40 solutions of the Stewart platform forward kinematics, where each solution is counted twice, because the two quaternions (a,b,c,d) and $(-a,-b,-c,-d)$ result in the same rotation matrix R_T. The next step is to calculate the so-called action matrix M of the linear mapping $T_{m_0} : p \mapsto pm_0$ for some monomial m_0. The columns of the action matrix are given by the image of the 80 basis monomials $(b_1,\ldots,b_{80})$. The image of a basis monomial is either another basis monomial, i.e., $b_j m_0 = b_i$, or a higher order monomial. If the image is a basis monomial, the j'th column of the action matrix is zero except in row i, i.e., $M_{i,j} = 1$ and $M_{k,j} = 0, \forall k \neq i$. If the image is a higher order monomial, it can be expressed by the basis monomials, i.e., $b_i m_0 = \sum_{k=1}^{80} \alpha_k b_k$. Then the j'th column of M is defined by $M_{kj} = \alpha_k$. The solutions to the original equations can then be calculated from the eigenvectors and eigenvalues of the action matrix M.

Simulations The solver implementation was tested on a set of 50 end-effector poses. For each of the 50 poses, the joint positions corresponding to the nominal configuration were calculated by the inverse kinematics. With the solver, alternative assembly modes were determined by finding all solutions to the forward kinematics for the calculated joint positions. The choice of end-effector poses was performed with the method described in Section 5.4 for chosing poses for kinematic calibration. Figure 4.16 shows the xy-coordinates of the end-effector poses together with the F1 prototype's workspace in a plane orthogonal to the actuators.

The solver does not always return the total number of 80 solutions that the problem theoretically has, see Figure 4.17. For one out of the 50 poses, only 18 solutions were found. For four poses, there were problems with the accuracy of the solutions.

For finding a reason for the numerical problems, besides the forward kinematics solution being complex in general, the influence of the Gantry-Tau's architecture on the numerical properties of the solver has to be examined. The links of a Stewart platform are typically rather evenly distributed along the base plate edge, while the Gantry-Tau's links are clustered into three groups. The Gantry-Tau's end-effector plate is much smaller than the triangle defined by the three carts. This triangle corresponds to the Stewart platform's base plate, which is often only slightly larger than the end-effector plate. However, the focus of this study was not to develop a new forward kinematics solver compared to the one presented in [Innocenti, 2001], but to examine the number and location of assembly modes of the F1-type Gantry-Tau.

A closer examination was done of the four poses for which 16 real solutions were found. As mentioned above, each solution is counted twice, because the same orientation can be represented by two different quater-

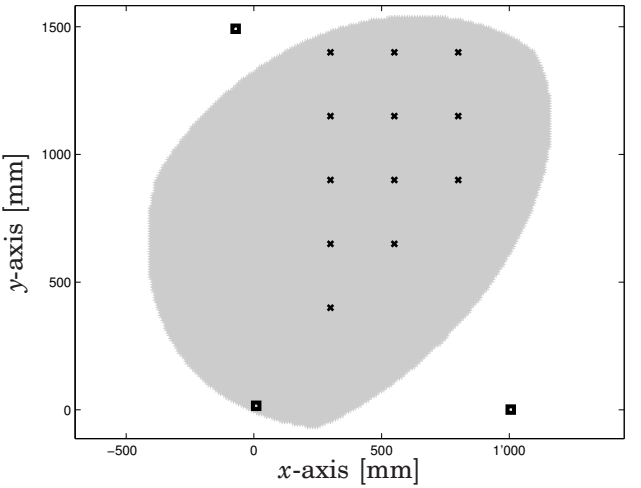


Figure 4.16 Robot poses for assembly mode study: The end-effector positions (black crosses) in the xy-plane orthogonal to the actuators. Each of the marked positions corresponds however to several of the 50 poses, which have equal xy-coordinates but a different z-coordinate and a different orientation. The grey area shows the kinematic workspace of the F1 prototype for the nominal end-effector orientation with $q_4 = q_5 = q_6 = 0$. The ball joint limits of 45° were taken into account, but possible collisions of the end-effector and/or links with the framework were not considered. The black squares indicate the guideway positions.

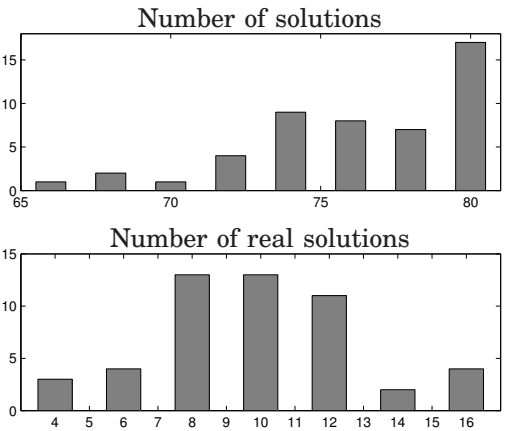


Figure 4.17 Histogram of found number of solutions found for each of the 50 poses: Total number of solutions (upper) and number of real solutions (lower). The theoretical number of 80 solutions was only returned for 16 poses. For one pose, only 18 solutions were returned by the solver, which is not shown in the histogram.

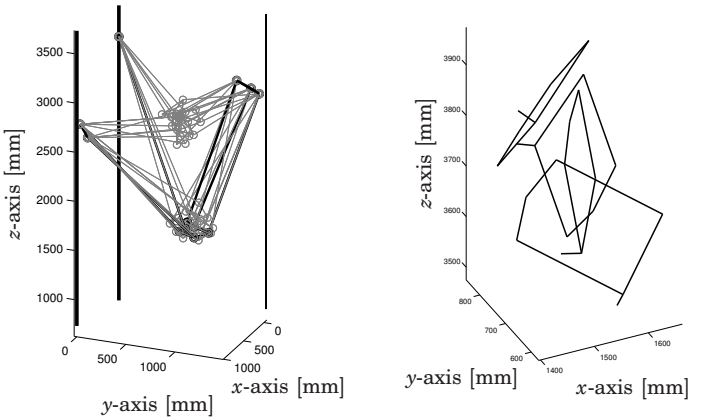


Figure 4.18 Assembly modes for one of the studied nominal poses: All eight assembly modes (left) with the nominal configuration (black). Detail of the lower four assembly modes (right) with only the end-effector plates, one of the corners attached to arm 2 is indicated by a small line to illustrate the orientation.

nions, (a,b,c,d) and $(-a,-b,-c,-d)$. For the studied poses, the eight resulting assembly modes formed two groups, each with four end-effector poses located close to each other. These two groups are related to the two solutions of the 3 DOF forward kinematics. Figure 4.18 illustrates the resulting solutions for one of the poses. Even though the end-effector positions within one group are close to each other, it can be seen in the right of the figure that the orientations within one group differ. Most of these found solutions correspond however to unfeasible link configurations, e.g., with crossed links, which cannot be reached from the original robot configuration without a re-assembly of the links.

Although the low number of poses studied here does not allow conclusions on the complete kinematic workspace, the results give an indication that other solutions of the forward kinematics differ at least in their orientation sufficiently from the desired configuration. With the chosen start value calculated by the implicit 5 DOF kinematics, a convergence to an undesired solution is unlikely. Furthermore, the forward kinematics algorithm implemented was successfully used in real-time control.

4.6 Conclusion

Based on an examination of the nominal model validity and a study of the resulting end-effector accuracy, a kinematic error model of the 3 DOF Gantry-Tau robot was presented. Kinematics for the 6 DOF F1-type Gantry-Tau were developed and algorithms for the solution of inverse and forward kinematics proposed. The number of assembly modes of the F1-type Gantry-Tau was examined.

Gantry-Tau kinematics including the kinematic Jacobian (see Chapter 6) were integrated in the ABB IRC5 controller software in cooperation with ABB Robotics. The same implementation was used for generation of Simulink S-functions and for an external IRC5 kinematics module, which can be activated to replace the internal IRC5 kinematics for development purposes. The F1-type kinematic were integrated in the ISG kernel software within the MONROE project.

For a high accuracy Gantry-Tau robot in industrial applications, the error modeling presented has to be further improved, in particular error models for the extended 5 DOF Gantry-Tau with serial wrist and the F1-type Gantry-Tau must be developed in the future.

5

Kinematic Calibration

This chapter covers several aspects of kinematic calibration. In general, the purpose of kinematic calibration is to increase the kinematic accuracy of the robot, which is affected by manufacturing tolerances. However, with the application of reconfigurable robots in SMEs, new challenges arise. Reconfigurations require repeated kinematic calibrations, affordable by SMEs. For that, the calibration has to be automated, so that non-expert staff can execute it, and rely on low cost measurement equipment affordable by SMEs.

After a summary of related work in Section 5.1, Sections 5.2 to 5.4 focus mainly on the calibration of different kinematic models derived in Chapter 4. Section 5.5 discusses camera vision for kinematic calibration and Section 5.6 presents a method for automated kinematic calibration. Section 5.7 concludes the chapter.

5.1 Related Work

Kinematic calibration of parallel robots has different requirements from serial robot calibration, e.g., in terms of sensor workspace [Merlet, 2000]. Three types of kinematic calibration can be distinguished: External, constrained and self-calibration.

External calibration relies on measurements by a separate device, identifying fully or partially the pose of the end-effector or other robot parts. Most measurement devices can be categorized as either mechanical or optical systems. Mechanical measurement systems are often physically connected to the robot. Examples are the wire-based system presented in [Tavolieri *et al.*, 2002], or a double-ball-bar system [Huang *et al.*, 2006], which is delivering only partial pose information. The alternative is to use optical measurement devices as laser trackers or computer vision including appropriate image processing software. In [Renaud *et al.*, 2006] a vision system for kinematic calibration is proposed and its accuracy is

evaluated to approximately 10 µm for position and 0.0015° for orientation, respectively. This proves adequate computer vision to be competitive to laser trackers regarding accuracy. A common way to perform vision-aided calibration is to mount a target with known pattern geometry on the robot end-effector and to take images of the target with a camera fixed in the robot environment as in [Renaud *et al.*, 2006]. It is as well possible to mount the camera on the robot flange and to take images of a target fix in the environment [Motta *et al.*, 2001]. A method presented in [Renaud *et al.*, 2005] relies on the observation of the robot's leg direction and no robot-mounted target is necessary. The image processing can be performed manually, e.g., using a camera calibration toolbox for Matlab [Bouguet, 2007], or more preferably in an automated way as presented in [Renaud, 2003].

For constrained calibration, the movement of the end-effector or of other robot parts is restricted mechanically as in [Daney, 1999]. A disadvantage of this method is that the mechanical constraint might be difficult to mount or may even damage the robot. The mechanical constraint might as well lack the stiffness to apply the geometric constraint with the accuracy required for a good calibration result.

Self-calibration is suitable for robots that can be equipped with redundant sensors, located in the passive joints or in other robot parts as in [Baron and Angeles, 2000; Merlet, 1993]. By the use of additional sensors, the kinematic model and the calibration problem can be simplified. In [Miermeister and Pott, 2012], a method using force sensors to perform self-calibration of cable-driven PKMs is presented.

Different forms of the kinematic model can be used for calibration. In [Everett, 1989], the objective function to minimize is based on the forward kinematics. For PKMs, the forward kinematics is difficult to solve and might not have a solution for all possible values of the kinematic parameters, obtained while iterating towards the optimal solution. The existence of a solution might as well be a problem for the inverse kinematics, but to a lower degree if at all for the closure equations. In [Renaud *et al.*, 2006] it is concluded that for the H4 PKM, the usage of the the implicit closure equation is slightly preferable to the inverse kinematics.

For a successful calibration, a good sensitivity of the parameters towards the model and chosen data is important [Merlet, 2000]. As the sensitivity is dependent on many aspects, e.g., the robot architecture and actual dimensions, it is difficult to compare calibration results directly. Before calibrating a robot, it is necessary to examine which parameters are identifiable [Renaud *et al.*, 2006]. Observability of geometric parameters may also depend on the calibration method chosen [Merlet, 2000]. In [Besnard and Khalil, 2001] a method is presented for determining the identifiable parameters of a parallel robot. In [Renaud *et al.*, 2003] a crite-

ria for selecting calibration poses with a maximum sensitivity is proposed and compared to several alternative criteria. An important question is also how many measurements need to be taken for an accurate result. In [Bai and Teo, 2002] it was found that the maximum accuracy was reached with ten six-dimensional pose measurements and could not be improved further with more measurements. The considered problem was to determine 12 parameters, and the measurement device's accuracy was 0.1 mm.

The calibration of a modular robot is studied in [Ji and Li, 1999]and it is proposed to calibrate complete, unchangeable modules beforehand, e.g., at the manufacturer, and to determine only the geometric parameters changed by the reconfiguration.

5.2 Introduction and Nominal Kinematic Model

This section discusses the procedure of the Gantry-Tau calibration in general and illustrates it with the calibration of the nominal kinematic model.

The Gantry-Tau kinematics was identified by external calibration, i.e., using measurements of the end-effector pose. The Gantry-Tau prototypes do not have any built-in redundant sensors for self-calibration, and the Gantry-Tau architecture and large workspace make it difficult to immobilize parts of the robot for constrained calibration. Optical measurement devices, laser trackers and camera vision, were used to record the end-effector pose. For the ideal, simplified kinematic model (4.9), 3-D measurements of the end-effector position are sufficient, but for the error model (4.14) and the F1-type kinematics, 6-D measurements of end-effector position and orientation are necessary.

The minimum number of measurements necessary depends on the number of parameters of the kinematic model to be calibrated. Optimizing the number of measurements to obtain a maximum accuracy with as few measurements as possible is an important issue for a fast automated kinematic calibration. The choice of measurement poses has to be optimized with the aid of an appropriate criteria [Renaud *et al.*, 2003]. This work focuses however on other aspects of calibration. Therefore, in most cases, no measurement pose optimization was performed and a large number of measurement poses were chosen as grid corner points distributed evenly in the complete Cartesian workspace of the robot.

Given a set of N joint positions q_k and the corresponding measured end-effector poses $(T_{m,k}, R_{T,m,k})$, the kinematic parameter set s is identified by minimizing a cost function V based on the kinematic closure

equations for each link $j = \{1,\dots 6\}$

$$\min_s V(q_k,(T_{m,k},R_{T,m,k}),s) = \min_s \sum_{j=1}^{6}\sum_{k=1}^{N}(\|A_{j,k}(q_k,s)-B_{j,k}(T_{m,k},R_{T,m,k},s)\|^2-L_j(s)^2)^2 \quad (5.1)$$

For the simplified nominal kinematic model (4.9), the cost function V considers only three links and the end-effector position

$$V=\sum_{j=1}^{3}\sum_{k=1}^{N}\left(\|A_j^s+q_{j,k}\cdot v-T_{m,k}\|^2-L_j^2\right)^2 \quad (5.2)$$

In contrast to the forward or inverse kinematics, the closure equation has for the Gantry-Tau robot the advantage that it always has a real valued solution for any real values of the kinematic parameters. On the other hand, the optimal parameters based on the closure equation might not be exactly the ones that minimize the inverse or forward kinematics error. Although the chosen objective function $V(s)$ is not convex on the domain $\mathbb{R}^n$, with the set s consisting of n parameters, it is assumed that the nominal values as initial values are close enough to the optimum that the algorithm does not iterate towards an undesired local minimum.

The end-effector poses $(T_{m,k},R_{T,m,k})$ are given in the coordinate frame of the measurement device, while the nominal kinematic parameters are expressed in the robot frame. Rather than optimizing the transformations between the different coordinate systems together with the kinematic parameters as suggested in [Renaud *et al.*, 2006], we determine the optimal parameters in the measurement frame. The robot frame can then be found with the aid of the optimal parameters, e.g., the base frame x axis along the identified actuator axes. To obtain a good initial guess, the nominal parameters s_0 are transformed with $H_{\mathrm{rb}}^{\mathrm{mb}}$ and $H_{\mathrm{re}}^{\mathrm{mt}}$. $H_{\mathrm{rb}}^{\mathrm{mb}}$ is the transformation between robot base and measurement device base frame and $H_{\mathrm{re}}^{\mathrm{mt}}$ between the robot end-effector and the frame of the measurement target, which is mounted on the end-effector, such that

$$(T_{m,k},R_{T,m,k}) \approx H_{\mathrm{rb}}^{\mathrm{mb}}\, f_{\mathrm{fk}}(q_k,s_0,c)\, H_{\mathrm{mt}}^{\mathrm{re}} \quad (5.3)$$

For the 3 DOF Gantry-Tau, only 3-D measurements of the end-effector position are necessary and (5.3) simplifies to

$$T_{m,k} \approx H_{\mathrm{rb}}^{\mathrm{mb}}\, f_{\mathrm{fk,\,tr}}(q_k,s_0,c) \quad (5.4)$$

with only the translational part $f_{\text{fk, tr}}$ of the forward kinematics. The offset between end-effector frame and measurement target origin cannot be identified with a constant end-effector orientation. Consequently, the identified parameters A_i^s and calibrated forward kinematics relate to the target origin. With a subsequent wrist or tool-workpiece calibration, this offset can however be identified or compensated for.

For evaluation, the measurements are divided such that each second grid point is used for calibration and the remaining points for evaluation. In that way, both measurement sets are distributed over the complete Cartesian workspace of the robot. The resulting accuracy is evaluated using the forward kinematics:

$$e_k = \| T_{m,k} - f_{\text{fk, tr}}(q_k, s_{\text{opt}}, c) \| \tag{5.5}$$

where e_k is the absolute positioning error and s_{opt} the identified kinematic parameters. For the F1-type kinematics and the kinematic error model, corresponding orientation errors are considered using an angular representation of the end-effector orientation.

In the remainder of this section, kinematic calibration of the 3 DOF part of the L1 prototype is discussed. Measurements were taken with a laser tracker [Leica Geosystems, 2009] and appropriate reflector for 3-D position measurements. Altogether 779 end-effector positions were recorded and despite the exceeding number, the complete set was used for calibration and evaluation.

The kinematic model of the ideal Gantry-Tau has 15 scalar parameters: L_i, A_i^s , $i = \{1,2,3\}$, and the guideway direction v. With v consisting of three scalar parameters, i.e., not being a unit vector, an adjusting factor for a not perfectly known gear box transmission is included. A modified nominal model that allows for nonparallel guideways, similar to the error kinematic model presented in [Williams *et al.*, 2006], is however both more accurate and more practical to calibrate. With each guideway having its own direction v_i, each kinematic chain can be studied and calibrated separately. The modified nominal model has thus 21 parameters: L_i, A_i^s and v_i, $i = \{1,2,3\}$.

Table 5.1 shows the identified kinematic parameters for both kinematic models. For the ideal model, the guideway directions were identified separately and assumed known. These earlier measurements showed that the guideways of the L1 prototype deviate around 0.1° from being parallel to each other. Figure 5.1 shows the absolute positioning error according to (5.5) for the evaluation data. The absolute positioning accuracy is improved considerably by using the modified nominal model. For the ideal model, the positioning errors were mostly smaller than 1.5 mm with two peaks above 2 mm and a mean value of 0.7 mm. The modified nominal

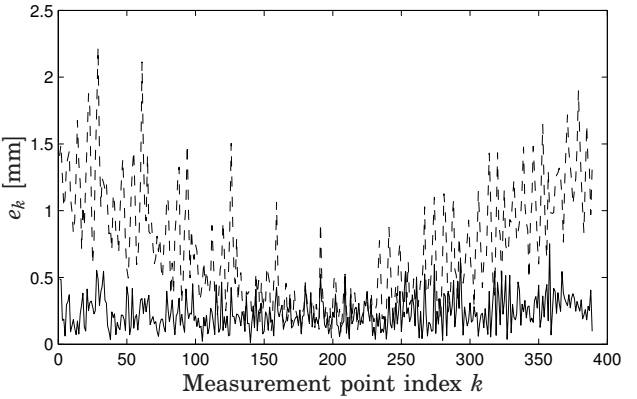


Figure 5.1 Absolute positioning error e_k for ideal kinematic model (dashed) and modified nominal model with nonparallel tracks (solid) for the L1 prototype. The regular peaks for the ideal kinematic model correspond to positions close to the workspace limit.

model gives a mean positioning error 0.23 mm and is mostly below 0.5 mm. For both models, the error along the actuator axes is smaller along the actuator direction than in the plane orthogonal to it. The overall u-shaped form of the curve for the ideal model in Figure 5.1 is due to a better data fit in the center of the workspace, where the imaginary, perfectly parallel guideways intersect with the actual, slightly misaligned guideways. The regular peaks in the curve for the ideal kinematic model correspond to poses close to the workspace boundary, where the angle between links and guideways is largest and an actuator modeling error therefore has largest influence. The measurement poses are arranged in 12 grid layers orthogonal to the guideways' axes, which is the global x-axis, i.e., with an increasing measurement point index, the x coordinate of the end-effector position increases.

The architecture of the Gantry-Tau's framework has a large influence on how difficult it is to achieve accurate, parallel mounting of the guideways. A modular construction of the framework and the possible reconfiguration by non-expert staff makes a misalignment probable. As the geometry of the end-effector plate and the carts match to a parallel arrangement of the actuators, the misalignment might modify the Tau configuration and thus cause the end-effector orientation to vary around the workspace. In this case, the simplified kinematic model considering only three links is not valid any more, and an error kinematic model as presented in Sections 4.4 and 5.3 is required.

Table 5.1 Nominal model calibration results for the L1 prototype, given in measurement frame.

	i	L_i [m]	A_i^s [m]	v_i [–]
Nominal model	1	2.0613	$\begin{pmatrix} -1.8093 \\ 3.5095 \\ 0.0041 \end{pmatrix}$	–
	2	2.0662	$\begin{pmatrix} -1.8688 \\ 2.2455 \\ 0.2075 \end{pmatrix}$	–
	3	2.0632	$\begin{pmatrix} -1.8109 \\ 3.2774 \\ -1.4436 \end{pmatrix}$	–
Modified model	1	2.0608	$\begin{pmatrix} -1.8089 \\ 3.5093 \\ 0.0039 \end{pmatrix}$	$\begin{pmatrix} 0.9994 \\ -0.0293 \\ -0.0022 \end{pmatrix}$
	2	2.0633	$\begin{pmatrix} -1.8656 \\ 2.2456 \\ 0.2074 \end{pmatrix}$	$\begin{pmatrix} 0.9990 \\ -0.0264 \\ -0.0045 \end{pmatrix}$
	3	2.0611	$\begin{pmatrix} -1.8086 \\ 3.2774 \\ -1.4434 \end{pmatrix}$	$\begin{pmatrix} 0.9997 \\ -0.0296 \\ -0.0014 \end{pmatrix}$

5.3 A Prototype-Optimized Error Model

In the previous section, the mean absolute positioning error of the L1 prototype after calibration of the nominal model was shown to be around 200 µm. Measurements on the D1 prototype show that the robot's repeatability is considerably higher. In [Crothers *et al.*, 2010], the omni-directional repeatability of the TCP was shown to be 13 µm in the mean, with a maximum of 50 µm. To benefit to a larger extent from the Gantry-Tau robot's high accuracy, a kinematic error model was developed in Section 4.4. The kinematic error models of the Scara Tau and Queensland Gantry-Tau robots presented in [Cui *et al.*, 2005] and [Williams *et al.*, 2006], respectively, were validated by simulation. This section presents calibration of

the kinematic error model developed in Section 4.4. The actuator model is further developed to improve accuracy. The focus lies on identifying the maximum achievable end-effector positioning accuracy for the D1 prototype in practice. Additionally, a parameter sensitivity analysis is performed.

Measurements

The following measurement sets were recorded using a laser tracker with a Leica T-Mac and a corner cube reflector, respectively [Leica Geosystems, 2009]. With the T-Mac, both position and orientation can be measured, whereas with the corner cube reflector it is only the position.

1. For each of 176 TCP poses lying on a grid filling the robot workspace, the position and orientation of the 3 DOF end-effector were recorded with the T-Mac.
2. Independently from the above measurements, and with a new laser tracker positioning and thus expressed in a different coordinate system, the cart position and orientation were measured for 28 points along the guideways with the T-Mac for each of the three carts. During these measurements, all carts were moved equally much, so that the configuration of the links, i.e., the angles between the link clusters, and consequently the load on the carts did not change throughout the measurements. The TCP was located in the center of the workspace in yz-plane.
3. In addition to the above actuator measurements, and with a third laser tracker positioning, the cart positions were recorded with a corner cube reflector while the TCP was moving to a set of 150 random poses.

Actuator Modeling

To examine the actuator linearity and positioning accuracy, measurement set 2 was evaluated. Figure 5.2 shows the absolute value of the residuals $\|\epsilon_i\|$ when fitting a linear function according to (4.13) between commanded cart positions $q_i(k)$ and k-th measurements $A_{m,i}(k)$ for cart $i = \{1,2,3\}$

$$A_{m,i}(k) = A_i^0 + q_i(k) \cdot v_i + \epsilon_i(k) \tag{5.6}$$

It can be seen that the linearity varies among the carts. The largest deviations are obtained for cart 2 and were between 25 µm and 140 µm. In order to model these variations more accurately, measurements with a higher spatial resolution along the guideway are necessary. The carts were mounted on roller blocks and driven by a rack-and-pinion system.

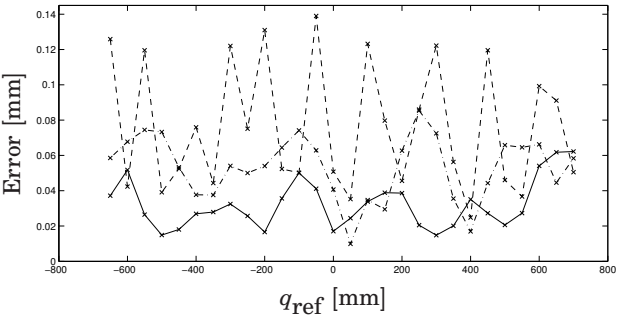


Figure 5.2 Linearity error of the D1 prototype actuators: Absolute value of residuals when fitting a linear function to the movement of cart 1 (solid), cart 2 (dashed) and cart 3 (dashed-dotted). Along the abscissa, the commanded cart position is shown.

According to the manufacturer, variations in cart position are expected with a period corresponding to the the length between the rack teeth, which is much shorter than the resolution of the measurements. Nevertheless, an attempt was made to derive actuator models based on the limited measurement available.

In Figure 5.3, the residuals are decomposed as defined by (4.11) and (4.12) into a part parallel to the guideway direction, i.e., projected on the actuator axis, and a part orthogonal to it, i.e., the distance of the measured point to the modeled guideway, and shown with the respective values for measurement set 3. The angle, i.e., where on a circle around the actuator axis with a given distance the measured point was, is not shown here. The laser tracker position was different for measurement sets 2 and 3 and the exact coordinate transformation between the two sets can be determined except for the angle around the actuator axis.

It can be seen that cart 2, and to a smaller extent cart 3, has a shifting error along the guideway direction. With only 28 measurement points along the guideway it is not possible to identify a spatial high frequency variation of the residuals along the guideway axis. It was impossible to perform further measurements within the scope of this work, since the robot was dismantled after the measurements. Cart 1 is most linear and least sensitive to the small load changes induced by different TCP positions in the yz plane, which are present in measurement set 3 (grey stars). Actuator 2 exhibits the largest errors. The difference between measurements 2 (black line) and 3 (grey stars) is larger for the orthogonal errors than for the errors parallel to the actuator axis.

A piecewise linear model using measurement set 2 was adopted instead of the nominal model (5.6). As the 28 measurements per cart did not cover

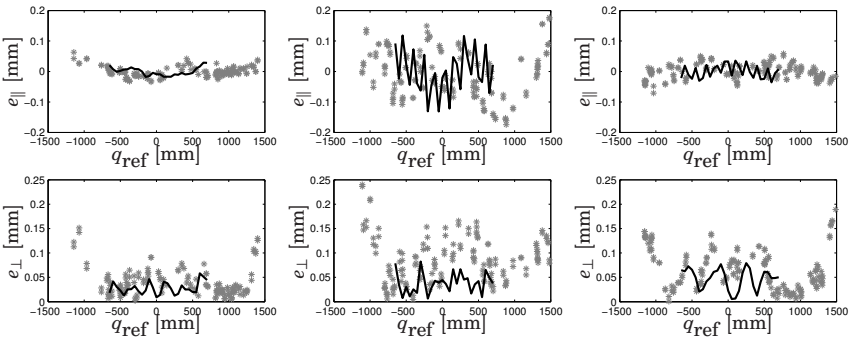


Figure 5.3 Linearity errors of the D1 prototype actuators: Residuals when fitting a linear function to the movement of cart 1 (left), cart 2 (center) and cart 3 (right). The upper row shows the residual vectors projected on the track direction, the lower row the absolute value of the residual component orthogonal to the track direction. The solid lines correspond to a TCP movement parallel to the track directions, while the grey stars represent measurements taken while the TCP was moving randomly through the workspace.

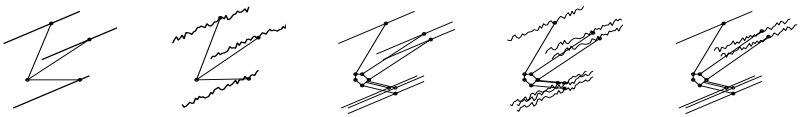
the complete length of the guideway, the model is only valid within the measured range from -650 mm to 700 mm. The cart position A_i for the commanded actuator position q_i is interpolated linearly between the two cart measurements $A_{m,i}(k)$ and $A_{m,i}(k+1)$ whose corresponding actuator positions $q_i(k)$ and $q_i(k+1)$ are closest to q_i

$$ {}^{\mathrm{pw}}A_i\,(q_i) = A_{m,i}(k) \;+\; \frac{q_i - q_i(k)}{q_i(k+1) - q_i(k)} \;\left(A_{m,i}(k+1) - A_{m,i}(k) \right) \qquad (5.7) $$

In (5.7) ${}^{\mathrm{pw}}A_i$ is, unlike A_i in (5.6), expressed in the coordinate system used for measurement set 2. Instead of optimizing the actuator direction v_i and offset ${}^{s}A_i^0$ as for the nominal model, the coordinate frame transformation ${}^{\mathrm{pw}}H_i$ between the guideway i and the TCP measurement frame has to be calibrated. The reflector was for practical reasons not mounted at the spherical joint center, but provided that the cart's orientation is constant along the guideway, the translation between joint center and reflector is constant in the global frame and is included in the coordinate frame transformation mentioned above. This was confirmed by measurements: The orientation of the carts when moving along the guideways for data set 2 did change at most 0.03°.

Table 5.2 Cost functions for the kinematic models 1 to 4 sketched in Figure 5.4.

Model	Cost function V_i for kinematic chain i
1	$\sum_{k=1}^{N}(\parallel T_{m,k} - ({}^{s}A_i^0 + q_{i,k} \cdot v_i) \parallel_2^2 - L_i^2)^2$
2	$\sum_{k=1}^{N}(\parallel T_{m,k} - {}^{\mathrm{pw}}H_i^s \cdot {}^{\mathrm{pw}}A_i(q_{i,k}) \parallel_2^2 - L_i^2)^2$
3	$\sum_{k=1}^{N}(\parallel (T_{m,k} + R_{T,m,k} \cdot B_i^0) - (A_i^0 + q_{i,k} \cdot v_i) \parallel_2^2 - L_i^2)^2$
4	$\sum_{k=1}^{N}(\parallel (T_{m,k} + R_{T,m,k} \cdot B_i^0) - {}^{\mathrm{pw}}H_i \cdot {}^{\mathrm{pw}}A_i(q_{i,k}) \parallel_2^2 - L_i^2)^2$



(a) Model 1 (b) Model 2 (c) Model 3 (d) Model 4 (e) Model 5

Figure 5.4 Illustration of the kinematic models compared.

Calibration Results

Different combinations of actuator and link cluster models are evaluated in the following. Figure 5.4 illustrates the different models. Model 1 is the simplified, nominal kinematics with linear actuators and only three links. Model 2 includes the piecewise linear actuator models, combined with the simplified three-link structure. Model 3 assumes linear actuators and a six-link structure, where one distinct linear path (5.7) is considered for each of the spherical joints connected to the six links. Model 4 combines the six-link structure model with the piecewise linear actuator model, even here with one distinct path per link. To obtain the highest possible accuracy, a fifth model was introduced which uses for each arm the most accurate model: Linear models for actuators 1 and 3 and a piecewise linear model for actuator 2.

The cost function for the kinematic calibration was based on the closure equation (4.9) for the simplified link model and (4.14) for the six-link model, respectively, with (5.6) or (5.7) for the actuator model. Table 5.2 specifies the cost functions for models 1 to 4, and gives thus as well an illustration of the kinematic models and parameters used. Note that the coordinate transformation ${}^{\mathrm{pw}}H_i^s$ includes in contrast to ${}^{\mathrm{pw}}H_i$ the ball joint position B_i^0 on the end-effector, similarly to ${}^{s}A_i^0$ and A_i^0. Each link was optimized individually.

The measurements available for calibration and validation consist of measurement set 1 with 176 TCP poses. Removing the robot poses whose

Table 5.3 Absolute TCP positioning error of the D1 prototype.

Model	1	2	3	4	5
Mean [µm]	140	120	110	100	90
Max [µm]	410	440	260	340	240

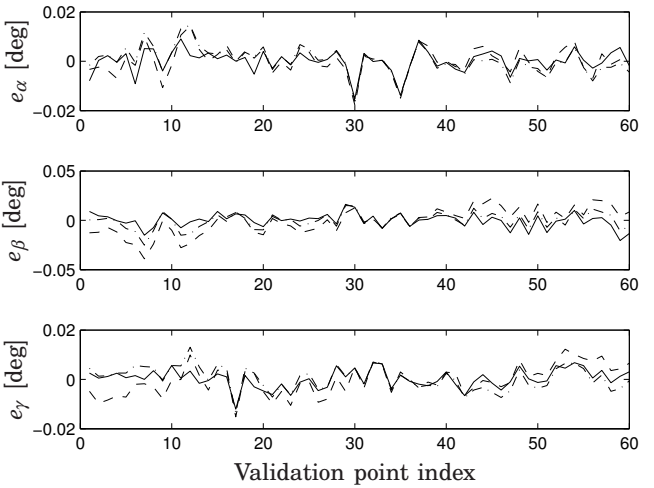


Figure 5.5 Modeling errors of end-effector orientation for the D1 prototype given in ZYZ Euler angles (α, β, γ) for model 3 (solid), model 4 (dashed) and model 5 (dashed-dotted). The remaining errors are roughly 10% of the end-effector orientation variations shown in Figure 4.7.

joint positions exceed the range of the piecewise linear actuator models and reserving half of the measurements for validation, 61 measurement poses were available for calibration.

Table 5.3 shows the absolute TCP positioning error of models 1 to 5. The mean positioning error was decreased from 140 µm (model 1) to 90 µm (model 5) and the highest peak decreased from 410 µm (model 1) and 440 µm (model 2) to 240 µm (model 5).

Tables 5.8 and 5.9 at the end of this chapter show the calibrated parameters for models 1 to 4. Model 5 is a combination of model 3 (kinematic chains 1 and 3) and model 4 (kinematic chain 2). For models 1 and 3, the actuator offsets ${}^sA_i$ and directions v_i are given, while for models 2 and 4, the coordinate frame transformations between the track measurements from set 2 and the TCP measurements from set 1 are given.

Figure 5.5 shows the modeling errors of the end-effector orientation

for models 3 to 5. All six-link models with capture the varying orientation with only small errors. This is in accordance with the assumptions that the orientation errors are caused by kinematic errors in the links, end-effector or ball joints, and not because of variations of the orientation of the carts as explained in Section 4.2.

Parameter Sensitivity Analysis

The sensitivity of the end-effector positioning and orientation error on the kinematic parameters was examined. Based on the optimized parameters, each of the 60 parameters of model 5 was changed with $\pm 5 \times 10^{-5}$. For each parameter, the changes in end-effector positioning ($S_{j,\text{pos}}$), orientation ($S_{j,\text{rot}}$) and of the cost function ($S_{j,\text{cost}}$) are accumulated for all validation points and shown in Table 5.10 at the end of this chapter

$$S_{j,\text{pos}} = \sum_{j=1}^{N} (\| T_j - T_j^+ \|_2 + \| T_j - T_j^- \|) \tag{5.8}$$

where T_j is the end-effector position for the identified model and for the validation point j and T_j^+ and T_j^- the corresponding positions for the modified models. Similarly, the orientation changes, expressed in ZYZ Euler angles, are accumulated in $S_{j,\text{rot}}$ and the cost function contribution of each measurement point in $S_{j,\text{cost}}$.

Besides Table 5.10, Figures 5.6 and 5.7 show the variations in the cost function and the end-effector pose per validation pose, divided into positive and negative changes, for selected parameters. Within the given range of [-650 mm, 700 mm] for the actuators, the modified actuator directions v_i, $i = \{1,2,3\}$, result in the smallest positioning changes. The rotation matrix of the transformation ${}^{\text{pw}}H_2$ between the two different laser tracker positions gives larger variations, as the distance between laser tracker and the cart position magnifies the modification by 5×10^{-5}. For both the joint offsets on the carts and on the end-effector plate, the cost function is much more sensitive to the x-component than to the y- and z-components. The link lengths L_i, see Figure 5.6 for L_1, exhibit the largest sensitivity.

Discussion

Considering Figure 5.2, it appears reasonable that actuator 1 gains the least by a piecewise linear instead of a linear actuator modeling, while improved results may be expected for actuators 2 and 3, where actuator 2 is the least linear with a large error along the actuator direction. On the other hand, only few actuator measurements were available for the modeling, and the actuator model for joint positions between these measurements is not definitely known. The ball joint position may vary

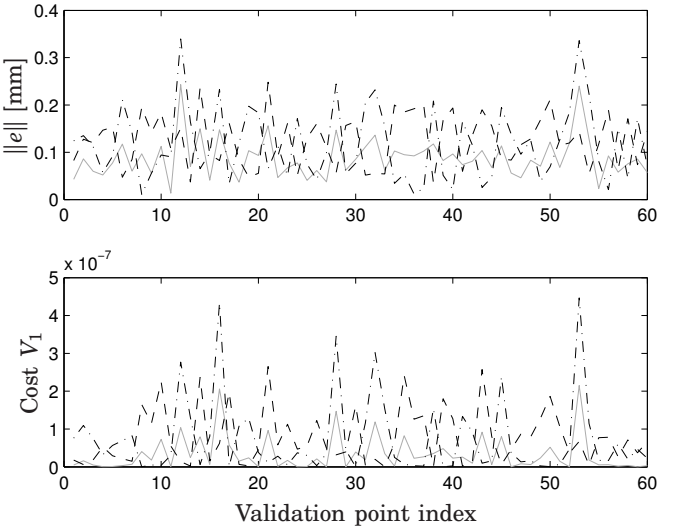


Figure 5.6 Sensitivity with respect to L_1: Changed cost (lower) and absolute positioning error $\|e\|$ (upper) for increasing (dashed) and decreasing (dashed-dotted) L_1 with 50 µm. The original cost is shown in grey; the maximum positioning error peak can be found in the cost function for arm 1, but not in the other arms' cost functions.

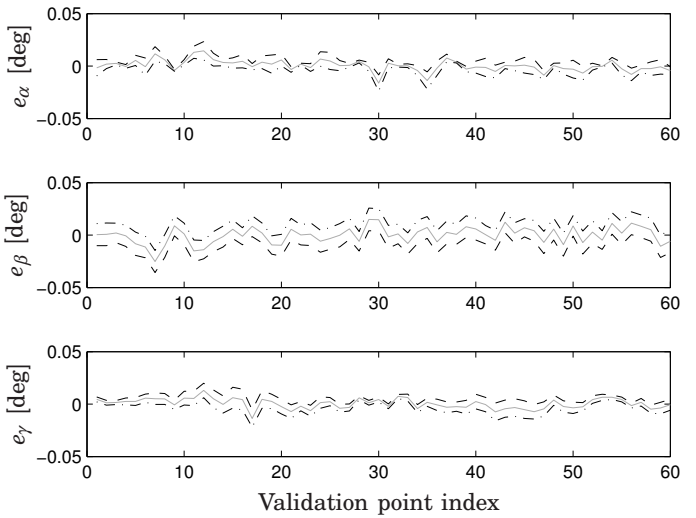


Figure 5.7 Changed orientation errors for increasing (dashed) and decreasing (dashed-dotted) $B_{2b,x}$ about 50 µm; original angular error of Model 5 shown in grey.

considerable from the model, in particular if the actuator has large errors in general. From this point of view, it appears rather unexpected that arm 2 is modeled more accurately with the piecewise actuator model.

For actuator 2, the measured position for one commanded cart position varied significantly for different link angles , i.e., if the other two carts were in different positions. This may be due to lacking stiffness, which causes the cart to move with a changed load from the links and makes it difficult to model the actuator as presented. Since the structural stiffness was not modeled, this error source could not be compensated for.

A comparison with Tables 5.8 and 5.9 shows as expected that a piecewise linear modeling of cart 1 does not decrease the cost function, but on the contrary increases it somewhat. The results for cart 2 in Figure 5.2 indicate that a piecewise linear actuator model would not catch the cart position's dependency on the TCP's *yz*-position. Nevertheless, the cost function could be decreased by using a piecewise linear actuator model. In model 4, link 2*b* has a slightly lower cost function than link 2*a*. This can be explained by the fact that the T-Mac was mounted closer to the link 2*b* ball joint during the measurements, which model thus the link 2*b* ball joint movement better than that of the link 2*a* joint on the same cart. The cost function of cart 3 on the other hand decreases when including a piecewise linear actuator model. While the cost function for link 3*c* in cart 3, which was the one nearest to the T-Mac during measurements, could be reduced somewhat from model 3 with linear actuators to model 4, the cost functions for links 3*a* and 3*b* increased to 2.6 and 4.2 times, respectively, their values for model 3.

Even though the resulting accuracy is still approximately ten times the D1 robot's repeatability and may be further improved, these results show the potential of an improved actuator modeling, where attention to the exact position of all spherical joints on a cart, the dependency on the movement and position of the other carts and stiffness should be paid. Using actuator measurements of a higher spatial resolution, a more complex model may improve the accuracy. For the cyclic behaviour found for actuators 2 and 3, a sinusoidal model might be suitable. However, with the few measurements available, more complex models resulted in a lower accuracy.

A comparison of the arms shows that the cost of arm 3 is in general about a factor ten lower than that of arms 1 and 2, among which arm 1 tends to perform worse than arm 2. An explanation for this may be, together with the actuator performance from Figure 5.2, that in general, the more links an arm consists of, the stiffer it is.

The kinematic error model of the arm structure gave better results than the nominal model in all cases except for cart 3.

The models including all six links (models 3 to 5) manage to capture

the varying end-effector orientation. Overall, the modeling is better for the end-effector orientation and gives a higher error reduction for the end-effector orientation than for the end-effector positioning. Another point worth to note is that the angular errors of the nominal kinematics, see Figure 4.7, exhibit a repeating pattern, which may be caused by errors in the link structure modeling, whereas the positioning errors do not exhibit any pattern. This may indicate that the remaining positioning errors can be decreased further by a better actuator modeling. The peak positioning error increases for the models with a piecewise linear actuator modeling, which may indicate that the corresponding pose, which is at the edge of the robot workspace, is particularly sensitive to the actuator modeling.

The parameter sensitivity analysis shows that more attention needs to be paid to the excitation of the different parameters by the chosen measurement points. In particular, the TCP measurement points should cover a larger range in x-direction to give a better excitation for the track directions v_i.

Conclusion and future work

A kinematic error model of the D1 Gantry-Tau robot was calibrated. The main purpose was to evaluate the maximum possible TCP positioning accuracy which varied from 140 µm to 90 µm between the different models. To increase accuracy, a piecewise linear model of the prismatic actuators was developed. Different combinations of nominal kinematics and error kinematic model parts were evaluated by their calibration results.

With stiffness modeling taking into account the gravity-induced axial forces in the links and the elasticities in links and joints, and with additional measurements on the actuators, with a higher positional resolution, the positioning error can be reduced even further in the future to fully benefit from the Gantry-Tau's high accuracy. For accurate movements at high speed, dynamic modeling of the Gantry-Tau will be developed further.

5.4 Calibration of the F1-type Gantry-Tau

Kinematic calibration of the F1-type Gantry-Tau is considerably more challenging than the calibration of the basic 3 DOF Gantry-Tau robot. For the 3 DOF Gantry-Tau, the measurement poses were distributed on a regular grid throughout the workspace. For the 6 DOF F1 Gantry-Tau, practical reasons make such a simple strategy difficult, because an exceedingly large data set would be necessary to cover the workspace with a reasonable small grid interval. The F1-type kinematics with their parallel combinations of serial kinematic chains are more complex, both to model and to calibrate.

Besides the additional rotational joints, the differences to the L1 and D1 prototypes are that the guideways are equipped with double motors and are mounted vertically instead of horizontally. This section examines how these differences influence the kinematic accuracy. The calibration measurements were performed with aluminum links mounted instead of the original carbon fibre links, which were still in production at the time of the measurements.

Pose Selection and Measurements

For the 6 DOF F1 prototype, an elaborate choice of measurement points is required. The measurement poses for the 3 DOF Gantry-Tau were distributed evenly in the complete workspace along the corners of a regular grid. This possibly suboptimal choice of poses was compensated for by a relatively high number of measurements, as an optimization of the number of measurements was not the focus of the work. A similar approach for 6 DOF with a reasonable coverage of the workspace would however result in an impracticably large amount of measurements.

From a grid of 6 DOF poses filling the F1 prototype's workspace, the 200 best calibration poses were chosen. With a grid length of 25 cm and 10°, respectively, 1347 grid poses were found to lie within the kinematic workspace. From this set, 200 poses were then chosen using the condition number based criteria

$$C_k = \mathrm{cond}(\mathbf{J}_{s,k}), \tag{5.9}$$

where $\mathbf{J}_s$ is the Jacobian matrix, i.e., the partial derivative of the cost function with respect to the kinematic parameters, and $\mathbf{J}_{s,k}$ the Jacobian matrix without the row corresponding to measurement pose k. In [Williams *et al.*, 2006], the criteria for the Queensland Gantry-Tau calibration was based on the lowest singular value instead of the condition number. The poses with the largest C_k are chosen for calibration. In [Renaud *et al.*, 2003] several pose selection criteria are compared and it is concluded that the criteria chosen here does not reduce measurement noise influence optimally. However, this is not expected to affect the calibration results noticeably as a laser tracker was used for very accurate measurements. Due to the large workspace of the Gantry-Tau and the computational cost of the iterative inverse kinematics, the distances between the grid points are rather large, resulting in only four to six values per dimension. However, the aim of the measurement pose selection was more to make a sensible choice than to optimize the location of a minimal set of poses.

The end-effector poses were measured in 6 DOF with a laser tracker and a T-Mac target [Leica Geosystems, 2009]. Of the 200 selected poses, 185 were inside the laser tracker's measurement range.

Parameters and Optimization

As for the 3 DOF Gantry-Tau, each kinematic chain can be calibrated separately. Unlike for the D1 error kinematic model in Section 5.3, here, the links of one arm are calibrated together for a minimum number of parameters. For the D1, each link within one arm had its own guideway parameters expressed by (A_i^0, v_i) or ${}^{\mathrm{pw}}H_i$ with $i = \{1, 2a, 2b, 3a, 3b, 3c\}$ to achieve a higher accuracy.

For the linear actuators, the gear ratio can be adjusted with the length of the identified direction vector v_i. The rotational joints are driven by ball screws. The nonlinear relation between the actuator position in [mm] and the joint angle in radians is assumed to be perfectly known.

Orientations are expressed by quaternions to avoid the singularities in angular representations. The inconvenience of using quaternions for optimization is having four instead of a minimal set of three parameters. Obtaining a unit quaternion can be assured by adding a corresponding nonlinear constraint to the optimization problem. In [Schmidt and Niemann, 2001] other possibilities of treating quaternions in an optimization are discussed. A method is presented that parameterizes a quaternion with only three scalar parameters as the deviation on the unit sphere in $\mathbb{R}^4$ from an initial guess of the quaternion. However, for the F1 calibration, the best results were obtained for a free optimization of the quaternion, i.e., with optimizing all four parameters without additional constraint.

The three kinematic chains have due to the varying number of actuators a different number of parameters, and the calibration has a different level of complexity. An analysis of the parameter identifiability as in [Besnard and Khalil, 2001; Khalil *et al.*, 2000] using the *QR* factorization of the Jacobian matrix of the cost function was carried out. The parameters used are described in Section 4.5 with illustrations in Figures 4.13 and 4.14. Figure 5.8 illustrates the base and end-effector frames of the F1 prototype.

Arm 1 has ten scalar parameters, similar to the kinematic error model: The link length L_1, the guideway base point A_1^0, the ball joint position on the end-effector plate B_1^0 and the actuator axis v_1. All ten parameters can be identified independently with the chosen measurement poses. The cost function for kinematic chain 1 is

$$V_1 = \sum_{k=1}^{N} \left(\| \left(T_{m,k} + R_{T,m,k} \cdot B_1^0\right) - \left(A_1^0 + q_{1,k} \cdot v_1\right) \|^2 - L_1^2 \right)^2 \qquad (5.10)$$

The arm 2 kinematics is described in Section 4.5 by altogether 24 scalar parameters: The guideway base point A_2^0 and direction v_2, the orientation ${}^1u_2$ of the rotational joint q_4 with respect to the base frame, the ball

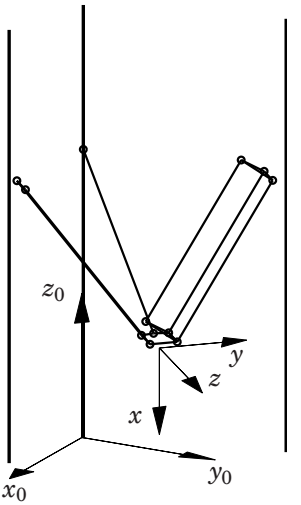


Figure 5.8 Robot base frame and end-effector frame of the F1 Gantry-Tau.

joint positions A_i^{bj} on cart 2, the link lengths L_i and the ball joint positions B_i^0 on the end-effector plate, $i = \{2a, 2b\}$. The orientation ${}^1u_2$ is parameterized as a quaternion and describes the orientation of the coordinate frame S_2, which defines the q_4 rotation by its y-axis. The cost function is

$$V_2 = \sum_{\substack{i \in \\ \{2a,2b\}}} \sum_{k=1}^{N} \Big(\| (T_{m,k} + R_{T,m,k} \cdot B_i^0) - (A_2^0 + q_{2,k} \cdot v_2 + R_{1_{u_2}} \cdot R_{q_4} \cdot A_i^{\text{bj}}) \|^2 - L_i^2 \Big)^2 \quad (5.11)$$

where $R_{1_{u_2}}$ is the rotation matrix associated with the quaternion ${}^1u_2$ and R_{q_4} the rotation matrix associated to the rotational joint q_4. An identifiability analysis showed that the y- and z-components of A_{2b}^{bj} cannot be determined uniquely. The reason for this is that the origin position of S_2 along the q_4 axis and its orientation around the same axis are not uniquely defined. The non-identifiable parameters and the parameters they depend on are shown in Table 5.4. The column of the cost Jacobian matrix corresponding to the redundant parameter is a linear combination of the columns corresponding to the dependent parameters given in the table. Or, a modified redundant parameter can be compensated for with a

Table 5.4 Non-identifiable parameters and dependencies for the nominal parameter values. The components of positions and quaternions are indicated in parentheses, e.g., (x) for the x-coordinate or (1) for the first element of a quaternion.

Redundant parameter	Dependent parameters
$A_{2b}^{\mathrm{bj}}(y)$	$A_2^0(x)$, $A_2^0(y)$, $A_{2a}^{\mathrm{bj}}(y)$
$A_{2b}^{\mathrm{bj}}(z)$	$A_{2a}^{\mathrm{bj}}(x)$, $A_{2a}^{\mathrm{bj}}(z)$, $A_{2b}^{\mathrm{bj}}(x)$, ${}^1u_2(2)$, ${}^1u_2(3)$
$r_3(y)$	$A_3^0(x)$, $A_3^0(y)$
${}^2u_3(2)$	$r_3(x)$, ${}^1u_3(2)$, ${}^1u_3(3)$, ${}^2u_3(1)$
$A_{3c}^{\mathrm{bj}}(x)$	$r_3(x)$, $A_{3a}^{\mathrm{bj}}(x)$, $A_{3b}^{\mathrm{bj}}(x)$
$A_{3c}^{\mathrm{bj}}(z)$	$A_{3a}^{\mathrm{bj}}(z)$, $A_{3b}^{\mathrm{bj}}(z)$, $A_{3c}^{\mathrm{bj}}(y)$, ${}^2u_3(3)$, ${}^2u_3(4)$

suitable change in the dependent parameters, resulting in the same robot geometry.

Arm 3 is the most complex one, with kinematics described by 38 scalar parameters. In comparison to arm 2, it has additional parameters corresponding to the third link and parameters describing the geometry of the q_6 axis: Offset r_3 and orientation ${}^2u_3$ of the coordinate frame ${}^2S_3$ with respect to frame ${}^1S_3$, see Figure 4.14. The identifiability analysis showed that, like for arm 2, the position the coordinate systems attached to joints q_5 and q_6 along the joint rotation axis is not uniquely determined, as well as the orientation around the same axis. The redundant parameters and the dependencies to the remaining parameters can be found in Table 5.4. The cost function for arm 3 is

$$V_3 = \sum_{\substack{i \in \\ \{3a,3b,3c\}}} \sum_{k=1}^{N} \Big(\| (T_{m,k} + R_{T,m,k} \cdot B_i^0) - (A_3^0 + q_{3,k} \cdot v_3 + R_{1_{u_3}} \cdot R_{q_5} \cdot (r_3 + R_{2_{u_3}} \cdot R_{q_6} \cdot A_i^{\mathrm{bj}})) \|^2 - L_i^2 \Big)^2 \quad (5.12)$$

The optimization was carried out with standard Matlab algorithms. For a faster convergence, the analytic gradients were provided and a manual normalization of the variables implemented. To avoid the parameter redundancy, $A_{2b}^{\mathrm{bj}}(y)$, $r_3(y)$, $r_3(z)$, $A_{3c}^{\mathrm{bj}}(x)$ and $A_{3c}^{\mathrm{bj}}(z)$ were fixed to their nominal values and $A_{2b}^{\mathrm{bj}}(z)$ was set equal to $A_{2a}^{\mathrm{bj}}(z)$.

Results

Table 5.5 and Figures 5.9 and 5.10 show the identified parameters and end-effector positioning error of the optimized model. In Table 5.5, for each link j, the mean absolute value of the length error over all N measurement poses k is given as

$$e_{L,j} = \frac{1}{N} \sum_{k=1}^{N} \left| \|A_{j,k} - B_{j,k}\| - L_j \right| \tag{5.13}$$

Arm 1 has the lowest value of 33 µm, while chain 3 has the largest errors of up to 100 µm, especially for link 3c, which has its attachment point furthest away from the q_6 axis.

The resulting mean absolute end-effector positioning error is 240 µm, see Figure 5.9. This is considerably lower than the corresponding mean value of 62 mm for the nominal parameters, based however on the manual, low accuracy measurements of the linear guideway positions. The angular error was reduced from maximal values of 6.7° to maximal values of 0.2°.

A principal component analysis shows in which direction the end-effector positioning error is largest. For that, the values were transformed from the laser tracker frame to the robot coordinate system shown in Figure 5.8. The error covariance is calculated from the matrix E, whose k-th row is the three-dimensional positioning error $\|T_{m,k} - f_{\mathrm{fk}}(q_k, s_{\mathrm{opt}}, c)\|$ for measurement k. The eigenvalues of the resulting covariance matrix $E^T E$ show then how large the error is in the directions given by the eigenvectors of the matrix. For the optimized model, the matrix D containing the eigenvalues and the matrix W with the eigenvectors are

$$D = \begin{bmatrix} 2.34 & 0 & 0 \\ 0 & 4.17 & 0 \\ 0 & 0 & 7.52 \end{bmatrix}, \quad W = \begin{bmatrix} 0.27 & 0.72 & 0.74 \\ -0.36 & -0.62 & 0.69 \\ 0.90 & -0.44 & 0.07 \end{bmatrix}$$

The results show that the error is lowest roughly in the direction parallel to the linear actuators, which is along the z-axis for the F1 robot's base frame. The error is largest in the direction $(0.74, 0.69, 0.07)^T$.

Discussion

The end-effector positioning accuracy was reduced significantly from a mean value of 62 mm to 240 µm for the position and from 6.7° to 0.2° for the maximal ZYZ Euler angle error. Even though the 3 DOF D1 prototype has a better accuracy with a positioning error of 140 µm for the nominal model, the F1 prototype has a more complex kinematic structure, which can increase the number of possible geometric errors, and thus decrease

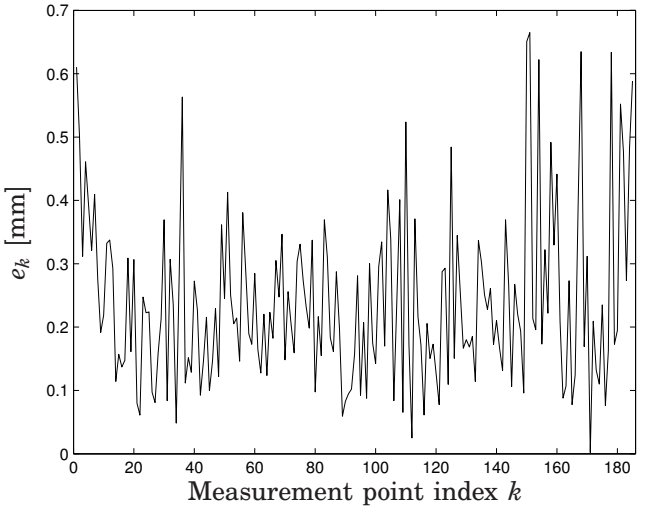


Figure 5.9 Absolute positioning error e_k for identified F1 model. The corresponding mean value for the nominal parameters was 62 mm.

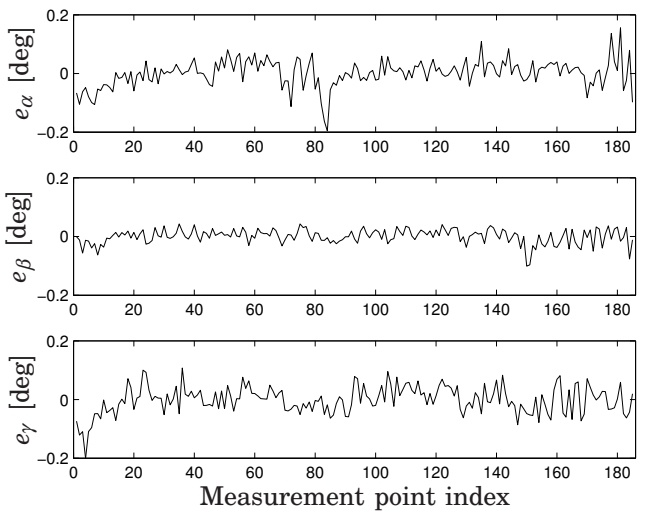


Figure 5.10 Angular errors $(e_\alpha, e_\beta, e_\gamma)$ for the identified F1 model, represented in ZYZ Euler angles. The corresponding values for the nominal parameters reached up to 6.7°.

Table 5.5 Identified parameters of the F1 prototype calibration. All lengths and position values are given in [mm], where the base frame was the laser tracker frame. Parameters in grey were fixed during the optimization to avoid redundancy.

i	1	2a	2b	3a	3b	3c
L_i	1922.58	1501.86	1504.27	1501.99	1501.19	1505.24
A_i^0	$\begin{pmatrix}-235.72\\3298.90\\1218.94\end{pmatrix}$	$\begin{pmatrix}-974.38\\2751.12\\1314.56\end{pmatrix}$			$\begin{pmatrix}719.76\\2238.66\\1315.63\end{pmatrix}$	
v_i	$\begin{pmatrix}0.00\\0.00\\0.99\end{pmatrix}$	$\begin{pmatrix}0.00\\0.00\\0.99\end{pmatrix}$			$\begin{pmatrix}0.00\\0.00\\0.99\end{pmatrix}$	
${}^1u_i$	–	$\begin{pmatrix}0.6308\\0.0054\\0.0054\\0.7467\end{pmatrix}$			$\begin{pmatrix}0.7056\\0.0010\\0.0036\\0.6715\end{pmatrix}$	
r_i	–	–			$\begin{pmatrix}0.18\\121.90\\-70.00\end{pmatrix}$	
${}^2u_i$	–	–			$\begin{pmatrix}0.0007\\0.0015\\0.7420\\0.6746\end{pmatrix}$	
A_i^{bj}	–	$\begin{pmatrix}-106.85\\86.33\\-86.32\end{pmatrix}$	$\begin{pmatrix}100.92\\82.52\\-86.32\end{pmatrix}$	$\begin{pmatrix}-103.54\\-4.04\\0.02\end{pmatrix}$	$\begin{pmatrix}99.86\\-4.19\\-0.05\end{pmatrix}$	$\begin{pmatrix}0.00\\55.19\\187.36\end{pmatrix}$
B_i^0	$\begin{pmatrix}-134.75\\-0.48\\-309.28\end{pmatrix}$	$\begin{pmatrix}-133.33\\-89.69\\-298.89\end{pmatrix}$	$\begin{pmatrix}-135.79\\-104.26\\-101.43\end{pmatrix}$	$\begin{pmatrix}-142.70\\104.64\\-285.28\end{pmatrix}$	$\begin{pmatrix}-140.53\\92.76\\-87.75\end{pmatrix}$	$\begin{pmatrix}-184.92\\-86.25\\-193.93\end{pmatrix}$
$e_{L,i}$	0.033	0.055	0.053	0.070	0.077	0.100

the accuracy. The link length errors in Table 5.5 show that the more complex the kinematic chain was, the larger the link length error was. The values within one link cluster are similar except for link 3*c*. As the link had a larger distance to the q_6 rotation axis compared to links 3*a* and 3*b*, a recalibration of the ball screw geometry was attempted, which did however not improve the results.

Another measurement series recorded on the same day, before the measurements used for calibration in this section, indicate a larger end-effector repeatability error at that occasion than the 1 µm to 5 µm which were succesfully demonstrated at the Automatica Fair 2012 in Munich.

A factor that might have had a negative influence on repeatability and accuracy was the usage of aluminum links instead of the much stiffer, original carbon fibre links. However, as the F1 prototype has a vertical direction of the linear actuators and the weight of the end-effector plate

and T-Mac target was small, the influence was probably minor. The vertical arrangement of the guideways and the resulting advantage that the gravity force is along the stiffest direction of the robot, were not improving the robot accuracy compared to the prototypes with horizontally arranged linear actuators.

The end-effector positioning error is smaller along the direction of the double-motor controlled linear actuators. However, as the principal component analysis shows, only by a factor of 2–3. A kinematic error model of arms 2 and 3 has to be developed in order to take advantage of the accuracy of the linear actuators and increase the positioning accuracy of the robot in general.

As no additional constraints on the quaternion parameters was introduced, the resulting parameters do not represent an orientation in the strict sense. The norm of the three quaternion parameters identified is between 0.9741 and 1.0028. The calibration algorithm should be improved to handle unit quaternions efficiently.

5.5 Calibration Using Camera Vision

Camera vision has advantages over other measurement devices used in kinematic calibration, in particular in view of automated calibration executed by non-experts in SMEs. Mechanical measurement systems like the wire-based system presented in [Tavolieri *et al.*, 2002] might be difficult to setup and handle. Moreover, as the device needs to be physically connected to the robot, it is difficult to exploit it further for industrial applications and needs thus to be stored away between calibrations. Laser trackers are highly accurate, but very expensive. A solution is therefore to use computer vision including appropriate image processing software.

Vision is being used in robotics for various applications, e.g., as feedback in control or for kinematic calibration. In previous work kinematic calibration of robots using vision has been performed by determining the location of a pattern plate [Renaud *et al.*, 2006] mounted on the robot or by observing the robot's legs [Renaud *et al.*, 2004; Andreff and Dressler, 2008]. In most cases, a single camera is used. Stereo or multi-camera vision has the potential to improve the measurement accuracy, as the depth information is increased.

This section evaluates the use of camera vision for kinematic calibration, in particular for automated calibration of the Gantry-Tau robot. A method for automated robot pose reconstruction from image data of two cameras is presented as well as an evaluation of the method's accuracy compared to laser tracker measurements, including a comparison of the accuracy of the calibrated model.

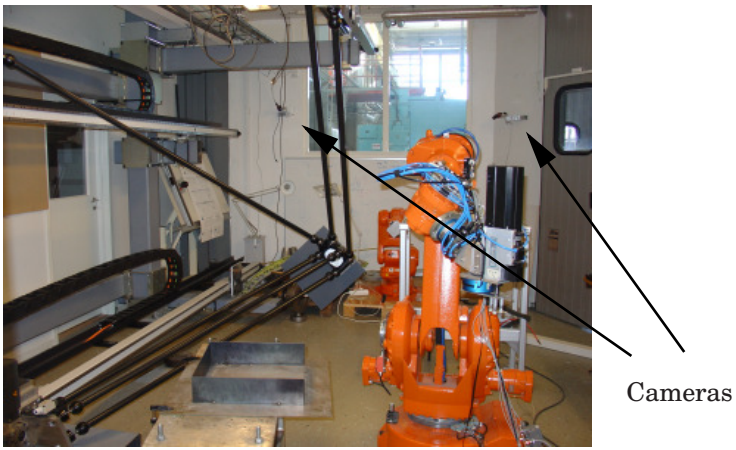


Figure 5.11 L1 prototype with camera setup for stereo vision.

Automated Image Processing Algorithm

Below, the automated pattern detection algorithm used for kinematic calibration in this work and developed by Carl Olsson and Karl Åström at the Mathematic Imaging Group, Lund University, is presented.

Figure 5.11 shows the experimental setup with the L1 prototype. Two synchronized firewire cameras are mounted on the wall in front of the robot end-effector. A special tool with a calibration pattern consisting of five black squares is mounted on the end-effector. Changing the distance and angle between the cameras, the measurement accuracy and range can be adjusted.

The algorithm consists of three steps: First, the calibration pattern is localized in the images. Next, the relative orientation between camera frame and calibration pattern is determined, so that the 3-D calibration pattern position can be estimated in the third step. The version of the algorithm which was used for the work in this thesis returns the positions in a pixel-based coordinate system. Consequently, the results have to be scaled to metric units before they can be used for robot calibration.

Pattern detection The automated detection of the pattern is based on a sequence of algorithms with increased computational complexity working on decreased portions of the image.

First, the image background is estimated using a median image of ten images throughout the entire sequence of calibration poses, see Figure 5.12. In later steps, only the foreground, i.e., pixels with a minimum deviation from the median, needs to be processed.

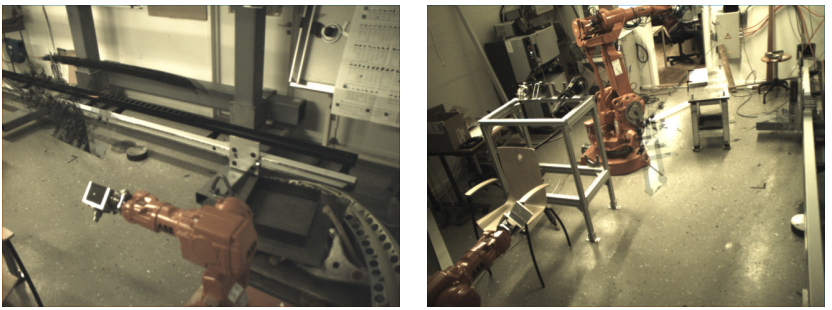


Figure 5.12 Median images used for background estimation.

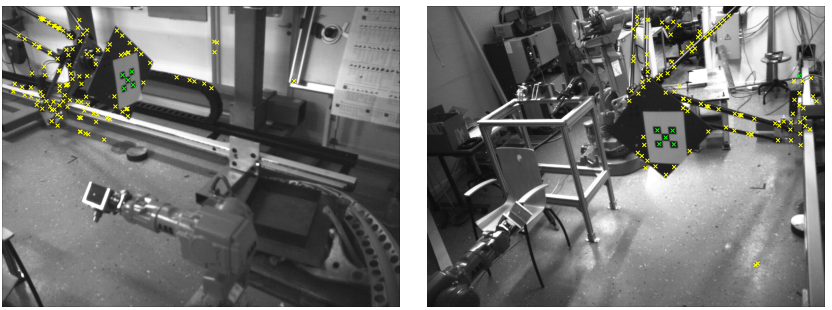


Figure 5.13 Validation of the square candidates. A yellow cross indicates a square candidate which failed validation, whereas a green cross indicates a successfully validated square.

Next, the image foreground is searched for elements resembling black squares using spatial derivatives. The result is typically that roughly 500 such square candidates are found in an image.

These square candidates are then validated by finding the positions and normals of all four edges of the square. The result is typically that roughly ten such squares remain after validation, see Figure 5.13. A yellow cross indicates a square candidate that failed the validation, whereas a green cross indicates a successfully detected square.

Each of the squares is then used to hypothesize the calibration pattern position. A validation based on searching for all 20 sides of the five black squares is made using sub-pixel edge detection [Åström and Heyden, 1999]. Typically, the pattern position is uniquely determined after this step. Figure 5.14 shows the detected pattern.

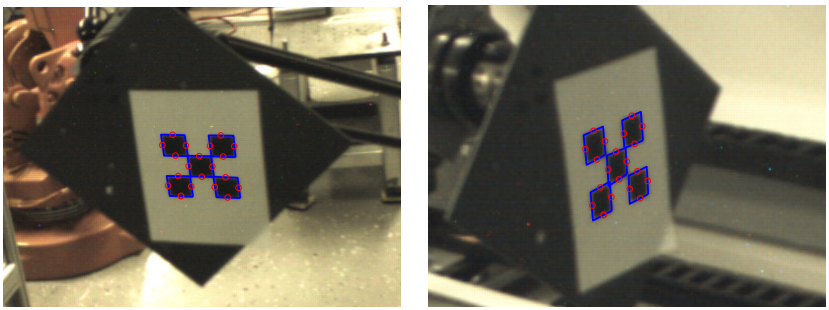


Figure 5.14 The detected chessboard pattern.

Relative orientation between cameras and robot Each corner point of the calibration pattern, as seen in both views for each calibration position, is used to estimate the relative orientation of the cameras. Standard algorithms for finding initial estimates based on five [Stewénius *et al.*, 2006], six [Philip, 1998] or eight points [Longuet-Higgins, 1981] have been tried. Bundle adjustment [McGlone *et al.*, 2004] is then used to estimate the parameters with a minimal reprojection error. It is assumed that the local optimum that we find is the global one.

Calibration pattern position Once the relative orientation between the cameras and the calibration pattern is known, the center point of the calibration pattern is estimated, together with an estimate of its covariance, by intersection [McGlone *et al.*, 2004]. Figure 5.15 shows the resulting reconstruction of the scene. The arrows represent the camera positions and orientations, the blue stars are the reconstructed TCP positions, and the red lines represent the movement of the robot end-effector.

Scaling to metric units The algorithm returns the TCP positions given in a pixel-based coordinate frame. In order to use the measurements for robot calibration, the results need to be scaled to metric units. To identify the scaling factor, several possibilities exist. The simplest, but maybe not most accurate way is to measure an object with known dimension in an image. It is however advantageous to use the already identified TCP measurements. As earlier measurements proved a sufficiently parallel alignment of the linear actuators and identified the transmission factor accurately, the fact can be used that the TCP measurement poses are arranged at the corners of a grid aligned with the robot base frame axes. Even though the exact yz-position of the TCP poses with respect to the robot frame is not known before an accurate kinematic calibration is

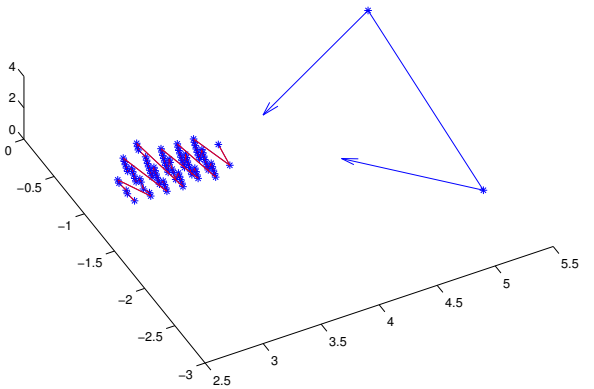


Figure 5.15 The resulting reconstruction of the measured end-effector poses (blue stars). The arrows represent the camera positions and orientations. Coordinate system see Figure 5.16.

made, the distance of the points in the x-direction is known and can be used to find the scaling factor.

Accuracy Evaluation

To evaluate the accuracy of the automated stereo vision measurement system, simultaneous measurements with the vision system and a laser tracker were performed on the L1 prototype. Figure 5.16 illustrates the experiment setup and coordinate frames. A set of 282 robot poses was chosen for calibration. The image processing was able to automatically detect the calibration pattern in 125 out of the 282 images. It is worth to note that a first measurement run resulted in images which were too dark for the automated pattern detection script. This indicates that in case of an industrial application, special care has to be taken to good illumination of the robot cell. In the following, the measurements of laser tracker and vision system are compared directly as well as in terms of calibration results.

Measurement accuracy Figures 5.17 and 5.18 compare the vision system and laser tracker measurements. The mean absolute value of the difference between the measurements is 1.7 mm. This is much higher than the laser tracker's accuracy of a approximately 10 µm in this case, so that the difference between the measurements can be regarded as the vision system's measurement error. Figure 5.18 shows that the measurement error is depending on the end-effector position within the workspace, and

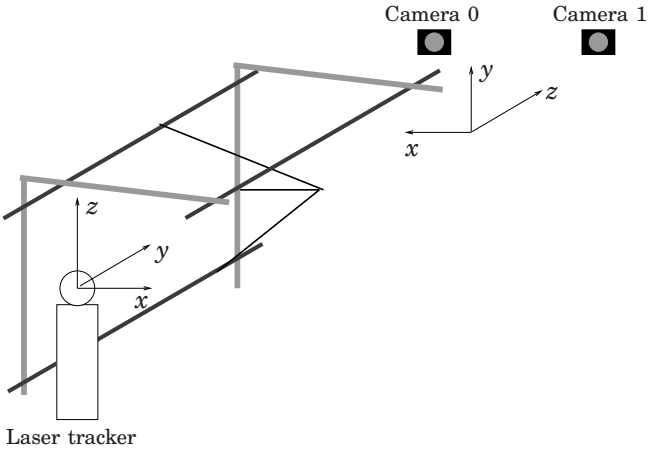


Figure 5.16 Experimental setup overview. Laser tracker and stereo vision system with the corresponding coordinate frames.

it is considerable smaller in the center than at the borders of the robot workspace or image. The error is to a large part in the xz-plane of the laser tracker, see Figure 5.16. This means that the error is smallest along the depth direction of the cameras, which is usually, at least for a single camera, the weakest direction for reconstructing 3-D positions from 2-D images. On the other hand, this was the direction that was used for finding the scaling factor, and errors might have been compensated for by adjusting the scaling. However, if the laser tracker measurements are used for finding the scaling factor, very similar results are obtained with only a minimal improvement of the measurement accuracy.

Calibration accuracy A calibration was performed with laser tracker and vision system measurements, respectively, to study the accuracy of the resulting models. As the laser tracker measures much more accurately than the vision system, the laser tracker data was used for evaluation in both cases. As usual, every second of the measurements along the grid was used for calibration and the remaining measurements for evaluation.

Figure 5.19 shows the resulting positioning accuracy and Table 5.6 the identified parameters. For the vision data, the mean positioning error is 1.5 mm and the pose dependency is very similar to the measurement error in Figure 5.18. The mean positioning error for the laser tracker based calibration is 125 µm. For comparison, the robot's unidirectional Cartesian repeatability was evaluated between two consecutive runs to be at most 68 µm. The parameters in Table 5.6 differ up to 9 mm between the two calibrations.

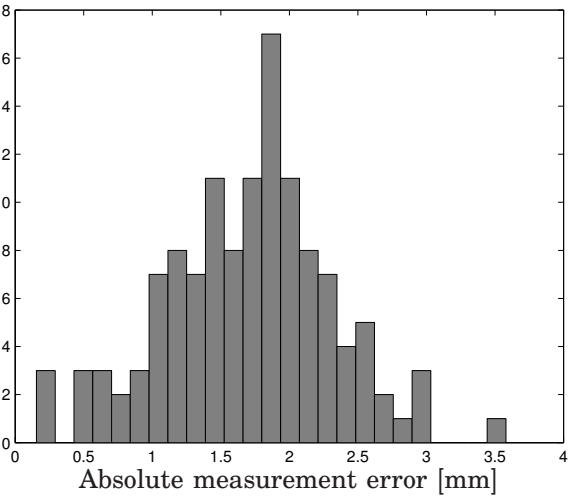


Figure 5.17 Histogram of absolute measurement error for stereo vision system, compared to the laser tracker measurements.

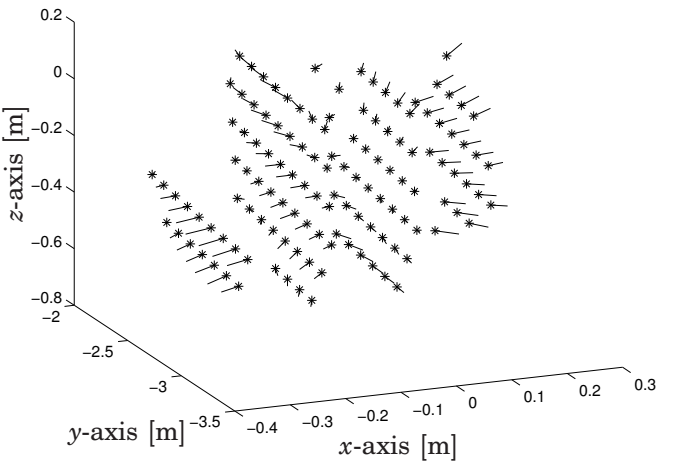


Figure 5.18 Measured wrist mount positions (stars) with three-dimensional measurement errors scaled by 20 (lines), given in the laser tracker frame, see Figure 5.16.

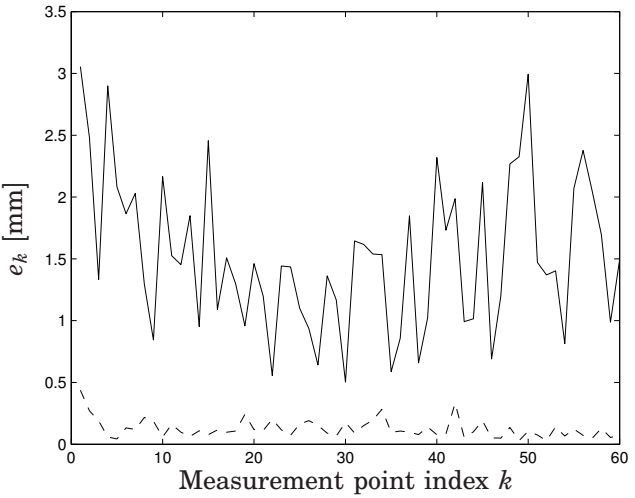


Figure 5.19 Absolute positioning error e_k for the L1 prototype model identified using vision data (solid) and laser tracker data (dashed). The corresponding mean values are 1.7 mm and 125 μm for vision and laser tracker data, respectively.

Discussion

This section evaluated the use of an automated stereo vision system for kinematic calibration of the Gantry Tau robot. The system is easy to use and thus convenient for non expert SME staff. A concern in an industrial context might be the dependence on good lighting conditions.

The mean absolute measurement error of the vision system was evaluated to 1.7 mm. Positioning and measurement errors are to a large part systematic, see e.g., Figure 5.18. The resulting errors in the kinematic model can therefore not be compensated for by more measurements, and the model's end-effector positioning accuracy is only slightly better than the measurement accuracy.

In [Dressler *et al.*, 2008], kinematic calibration of the Gantry-Tau robot was performed using only one camera. A mean absolute positioning error of 2.28 mm was obtained, with errors mostly below 3 mm. This indicates nevertheless a major improvement of the stereo vision method used here.

However, for a future usage, the origin of the measurement errors has to be studied further. Accuracy can be improved with a better camera modeling, and a better camera. The accuracy can be further improved with an optimization of the camera positioning with respect to the robot's workspace, such that the pattern plate covers a part as large as possible of the image, but still such that the complete end-effector workspace can be

Table 5.6 Identified parameters [m] of the L1 prototype, given in the laser tracker frame.

	Arm	L_i	${}^{s}A_i$	v_i
Vision system	1	2.041	$\begin{pmatrix} 1.031 \\ -1.108 \\ 0.512 \end{pmatrix}$	$\begin{pmatrix} -0.040 \\ -1.000 \\ 0.000 \end{pmatrix}$
	2	1.805	$\begin{pmatrix} -0.236 \\ -1.027 \\ 0.711 \end{pmatrix}$	$\begin{pmatrix} -0.029 \\ -0.997 \\ 0.003 \end{pmatrix}$
	3	1.810	$\begin{pmatrix} 0.968 \\ -1.474 \\ -0.944 \end{pmatrix}$	$\begin{pmatrix} -0.037 \\ -1.002 \\ 0.009 \end{pmatrix}$
Laser tracker	1	2.048	$\begin{pmatrix} 1.039 \\ -1.107 \\ 0.519 \end{pmatrix}$	$\begin{pmatrix} -0.046 \\ -0.999 \\ 0.001 \end{pmatrix}$
	2	1.810	$\begin{pmatrix} -0.236 \\ -1.026 \\ 0.720 \end{pmatrix}$	$\begin{pmatrix} -0.042 \\ -0.999 \\ 0.001 \end{pmatrix}$
	3	1.814	$\begin{pmatrix} 0.975 \\ -1.475 \\ -0.947 \end{pmatrix}$	$\begin{pmatrix} -0.047 \\ -0.999 \\ 0.005 \end{pmatrix}$

recorded. This may however have practical limitations if a permanently installed camera is used. Another possibility may be to use even more than two cameras with overlapping views.

The camera vision system presented was used for automated estimation of the TCP position. Another possible application could be dynamic tracking, which can optionally be used for feedback control.

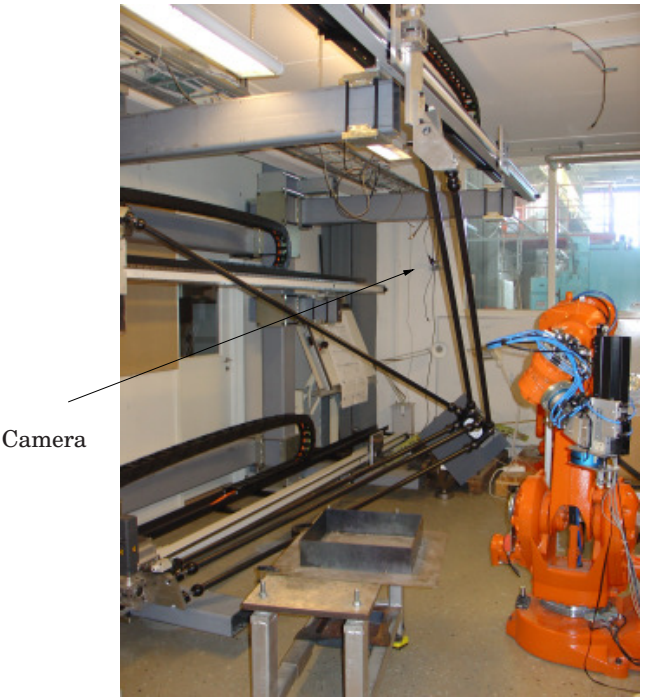


Figure 5.20 L1 prototype with mounted pattern plate for calibration using a camera mounted to the left of the window. The picture shows as well other experiment equipment narrowing the Gantry-Tau's workspace, e.g., an IRB 2400 robot.

5.6 Automated Calibration

The intended, but not exclusive application of the automated calibration method presented is to enable non-expert users perform kinematic calibration after reconfiguring a modular robot. A tool developed assists experiment preparation, measurement execution and optimization, and is experimentally tested. The tool performs exactly the same steps as an expert would do, but the challenge is to transfer the expert's knowledge and experience to numbers in order to assure a successful outcome.

Algorithm

In [Ji and Li, 1999] it was suggested to recalibrate only changed kinematic parameters after reconfiguring a modular robot. In spite of the larger number of parameters, the calibration of the complete kinematic model was adopted here. As models never simulate reality perfectly, a parameter value once identified might not be equal to the real geometric property

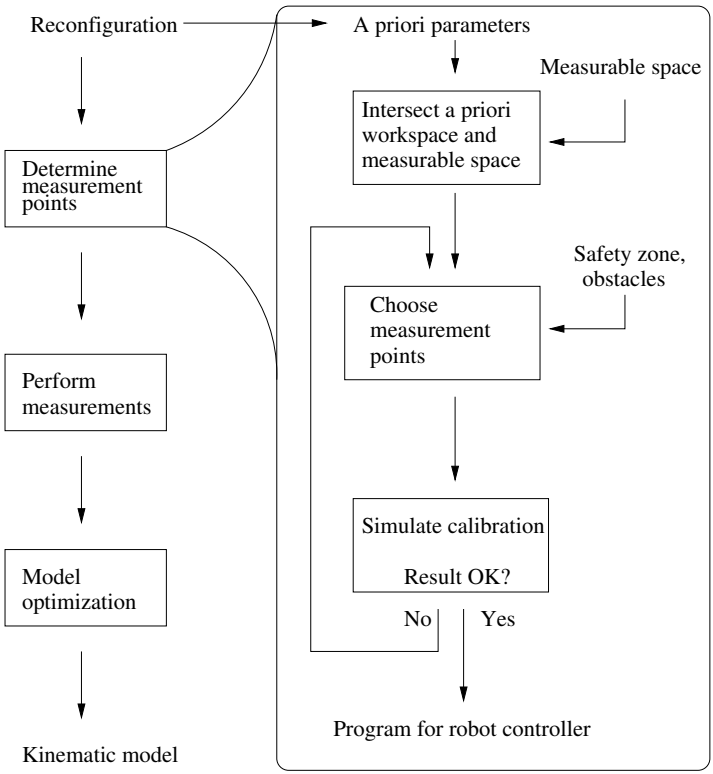


Figure 5.21 Flow of the calibration algorithm with the measurement point selection presented.

and may no longer give an optimal accuracy of the kinematic model after reconfiguration.

Figure 5.21 shows the flow of the developed method. In the first step, the measurement poses are selected according to the chosen measurement device and the robot's *a priori* workspace. This workspace is based on initial knowledge of the approximate values of the kinematic parameters, resulting e.g., from a tool optimizing the robot geometry for a specific task. Next, a simulated calibration evaluates whether the choice of measurement poses results in accurate parameters. The tool then generates a trajectory in a form readable by the robot controller, and the kinematic calibration is performed.

Workspace calculation With the aid of *a priori* kinematic parameters, the robot's working range is calculated. These parameters can result from manual measurements or as a nominal parameter output from an optimization tool which reconfigures the robot geometry according to a new desired task. The kinematic workspace, i.e., all poses for which the inverse kinematics problem has a solution, is reduced by actuator and passive spherical joint limits, as well as Cartesian limits preventing collisions of end-effector and robot framework. The working range is determined by a grid search in a maximally conceivable workspace, which is roughly determined by the *a priori* parameters and the room dimensions.

Safety zones The resulting kinematic working range can additionally be narrowed by obstacles such as walls or other equipment in the workshop, see Figure 5.20. Grid points colliding with obstacles are deleted from the workspace. In doing so, even the dimensions of extra equipment, such as a pattern plate for calibration with a camera, has to be considered.

To prevent damage of the robot due to the uncertainty of the *a priori* parameters, the working range for calibration excludes a safety zone around obstacles, in the range of the spherical joint angles and at the limit of the kinematic working range.

Measurable area The resulting working range is then intersected with the area the measurement device can record. This area is not exclusively determined by the measurement device's properties, but can possibly be narrowed by occultations for optical devices or collisions for mechanically connected devices. The measurement accuracy can vary around the measurable area, e.g., with the distance between the end-effector and measuring device, or with respect to the angle between a target and a camera's optical axis. If applicable, this dependency has to be taken into account to utilize the best achievable measurement accuracy. Here, a camera with fixed position and orientation is considered, with the pose and intrinsic camera parameters assumed to be known beforehand. Using a pinhole model of the camera, the tool tests for each workspace point whether the projection of the complete calibration pattern is inside the picture boundaries.

Measurement poses The measurement poses, i.e., the robot configurations in which the camera takes an image, are then distributed in a regular pattern over the remaining workspace obtained by intersecting the robot workspace and the recordable area. This is achieved by estimating the volume of the workspace and calculating a new grid constant based on a desired number of measurement points.

Simulation and experiment Using the *a priori* parameters, the choice of measurement poses is evaluated both by examining the Jacobian matrix of the cost function and by a simulated kinematic calibration.

The Jacobian matrix $\mathbf{J}$ is calculated, i.e., the partial derivative of the cost function with respect to the kinematic parameters for the chosen measurement points. For evaluating the Jacobian matrix, the criterion presented in [Nahvi and Hollerbach, 1996] was adapted and approved if

$$\frac{\sigma_{max}^2}{\sigma_{min}} = \frac{\sigma_{max}}{\text{cond}(\mathbf{J})} < C \tag{5.14}$$

where σ_{max} and σ_{min} represent the largest and smallest singular value of the Jacobian matrix, respectively, and $\text{cond}(\mathbf{J})$ is the conditioning number. Compared to other criteria, (5.14) is more sensitive to calibration errors. The threshold constant C is to be chosen according to the robot architecture and selected as 200 here.

The simulated calibration is performed with assumptions on measurement noise according to the chosen device. The resulting parameters are compared to the *a priori* parameters.

If the evaluation of the Jacobian matrix and the results of the optimization are approved, the real calibration is executed, if not, the procedure is repeated with more measurement points.

It is questionable that choosing more measurement points is resulting in a more accurate calibration in all cases. All parameters might not be identifiable, depending on the measurement device and the modified parameter values. Another possibility is that the area covered by the measurement device is not large enough to give accurate results for all parameters. An analysis on the identifiability of the kinematic parameters should be included in a later version of the tool.

Calibration results

An experiment was performed on the L1 prototype to validate the method presented. Table 5.7 shows the robot's *a-priori* kinematic parameters, previously determined using a measuring tape. The measurements for the validation experiment were performed using a camera and processed with a camera calibration toolbox for Matlab [Bouguet, 2007]. To be able to precisely transform results from the camera frame to the robot frame, a robot frame different from the convention in this thesis was defined: The x-axis was chosen along the axis of actuator 3 and the z-axis was intersecting with the axis of actuator 1. The origin was defined such that ${}^{s}A_3 = 0$.

Figure 5.22 shows the robot's kinematic workspace identified, together with the region in which the grid search was executed. The kinematic workspace was calculated using the *a priori* parameters resulting from

Table 5.7 Kinematic parameters of the L1 prototype, all values except for the dimensionless actuator direction v_i given in [m] in the robot base frame defined: The x-axis was chosen along the axis of actuator 3 and the z-axis is intersecting with the axis of actuator 1. The origin was defined such that ${}^sA_3 = 0$.

	Arm	L_i	${}^sA_i$	v_i
Manually measured	1	2.05	$\begin{pmatrix} 0.04 \\ 0 \\ 1.44 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
	2	1.82	$\begin{pmatrix} 0.25 \\ -1.31 \\ 1.46 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
	3	1.81	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$
Calibrated	1	2.0440	$\begin{pmatrix} -0.0396 \\ 0.0009 \\ 1.4243 \end{pmatrix}$	$\begin{pmatrix} 0.9641 \\ -0.0209 \\ -0.0087 \end{pmatrix}$
	2	1.7547	$\begin{pmatrix} 0.2443 \\ -1.2639 \\ 1.3984 \end{pmatrix}$	$\begin{pmatrix} 0.9768 \\ -0.0331 \\ -0.0216 \end{pmatrix}$
	3	1.7770	$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$

manual measurements, see Table 5.7. A grid with 10 cm interval was used. The spherical joint angles were limited to 30°, which resulted in a safety interval of nearly 15°. The cart movement was limited to ±1 m from their initial position. The safety distance from the workspace limit was 10 cm, but is not taken into account in Figure 5.22.

In Figure 5.20, the obstacles present in the robot lab at Lund University can be seen, such as experiment equipment and an ABB IRB 2400 robot. Areas corresponding to these obstacles were eliminated from the workspace. Figure 5.23 shows the remaining workspace points as well as the obstacle regions. They correspond to an experiment table, the ABB IRB 2400 robot, experiment equipment behind the robot and a palette with various workpiece samples in the corner of the room. The ratio between

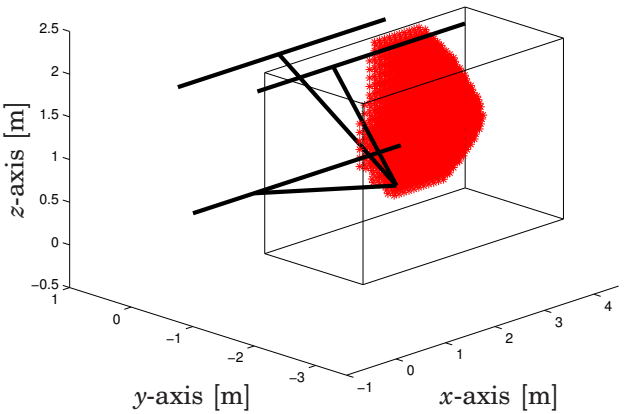


Figure 5.22 Kinematic workspace of the L1 robot (red star cloud), taking into account joint limits. The approximate actuator positions and links are represented by the thick lines. The unconstrained initial workspace, for which the grid search was performed, is indicated by the black wire-box.

the workspace remaining for calibration and the kinematic workspace in Figure 5.22 is 0.4.

Figure 5.24 shows the area which is covered by the camera view in relation to the robot workspace calculated above. The measurable area was estimated based on a pinhole camera model. The intrinsic camera parameters as well as the approximate camera pose in relation to the robot were determined beforehand. As can be seen, the camera could record the complete robot workspace.

The desired number of measurement points was 100, which included 50 measurements for model validation. This is a sufficient number to estimate seven parameters for each arm independently. However, the optimal number of measurements, which is likely to depend on the measurement device, should be identified, which was however not the focus here.

Next, the measurement points for the calibration were chosen, see Figure 5.24. For that, the workspace was filled with a new grid with a spacing of 0.1367 m, which was calculated using the desired number of measurement points, 100, and an estimation of the workspace volume. 112 Measurement points were obtained, for which the joint positions were calculated using the *a priori* parameters.

Next, a simulated kinematic calibration showed that the kinematic pa-

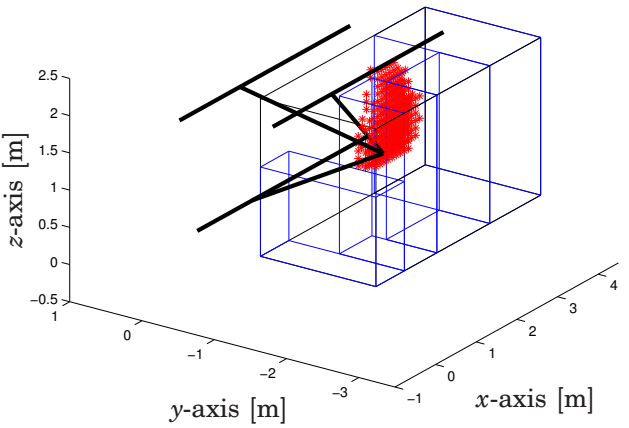


Figure 5.23 Remaining workspace points after eliminating the points which were colliding with obstacles or which were too close to the kinematic workspace limit. The obstacle regions are marked as blue boxes inside the maximally conceivable workspace, marked with a black box. The ratio between the remaining workspace and the kinematic workspace in Figure 5.22 is 0.4.

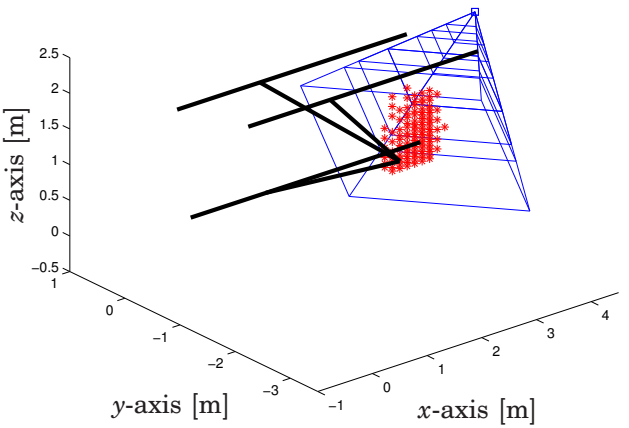


Figure 5.24 The measurement points chosen together with the area covered by the camera view.

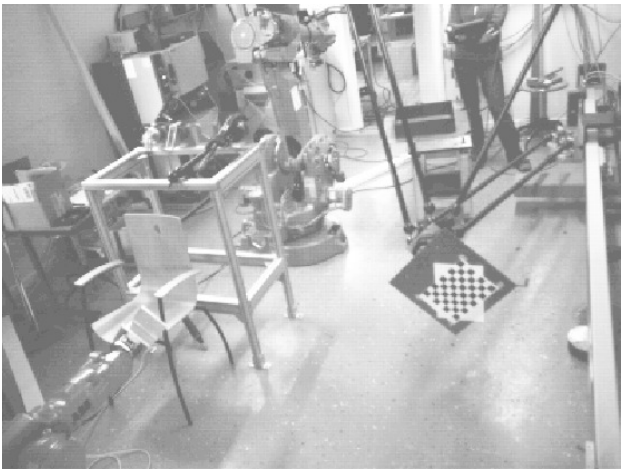


Figure 5.25 View out of the camera: The robot with the calibration pattern (to the right in the picture). To the left in the picture, obstacles, e.g., experiment equipment and and ABB IRB 2400 robot can be seen.

rameters can be estimated sufficiently well with the chosen measurement points. The evaluation of the Jacobian matrix, which was done for each arm separately, gave singular values between 0.3 and 64.

The RAPID program generated by the software tool was executed on the robot controller and simultaneously, the images were recorded by the camera, see Figure 5.25. The image processing was done manually using a camera calibration toolbox for Matlab [Bouguet, 2007], as the automated vision system described in Section 5.5 was not yet available. Every second measurement point was used for the parameter optimization, the remaining points to validate the kinematic model.

The calibrated parameters are listed in Table 5.7. Figure 5.26 shows the absolute end-effector positioning error. The error was of the same order of magnitude as the camera's measurement accuracy. It reached 9.5 mm at one point but was mostly below 3 mm, its mean value being 2.3 mm. The peaks were distributed randomly in the robot's workspace and do not correspond to a certain direction or region in the workspace, even though they appear to form a pattern in the figure.

Discussion

Section 5.5 evaluated the measurement accuracy of a stereo vision system with two cameras, which were identical to the one used here, to approximately 1.7 mm and identified the end-effector's mean absolute positioning error of the L1 Gantry-Tau as 125 µm. It can thus be expected that

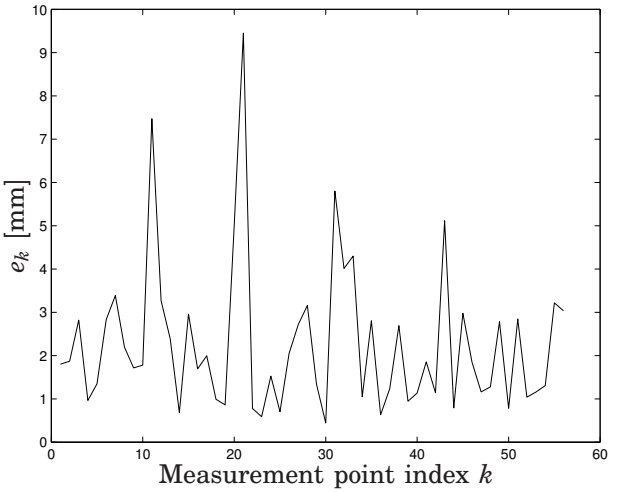


Figure 5.26 Absolute end-effector positioning error e_k of the identified kinematic model for the L1 robot.

an important part of the positioning error in Figure 5.26 was due to the camera's measurement uncertainty rather than caused by the kinematic modeling error. Comparing the calibrated parameters L_2 and L_3 in Tables 5.6 and 5.7 shows differences of approximately 5 mm between the values identified using laser tracker measurements and the values identified in this section. However, it is difficult to tell if these rather large errors are caused by an inadequate choice of the measurement poses or by inaccurate camera measurements. For that, the single camera vision has to be evaluated and the automated calibration method performed using a more accurate measurement device, which was not available at the time of the experiments.

The advantage of the method presented is the automated selection of measurement points. This makes it possible to execute calibration without choosing appropriate measurement poses manually and thus, kinematic calibration can be performed by non-experts. When selecting the measurement points, the tool has to find a compromise between safety and spreading the measurement points as wide as possible and thus probably obtaining a more accurate calibration result. Comparing Figures 5.22 and 5.23, it can be seen that the workspace is reduced considerably. Moreover, a 15° safety interval for the joint angles seems rather large. During the experiment, some joints approached their mechanical limit up to about 10° at a few poses, whereas collisions were not risked. Mechanical touch

sensors can prevent damage of the joints, but triggering one of the sensors leads to experiment interruptions and possibly fewer measurements.

In principle, the calibration method presented can be applied for any kind of robot or measurement device. Both the calculation of the measurable space of the camera and the robot's kinematic model can be exchanged for corresponding calculations for other robots or measurement devices. Such adaptations include also the determination of how different robot geometries influence different parameter and criteria values. The range of singular values and condition numbers of the Jacobian matrix in which the optimization gives a good result may vary from robot to robot. Different robot geometries may also have a different workspace edge sensitivity on parameter changes and thus need a different safety zone.

Further research might include a tool for camera pose optimization. For a freely movable camera, the tool could easily be adapted to calculate a pose where the largest possible area of the robot workspace could be recorded. For a partly movable camera, the tool could, for example, optimize the orientation. This tool would then need to include an assistance for the camera placement, e.g., the superposition of the camera picture with a calculated desired picture. The tool could also support other types of measurement tools.

A possibility to increase the calibration accuracy would be to determine more measurement points than needed and then choose an optimal set of poses, both with respect to the number and the location. Optimizing the location can easily be done by examining the influence of a certain measurement point on the singular values of the Jacobian matrix, which is already calculated for the measurement poses chosen for calibration. It would as well be interesting to have an estimate of the model accuracy, both before and after calibration. Before calibration it would allow changes to be made before executing the calibration if the desired accuracy cannot be achieved, e.g., by increasing the number of measuring points or choosing a more accurate measuring device.

5.7 Conclusion

This chapter focuses on two aspects of kinematic calibration: Evaluation of the kinematic modeling in Chapter 4 and supporting non-expert operators calibrating a reconfigurable robot.

The resulting Cartesian positioning accuracy, which was experimentally evaluated for different prototypes and kinematic models, has mean values between 90 µm and 230 µm. With a kinematic error model, the orientation error for the D1 prototype was below 0.05°, which is approximately ten times less than for the nominal model. There is however still

room for improvement, as the D1 prototype's omnidirectional Cartesian end-effector repeatability of 15 µm proves. Results indicate that improvements can be made by a more thorough actuator modeling, by taking into account stiffness modeling and possibly by arm-side measurements for the linear actuators or dual-motor control.

A stereo vision system, which has the advantage of being an affordable measurement device possibly reusable for other applications, gave in principle good results. However, the measurement accuracy has to be further studied and improved. A method to lead non-expert operators through kinematic calibration of a reconfigurable robot was successfully tested.

In ongoing projects, the kinematic calibration of the L2 and F1 prototypes is studied.

Table 5.8 Calibration results for the kinematic error model of the D1 prototype in Section 5.3: Models 1 and 2.

Model	Link	L_i [m]	Actuator offset	Actuator orientation	Final cost
			${}^{s}A_i^0$ [m]	v_i	
1	1	2.04423	$\begin{pmatrix} -3.12999 \\ -4.88145 \\ 0.70473 \end{pmatrix}$	$\begin{pmatrix} 0.99779 \\ -0.06634 \\ -0.00530 \end{pmatrix}$	5.458e-06
	2	2.04292	$\begin{pmatrix} -3.07865 \\ -3.47376 \\ 0.66795 \end{pmatrix}$	$\begin{pmatrix} 0.99752 \\ -0.06598 \\ -0.00544 \end{pmatrix}$	3.661e-06
	3	2.04350	$\begin{pmatrix} -3.13955 \\ -4.81373 \\ -0.68032 \end{pmatrix}$	$\begin{pmatrix} 0.99769 \\ -0.06603 \\ -0.00509 \end{pmatrix}$	3.515e-07
			Frame offset [m]	Frame Euler angles [deg]	
2	1	2.04423	$\begin{pmatrix} 1.81040 \\ -4.29494 \\ -0.08075 \end{pmatrix}$	$\begin{pmatrix} 137.94642 \\ -0.49443 \\ 0.36210 \end{pmatrix}$	5.506e-06
	2	2.04291	$\begin{pmatrix} 1.41419 \\ -3.86450 \\ 0.08531 \end{pmatrix}$	$\begin{pmatrix} 149.73731 \\ -0.85832 \\ 0.30077 \end{pmatrix}$	2.4647e-06
	3	2.04350	$\begin{pmatrix} 0.53097 \\ -2.08460 \\ 0.03851 \end{pmatrix}$	$\begin{pmatrix} 167.03480 \\ -0.41249 \\ 0.06433 \end{pmatrix}$	5.5341e-07

Table 5.9 Calibration results for the kinematic error model of the D1 prototype in Section 5.3: Models 3 and 4.

Model	Link	L_i [m]	B_i^0 [m]	Act. offset A_i^0 [m]	Act. orient. v_i	Final cost
3	1	2.04422	$\begin{pmatrix}-0.04871\\-0.05363\\-0.10361\end{pmatrix}$	$\begin{pmatrix}-3.18514\\-4.96829\\0.77830\end{pmatrix}$	$\begin{pmatrix}0.99773\\-0.06636\\-0.00544\end{pmatrix}$	2.5339e-06
	2a	2.04291	$\begin{pmatrix}-0.05079\\-0.11527\\-0.08175\end{pmatrix}$	$\begin{pmatrix}-3.13100\\-3.52723\\0.79856\end{pmatrix}$	$\begin{pmatrix}0.99761\\-0.06584\\-0.00544\end{pmatrix}$	2.9384e-06
	2b	2.04294	$\begin{pmatrix}-0.05864\\0.00230\\0.13871\end{pmatrix}$	$\begin{pmatrix}-3.12578\\-3.33374\\0.64030\end{pmatrix}$	$\begin{pmatrix}0.99744\\-0.06589\\-0.00562\end{pmatrix}$	2.8946e-06
	3a	2.04351	$\begin{pmatrix}-0.18926\\-0.09046\\0.04200\end{pmatrix}$	$\begin{pmatrix}-3.32073\\-4.73900\\-0.59428\end{pmatrix}$	$\begin{pmatrix}0.99766\\-0.06600\\-0.00515\end{pmatrix}$	4.4596e-07
	3b	2.04352	$\begin{pmatrix}-0.04625\\-0.00142\\-0.14303\end{pmatrix}$	$\begin{pmatrix}-3.19764\\-4.94939\\-0.65117\end{pmatrix}$	$\begin{pmatrix}0.99764\\-0.06607\\-0.00507\end{pmatrix}$	3.108e-07
	3c	2.04352	$\begin{pmatrix}-0.05438\\0.11726\\0.07666\end{pmatrix}$	$\begin{pmatrix}-3.19209\\-4.75506\\-0.80852\end{pmatrix}$	$\begin{pmatrix}0.99774\\-0.06597\\-0.00507\end{pmatrix}$	5.6626e-07
				Frame offset [m]	Frame Euler angles [deg]	
4	1	2.04422	$\begin{pmatrix}-0.04871\\-0.05363\\-0.10361\end{pmatrix}$	$\begin{pmatrix}1.75525\\-4.38178\\-0.00718\end{pmatrix}$	$\begin{pmatrix}137.94642\\-0.49443\\0.36210\end{pmatrix}$	2.6253e-06
	2a	2.04291	$\begin{pmatrix}-0.05079\\-0.11524\\-0.08174\end{pmatrix}$	$\begin{pmatrix}1.36183\\-3.91799\\0.21592\end{pmatrix}$	$\begin{pmatrix}149.73731\\-0.85832\\0.30077\end{pmatrix}$	1.6629e-06
	2b	2.04294	$\begin{pmatrix}-0.05865\\0.00234\\0.13867\end{pmatrix}$	$\begin{pmatrix}1.36706\\-3.72450\\0.05766\end{pmatrix}$	$\begin{pmatrix}149.73731\\-0.85832\\0.30077\end{pmatrix}$	1.5159e-06
	3a	2.04351	$\begin{pmatrix}-0.18926\\-0.09045\\0.04202\end{pmatrix}$	$\begin{pmatrix}0.34979\\-2.00983\\0.12452\end{pmatrix}$	$\begin{pmatrix}167.03480\\-0.41249\\0.06433\end{pmatrix}$	1.1687e-06
	3b	2.04353	$\begin{pmatrix}-0.04625\\-0.00144\\-0.14303\end{pmatrix}$	$\begin{pmatrix}0.47288\\-2.22024\\0.06763\end{pmatrix}$	$\begin{pmatrix}167.03480\\-0.41249\\0.06433\end{pmatrix}$	1.3091e-06
	3c	2.04353	$\begin{pmatrix}-0.05434\\0.11733\\0.07667\end{pmatrix}$	$\begin{pmatrix}0.47844\\-2.02590\\-0.08971\end{pmatrix}$	$\begin{pmatrix}167.03480\\-0.41249\\0.06433\end{pmatrix}$	4.3551e-07

Table 5.10 Results of the parameter sensitivity study in Section 5.3: $\text{Offset}_{\{x,y,z\}}$ and "$\text{Orient.}_{\{1,2,3\}}$" refer to the actuator offset and orientation parameters, respectively. For arms 1 and 3, the corresponding parameters are A_i^0 and v_i, respectively, $i = \{1, 3a, 3b, 3c\}$. For arm 2, the offset and orientation are described by the position and orientation of ${}^{\text{pw}}H_i$, respectively, with the orientation expressed by ZYZ Euler angles. The accumulated changes $S_{j,\text{pos}}$, $S_{j,\text{rot}}$ and $S_{j,\text{cost}}$ of the TCP position, TCP orientation and cost function, respectively, are defined as in (5.8) for $S_{j,\text{pos}}$.

Link	1	2a	2b	3a	3b	3c
TCP position change $S_{j,\text{pos}}$						
L_i	11.6	5.9	3.8	2.0	5.6	2.9
$B_{i,x}$	9.1	5.1	3.4	1.6	4.6	2.4
$B_{i,y}$	3.2	2.3	1.4	0.8	2.3	1.2
$B_{i,z}$	5.7	1.2	0.8	0.7	1.9	1.0
Offset_x	9.5	4.9	3.2	1.7	4.8	2.5
Offset_y	4.2	2.0	1.3	0.6	1.9	0.9
Offset_z	4.4	2.1	1.4	0.6	1.7	0.9
Orient._1	2.7	7.0	4.5	0.5	1.3	0.7
Orient._2	1.3	5.9	3.7	0.2	0.5	0.2
Orient._3	1.3	3.6	2.2	0.2	0.5	0.3
TCP orientation change $S_{j,\text{rot}}$						
L_i	3.7e-05	0.048	0.048	0.038	0.049	0.05
$B_{i,x}$	2.9e-05	0.042	0.042	0.031	0.04	0.042
$B_{i,y}$	1.2e-05	0.018	0.018	0.015	0.019	0.019
$B_{i,z}$	1.8e-05	0.0094	0.0094	0.013	0.017	0.017
Offset_x	3e-05	0.04	0.04	0.033	0.042	0.044
Offset_y	1.3e-05	0.015	0.015	0.012	0.017	0.016
Offset_z	1.6e-05	0.017	0.017	0.011	0.014	0.014
Orient._1	8.5e-06	0.053	0.053	0.009	0.012	0.012
Orient._2	3.9e-06	0.046	0.046	0.0032	0.0043	0.0042
Orient._3	4.8e-06	0.028	0.027	0.003	0.0039	0.0039
Cost function change $S_{j,\text{cost}}$						
L_i	8.2e-06	7.4e-06	8.2e-06	5.8e-06	5.3e-06	5.8e-06
$B_{i,x}$	6e-06	6.2e-06	7e-06	4.3e-06	3.9e-06	4.4e-06
$B_{i,y}$	2.1e-06	2.3e-06	2.7e-06	1.8e-06	1.4e-06	1.9e-06
$B_{i,z}$	3.9e-06	1.2e-06	1.5e-06	1.4e-06	1.2e-06	1.6e-06
Offset_x	6.2e-06	5.9e-06	6.6e-06	4.6e-06	4.2e-06	4.7e-06
Offset_y	2.8e-06	2e-06	2.4e-06	1.3e-06	1.1e-06	1.5e-06
Offset_z	3e-06	2.2e-06	2.6e-06	1.4e-06	1e-06	1.4e-06
Orient._1	1.5e-06	1e-05	1.1e-05	9.7e-07	7.8e-07	1e-06
Orient._2	8.6e-07	8.7e-06	9e-06	3.2e-07	2.5e-07	3.6e-07
Orient._3	8.8e-07	4e-06	4.5e-06	3.5e-07	2.4e-07	3.5e-07

6

Dynamic Modeling

To fully exploit the high accelerations achievable by parallel robots, accurate dynamic models for model-based control are essential. A high-bandwidth robot-workpiece interaction requires a stiff robot without resonances in the concerned frequency interval. It is therefore important not only to identify accurate rigid-body models, but also to study the compliance dynamics for force-controlled applications like grinding. This chapter presents a rigid body model of the Gantry-Tau robot as well as results on the compliance dynamics.

6.1 Related Work and Background

The dynamic model of an open-chain robot manipulator can be expressed as

$$M(\theta)\,\ddot{\theta} + C(\theta,\dot{\theta})\,\dot{\theta} + N(\theta,\dot{\theta}) = \Gamma \tag{6.1}$$

where $\theta \in \mathbb{R}^n$ is the vector of the n joint variables and $\Gamma \in \mathbb{R}^n$ the vector of actuator torques [Murray *et al.*, 1994]. $M(\theta) \in \mathbb{R}^{n\times n}$ is the inertia matrix, $C(\theta,\dot{\theta}) \in \mathbb{R}^{n\times n}$ the Coriolis matrix and $N(\theta,\dot{\theta}) \in \mathbb{R}^n$ describes the non-inertial forces on the joints, e.g., gravity or friction. The inverse dynamic model of parallel robots is expressed in a similar form, but rather as a function of the end-effector variables than the joint variables [Merlet, 2000].

A general solution for closed-loop manipulator dynamics was presented in [Khalil and Ibrahim, 2007]. The dynamic model of a parallel robot with m arms can be written as

$$\Gamma = J_P^T \left(F_p + \sum_{i=1}^{m} J_{vi}^T J_i^{-T} H_i \right) \tag{6.2}$$

where

J_P is the kinematic Jacobian matrix of the robot

F_p is the vector of forces and moments acting on the end-effector plate

J_{vi} relates the velocities of the end-effector and arm i

J_i is the kinematic Jacobian of arm i

H_i is the inverse dynamic model of arm i

Elastic dynamic models of Stewart platforms were developed in e.g., [Lee and Geng, 1993; Zhaocai and Yueqing, 2008]. In [Abdellatif *et al.*, 2007], friction modeling in both active and passive joints of parallel robots was presented.

The modeling of [Khalil and Ibrahim, 2007] was also applied in [Lyzell and Hovland, 2007; Hovland *et al.*, 2007b], where a dynamic rigid body model of the Queensland Gantry-Tau, including friction in the case of the second paper, was verified in experiments. Elastic dynamic modeling of the Queensland Gantry-Tau is presented in [Tyapin *et al.*, 2008]. The master's thesis [Cescon, 2008] studies compliance models of the L1 Gantry-Tau prototype.

For experimental identification of the dynamics, the choice of a trajectory exciting all parameters is important. While [Nabat *et al.*, 2006] heuristically chose a combination of slow motion for friction identification and fast motion for inertia identification, [Farhat *et al.*, 2008] parameterized the trajectory with finite Fourier series in the joint space and minimized the condition number of the observation matrix. Cartesian and passive joint limits add nonlinear constraints to the optimization problem.

There exist various model-based robot control schemes utilizing a dynamic of the robot, see e.g., Figure 6.1. In [Murray *et al.*, 1994], a summary of the computed torque control law can be found. It combines a feedforward term calculated by the inverse dynamic model and a feedback PD controller. In [Vivas *et al.*, 2005], the performances of a computed torque control law, a PID controller and model predictive control (MPC) were compared when applied to a parallel robot. Various publications consider the usage of the inverse dynamics modeled in Modelica for feedforward control [Thümmel *et al.*, 2001; Krabbes and Meißner, 2006].

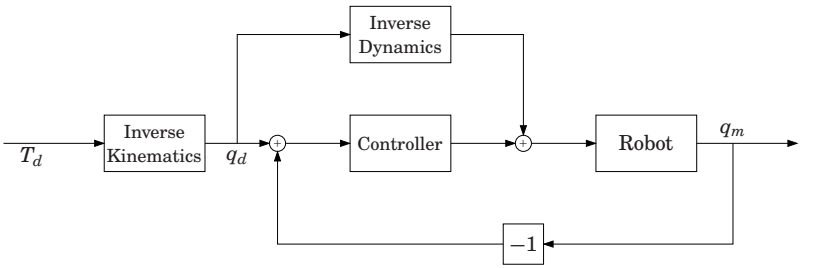


Figure 6.1 Control scheme with inverse dynamic model for feedforward compensation and feedback controller. T_d denotes the desired TCP trajectory, q_d the corresponding desired actuator trajectory and q_m the measured actuator positions.

6.2 Rigid Body Model

The dynamic model of the Gantry-Tau robot was obtained in an automated way from a Modelica model as described in [Dressler *et al.*, 2009]. The nominal 3 DOF Gantry-Tau robot with parallel actuator axes was modeled using the MultiBody library [Otter *et al.*, 2003]. Carts, links and end-effector plate are represented by point masses in their center of gravity. A link cluster is represented by one link with the mass of all links of the cluster. The spherical joints are modeled without inertia, but the weight of the joints is added to the masses of carts and end-effector plate. The model includes kinematics and dynamics of the Gantry-Tau as a rigid body system. Elasticities were not taken into account. Friction was not considered at this stage, but is manually added to the actuator forces in the next section. From the Modelica model, the following four types of equation systems are obtained. Where not indicated, the parameter and variable notation introduced in Chapter 4 is used. The global x-axis is assumed to be aligned with the actuator axes, i.e., v in (6.4) is equal to $(1,0,0)^T$.

1. The kinematic constraint equations for arms 1 to 3 are given by

$$L_i^2 - (\Delta X_i^2 + \Delta Y_i^2 + \Delta Z_i^2) = 0 \tag{6.3}$$

where the vector along link i is

$$\begin{pmatrix} \Delta X_i \\ \Delta Y_i \\ \Delta Z_i \end{pmatrix} = T - ({}^sA_i + q_i \cdot v) \tag{6.4}$$

and the end-effector position is $T = (T_x, T_y, T_z)^T$.

2. The inverse velocity Jacobian, relating joint and Cartesian velocities is

$$J^{-1} = \begin{pmatrix} 1 & \frac{\Delta Y_1}{\Delta X_1} & \frac{\Delta Z_1}{\Delta X_1} \\ 1 & \frac{\Delta Y_2}{\Delta X_2} & \frac{\Delta Z_2}{\Delta X_2} \\ 1 & \frac{\Delta Y_3}{\Delta X_3} & \frac{\Delta Z_3}{\Delta X_3} \end{pmatrix} \tag{6.5}$$

3. The relation between joint and Cartesian accelerations is

$$\ddot{T} = J \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} - \begin{pmatrix} \Delta X_1 & \Delta Y_1 & \Delta Z_1 \\ \Delta X_2 & \Delta Y_2 & \Delta Z_2 \\ \Delta X_3 & \Delta Y_3 & \Delta Z_3 \end{pmatrix}^{-1} \begin{pmatrix} (\dot{T}_x - \dot{q}_1)^2 + \dot{T}_y^2 + \dot{T}_z^2 \\ (\dot{T}_x - \dot{q}_2)^2 + \dot{T}_y^2 + \dot{T}_z^2 \\ (\dot{T}_x - \dot{q}_3)^2 + \dot{T}_y^2 + \dot{T}_z^2 \end{pmatrix} \tag{6.6}$$

4. The dynamic equations are then

$$M_1 \begin{pmatrix} \ddot{T}_x \\ \ddot{T}_y \\ \ddot{T}_z \end{pmatrix} + M_2 \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{pmatrix} + G = J^{-T} \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \tag{6.7}$$

where f_i is the force acting on cart i and

$$M_1 = \begin{pmatrix} m_p + \sum_{i=1}^{3} \frac{m_{a,i}}{2} & 0 & 0 \\ \sum_{i=1}^{3} \frac{m_{a,i}}{4} \frac{\Delta Y_i}{\Delta X_i} & m_p + \sum_{i=1}^{3} \frac{m_{a,i}}{4} & 0 \\ \sum_{i=1}^{3} \frac{m_{a,i}}{4} \frac{\Delta Z_i}{\Delta X_i} & 0 & m_p + \sum_{i=1}^{3} \frac{m_{a,i}}{4} \end{pmatrix}$$

$$M_2(:,i) = \begin{pmatrix} m_{c,i} + \dfrac{m_{a,i}}{2} \\ \dfrac{\Delta Y_i}{\Delta X_i}(m_{c,i} + \dfrac{m_{a,i}}{4}) \\ \dfrac{\Delta Z_i}{\Delta X_i}(m_{c,i} + \dfrac{m_{a,i}}{4}) \end{pmatrix}$$

$$G = g \begin{pmatrix} 0 \\ 0 \\ m_p + \sum_{i=1}^{3} \dfrac{m_{a,i}}{2} \end{pmatrix}, \quad i = \{1,2,3\}$$

The masses of cart i, link cluster i and end-effector plate are $m_{c,i}$, $m_{a,i}$ and m_p, respectively, and g is the gravitational acceleration.

Friction Modeling

Coulomb and viscous friction in the actuators were added to the dynamic model generated from the Modelica model. The force f_i on cart i in (6.7) is then

$$f_i = \tau_i - f_{c,i}\,\mathrm{sign}(\dot{q}_i) - f_{v,i}\,\dot{q}_i \tag{6.8}$$

with τ_i being the actuator torque, $f_{c,i}$ the Coulomb friction coefficient and $f_{v,i}$ the viscous friction coefficient. Friction in the passive spherical joints is not considered.

The friction model was verified with a similar method as presented in [Marton and Lantos, 2009]. The actuators were moved with constant velocity and the actuator torque measured to obtain the curve exemplified in Figure 6.2.

In the case of all carts moving with the same constant velocity, i.e., actuator and Cartesian accelerations are zero, the actuator torques τ_i are calculated as

$$\begin{pmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{pmatrix} = J^T \cdot G + \begin{pmatrix} f_{c,1}\,\mathrm{sign}(\dot{q}_1) + f_{v,1}\,\dot{q}_1 \\ f_{c,2}\,\mathrm{sign}(\dot{q}_2) + f_{v,2}\,\dot{q}_2 \\ f_{c,3}\,\mathrm{sign}(\dot{q}_3) + f_{v,3}\,\dot{q}_3 \end{pmatrix} \tag{6.9}$$

With this method, the viscous friction coefficients can be determined uniquely. If symmetrical Coulomb friction is assumed, i.e., the Coulomb friction term is the same for both positive and negative velocities, the Coulomb friction coefficients and the accumulated end-effector and link masses are also uniquely determined. Without this assumption, the dotted line in Figure 6.2 marking the gravity term can be moved vertically

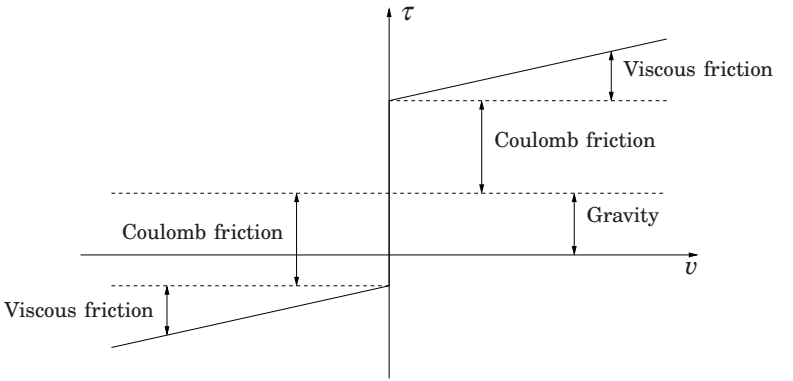


Figure 6.2 Relation between constant cart velocity v and actuator torque τ for friction identification experiments.

thus changing the Coulomb friction terms but still resulting in the same $\tau(v)$ curve. As the friction modeling had a more qualitative character to validate the model structure, the robot was not taken apart to measure the gravity force.

In [Hovland *et al.*, 2007b] it is described how the actuator friction of the Queensland Gantry-Tau was measured without mounted arm system, thus overcoming the problem of the gravity term. A so-called saddle friction term, caused by moments of the link forces, was also introduced.

To identify the friction coefficients, the actuator torques were recorded for 15 different actuator velocities up to $1\,\mathrm{m\,s^{-1}}$. The end-effector position for these measurements was constant in the yz-plane and is indicated with a star in Figure 6.3. To study the influence of the link angles and to evaluate the validity of the model in the complete robot workspace, measurements were as well taken in other end-effector positions shown in Figure 6.3. Unsymmetrical friction was assumed, i.e., different friction coefficients for negative and positive actuator velocity were adopted. The gravity term was chosen such that the Coulomb friction was as symmetric as possible.

Figure 6.4 and Table 6.1 show the identification results. The accumulated masses, i.e., $m_p + \sum_{i=1}^{3} \frac{m_{a,i}}{2}$, were determined to 22.38 kg. Except for the viscous friction $f_{v,3}$ of actuator 3, the coefficients for positive and negative velocities vary considerably. The behaviour of actuator 3 is the most linear one, i.e., the measurement data fits best to the identified piecewise linear model in Figure 6.2. Actuator 2 is the least sensitive to the end-effector position and link angles. This can be explained by Table 6.2, which shows how the gravity force is divided between the actuators for the different end-effector positions. Actuator 2 has a much smaller dis-

Table 6.1 Identified coefficients for Coulomb friction (subscript c) and viscous friction (subscript v). The superscript indicates negative ($-$) and positive ($+$) velocities.

Actuator i	$f_{c,i}^{-}$ [N]	$f_{c,i}^{+}$ [N]	$f_{v,i}^{-}$ [kg/s]	$f_{v,i}^{+}$ [kg/s]
1	145.69	189.12	158.01	205.03
2	119.23	262.59	133.76	190.00
3	177.22	239.06	197.52	185.94

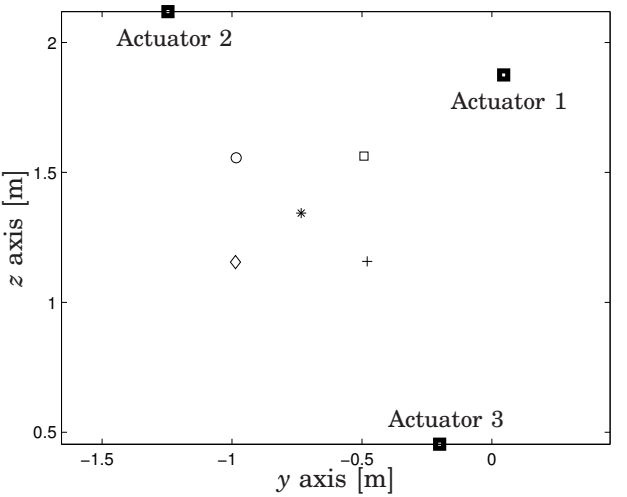


Figure 6.3 End-effector positions in yz workspace section used for friction identification; the symbols used correspond to the ones used for the measurement data in Figure 6.4; the thick squares indicate the guideway positions.

tributed load from the end-effector and link weights than the remaining two actuators and is therefore less sensitive to the link angles. The end-effector position in the lower left corner indicated by a diamond (◊) differs most from the identified model for actuators 1 and 3. Table 6.2 shows that the gravity contribution for this end-effector position is higher for actuators 1 and 3 than for the other end-effector positions. Similarly as shown in [Hovland *et al.*, 2007b], a different link angle and load can modify the friction behaviour of the actuator.

Table 6.2 Contribution $J^T \cdot G$ in [N] of gravity to actuator forces for the end-effector positions shown in Figure 6.3.

Actuator i	$*$	$\square$	$\circ$	$\Diamond$	$+$
1	-176.9	-159.1	-180.3	-261.8	-201.6
2	-14.6	-23.7	-3.6	-21.5	-33.7
3	191.5	182.8	183.9	283.3	235.3

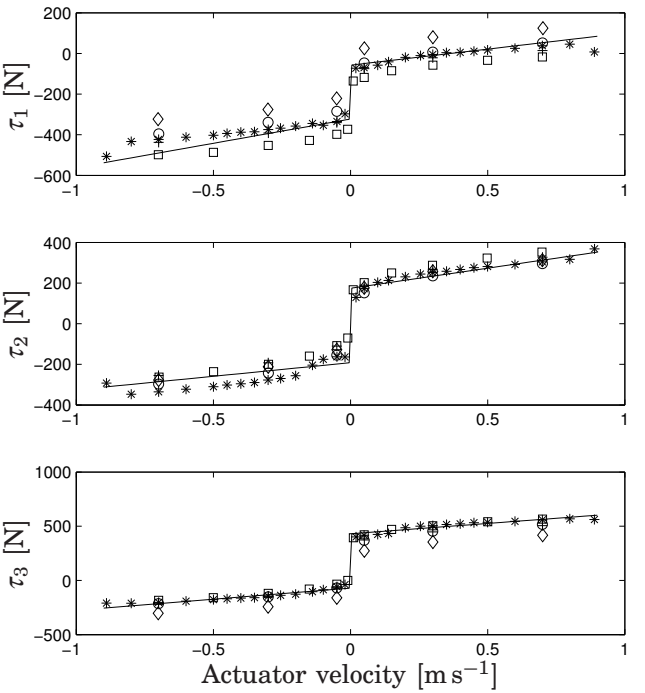


Figure 6.4 Friction modeling for the three actuators: The curves show the motor torques as in Figure 6.2, including both friction and a contribution of the gravity force. The solid curves are the identified models and the measurement data is represented by the symbols $(\star, \Diamond, \square, \circ, +)$, corresponding to the end-effector positions in Figure 6.3. The model was identified using the center position only $(\star)$, the remaining measurements were compensated for the different gravity contribution.

Table 6.3 Identified model parameters and sum of residuals $e_i = \sum_{k=0}^{N}(\tau_{i,k,\mathrm{mod}} - \tau_{i,k,\mathrm{meas}})^2$ between modeled actuator torque $\tau_{i,k,\mathrm{mod}}$ and measured actuator torque $\tau_{i,k,\mathrm{meas}}$ for actuator i and measurement k over all N measurements. The performance of the non-symmetric function model from Table 6.1 was tested with the friction parameters marked with $\star$. The grey **0** indicates that the corresponding parameter was not taken into account.

Model	m_p [kg]	Arm i	$m_{c,i}$ [kg]	$f_{c,i}$ [N]	$f_{v,}$ [kg/s]	e_i [10^7 (Nm)^2]
1	8.76	1	14.33	152.13	**0**	1.11
		2	14.05	259.09	**0**	2.33
		3	38.10	291.24	**0**	1.98
2	9.13	1	13.58	128.62	102.43	1.03
		2	13.64	145.13	355.25	1.29
		3	38.23	194.57	388.77	1.05
3	8.79	1	13.85	$\star$	$\star$	2.38
		2	14.85	$\star$	$\star$	3.07
		3	38.47	$\star$	$\star$	1.77
4	9.67	1	12.51	109.97	$\star$	1.11
		2	12.73	207.13	$\star$	1.60
		3	37.70	243.75	$\star$	1.31

Identification of Inertial Parameters

The dynamic model (6.9) includes seven masses for end-effector plate (m_p), carts ($m_{c,i}$) and links ($m_{a,i}$) and six friction parameters for the active joints ($f_{c,i}$ and $f_{v,i}$), $i = \{1,2,3\}$. An experiment combining fast and slow motion was performed. Using least squares estimation, several models including different combinations of masses and friction parameters were identified. Table 6.3 presents the identified models. Figures 6.5 and 6.6 show the data fit for identification and evaluation data, respectively.

For the inertial parameters, the best results were obtained for only estimating end-effector plate and cart masses and neglecting the link masses. Some of the link masses were even erroneously identified as negative when including them in the estimation. As the links are made of carbon fibre and therefore very light-weight, their acceleration force is for any motion very small. An explanation may be that the link acceleration term rather tries to compensate for unmodeled forces, e.g., friction in the passive joints, than model the small link inertial forces.

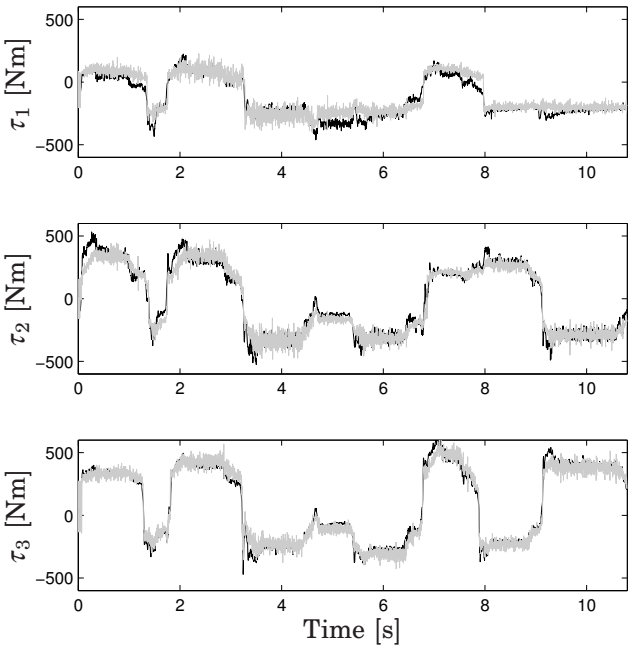


Figure 6.5 Fit for estimation data: Measured torque (black) for actuators 1 to 3 with torque calculated by Model 2 in Table 6.3 (grey).

The best data fit was achieved for a model with symmetric Coulomb and viscous friction terms (Model 2). The asymmetric friction models from above, which were identified at a separate occasion, gave a worse data fit. Attempts to identify different friction parameters for negative and positive velocities resulted in unrealistic values for the end-effector mass and/or the corresponding friction parameters. That might be an indication that the motion performed for identification did not excite the end-effector inertia sufficiently.

Figure 6.6 shows the model output for Model 2 and the measured actuator torque for a slow validation motion, in which the end-effector was traversing a 1 cm circle. Due to the slow velocity, the friction forces dominate clearly. At such low velocities, the friction force is subject to the Stribeck effect, which was not included in the model and thus causes modeling errors. For actuator 1, the Coulomb friction level does not coincide between model and measurement. Also, the modeled torque seems to miss a hardly noticeable sinusoidal trend, which could be an underestimated viscous friction force, as the velocity curve in Figure 6.7 suggests.

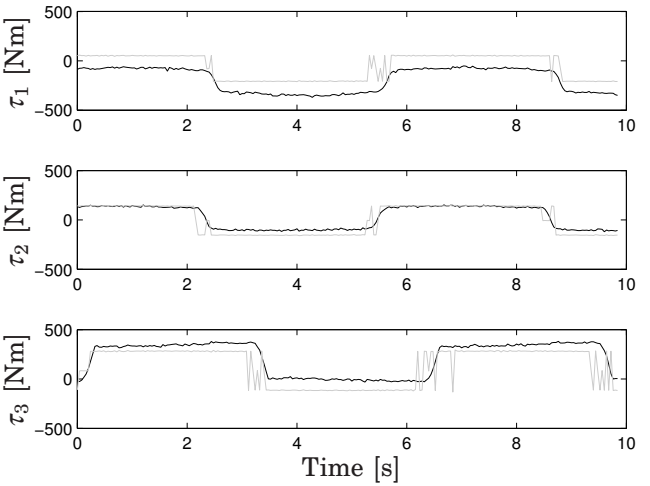


Figure 6.6 Model validation with a slow circular motion. Measured (black) and modeled (grey) torque for actuators 1 to 3. The oscillations in the modeled torque when changing direction are due to measurement noise. The high-frequency noise causes the velocity signal and consequently the modeled Coulomb friction to change sign when close to 0.

For actuator 2, the modeled torque follows the measurements closely. For actuator 3, it seems that the basis torque level, i.e., the Coulomb friction force, is estimated accurately, but that a slight linear trend is missed in the model. A comparison with the cart velocities in Figure 6.7 indicates that the linear trend is not related to the actuator movement, but either to an underestimation of the end-effector inertia or to unmodeled effects.

Discussion

The friction in the Gantry-Tau linear guideways was shown to be adequately modeled by Coulomb friction and viscous damping. It can be seen in Figure 6.4 that a different end-effector positioning changes the friction coefficients only slightly. A comparison of the friction coefficients in Tables 6.1 and 6.3 shows that the identification results differ between the two experiments. Especially the viscous friction coefficients for Model 2 are larger than the corresponding coefficients identified in the designated friction experiments in Table 6.1. The two experiments were performed with one year's interval, and other factors as temperature and lubrication can influence the friction model. The results of the friction modeling are therefore mostly qualitative.

The identification of the complete dynamic model gave satisfactory re-

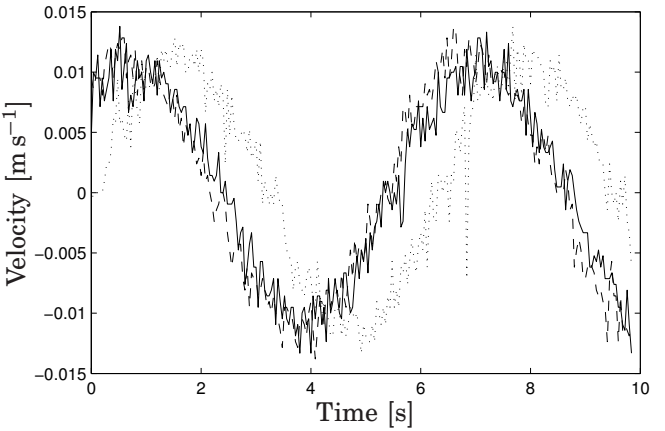


Figure 6.7 Velocity for carts 1 (solid), 2 (dashed) and 3 (dotted).

sults. The identified model is sufficiently good to be used for e.g., model-based control and feedforward control, even though the actuator torques are not followed with a high accuracy. The model can certainly be improved by modifying the identification trajectory for an optimized parameter excitation. The results still show the validity of the dynamic rigid model presented.

6.3 Compliance Dynamics

Parallel robots are generally considered as stiffer than serial robots. In [Tyapin *et al.*, 2008], the resonance frequency of the Queensland Gantry-Tau is calculated to be larger than 50 Hz. However, the findings in [Cescon, 2008] and experiences from the ILC experiments in Chapter 7 showed notable compliant behaviour with lower resonance frequencies for the L1 prototype. A rigid body model is thus not sufficient for model-based control at higher frequencies. A flexible dynamics model is required to access the performance in applications like force control. A new modeling compared to [Cescon *et al.*, 2009] was required to fit more specifically the needs for ILC, but as well because of the serial wrist, which had been added since the first experiments and thus had changed the dynamic properties.

First, a study of the actuator dynamics is shortly presented, and then the dynamics of the complete L1 prototype is discussed. The work presented was mainly done for the ILC experiments presented in Chapter 7 and is completed by the findings presented in [Cescon *et al.*, 2009] and [Cescon, 2008].

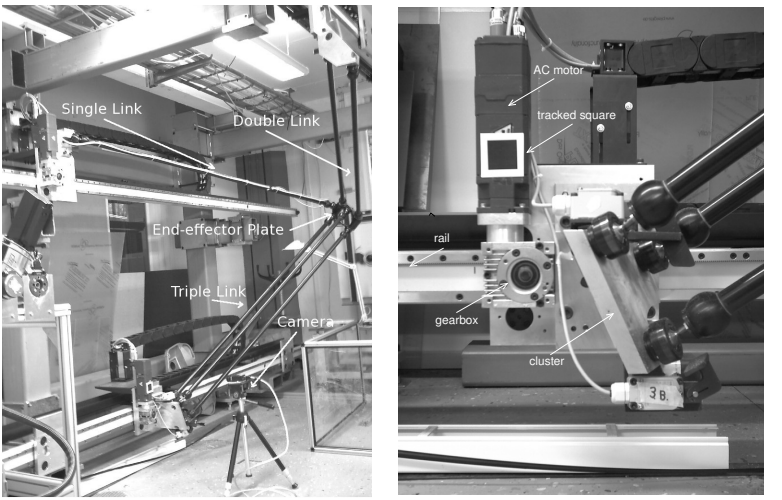


Figure 6.8 L1 prototype with camera tracking system (left) and actuator detail with tracking target (right).

The identification is based on measurements using linear encoders [Heidenhain, 2010]. The experimental set-up is described more in detail in Section 7.3. For additional measurements, a laser tracker [Leica Geosystems, 2009] and a camera vision system as shown in Figure 6.8 were used. The vision tracking system is based on the one described in [Olsson, 2007] and was enhanced in [Cescon, 2008] with a predictor to be able to track higher velocities. The vision system and linear encoders are sampled at 250 Hz via the extended robot control system presented in Chapter 3. The laser tracker has a sampling rate of 1 kHz. The linear encoders and laser tracker have an accuracy in the order of 1 µm and 10 µm, respectively. The vision system's accuracy depends on the distance between camera and target.

Actuator Models

The L1 prototype, like most industrial robots, has its actuator position measured from the motor side, with so-called co-located measurements. To identify the dynamics of only the mechanical arm structure, it would be necessary to both measure the end-effector and the arm-side actuator movement. As not enough sensors were available for that, the actuator dynamics were studied separately. Additionally, the identified actuator models were used for the motor-side ILC in Chapter 7.

The ILC algorithms require models with the position reference as input, i.e., models of the controlled, closed-loop system. Therefore, single-

input-single-output (SISO) models for the three actuators, with motor-angle reference as input and measured motor angle as output, were derived

$$q_m = G_{a,i}(q)\, q_r, \quad i = \{1,2,3\} \tag{6.10}$$

where q without subscript denotes the forward shift operator. This should not be confused with q_r and q_m, which represent the reference and the measurement of the actuator position, respectively.

In order to disregard the non-linear coupling effects between the actuators, all three actuators performed the same motion simultaneously, such that the tool was moving in the x-direction in parallel to the actuator axes, and the passive ball joints did not change their angles. This procedure facilitates obtaining linear SISO models, but it limits as well the validity of the identified models for movements involving coupling effects.

The tuning of the very simple ILC algorithm used for motor-side ILC in Chapter 7 only requires the system's stationary gain and delay, and experimental results prove that delayed first order systems adequately model the controlled motors. Three step response experiments with amplitudes of 1 mm, 1.5 mm and 2 mm, respectively, were performed. This choice is motivated by the ILC experiments described in Chapter 7, where the robot makes small movements within a few millimeters from the end-effector position and thus also chosen here.

The three motor models are identified using data from all three step response experiments, resulting in

$$\begin{aligned} G_{a,1}(q) &= q^{-5}\frac{0.03527}{1-0.9647\,q^{-1}} \\ G_{a,2}(q) &= q^{-5}\frac{0.03425}{1-0.9656\,q^{-1}} \\ G_{a,3}(q) &= q^{-5}\frac{0.02748}{1-0.9728\,q^{-1}} \end{aligned} \tag{6.11}$$

The sampling time of the identified discrete-time systems is 4 ms. The controlled linear actuators have a bandwidth of approximately 1.1 Hz. Note that this does not directly correspond to the tracking performance, where e.g., feedforward control is applied. Also, with a better tuning of the position and speed controllers, a much faster response could be obtained. The models for actuator 1 and actuator 2 are close to each other, while the model for actuator 3 is somewhat different. An explanation for that might be that, as identification experiments in Section 6.2 showed, cart 3 is heavier than the other two carts. The controlled actuators can be approximated by a low-pass filter. The five samples delay are presumably caused by internal data communication in the L1 prototype's IRC5 system.

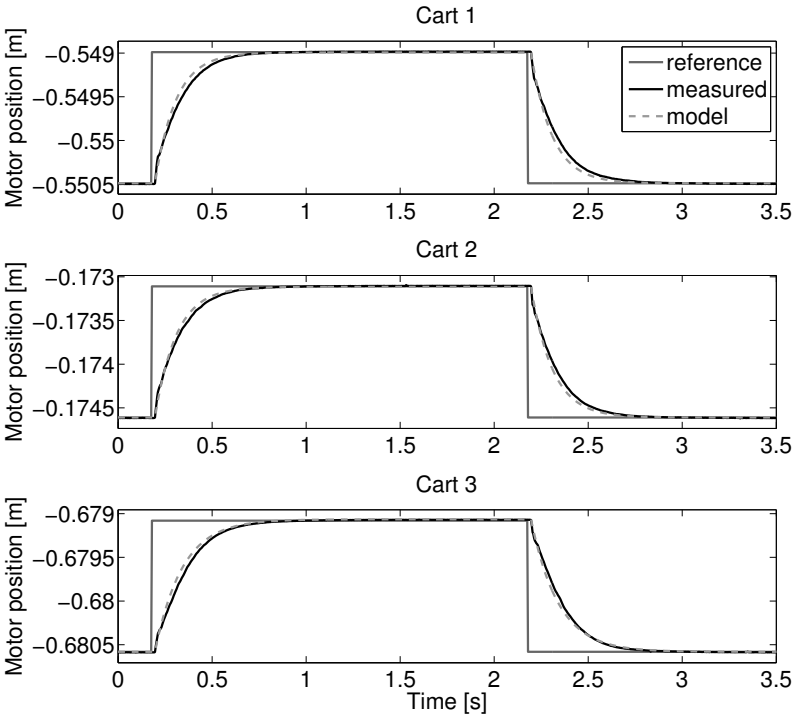


Figure 6.9 Model validation: Measured and modeled actuator positions together with position reference for all 3 actuators.

The models are validated using the part of the data from the 1.5 mm step response experiment which has not been used for identification. Figure 6.9 shows a good approximation of the controlled motors with a delayed first order system. However, even if different data is used for identification and validation, still the same simple movement is performed. A validation with the rectilinear ILC trajectory from Chapter 7 shows that the identified models are too slow for modeling the robot performing this different type of movement, where stronger coupling effects appear between the actuators, which were not modeled here.

Table 6.4 Results of resonance frequency measurement for the L1 prototype.

		Lowest resonance frequency in	
		x-direction	y-direction
Force direction	x	7.4 Hz	7.3 Hz–7.5 Hz
	y	10.4 Hz–10.7 Hz	11.4 Hz–11.5 Hz

Modeling of the Complete Robot Structure

Resonance frequency To determine the new resonance frequency of the L1 robot including the serial wrist, impulse response experiments were performed. A force impulse was applied to the 3 DOF end-effector plate, and the resulting position was measured with the linear encoder in x- and y-direction. Different experiments were carried out with a force applied along the x- and y-direction, respectively.

When the force was applied in x-direction, a resonance of 7.4 Hz was observed in x-direction, while the end-effector started a vibration with a frequency between 11.4 Hz and 11.9 Hz in y-direction, which after a few oscillations turned into a 7.3 Hz–7.5 Hz vibration. When the force was applied in y-direction, the resonance frequency in y-direction was between 11.4 Hz and 11.5 Hz, and the resonance frequency in x-direction between 10.4 Hz and 10.7 Hz. Table 6.4 summarizes the results.

Linear black box models Besides knowledge about the robot properties in general, the models were needed for tuning the ILC filters and designing a Kalman filter for estimating the tool position. A pseudo random binary sequence (PRBS) signal with an amplitude of 1 mm and adjustable frequency content was added to each of the motor references. The tool position was measured by the Heidenhain linear gauges as described for the resonance experiments above. No compensation of the friction was made, which makes it more difficult to obtain accurate models.

Two different models were identified. The output signal for both models is the tool position in the measured x- and y-directions. In contrast to the notation introduced for the Gantry-Tau kinematics, the measured tool position is denoted z_m, which is in line with the notation utilized for ILC. As input signal for one model, the xy-reference for the tool position, z_r, is used.

$$z_m(t) = G_r(q) \cdot z_r(t) \tag{6.12}$$

Another possibility is to use the actuator position measurements trans-

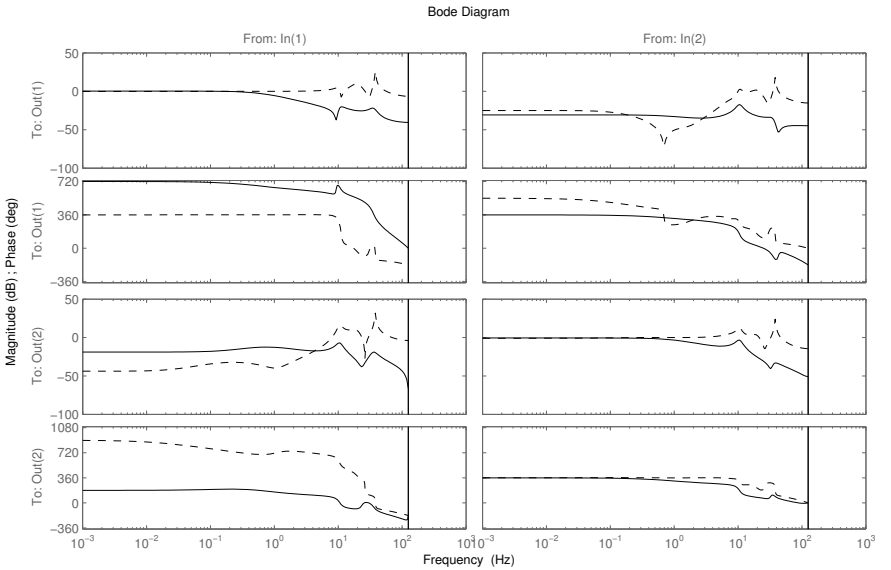


Figure 6.10 Bode diagrams: Model from tool position reference to measured tool position (solid), model from tool position calculated by kinematics to measured tool position (dashed).

formed to the tool side by the forward kinematics, z_c.

$$z_m(t) = G_c(q) \cdot z_c(t) \tag{6.13}$$

This signal has no direct physical correspondence, but can be considered closer connected to the output signal and the resulting model performs better for estimation than models with the actuator positions as input signals. Figure 6.10 shows the Bode plots of the two models which are used for the ILC tuning and the Kalman estimator. The resonances around 10 Hz can clearly be seen. The model from tool position reference to measured tool position was a seventh order model and the model from the tool position calculated by kinematics to the measured tool position was a tenth order model.

Black box models of robot without wrist Different experiments were made for subspace-based identification of the L1 dynamics before the wrist was added. Cart 3 was performing a motion of overlayed sinus signals while the other actuators were standing still and the tool motion tracked by the vision system. Additionally, an experiment similar to the resonance

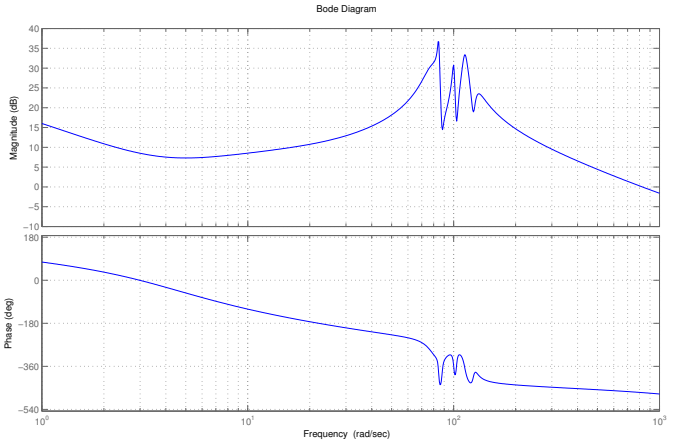


Figure 6.11 Bode diagram of the estimated 12th order model based on laser tracker measurements.

frequency experiments described above was carried out. A force impulse was applied to the end-effector while the tool motion was measured with a laser tracker.

From the experiment where only cart 3 was moved, a forth order model with a pair of complex poles within the unit circle corresponding to a frequency of 14 Hz was obtained. This model was transformed into a physically meaningful form as described later in this section. From the laser tracker measurements, a 12th order model was estimated and showed closely spaced resonances centered around a frequency of 100 rad s^{-1}, or 16 Hz, see Figures 6.11 and 6.12. Figure 6.13 shows the impulse response of the identified model. More impulse response experiments were performed resulting in very similar resonances [Cescon *et al.*, 2009].

Physical modeling An black box model identified above was fit to the physically meaningful form in the Laplace domain

$$X(s) = (M\ s^2 + D\ s + K)\ B\ U(s) \tag{6.14}$$

in order to identify the inertia M, the damping D and the stiffness K, with X being the position and U the input signal. The matrix B is relating the input signal to the physical spring-damper system. A description of how this was done can be found in [Cescon *et al.*, 2009].

From the fourth order model based on vision measurements from the

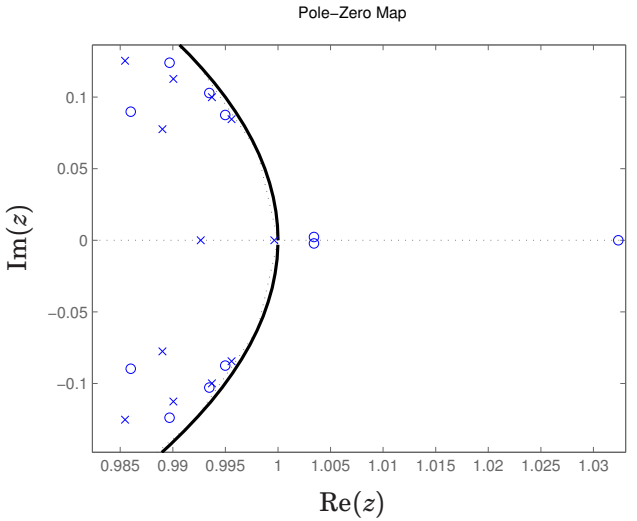


Figure 6.12 Pole-zero map for the 12th order model based on the laser tracker data with poles('x') and zeros('o').

experiment where only cart 3 was moved, the matrices were identified as

$$K = 10^3 \begin{bmatrix} -3.7326 & -1.5671 \\ 6.9209 & 0.5560 \end{bmatrix}$$

This is a positive definite matrix, which has the eigenvalues $(-1.5883 \pm 2.4996i) \cdot 10^3$. The inertia and damping matrices were identified as

$$M = \begin{bmatrix} 0.0278 & -0.0331 \\ -0.1759 & 0.3086 \end{bmatrix}, \quad D = \begin{bmatrix} 3.7432 & -4.7826 \\ -9.7408 & -3.4976 \end{bmatrix}$$

Further, the generalized eigenvalue problem

$$\det(M\lambda + K) = 0 \tag{6.15}$$

determined the resonance modes $\pm\sqrt{\lambda}$ to 14 Hz and 103 Hz, matching the results found above.

Discussion

The compliance dynamics of the L1 Gantry-Tau prototype was identified based on two types of experiments: The robot structure compliance was excited by force impulses applied to the end-effector and by moving one or

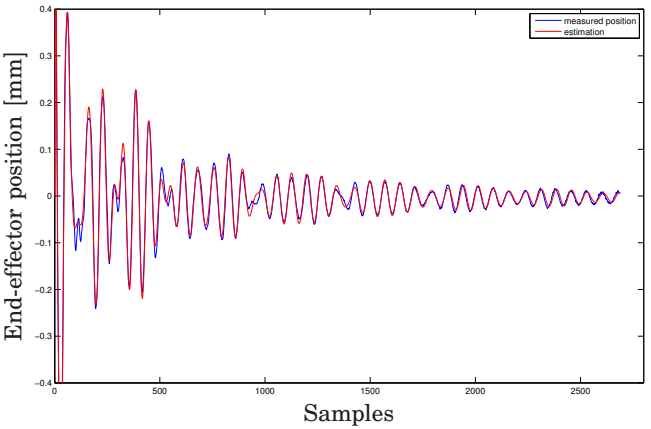


Figure 6.13 Impulse response of the identified 12th order model (red) compared with the laser tracker measurements (blue). The laser tracker's sampling rate was 1 kHz.

several actuators. Experiments were carried out both without and later with a mounted wrist.

The main result is that the resonance frequency is much lower than expected, approximately 10 Hz and 15 Hz, with and without wrist, respectively. Both type of experiments are consistent in this frequency range.

For high-bandwidth applications like force control, it is crucial to identify the elasticities to further optimize the Gantry-Tau design for these applications. The newly developed spherical joints are extremely stiff and a comparison of motor-side and arm-side measurements on the actuators shows considerably less flexibility than the tool motion. Thus, the flexibility can be assumed to lie in the links and the framework. In Figures 5.20 or 6.8 one can see a long lever from cart 2 to link cluster 2, which considerably decreases the stiffness and introduces unnecessary flexibility. For the L2 prototype, this has been significantly improved in addition to stiffer spherical joints and links.

A physical parameterization of one of the black box models gave positive definite stiffness and inertia matrices, but the damping matrix has, despite a positive trace, a negative determinant, which is difficult to interpret.

6.4 Conclusion

This chapter presented dynamic modeling results of the Gantry-Tau robot. Both rigid body dynamics and compliance dynamics were modeled. It was found that the lowest resonance frequency is roughly between 10 Hz and 16 Hz, and thus that for high-bandwidth applications, the compliances cannot be neglected.

The results in this chapter have been an important input for the design of the L2 prototype. First experimental results for the L2 prototype indicate an improved stiffness and improved dynamical properties compared to the L1 prototype.

7

Iterative Learning Control

This chapter presents experimental results of ILC applied to the Gantry-Tau robot. The ILC algorithm is based on estimates of the robot tool position. The tuning of the estimator and the ILC algorithm is based on the dynamic models developed in Chapter 6. Learning is enabled up to and above the robot's dominating resonance frequencies.

7.1 Introduction and Related Work

Chapters 5 and 6 present kinematic calibration and the identification of dynamic models of the Gantry-Tau robot. Accurate modeling is necessary for using parallel robots in applications where the present serial robot technology is unsatisfactory. To achieve high accuracy, kinematically as well as dynamically, expensive high precision measurement devices, components and assembly are often necessary. In particular for a reconfigurable robot used at SMEs for small production lots, this is not cost-efficient. A possibility for handling this situation could be to use ILC to compensate for errors. ILC is a control method that compensates for repetitive errors, and originates from the robotics field. Since the first academic publications in 1984 [Arimoto *et al.*, 1984; Casalino and Bartolini, 1984; Craig, 1984], it has developed into an intense research area as can be seen in the survey papers [Bristow *et al.*, 2006; Ahn *et al.*, 2007] among others.

ILC algorithms for robots can easily be made based on motor angle measurements, as they are in general the only measurements available in commercial robot systems. The control objective is however to follow a desired tool path. At high velocities, the flexibility of the robot structure gains more importance, and the tool accuracy is coupled to a lower degree to the joint accuracy. A difficulty in industrial applications is to measure the actual tool position. High precision measurement devices

are either very expensive, e.g., laser trackers, or need to be in physical contact with the end-effector, which is disturbing for the industrial application. Cheaper optical solutions, e.g., cameras, suffer from possible occultations. It is therefore preferable to use additional sensors, e.g., accelerometers, in combination with signal processing algorithms to obtain accurate estimates of the relevant signals [Norrlöf and Karlsson, 2005]. These estimates can then be used in the ILC algorithm to improve the tool performance and hence be able to compensate for errors originating from the flexible robot structure, as well as rigid body dynamic errors.

This chapter presents experimental results of ILC using tool position estimates, applied to the Gantry-Tau robot. For comparison and evaluation, experiments with ILC based on motor angle measurements and tool position measurements were performed as well. The estimates are derived from motor angle and tool acceleration measurements and two estimation methods are compared: Complementary filtering and Kalman filtering.

The learning is often limited to frequencies below the resonance, which limits the accuracy at high frequencies. In the experiments presented in this thesis, learning is enabled up to and above the first dominant resonance frequency of the robot.

Only in a few earlier studies, ILC is applied to PKMs. In [Abdellatif *et al.*, 2006; Abdellatif and Heimann, 2010], linear ILC algorithms are applied to the direct-driven hexapod PaLiDA. The main objective was to increase accuracy of high velocity motions. The ILC algorithms were based on the measured motor positions and the tool performance was evaluated by transforming the actuator positions by the forward kinematics. In [Cheung and Hung, 2009] an ILC algorithm based on measured joint positions is applied to a planar parallel manipulator prototype with a small workspace (4 cm $\times$ 2 cm) intended for semiconductor packaging. The ILC algorithm was intended to reduce the error at lower frequencies, and therefore the higher frequency contents of the control signal are removed by an appropriate filter, in order not to excite the robot at around the resonance frequencies. A high-precision laser measurement system provided measurements of the resulting tool position. In [Burdet *et al.*, 2001] ILC algorithms based on measured motor angles were applied to a 3 DOF and two 2 DOF parallel robots. To overcome the problem with a flexible robot structure, the learning algorithm has a low bandwidth to prevent instability. Another example for ILC based on measured motor angles applied to a parallel robot, here a 3-PRPS platform, can be found in [Chuang and Chang, 2001].

Estimation and ILC have been combined in few publications. In [Ratcliffe *et al.*, 2006] norm-optimal ILC is performed on a gantry robot, using the system states estimated by an observer. In [Gunnarsson *et al.*, 2007] ILC was carried out on a flexible 1 DOF robot arm. The ILC algorithm

uses an arm-side estimate based on motor angle and arm-side acceleration measurements. In [Schöllig and D'Andrea, 2009] the error of a state-space model, linearized along the desired trajectory, is estimated by a Kalman filter in the iteration-domain. The ILC input signal is derived by minimizing the deviation of the states from the desired trajectory. Another example is [Tayebi and Xu, 2003], where the estimated states for a class of time-varying nonlinear systems are used in an ILC algorithm.

The above references concentrate on specific estimation and ILC algorithms. This chapter presents experimental results based on the previous work [Wallén *et al.*, 2009; Wallén *et al.*, 2011], where the main focus was to obtain a good estimate of the controlled variable, usually the tool pose, by using sensor fusion.

7.2 Theoretical Background

ILC Algorithms

The following section summarizes the ILC algorithm presented in [Wallén, 2011]. ILC is applied to the following discrete-time LTI system

$$\begin{aligned} y_k(t) &= T_{ry}(q)r(t) + T_{uy}(q)u_k(t) \\ z_k(t) &= T_{rz}(q)r(t) + T_{uz}(q)u_k(t) \end{aligned} \tag{7.1}$$

with measured variable y_k and controlled variable z_k at iteration k, and the forward shift operator q. The reference input variable is denoted r, which is independent of the current iteration, and u_k is an iteration dependent input, which is modified by the ILC algorithm. The equations are assumed to describe a stable closed loop system, stabilized by a controller.

A linear discrete-time ILC algorithm is applied according to

$$u_{k+1}(t) = Q(q)\big(u_k(t) + L(q)\epsilon_k(t)\big) \tag{7.2}$$

with possibly non-causal ILC filters Q and L. The error ϵ_k in the ILC update equation is given by

$$\epsilon_k(t) = r(t) - \hat{z}_k(t) \tag{7.3}$$

using the estimate $\hat{z}_k$ of the controlled variable given by

$$\hat{z}_k(t) = F_r(q)r(t) + F_u(q)u_k(t) + F_y(q)y_k(t) \tag{7.4}$$

with the stable filters F_r, F_u and F_y.

The system (7.1) and ILC algorithm (7.2) can be expressed in matrix form, similarly to the descriptions in [Phan and Longman, 1988; Moore, 1998]

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{T}_{ry}\boldsymbol{r} + \boldsymbol{T}_{uy}\boldsymbol{u}_k \\ \boldsymbol{z}_k &= \boldsymbol{T}_{rz}\boldsymbol{r} + \boldsymbol{T}_{uz}\boldsymbol{u}_k \end{aligned} \tag{7.5}$$

The ILC algorithm (7.2) in matrix form is

$$\begin{aligned} \boldsymbol{u}_{k+1} &= \boldsymbol{Q}(\boldsymbol{u}_k + \boldsymbol{L}\boldsymbol{\epsilon}_k) \\ \boldsymbol{\epsilon}_k &= \boldsymbol{r} - \hat{\boldsymbol{z}}_k \end{aligned} \tag{7.6}$$

with the estimate

$$\hat{\boldsymbol{z}}_k = \boldsymbol{F}_r\boldsymbol{r} + \boldsymbol{F}_u\boldsymbol{u}_k + \boldsymbol{F}_y\boldsymbol{y}_k \tag{7.7}$$

The vectors $\boldsymbol{r}, \boldsymbol{u}_k$, $\boldsymbol{y}_k$, $\boldsymbol{z}_k$ and $\hat{\boldsymbol{z}}_k$ are N-sample sequences of the corresponding signals, e.g., the reference sequence $\boldsymbol{r}$ is

$$\boldsymbol{r} = \begin{pmatrix} r(0) & \dots & r((N-1)T_s) \end{pmatrix}^T \tag{7.8}$$

where T_s is the system's sampling time. The transfer operators in (7.1) are replaced by the system matrices $\boldsymbol{T}_{ry}$, $\boldsymbol{T}_{uy}$, $\boldsymbol{T}_{rz}$, $\boldsymbol{T}_{uz}$, $\boldsymbol{F}_r$, $\boldsymbol{F}_u$ and $\boldsymbol{F}_y$, composed by pulse response coefficients. E.g., The matrix $\boldsymbol{T}_{ry}$ is formed by the pulse response coefficients $g_{T_{ry}}, t \in \{0,\dots,(N-1)T_s\}$ of the transfer operator T_{ry}. This results in the Toeplitz matrix

$$\boldsymbol{T}_{ry} = \begin{pmatrix} g_{T_{ry}}(0) & 0 & \dots & 0 \\ g_{T_{ry}}(1) & g_{T_{ry}}(0) & & 0 \\ \vdots & & \ddots & \vdots \\ g_{T_{ry}}(N-1) & g_{T_{ry}}(N-2) & \dots & g_{T_{ry}}(0) \end{pmatrix} \tag{7.9}$$

In the experiments, the filter L is chosen as $L(q) = \gamma q^{\delta}$, with a learning gain γ and a time shift of δ samples. This non-causal filter is implemented by letting

$$\epsilon_k(t) = \epsilon_k\big(T_s(N-1)\big), \quad t > T_s(N-1)$$

which in matrix form is represented as

$$\boldsymbol{L} = \begin{pmatrix} 0 & \dots & 0 & \gamma & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \gamma & \dots & 0 \\ \vdots & & & & & \ddots & \\ 0 & \dots & 0 & 0 & 0 & \dots & \gamma \\ 0 & \dots & 0 & 0 & 0 & \dots & \gamma \\ \vdots & & & \vdots & & \vdots & \\ 0 & \dots & 0 & 0 & 0 & \dots & \gamma \end{pmatrix}. \tag{7.10}$$

Q is chosen as a non-causal filter with zero-phase characteristics. A standard way to carry out such filtering is to use a causal filter $\bar{Q}$, and perform forward-backward filtering in order to obtain a zero-phase filter. The lower-triangular Toeplitz matrix $\bar{Q}$ is created similarly as in (7.9) from the pulse response coefficients of the filter $\bar{Q}$, giving

$$\bar{\bar{Q}} = \bar{Q}^T \bar{Q} \tag{7.11}$$

A more detailed discussion of the ILC implementation can be found in [Wallén *et al.*, 2010].

As explained in more detail below in Section 7.3, the ILC algorithm is only applied to a central part of the trajectory. This central part is preceded and followed by a phase where the ILC control signal is weighted in time-domain. The weight coefficients during this n samples long phase is given by the vector

$$w = \begin{pmatrix} w(1) & \dots & w(n) \end{pmatrix}^T \tag{7.12}$$

The matrix Q in (7.6) is now obtained as

$$Q = Q_w \bar{\bar{Q}} \tag{7.13}$$

with

$$Q_w = \text{diag}(0,\dots 0,\, w_1,\dots w_r,\, 1,\dots 1,\, w_r,\dots w_1,\, 0,\dots 0) \tag{7.14}$$

Stability and convergence of the system described is discussed in [Wallén *et al.*, 2011]. The system (7.5) controlled by the ILC algorithm (7.6) using the estimate (7.7) is stable if and only if

$$\rho\Big(Q\big(I - L(F_u + F_y T_{uy})\big)\Big) < 1 \tag{7.15}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix and I the identity matrix. If the system (7.5) controlled by the ILC algorithm (7.6) fulfills

$$\bar{\sigma}\Big(Q\big(I - L(F_u + F_y T_{uy})\big)\Big) < 1 \tag{7.16}$$

that is, if the largest singular value is smaller than one, then the system is stable and u_k converges to the limit value u_∞ with monotone exponential convergence.

Tool Position Estimation

Two different estimation methods were used, complementary filtering and a Kalman filter. Because of the measurement equipment limitations described in Section 7.3, he tool position estimate $\hat{z}$ included, like the tool position measurement, only the x- and y-coordinates. The models and observers were tuned and evaluated based on the specific trajectory chosen for the ILC experiments.

Complementary filter A pair of filters is called a complementary filter if the sum of their transfer functions is 1 for all frequencies. Complementary filtering is used to fuse noisy measurements of the same physical variable from two sensors with different frequency characteristics, i.e., measurements y_1 with high frequency noise and measurements y_2 with low frequency noise. A low-pass filter G can then be used to filter y_1, which is a good estimate at low frequencies. The measurements y_2, which are a good estimate at high frequencies, are filtered with the high-pass filter $(1-G)$, i.e., the complement to G. The estimate $\hat{y}$ is then the sum of the two filtered signals [Higgins, 1975].

This simple estimation technique is a common sensor fusion technique for the combination of gyro and accelerometer signals in flight control industry. In [Higgins, 1975] its relation to the steady-state Kalman filter for a class of filtering problems is discussed. No details about the noise processes are considered in complementary filtering, and the filters are derived based on a simple analysis in the frequency domain. A similar kind of filter pairs is widely used in communication systems [Vaidyanathan, 1993], however there the sum of the transfer functions does not necessarily have zero phase, since many communication systems can handle time delays.

In the present application, the tool position should be estimated from the motor positions and from measurements by an accelerometer mounted on the end-effector plate. Transformed by the forward kinematics, the motor position measurements result in $\hat{z}_c$, which is a good tool position estimate for low frequencies. The estimate $\hat{z}_a$ is obtained by double integration of the accelerometer output and is a sufficiently good tool position estimate for higher frequencies. These two estimates are then combined to the tool position estimate $\hat{z}$ as

$$\hat{z}(t) = G(q)\,\hat{z}_c(t) + \big(1 - G(q)\big)\,\hat{z}_a(t) \tag{7.17}$$

The filter G is a low pass filter with zero-phase characteristics. This is achieved by applying a second order Butterworth filter in both forward and reverse direction using the Matlab function `filtfilt`, [Matlab, 2011]. The cutoff frequencies f_n for the x- and y-directions are tuned experimentally by inspection to

$$f_n = (4.50 \quad 2.63)^T \text{ Hz} \tag{7.18}$$

A similar approach with double integration and high-pass filtering of the accelerometer signal is used in [Nordström, 2006]. Simulation examples and experimental evaluation prove that the industrial robot control is improved by using an accelerometer as an additional sensor.

In Figure 7.1 the resulting tool position estimation error $e = z - \hat{z}$ is shown. The actual tool position z is measured by length gauges in x- and

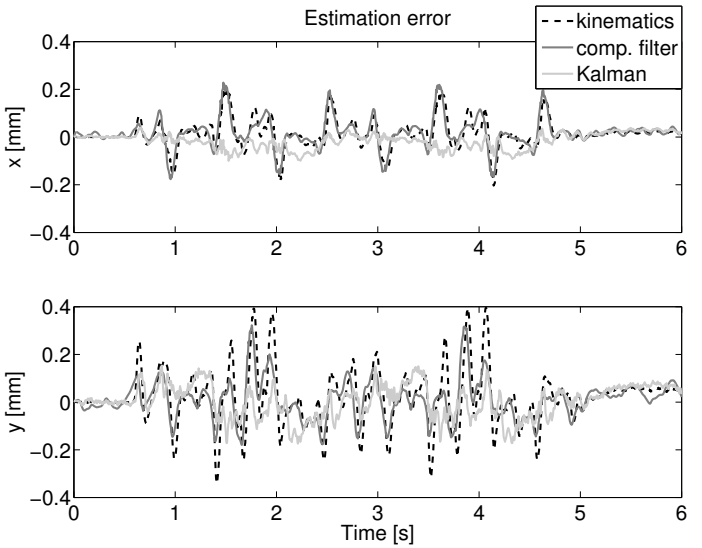


Figure 7.1 Estimation errors for different tool position estimates using forward kinematics ('kinematics', dashed), complementary filter ('comp. filter', black solid) and Kalman filter ('Kalman', grey solid).

y-direction. The error of the estimate (7.17) is compared to the error when using the estimate $\hat{z}_c$, which relies only on motor position measurements. The usage of the complementary filter together with the additional sensor mounted on the end-effector plate, the accelerometer, decreases the estimation error, especially in the y-direction. The main reason for this is the flexibility of the robot in y-direction compared to the much stiffer x-direction.

Kalman filter The stationary Kalman filter estimating the tool position is based on the identified model (6.13) from Section 6.3, which is extended to include the accelerometer measurements. The model is described in discrete-time state-space form as

$$\begin{aligned} x(t+T_s) &= Ax(t)+Bu(t) \\ z(t) &= Cx(t) \end{aligned} \tag{7.19}$$

The input u consists of the measured motor positions transformed by the forward kinematics to corresponding tool positions. The output z is the tool position, which is measured by length gauges in the xy-direction. New states for the tool velocity (x_v) and acceleration (x_a) are introduced

using the forward Euler approximation for the time derivative:

$$x_v(t+T_s) \approx \frac{z(t+T_s)-z(t)}{T_s} = \frac{C\big(Ax(t)+Bu(t)\big)-Cx(t)}{T_s} \tag{7.20}$$

Together with a corresponding expression for x_a, a state-space model with the extended state vector $\bar{x}$ and state-space matrices $\bar{A}, \bar{B}, \bar{C}$ is obtained

$$\begin{aligned}
\underbrace{\begin{pmatrix} x(t+T_s) \\ x_v(t+T_s) \\ x_a(t+T_s) \end{pmatrix}}_{\bar{x}(t+Ts)} &= \underbrace{\begin{pmatrix} A & 0 & 0 \\ \frac{1}{T_s}C(A-I) & 0 & 0 \\ \frac{1}{T_s^2}C(A-I) & -\frac{1}{T_s}I & 0 \end{pmatrix}}_{\bar{A}} \begin{pmatrix} x(t) \\ x_v(t) \\ x_a(t) \end{pmatrix} + \underbrace{\begin{pmatrix} B \\ \frac{1}{T_s}CB \\ \frac{1}{T_s^2}CB \end{pmatrix}}_{\bar{B}} u(t) \\
\begin{pmatrix} z(t) \\ y(t) \end{pmatrix} &= \underbrace{\begin{pmatrix} C & 0 & 0 \\ \mathbf{0} & 0 & 1 \end{pmatrix}}_{\bar{C}} \begin{pmatrix} x(t) \\ x_v(t) \\ x_a(t) \end{pmatrix}
\end{aligned} \tag{7.21}$$

The output of the extended model consists of the measured variable y and the controlled variable z.

A Kalman filter can now be designed to estimate the tool position z

$$\begin{aligned}
\hat{\bar{x}}(t+T_s) &= \bar{A}\,\hat{\bar{x}}(t) + \bar{B}\,u(t) + K\big(y(t) - \bar{C}\,\hat{\bar{x}}(t)\big) \\
\hat{z}(t) &= \begin{pmatrix} C & 0 & 0 \end{pmatrix} \hat{\bar{x}}(t)
\end{aligned} \tag{7.22}$$

Like in model (7.21), the input u consists of the actuator positions transformed by the forward kinematics to a corresponding tool position. The measurement y is the tool acceleration. The observer gain K is determined by solving the corresponding Riccati-equation described e.g., in [Åström and Wittenmark, 1997]. The matrices R_1 and R_2 correspond to the process and measurement noise variance, respectively. The covariance matrix R_2 is based on the output error covariance of model (7.21). The covariance matrix R_1 is then chosen as $R_1 = rI$. The factor r is determined by inspection to minimize the estimation error and has to be rather small in order to avoid drift in the estimate. Figure 7.1 compares the resulting Kalman filter estimation error to the corresponding errors of estimates $\hat{z}_c$ and complementary filter estimate $\hat{z}$. The Kalman filter performs slightly better than the complementary filter. It is likely that the quality of the estimate can be further increased by using a better, possibly nonlinear model.

For a Kalman filter estimating the tool position based on accelerometer data, the tool acceleration has to be one of the output signals. However,

better results are obtained when the model is manually extended with states corresponding to the acceleration, compared to fitting a black-box model to measured accelerometer data. A reason for this might be the rather noisy accelerometer signal.

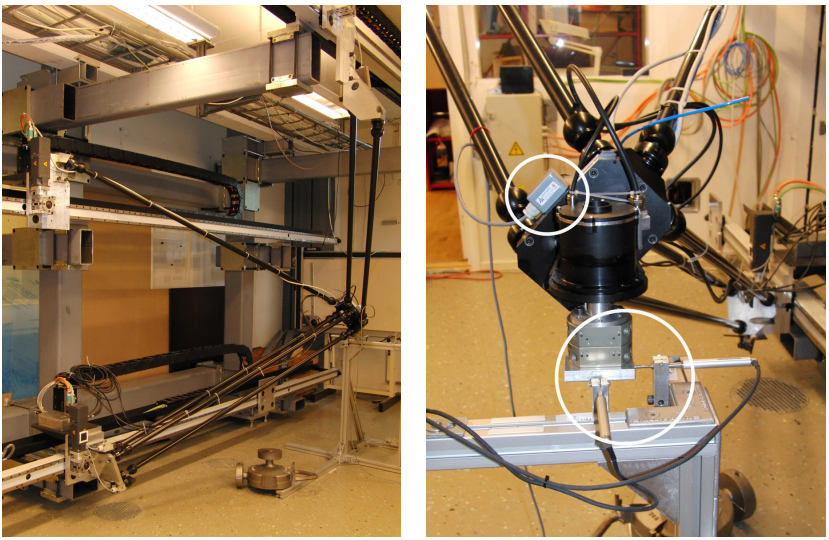
7.3 Experiment Description

Experimental Setup and System

Robot The experiments were performed on the L1 prototype in Lund. Only the parallel 3 DOF part of the robot was used, i.e., the serial wrist was fixed during the experiments. The robot controller is described in Chapters 2 and 3: A standard industrial ABB IRC5 system is extended with the possibility to change reference signals sent between main and axis computers and read synchronized sensor values. For the ILC experiments, a driver was implemented that reads values from a file, synchronized with the external controller. The robot is only controlled by the motor position controllers, independently for each motor. The lack of an inverse dynamic model for feedforward control or any other compensation of the coupling effects between the actuators leads to repeating control errors for repeated tasks, which can be compensated for by ILC.

3 DOF accelerometer A Freescale accelerometer [Freescale, 2010] is mounted on the 3 DOF wrist holder plate of the robot, see Figure 7.2(b). The accelerometer can measure translational accelerations in 3 DOF up to a bandwidth of 300 Hz within a range of ± 1.5 g. The standard deviation of the measurement noise for the different channels is measured to be approximately $0.07\,\mathrm{m\,s^{-2}}$. The orientation of the accelerometer with respect to the robot tool frame was difficult to determine with a good accuracy. The accelerometer was mounted on the wrist holder, which can not reorient but only move in 3 DOF, such that the gravity force could not be used for identifying the accelerometer orientation. An end-effector motion along the tool frame axes with a sufficiently high acceleration induces oscillations in the robot structure, which makes it difficult to identify these directions.

Heidenhain length gauges Two Heidenhain length gauges ST 3078 [Heidenhain, 2010] mounted on a support frame were used to measure the end-effector position, see Figure 7.2. As only two gauges were available, it was decided to measure the motion in xy-direction, i.e., the horizontal



(a) Robot. (b) External sensors.

Figure 7.2 Experimental set-up: L1 robot with the sensor support frame (left) and end-effector with mounted 3 DOF accelerometer (right, upper white circle) and the Heidenhain length gauges (right, lower white circle).

motion. The directions orthogonal to the guideways, the y- and z-direction, have similar dynamic properties compared to the x-axis, which is parallel to the guideways. In that way it was possible to capture the difference. The measurement range, 30 mm x- and y-direction, respectively, is located around the end-effector position in the workspace center chosen for ILC and the identification experiments in Chapter 6. The gauges actually measure the motion of a metal block attached to the 2 DOF serial wrist, which was assumed to be stiff enough, and not directly on the 3 DOF wrist holder platform. The gauges have a nominal accuracy of $\pm 1\,\mu$m. However, several error sources might decrease the accuracy, like unknown sensor dynamics, friction between the sensor tip and metal block or oscillations in the support frame of the gauges. The sensor tip engraved traces into the metal block in the course of repeated experiments. However, the traces were minor and changing the block during an experiment would have changed the experiment conditions.

System overview and signals Figure 7.3 gives a schematic overview of the robot system and signals, which are as well listed in Table 7.1. Experiments presented in Section 6.3 showed that the dynamics between

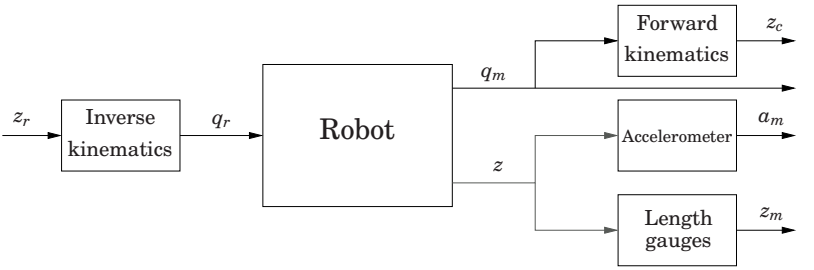


Figure 7.3 Robot system with signals according to Table 7.1. Signals available (measurements or input signals) in black, actual tool position $z(t)$, which is not available, in grey.

the motor-side and the arm-side position of the linear actuators can be neglected compared to the dynamics of the robot link structure. In the following, no distinction between the two signals will therefore be made and only the arm side will be considered. Thus, q_m denotes the measured arm-side actuator position, which is obtained by scaling the motor position measurement by the gear ratio. Similarly q_r is the arm-side reference position of the actuators, which is in practice sent to the robot controller as motor-side reference scaled by the gear ratio. The actuator reference q_r is obtained from the tool position reference z_r via the inverse kinematics. With the forward kinematics and the measured actuator positions q_m, an estimate z_c of the tool position is obtained. This estimate is valid for low velocities, where the robot's mechanical flexibility has only a minor influence. The two external sensors, length gauges and accelerometer, output the measured tool acceleration a_m and position z_m. Due to the limited number of length gauges, z_m is only two-dimensional. The sensor input and actual tool position z, in grey in Figure 7.3, is not known.

Experiments

Trajectory As the tool position could only be measured within a square of 30 mm×30 mm in the xy-plane, a horizontal motion within a 10 mm square was chosen as reference trajectory. Figure 7.5 shows the rectilinear tool path. The motion starts at the indicated point in negative y-direction. To excite the robot flexibilities maximally, the velocity reference was set to zero in the corners of the tool path and the velocity was chosen as high as possible. With a nominal reference tool velocity of 100 mm s^{-1}, the velocity profile created by the IRC5 path generator reached a maximum tool velocity slightly below 100 mm s^{-1}.

Table 7.1 Signal description.

Signal	Description
q_r	Actuator position reference (arm side)
q_m	Measured actuator position (calculated to arm side)
z_r	Tool position reference
z	Tool position
z_c	Calculated tool position
z_m	Measured tool position
a_m	Measured tool acceleration

The ILC algorithm is not applied to the entire movement, but only to a central part of the trajectory which is preceded by a so-called lead-in part and followed by a lead-out part. This decreases the influence of varying initial conditions. Even though a pause to calculate the new ILC signal guarantees that no oscillations are propagated between iterations, and the motors are carefully reset to the same initial position before each iteration, the motor torque profile occasionally varied when the robot motion was started from stand-still. Lead-in is as well useful in practical applications, e.g., laser cutting, where a constant tool velocity along the path is important. The potential of lead-in is however not fully used here, as the reference velocity in the corners is zero. To have the ILC algorithm cover the complete rectilinear figure, the path is traversed in two consecutive rounds. The length of the lead-in and lead-out parts is chosen as $100 \cdot T_s = 0.4\,\text{s}$. The time-domain weighting of the ILC input signal during lead-in and lead-out is performed with a weighting vector (Figure 7.4) with coefficients taken from a 200-point Tukey window with $\alpha = 0.95$ according to the definition in [Harris, 1978].

ILC algorithms applied The ILC algorithms applied modify the actuator position and velocity references sent to the axis controllers. It would as well be possible to modify directly the motor torque reference, but as it is considerably more intuitive to anticipate instabilities by examining the modified position reference, this solution was chosen. The ILC algorithm determines a modified actuator position reference, which is then differentiated and added to the actuator velocity reference.

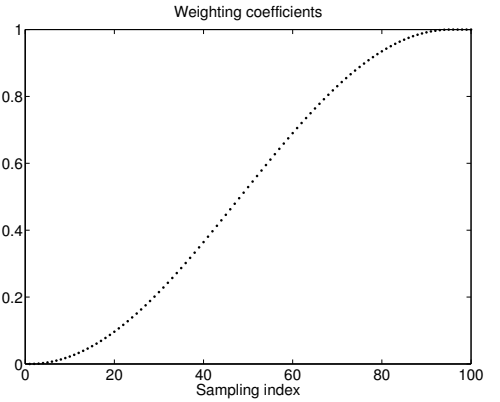


Figure 7.4 Weighting coefficients for time-domain weighting of the ILC input signal.

ILC algorithms based on three different signals are compared:

1) Actuator position measurements, $\epsilon_k = q_r - q_{m,k}$
2) Estimates of tool position, $\epsilon_k = z_r - \hat{z}_k$
3) Measurements of tool position, $\epsilon_k = z_r - z_{m,k}$

where ϵ_k is the error in (7.2), $q_{m,k}$ the measured actuator positions and $\hat{z}_k$ and $z_{m,k}$ the estimated and measured tool position, respectively, in iteration k. The tool position estimates for Case 2 are obtained using

A) Complementary filtering according to (7.17)
B) Kalman filtering according to (7.22)

Evaluation The performance of the ILC algorithms is evaluated by the control error at iteration k in relation to the initial error at iteration $k = 0$, when no ILC is applied.

For ILC based on actuator position measurements, the initial error is

$$e_{q,0} = q_r - q_{m,0} \tag{7.23}$$

Similarly the initial control error of the tool position is

$$e_{z,0} = z_r - z_{m,0} \tag{7.24}$$

where $e_{q,k} \in \mathbb{R}^3$ is a three-dimensional vector with values for all three actuators, while $e_{z,k} \in \mathbb{R}^2$ is a two-dimensional vector containing the x- and y-components of the error.

The 2-norm reduction of the error at iteration k is given in percentage of the initial error, as in

$$\bar{e}_{n,k} = 100 \cdot \frac{\|e_{n,k}\|_2}{\|e_{n,0}\|_2} \quad [\%] \tag{7.25}$$

where n symbolizes actuator (q) or tool position (z), respectively. $\bar{e}_{n,k}$ is based on the trajectory part where the ILC algorithm is fully applied, i.e., the lead-in and lead-out parts where the ILC input signal is weighted are not included.

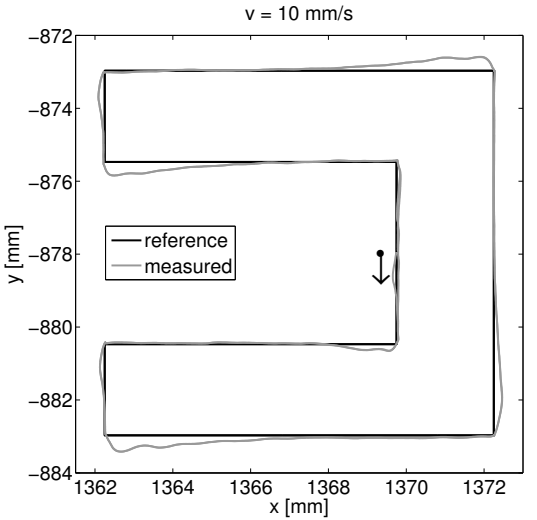
System Performance

In the following, the control performance of the robot system without ILC and the repeatability are studied.

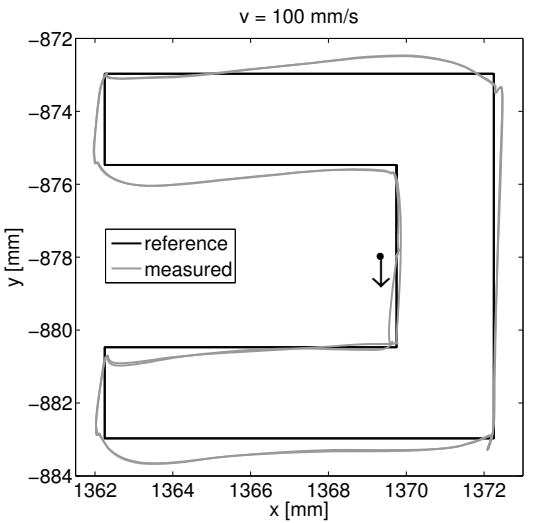
Control error Figure 7.5 compares the control performance when the robot traversed the reference path, first with a very low nominal velocity of $10\,\mathrm{mm\,s^{-1}}$ and then with the higher nominal of $100\,\mathrm{mm\,s^{-1}}$ velocity chosen for the ILC experiments.

Despite the low velocity, Figure 7.5(a) shows that position errors remained, especially after passing a corner. Besides possible oscillations in the links, which might have been induced by the robot's acceleration, another possible explanation of the errors may be friction in the motors and drivelines. The oscillatory behaviour, which was especially pronounced in the lower left corner, could be explained by static friction between the length gauge sensor tip and the metal block attached to the robot.

At high velocity, the position error was larger. This can be explained both by dynamic effects, which are not compensated for, and a larger control error at motor level. As is discussed in Chapter 6, the robot is stiffer in the x-direction, while it has a dominant resonance at around 11.5 Hz in the y-direction in this operating point. This coincides with the tendency of the overshoot after corners in x-direction being smaller than in y-direction. It can as well be seen in Figure 7.5(a) that the robot motion differed between the two consecutive turns the path was traversed. The differences are visible for the edge along which the motion is started and the succeeding one. For the edge where the motion starts, the reference velocity profiles differ clearly between the turns, which explains the differences. The following edge has however the same reference velocity profile in both turns, and the resulting differences confirm the necessity of adding a lead-in phase to the ILC experiment.



(a) Low velocity



(b) High velocity

Figure 7.5 Control performance of the robot for low ($v = 10\,\mathrm{mm\,s^{-1}}$, 7.5(a)) and high ($v = 100\,\mathrm{mm\,s^{-1}}$, 7.5(b)) velocity. The motion starts next to the indicated point in negative y-direction. Reference z_r (black) and measured tool position z_m (grey).

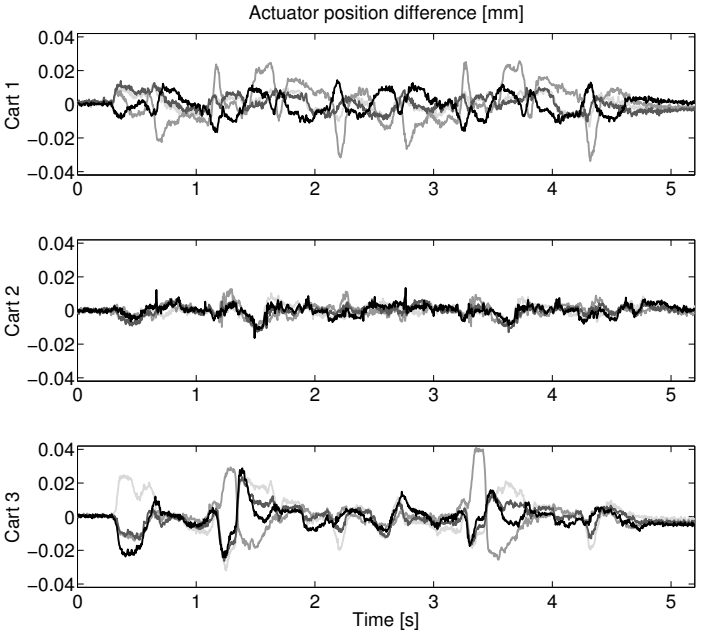


Figure 7.6 Repeatability experiments: Actuator position differences between iteration $k = 0$ and iterations $k = 2 \ldots 5$ according to (7.26). The curve for each iteration $k = 2 \ldots 5$ is plotted in a different shade of grey.

Repeatability A good dynamic repeatability is crucial for ILC experiments, since the ILC algorithm can only correct the repeatable error part. Five identical experiments were performed to evaluate the repeatability.

Figure 7.6 illustrates the actuator position differences between experiment 1 and j as

$$e_{\text{rep}}(t) = q_{m,1}(t) - q_{m,j}(t), \quad j = 2,\ldots,5 \tag{7.26}$$

As can be seen, the non-repeatable error parts are small compared to the complete control error, which is an order of magnitude larger, see compare Figure 7.5. A larger part of the control error can thus possibly be eliminated by ILC. Static friction might explain the non-repeatability partially. Different motor torques observed when standing still in the same robot configuration support this explanation. The controllers require then a varying amount of time to integrate the torques to overcome the static friction.

7.4 Results

Below, the ILC experiment results are summarized. ILC based on actuator position measurements (Case 1), tool position estimates (Case 2A and Case 2B) and tool position measurements (Case 3) are compared. Even though it is difficult to measure the tool position in industrial applications, Case 3 was performed to find a lower limit on the control error given the experiment setup and ILC algorithm (7.2) and supposing perfect estimates. The ILC experiment based on actuator position measurements was performed five times to assess repeatability. Besides an evaluation of the resulting control error according to (7.25), the stability of the ILC algorithm and the choice of filter Q in (7.2) are considered in each case. The filter L in (7.2) is for all experiments chosen as $L = \gamma q^{\delta}$. The time shift $\delta = 5$ and the learning gain $\gamma = 0.9$ are chosen based on the system's time delay and static gain.

Case 1: ILC Based on Actuator Position Measurements

Algorithm and stability To each of the three actuators, an ILC algorithm according to (7.2) is applied. The error $\epsilon_k(t)$ is based on the actuator positions and references, so that (7.3) becomes

$$\epsilon_k(t) = q_r(t) - q_{m,k}(t)$$

In this case, the measured variable y_k is the same as the controlled variable z_k. The system description (7.1) consists of only one equation

$$q_{m,k}(t) = T_{q_r q_m}(q) q_r(t) + T_{u q_m}(q) u_k(t)$$

and the estimation of the controlled variable reduces to $q_m = q_m$, i.e., $F_r = F_u = 0$, $F_y = 1$ in (7.4). The ILC input signal u_k is added to the actuator references q_r, so that $T_{q_r q_m} = T_{u q_m}$. As the three actuator models (6.11) are similar to each other, the same filter Q is applied to all actuators. The causal filter $\bar{Q}$, which is applied forward and backwards to give zero-phase characteristics as explained in Section 7.2, is chosen as a second-order Butterworth filter with cutoff frequency $f_n = 10\,\text{Hz}$.

After transforming required filters to matrix form, the criterion (7.16) for stability and monotone convergence

$$\bar{\sigma}_i = \bar{\sigma}\big(Q(I - L T_{u q_m, i})\big) < 1 \tag{7.27}$$

can be evaluated resulting in the following maximum singular values

$$\bar{\sigma}_1 \approx 0.90, \quad \bar{\sigma}_2 \approx 0.91, \quad \bar{\sigma}_3 \approx 0.93 \tag{7.28}$$

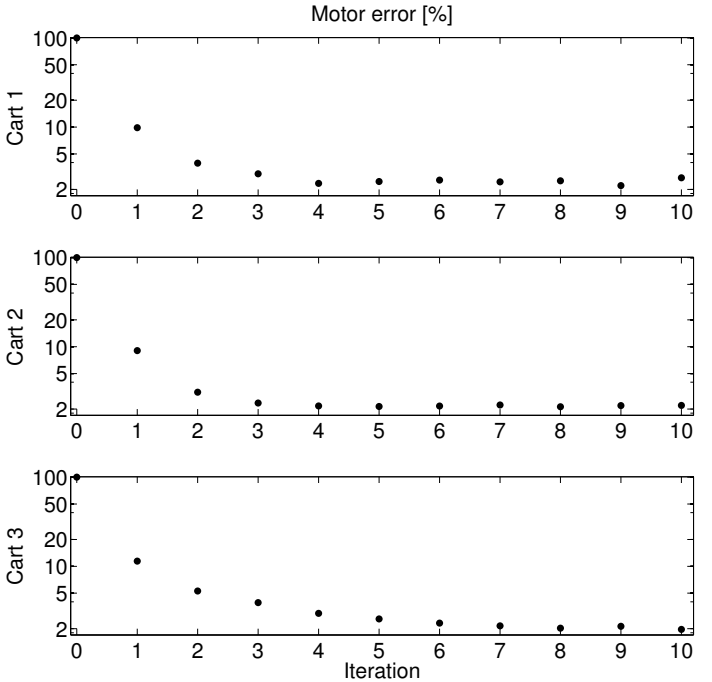


Figure 7.7 Case 1: Actuator control error (7.25)

for actuators 1 to 3. The matrices include the trajectory part fully corrected as well as the lead-in and lead-out parts. As the maximum singular values are below 1, the ILC algorithm is stable and converges monotonically.

Results Figures 7.7 and 7.8 show the resulting actuator control errors when applying the ILC algorithm for 10 iterations. Figure 7.7 shows that the relative error (7.25) is reduced to about 2 % of the initial error after five iterations, similarly for all 3 actuators. Figure 7.8 compares the initial error to the error at iteration 10.

The fast convergence is important in industrial applications, where time constraints are present. A small effort giving a substantial error reduction after only a few iterations is often sufficient.

The corresponding tool performance is shown in Figures 7.9 to 7.11. Figure 7.9 shows the tool error $z_r(t) - z_m(t)$ for iterations $k = 0$ and 10. Even though the error is visibly reduced, especially in x-direction, a comparison with Figure 7.8 shows that the tool error decreases less than on the motor side. This can also be seen in Figure 7.10, which shows the tool path in the xy-plane for iteration $k = 10$. While the actuator posi-

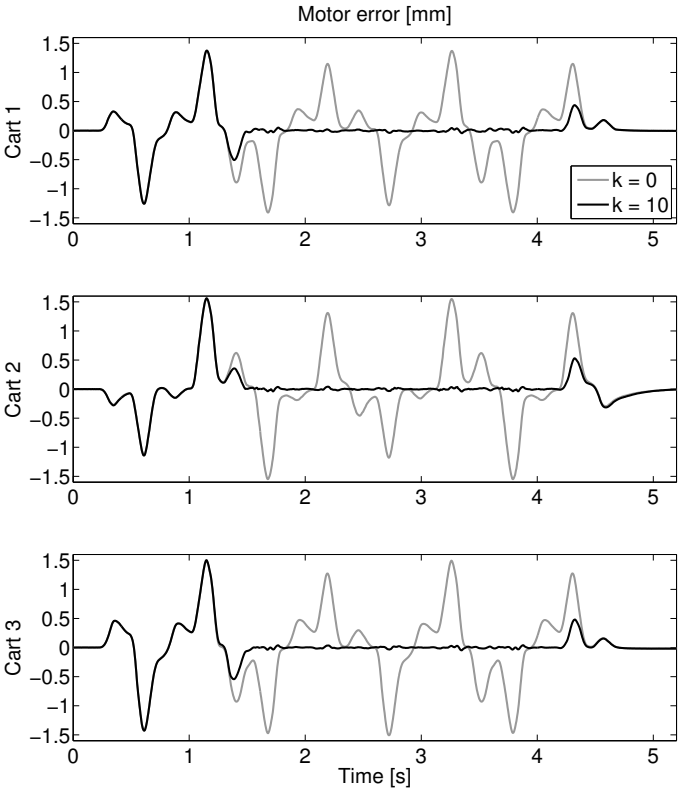


Figure 7.8 Case 1: Actuator control error after 10 iterations (black) compared to the initial error (grey). The ILC algorithm is fully applied between $t = 1.6 - 4.1\,\mathrm{s}$, preceded by a lead-in phase and followed by a lead-out phase.

tions transformed by the forward kinematics follow the reference closely, the measured tool position exhibits overshoots, mostly in y-direction. Figure 7.11 shows the relative error for iterations 0 to 10 on the tool side. The error in x-direction being reduced to 13 % of the initial error, whereas the error in y-direction is only decreased to 36 %.

The comparison between Figures 7.5(a) and 7.5(b) indicates that the influence of compliance dynamics is relatively small for low velocities, while it is important for the high velocity chosen for the ILC experiments. Thus, the tested ILC algorithm shows good results on the motor side, and would probably show results with a comparable accuracy on the tool side if a lower velocity would have been chosen. Using the actuator positions can be interpreted as using the estimate $\hat{z}_c$, which is an accurate estimate

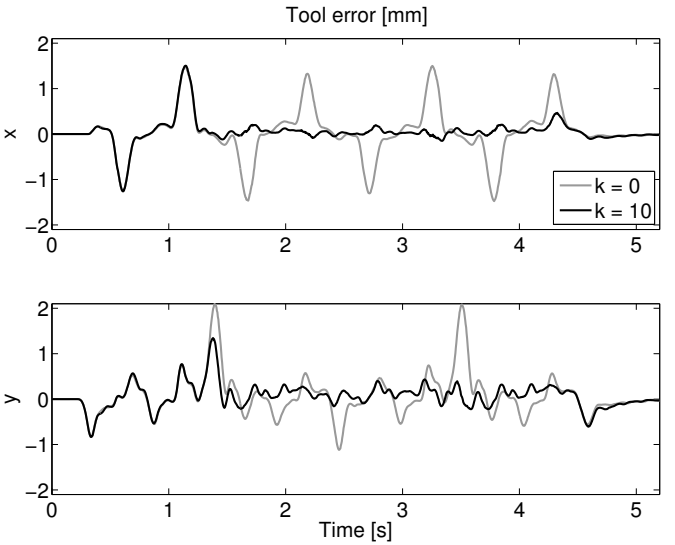


Figure 7.9 Case 1: Tool-side error after $k = 10$ iterations compared to the initial error for $k = 0$.

for low velocities. However, the elasticities in the link structure cause a dynamic tool behaviour that is not observable from the actuator position measurements and cannot be compensated for with the current ILC algorithm. As the robot is much stiffer in x-direction than in y- and z-direction, the error in x-direction is corrected to a higher degree than in y-direction. Kinematic errors can neither be observed from actuator measurements nor be corrected with the current ILC algorithm. Therefore, for further tool position improvements, more information on the tool position, either by measurement or observation, is needed.

Repeatability An ILC experiment was repeated five times based on the same iteration $k = 0$, i.e., each iteration $k = 1$ was performed with the same u_1 based on the same ϵ_0. The resulting relative errors are shown in Figure 7.12 for the motor side and in Figure 7.13 for the tool side. The spread shown in Figure 7.12 may explain the slightly non-monotone convergence seen e.g., in Figure 7.7.

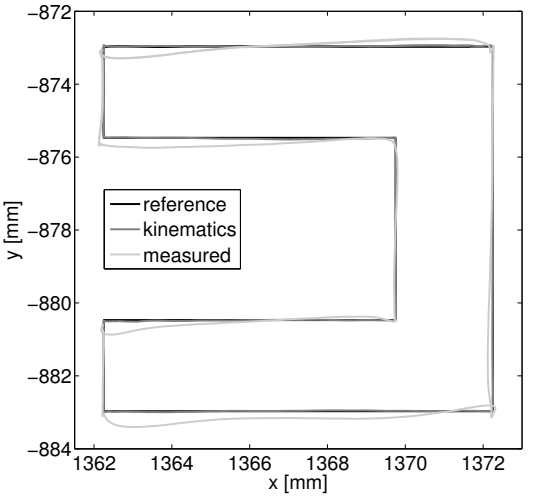


Figure 7.10 Case 1: Tool performance at iteration 10. Tool reference path (black), actuator positions transformed by forward kinematics (dark grey) and tool position measurements (light grey).

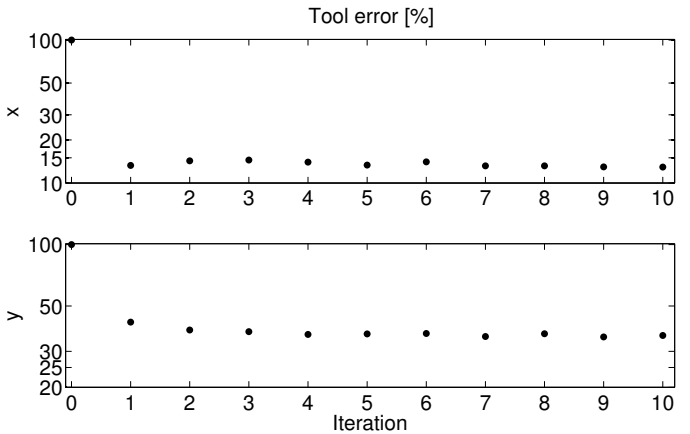


Figure 7.11 Case 1: Error (7.25) on tool side.

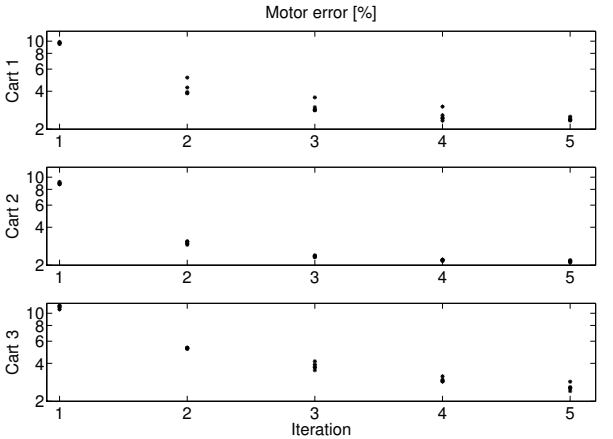


Figure 7.12 Case 1: Spread of error measure (7.25) on motor side for five experiments with identical inputs at first iteration.

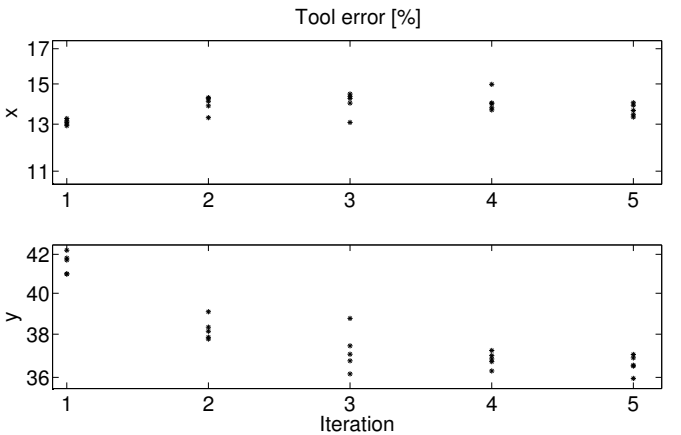


Figure 7.13 Case 1: Spread of error measure (7.25) on tool side for five experiments with identical inputs at first iteration.

Case 2: ILC Based on Tool Position Estimates

The ILC algorithm presented here is based on tool position estimates of a complementary filter (Case 2A) and a Kalman filter (Case 2B) using accelerometer and motor position measurements.

Algorithm and stability The ILC algorithm is applied on the tool side, i.e., the error $\epsilon_k(t)$ is now based on the tool position estimate $\hat{z}_k(t)$ in x- and y-direction:

$$\epsilon_k(t) = z_r(t) - \hat{z}_k(t)$$

The system description (7.1) consists of equations for the measured variable $y_k(t) = x_a(t)$, the tool acceleration, and the controlled variable $z(t)$, the tool position. The estimator (7.4) is not as trivial as in the previous case, where F_r, F_u and F_y are according to the observer descriptions (7.17) and (7.22).

The tuning of the filter Q in (7.2) is based on the identification experiments in Section 6.3. The robot has a distinct resonance at 11.5 Hz in the y-direction and a not so pronounced resonance at 7.4 Hz in the x-direction. If the filter $\bar{Q}$ is chosen as a low pass filter like in the previous case, the cutoff frequency has to be below the resonance frequencies of the closed-loop system to give a stable algorithm. This follows from the convergence analysis in frequency domain as described in e.g., [Norrlöf and Gunnarsson, 2002]. To correct also for errors at higher frequencies, Q is designed in a different way. The filter Q is tuned to be robust to large model errors especially around the resonance frequencies. Due to high frequency measurement noise, learning up to 30 Hz is chosen, which is above the lowest resonance frequencies of the system. The choice of Q is experimentally evaluated for both Case 2A and Case 2B to give a good error reduction. In Figure 7.14, the magnitude $|\bar{Q}^{-1}|$ is shown together with the magnitude $|1 - L(F_r + F_y T_{uy})|$ for the x- and y-direction, respectively. The robustness around the resonance frequencies of the system can be seen, together with the low-pass characteristics of the filter $\bar{Q}$ for higher frequencies.

The stability and convergence criterion (7.16) results in

$$\bar{\sigma}_{\text{Case 2A}} \approx 0.86$$
$$\bar{\sigma}_{\text{Case 2B}} \approx 0.93$$

Results In Figure 7.15 the tool performance can be seen for Case 2A. Similar results are achieved for Case 2B, and these figures are therefore omitted. Figure 7.16 and Table 7.2 summarize the resulting relative error (7.25) for all ILC approaches. It can be seen that the error on the tool

Table 7.2 Error reduction on the tool side: Mean value of the tool-side error in x- and y-direction for iterations $k = 5$ to $k = 10$ relative to the initial value for $k = 0$ as in (7.25).

	Case 1	Case 2A	Case 2B	Case 3
x	13.3 %	12.6 %	13.3 %	10.3 %
y	36.1 %	23.9 %	25.2 %	24.9 %

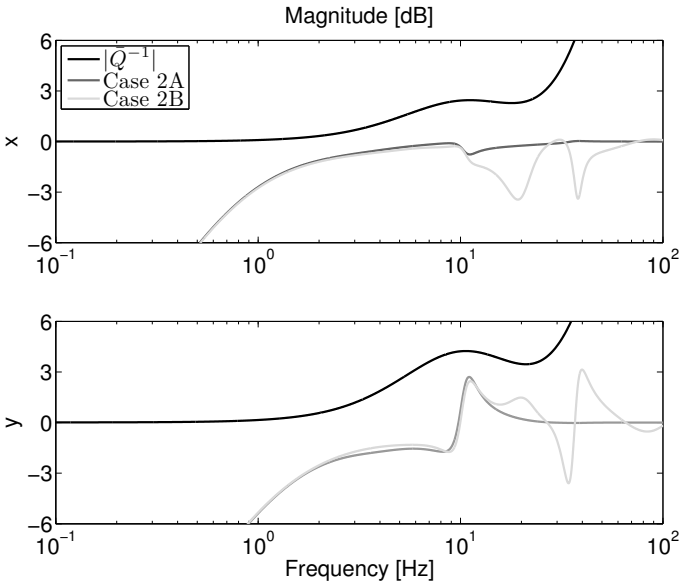


Figure 7.14 Case 2A, 2B: $|\bar{Q}^{-1}(e^{i\omega})|$ is illustrated together with the magnitude $|1 - L(e^{i\omega})(F_r(e^{i\omega}) + F_y(e^{i\omega})T_{uy}(e^{i\omega}))|$ in the x- and y-direction respectively.

side was decreased by using tool position estimates instead of actuator position measurements in Case 1. However, it is difficult to compare the approaches quantitatively, as different ILC algorithms were tuned and applied to different systems. Another tuning could for example give a slightly smaller error in x-direction and result in a larger error in y-direction. It can as well be seen that the improvement from Case 1 to Case 2 is larger in y-direction than in x-direction. This is explained with the robot being stiffer in x-direction, so most of the errors in the x-direction can be compensated for by actuator measurements. For the y-direction, the accelerometer signal can give more information about the tool position than from the actuator measurements, which improves the performance in that direction to a higher degree.

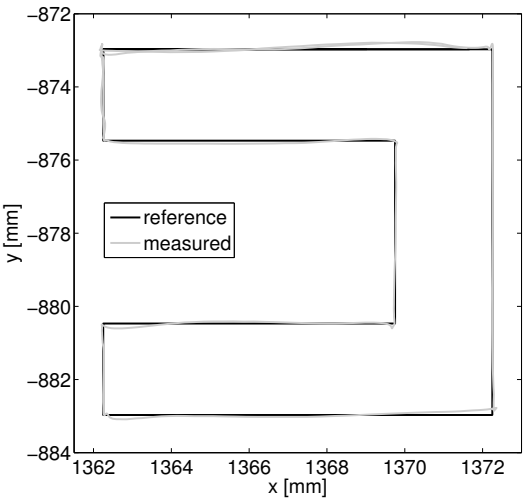


Figure 7.15 Case 2A: Tool performance after 10 iterations, reference path (reference, black) and measured tool position (measured, grey).

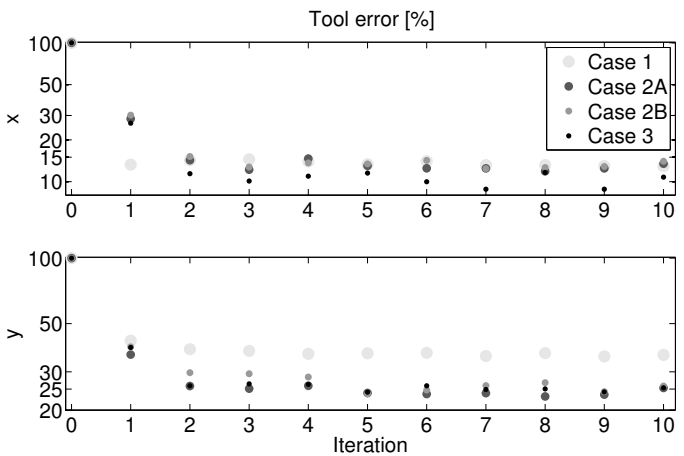


Figure 7.16 Relative error (7.25) on tool side.

Case 3: ILC Based on Tool Position Measurements

Algorithm and stability Like in Case 1, the measured and controlled variables are the same, but here they are the tool position in x- and y-direction. The error $\epsilon_k(t)$ is

$$\epsilon_k(t) = z_r(t) - z_{m,k}(t)$$

The estimator (7.4) simplifies to $\hat{z}_k(t) = y_k(t) = z_k(t)$ with $F_r = 0$, $F_u = 0$ and $F_y = 1$.

The filter Q is designed in the same manner as in Case 2. It is robust to modeling errors, especially around the resonance frequency, and corrects for errors up to 30 Hz. The tuning is based on the model identified in Section 6.3, which describes the closed-loop system from tool reference to tool position. The filter is experimentally evaluated to give a good error reduction. In Figure 7.17, the inverse $|\bar{Q}^{-1}|$ is shown together with the relation $|1 - L(F_u + F_y T_{uy})|$ for the x- and y-direction, respectively.

The criterion (7.16) for monotone convergence of the ILC algorithm results in

$$\bar{\sigma}\big(\boldsymbol{Q}(I - \boldsymbol{L}\boldsymbol{T}_{uz})\big) \approx 0.95 < 1$$

Results The resulting performance of the robot tool is shown in Figure 7.18. In Figure 7.16 and Table 7.2 it can be seen that the relative error (7.25) is slightly improved when using tool measurements instead of estimates. As mentioned previously, it is difficult to compare the experimental results for the different approaches quantitatively.

7.5 Conclusion

Three different ILC approaches were experimentally evaluated on the Gantry-Tau robot. The ILC algorithm was based on three different signals: Actuator position measurements, tool position estimates and tool position measurements. The tool position estimates were derived by both a complementary filter and a Kalman filter. In the experiments, the tool position was also measured, but only used for evaluation.

The experiments prove that the tool performance can be improved using estimates of the tool position instead of the standardly available motor measurements. The used additional sensor, an accelerometer mounted at the end-effector plate, is of low cost and does not obstruct the industrial application. This would however be the problem for the third ILC approach, which was evaluated with the purpose to find a lower limit on

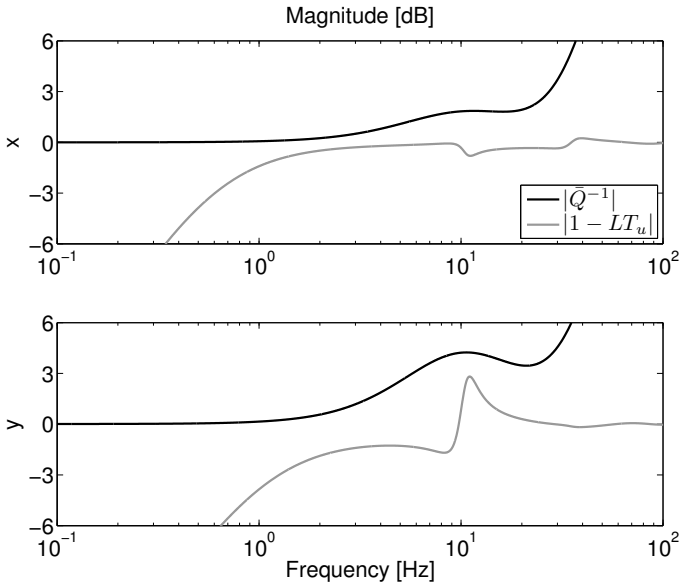


Figure 7.17 Case 3: Illustration of choice of filter $\bar{Q}$, when tuned based on the model identified. The inverse $|\bar{Q}^{-1}|$ is illustrated together with the relation $|1-LT_{uz}|$ in x- and y-direction, respectively.

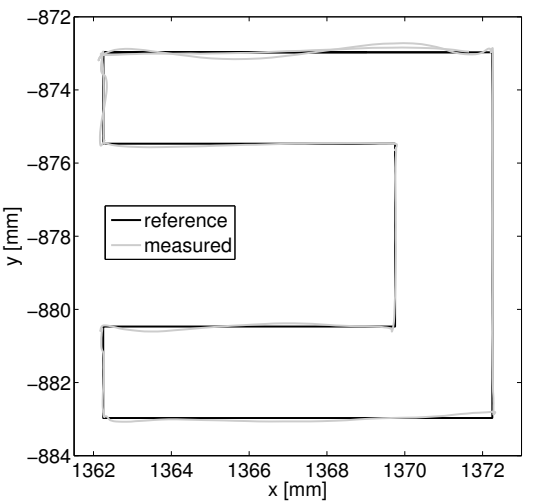


Figure 7.18 Case 3: Tool performance after 10 iterations.

the resulting accuracy with the given algorithms and robot system than in view of an industrial application.

The test trajectory was traversed at a relatively high velocity with respect to the robot's resonance frequencies, and the tuning of the ILC filters included learning above the resonance frequencies of the robot. The performance in the experiments is limited rather by the flexible behaviour of the robot than by the ILC algorithm in connection with the tool position observer.

8

Lead-Through Programming with Tool Force Feedback

8.1 Introduction

Robot programming today is mostly a cumbersome task requiring knowledge of robotics and robot manufacturer specific programming languages. Robots are to a large part utilized in large size companies, performing the same programmed movement for very long time spans, as e.g., a welding robot in automobile industry. To be able to use robots in SMEs, robot programming needs to be faster and more intuitive, so that it can be performed by SME staff and and which entitles robots to be used also for small lot size production applications. Lead-through programming, i.e., using force feedback to manually guide the robot tool along a desired trajectory or to certain distinct points in the workspace, is an intuitive and fast way of robot programming.

However, using just one force sensor for guiding the robot along the desired trajectory may lead to varying forces between tool and work object or even to occasionally losing contact with the work object. A remedy to this is the usage of a second force sensor for feedback and control of the tool force. Controlling and thus limiting the tool force with lead-through programming protects the tool and work object and enables a smoother guiding along the desired trajectory. As a deformation of the tool is avoided in this way, a higher accuracy can be achieved, e.g., in applications where lead-through is used for measurements or calibration of the workpiece position.

8.2 Related Work

An overview of force control is given e.g., in [Yoshikawa, 2000] and [Siciliano and Villani, 1999], where a distinction is made between direct and indirect force control. A very common approach of indirect force control is impedance control [Hogan, 1984]. Direct force control is often combined with position control in a hybrid approach like presented in [Craig and Raibert, 1979].

Several approaches to ease robot programming using force control and human demonstration have been implemented in the past. In [Asada and Izumi, 1989] it is described how position and force of the tool are measured when an operator performs a task. The measured data is then processed into a robot program for hybrid positon/force control, or as in [Asada and Asari, 1988], where an impedance control law is identified from the measurements. In this case, the tool is not mounted on the robot during demonstration.

In [Hirzinger and Landzettel, 1985] a flexibly applicable system for robot learning is described. A sensor ball [Heindl and Hirzinger, 1983] containing a force/torque sensor, which can optionally be mounted on the robot, is used to move the robot. The robot is equipped with a second force sensor mounted on the wrist. The wrist sensor can be used to control or to record the tool force, but it is unclear whether the authors controlled the tool force in the teaching phase. The wrist force sensor can be exchanged, e.g., with an inductive distance sensor, whose data is then transformed to a corresponding “pseudo-force” measurement. No experiment data is presented.

In [Wang *et al.*, 2008] and [Pan and Zhang, 2008], lead-through teaching is used to record a few characteristic points on the path to program. In a second phase, the complete path is recorded in a contour following process, in which the recorded points give shape information to better cope with edges etc. In [Pan and Zhang, 2008], in the application after teaching, the robot deformation due to the machining forces is compensated for to increase accuracy. In [Pan and Zhang, 2008], it is mentioned that changing the tool orientation with lead-through teaching was difficult and therefore only the translational DOF were changeable by lead-through, the orientation was changed by jogging the robot using the joystick. In [Qi *et al.*, 2009], a Stewart-platform-based measurement device is used instead of a force/torque sensor for lead-through teaching. The robot follows the motion exercised on the so-called RoboPuppet such that it keeps a constant configuration. In [Grunwald *et al.*, 2003] a lightweight robot with implemented lead-through teaching is presented. The torque sensors added to all robot joints are used for lead-through teaching instead of a wrist mounted force/torque sensor. Today, lead-through teaching is

available as a standard option for many industrial robot systems, e.g., [Ge *et al.*, 2008].

The idea to use both a tool force sensor and an additional force sensor for human/robot co-manipulation is attractive also for medical applications. [Kumar *et al.*, 2000] implemented a one-dimensional tool force control, where the desired tool tip force is scaled down from the tool handle force input of the human operator. [Cagneau *et al.*, 2008] present force amplification control of a 4 DOF surgical robot giving the surgeon force feedback from the force between tool and organ. [Lamy *et al.*, 2010] transfer the idea of force amplification to other applications where the robot assists a human in tasks where large forces are needed.

In robotics, sensor fusion of force and other signals is an interesting topic. [Hosoda *et al.*, 1998] and [Xiao *et al.*, 2000] combine the force measurement with visual servoing and implement a hybrid position/force control.

In the two master's theses [Friman, 2010] and [Gustafsson, 2008], lead-through programming using two force sensors was implemented. In [Gustafsson, 2008], impedance control on a force value which is a combination of the two sensor measurements is performed. In [Friman, 2010], the control switches between impedance control using the lead sensor measurement and direct force control of the tool force measurement with a PI controller. Both master's theses consider only 3 translational DOF.

[Yoshikawa and Sudou, 1993] extended the hybrid position/force control with an online estimation of an unknown constraint based on force and velocity measurements corresponding to the directions of normal and friction force, respectively. An example for the recognition of different contact types can be found in [Meeussen *et al.*, 2007].

8.3 Control Concept

Similar to hybrid position/force control, the basic control concept is to separately control subspaces of *SE(3)* with different controllers utilizing feedback from different sensors. Instead of visual and force feedback as in [Hosoda *et al.*, 1998], two force sensors are used.

The sensor placement is according to Figure 8.1. As follows the reasoning in [Lamy *et al.*, 2010], this force sensor placement is the best suitable for this application, as the respective sensors measure directly the forces between the robot and operator and tool, respectively.

Figure 8.2 gives a rough overview of the lead-through control structure. Typically, at the beginning of a lead-through teaching procedure, the robot tool is not in contact with the workpiece. The operator moves the tool towards the workpiece, performs the desired trajectory on the workpiece

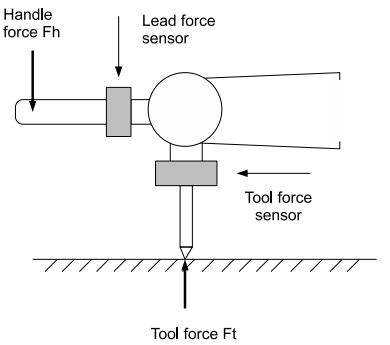


Figure 8.1 Force sensor placement

after taking contact, and then moves the robot back to a position in free space. Lead-through programming in free space requires different control compared to when the robot tool is in contact with the workpiece surface. As illustrated in Figure 8.2, the contact state, i.e., if the tool is in contact with the workpiece, determines which control law to use. Note that it is possible to have other than the two illustrated states if the tool and workpiece are such that the type of geometric constraint changes along the desired trajectory. A classification of contact types can be found in e.g., [Meeussen *et al.*, 2007].

Based on the force measurements from the tool sensor, $F_t = [f_t, \tau_t]$, and the handle sensor, $F_h = [f_h, \tau_h]$, and the corresponding control laws C_t and C_h, the two tool velocity references V^h_{tool} and V^t_{tool} are calculated

$$V^h_{\text{tool}} = C_h(F_h) \tag{8.1}$$

$$V^t_{\text{tool}} = C_t(F_t) \tag{8.2}$$

The full 6 DOF velocity references V^h_{tool} and V^t_{tool} are then, depending on the contact state c, projected into the chosen subspaces of *SE(3)* by P_h and P_t and the sum of both signals is then applied to the controlled robot

$$V_{\text{tool}} = P_h(V^h_{\text{tool}}, c) + P_t(V^t_{\text{tool}}, c) \tag{8.3}$$

The control concept presented is independent of the actual, chosen controller implementations. In this work, a simple P controller (tool force control C_t) and an impedance controller without position term (handle

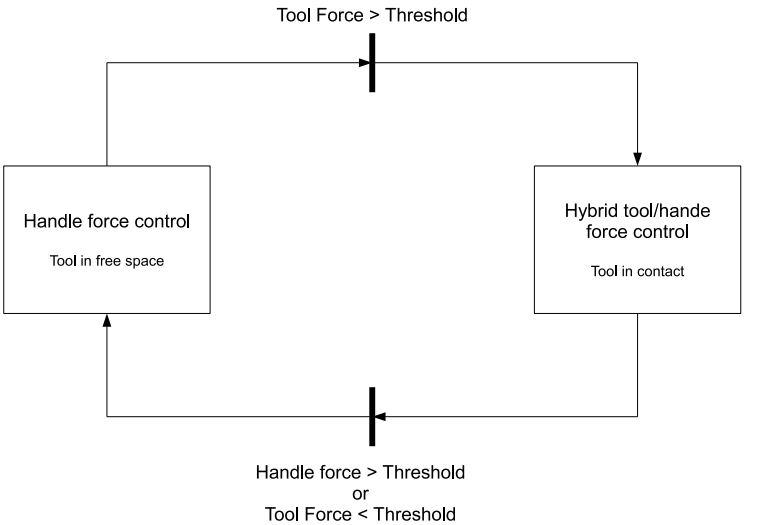


Figure 8.2 Lead-through control overview

force control C_h) were used to illustrate the concept.

$$V^t_{\text{tool}}(t_k) = K \cdot (F^{\text{ref}}_t - F_t(t_k)) \tag{8.4}$$

$$V^h_{\text{tool}}(t_k + 1) = \frac{T_s}{q-1}\ A^h_{\text{tool}}(t_k) \tag{8.5}$$

$$A^h_{\text{tool}}(t_k) = M^{-1}\,(F_h(t_k) - D \cdot V_{\text{tool}}(t_k)) \tag{8.6}$$

where K is the gain matrix of the P-controller, F^{ref}_t the tool force reference, $T_s = 4\,\text{ms}$ the sampling time and q the forward shift operator. The acceleration reference A^h_{tool} is integrated to obtain the corresponding velocity reference. M and D are mass and damping matrices, respectively. The applied tool velocity reference V_{tool} is used for velocity feedback rather than a measured real velocity value, as the bandwidth of the robot servo control was much faster than the lead-through control and the reference was considered a better estimate than the noisy joint velocity measurements transformed by the kinematics. As the impedance controller contains an integrator and the applicable velocity reference was limited for safety reasons, feedback of the applied velocity reference avoids wind-up and transient impulses with contact state changes. The modified impedance

controller was then

$$V^h_{\text{tool}}(t_k+1) = \frac{T_s}{q-1}\left(A^h_{\text{tool}}(t_k) - \frac{1}{T_s}(V^h_{\text{tool}}(t_k) - P_h(V^h_{\text{tool}}(t_k)))\right) \tag{8.7}$$

where the projection P_h includes as well a safety saturation of the signal.

In a typical case where lead-through programming is used, the geometric constraints of the tool are at most partially known. It is therefore desirable to estimate these constraints. Experiments showed that better results are obtained when friction tangential to the surface is not considered as in [Yoshikawa and Sudou, 1993], but only the normal force. The reason for this are low velocities with noisy measurements. Depending on the type of contact, the geometric constraint can be uniquely determined. An example is a point contact between a tool tip and one surface where the surface normal vector can be uniquely determined. In case of a tool tip constrained by more than one surface, e.g., a tool tip should be moved along a groove with triangular cross section, it is difficult to determine the constraints only by tool force measurements.

It is therefore not possible to implement one single controller (8.1)–(8.3) which is applicable for all possible cases. The projection laws P_h and P_t have to be manually adjusted for specific use cases, as will be demonstrated in the two example tasks in Section 8.4. As already mentioned in [Pan and Zhang, 2008] and which our experiments confirmed, orientation is more difficult to efficiently control by lead-through. We therefore only modified the orientation by the tool force controller.

8.4 Experiments

The force control concept presented was verified in two example applications. The first application of following a surface with a rolling tool was limited to the 3 translational DOFs. For a full 6 DOF application, the peg-in-hole problem was chosen.

Experimental Setup

The experiments were performed on a ABB IRB 140 robot controlled by the extended IRC5 system described in Chapter 3. The force sensors were mounted as shown in Figure 8.3. The tool sensor was a JR3 load cell [JR3 Inc, 2012], the handle sensor an ATI nano 25 [ATI Industrial Automation, 2012] sensor, both measuring force and torque in 6 DOF.

Surface Tracking

The first example task is to follow an unknown surface with a rolling contact point. An industrial application for this case might be measuring or

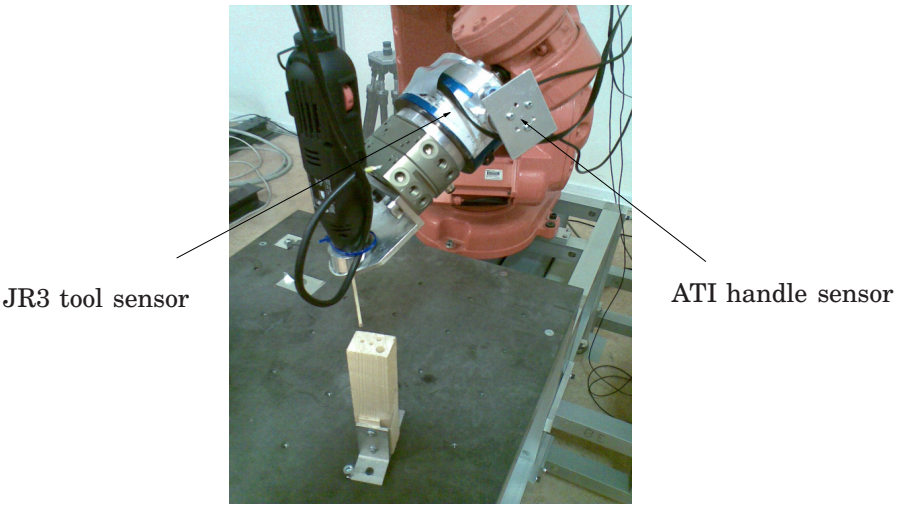


Figure 8.3 IRB 140 robot with tool and handle sensors (occluded by a plate representing the handle) and drilling tool with match as dummy drill

calibrating a workpiece, gluing or welding. For this application case, the robot was only controlled in 3 translational DOF, i.e., the tool orientation was kept constant. The surface-following functionality may be achieved using lead-through with an industrial robot system, but our concept provides separation of the operator's push-forces and the allowed contact and tool forces.

The contact plane can be defined by a point and a normal vector. While the point or positioning of the surface was not known to the lead-through controller, a nominal normal vector $n_n = (0,0,1)^T$ was assumed to be known. The lead-through controller estimated the normal vector during contact and took into account small variations from the nominal normal vector. Of course, the operator, forming the outermost control loop with the robot system, benefits from visual feedback which gives a rough estimation of the surface properties.

Surfaces with different properties were tested. For a safe comparison of lead-through programming with and without tool force sensing, a very compliant surface was chosen: A cardboard with rather compliant surface was chosen. To test the normal vector estimation, a metal plate was added on top of the box to give a more even and better defined surface.

The lead-through controller was implemented as follows: In case of contact, the tool force was controlled in the surface normal direction n, and the space orthogonal to the normal vector was controlled by the lead

force. The tool orientation was kept constant.

$$P_h\left(V^h_{\text{tool}} = \begin{bmatrix} v^h_{\text{tool}} \\ \omega^h_{\text{tool}} \end{bmatrix}\right) = \begin{cases} \begin{bmatrix} v^h_{\text{tool}} \\ 0 \end{bmatrix}, & \text{no contact} \\ \begin{bmatrix} v^h_{\text{tool}} - \left(n^T \cdot v^h_{\text{tool}}\right) \cdot n \\ 0 \end{bmatrix}, & \text{contact} \end{cases} \tag{8.8}$$

$$P_t\left(V^t_{\text{tool}} = \begin{bmatrix} v^t_{\text{tool}} \\ \omega^t_{\text{tool}} \end{bmatrix}\right) = \begin{cases} \begin{bmatrix} 0 \\ 0 \end{bmatrix}, & \text{no contact} \\ \begin{bmatrix} \left(n^T \cdot v^t_{\text{tool}}\right) \cdot n \\ 0 \end{bmatrix}, & \text{contact} \end{cases} \tag{8.9}$$

The tool force reference was set to $F_t^{\text{ref}} = 2 \cdot n\,\text{N}$. In this way, once contact is established between tool and surface, it is evenly kept while the tool can be moved in the surface tangential space by applying force on the handle. Contact can be released by applying a force larger than a specified limit in the surface normal direction.

To compensate for the change of the surface normal direction, the normal direction was estimated and corrections were made within an angle of φ_L from the assumed normal direction.

$$n = \begin{cases} n_{\text{est}} = \dfrac{f_t}{\|f_t\|}, & \arccos(n_n^T \cdot n_{\text{est}}) < \varphi_L \\ n_n, & \arccos(n_n^T \cdot n_{\text{est}}) > \varphi_L \end{cases} \tag{8.10}$$

In Figure 8.4a comparison of lead-through programming with and without tool force control, respectively, is shown. The correction range φ_l was 10° and a plain cardboard box was used as surface. Without tool force control, it is difficult to keep contact with the surface, while with tool force control, contact is kept and the force successfully controlled. Without tool force control, peaks in the tool force when taking contact reach up to 7 N. Of course the lead-through programming performance can be improved by a more skilled operator, but the advantage of the second sensor is evident.

In a second experiment, the normal vector estimation was studied. A metal plate was added on top of the cardboard box to achieve a more plane surface. At the same time, the range within which the surface's normal vector was corrected for was varied from 0° to 10°. As Figure 8.5 shows, oscillations with a frequency of 68 Hz appeared in contrast to the experiment with only the cardboard box. The oscillation amplitude was slightly increasing with increased correction range. For a 10° correction range, the oscillations were close to cause loose of contact.

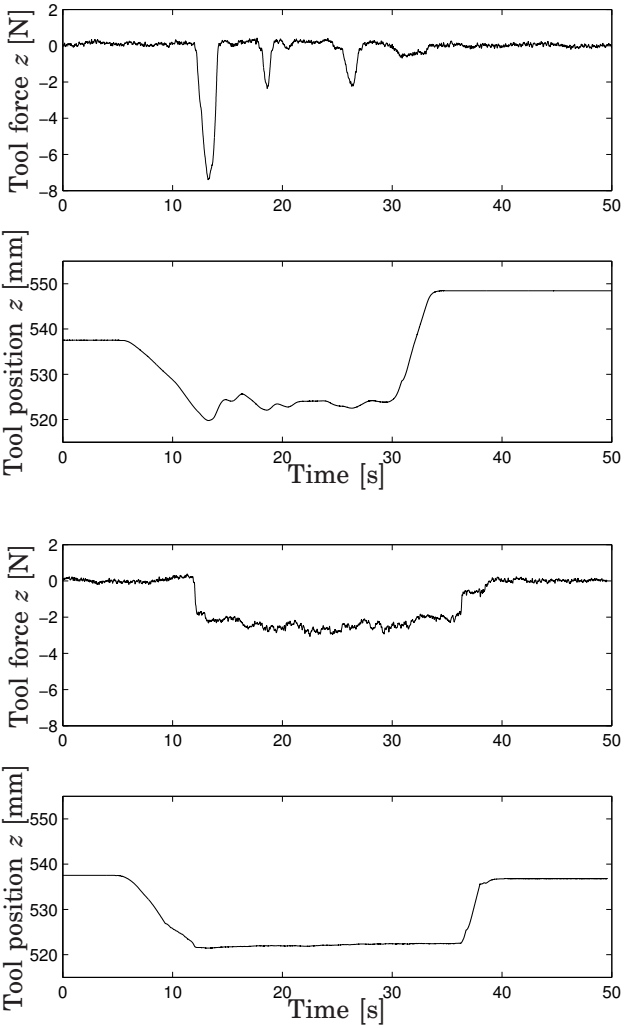


Figure 8.4 Surface tracking: Tool position and tool force in z direction. Without tool force control (upper) and with tool force control (lower).

Peg in Hole

Figure 8.3 shows the setup for this experiment. As a drill dummy, a 9.5 cm long match was used. The cross section of the match had 3 mm edges. The match was docked into a wooden block with holes in different sizes. The

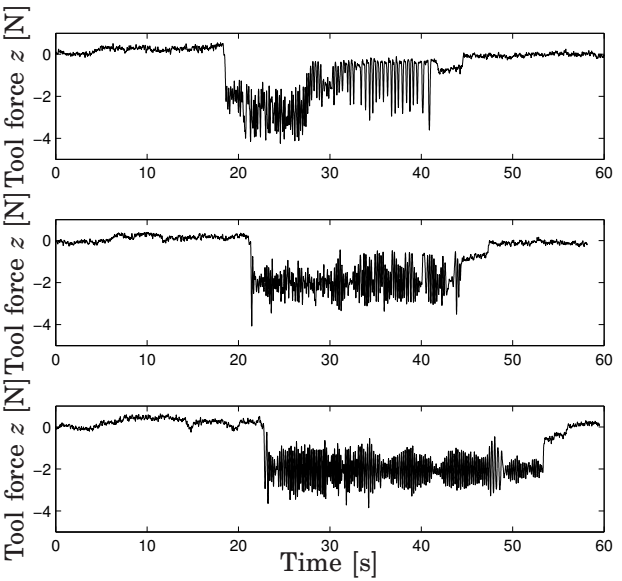


Figure 8.5 Surface tracking: Tool force in z direction. Range φ_L for correcting normal vector varying from nominal direction 10° (upper), 5° (center) and 0° (lower).

smallest hole, which was used for the data presented, had a diameter of 4 mm. Figure 8.6 shows the wooden block with a match used. The match length axis was misaligned with the hole's length axis.

Pre-experiments showed that is was difficult to efficiently control the tool orientation by lead-through. Therefore, the lead-through in this experiment was limited to the three translational DOF using the lead-through control law (8.8). The orientation is only controlled in contact state and then completely without projections by the tool force controller:

$$P_t\left(V^t_{\text{tool}} = \begin{bmatrix} v^t_{\text{tool}} \\ \omega^t_{\text{tool}} \end{bmatrix}\right) = \begin{cases} \begin{bmatrix} 0 \\ 0 \end{bmatrix}, & \text{no contact} \\ \begin{bmatrix} \left(n^T \cdot v^t_{\text{tool}}\right) \cdot n \\ \omega^t_{\text{tool}} \end{bmatrix}, & \text{contact} \end{cases} \tag{8.11}$$

The tool force/torque reference was set to 0 N and 0 N m, respectively. Unlike in the surface tracking case, where a continuous contact was desired, this leads to the tool being pushed away if it touches a geometric constraint. A consequence of this is that the hole has to be slightly larger than the peg in order not to be rejected from the hole. For the 3 mm match

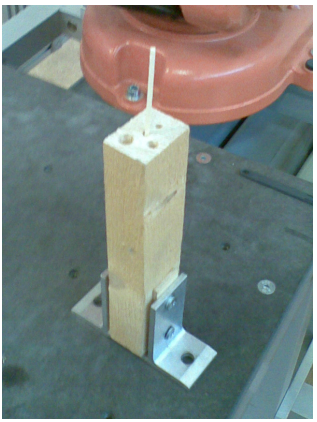


Figure 8.6 Wooden block and match used for peg-in-hole experiments

and 4 mm hole, no problems were encountered.

As a cylindrical tool is inserted in a cylindrical hole, the rotation around the length axis of tool and hole is not determined by the geometric constraint. With the current control implementation, the orientation of the robot tool around this axis is therefore arbitrary, or rather determined by measurement noise, during the experiment. This could be prevented in several ways with estimating the axis direction of the hole. The orientation around this axis can then be locked, specified or controlled by lead-through. However, the reorientation was not large and no problems arose during the experiments.

Figures 8.7–8.9 show the recorded data. Figure 8.7 shows the tool force and torque which is successfully controlled to zero exhibiting a few peaks of the force in z-direction when the tool is hitting the wooden block and the torque in y-direction, which is created by the misalignment of match and hole. Figure 8.8 shows the TCP trajectory as calculated by the joint positions and robot kinematics. It should be kept in mind here that the match is very compliant and the deviations from a straight vertical line are mostly due to a deformation of the match. Finally, Figure 8.9 shows the tool frame's ZYZ Euler angles. It can be seen that after the reorientation phase between $t = 20$ s and $t = 70$ s, when the tool is aligned with the hole, the changed orientation remains; it is not controllable by lead-through programming.

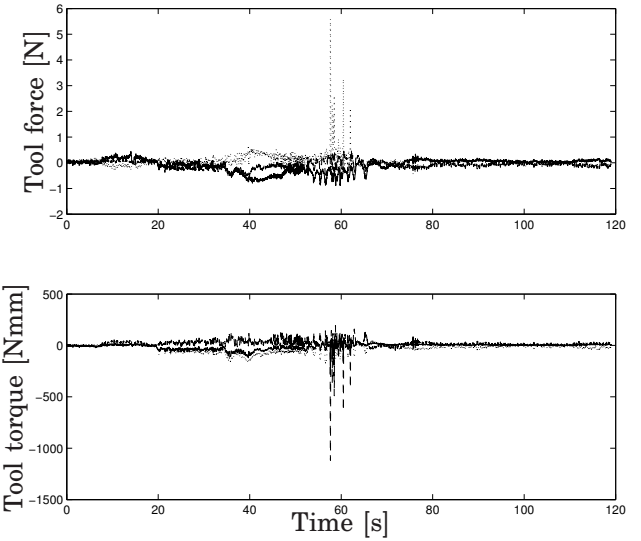


Figure 8.7 Tool force (upper) and torque (lower) in base frame x- (solid), y- (dashed) and z- (dotted) directions.

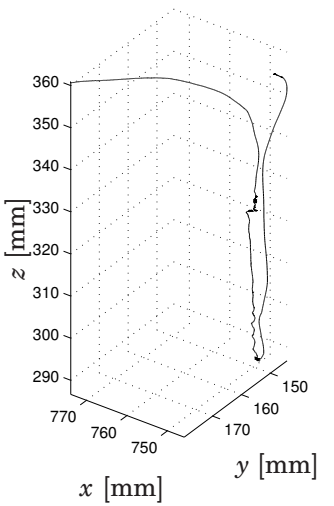


Figure 8.8 Tool trajectory in base frame as calculated by forward kinematics and joint positions. Deviations from a straight vertical line are mostly due to the deformation of the very compliant match.

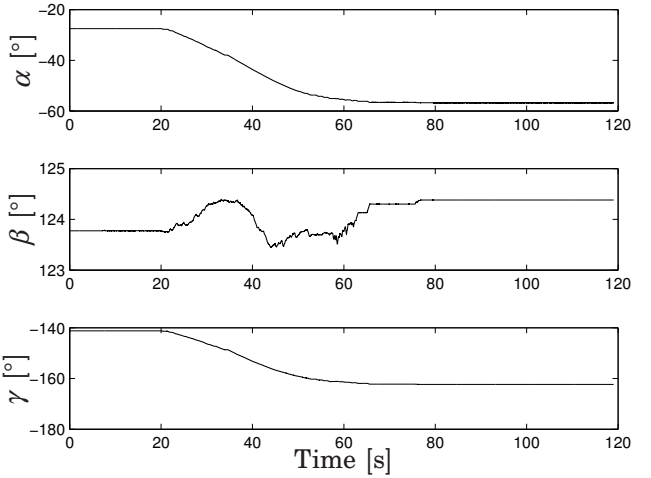


Figure 8.9 Tool frame ZYZ Euler angles.

8.5 Discussion

Lead-through control using tool force feedback from a second force sensor was successfully tested for two example use cases. The control concept was to use sensor fusion and to control different subspaces of *SE(3)* by different sensor feedback.

In the first example application of moving along a surface, contact with the surface was steadily kept with a contact force of 2 N, while the operator was moving the tool in the tangential space of the surface.

For the second test with a metal plate on top of the cardboard box, oscillations were observed. As the focus of the experiment was to prove the control principle and not to optimize a specific force control implementation, oscillations can be avoided by a force control implementation which is more robust to varying contact stiffness.

In a typical lead-through application, the geometric constraints are not perfectly known. In the surface tracking application, the height of the workpiece is detected by the contact force that is built up and successfully limited. The normal vector estimation was done according to simple state-of-the-art methods, only using the tool force measurement. [Yoshikawa and Sudou, 1993] used also the tool velocity to deduct the friction force from the measurement and obtain the pure contact force. Here, the raw velocity measurement was so noisy that the simpler method gave more accurate

results. The normal vector estimate is however still corrupted by the force measurement noise, which is the cause for the oscillations in Figure 8.5 increasing with increasing confidence in the estimated normal vector. By better signal filtering, a more advanced estimator or the additional usage of tool position data, the constraint estimation can be improved.

The surface tracking was performed only in 3 DOF with constant tool orientation. For surfaces with more curvature, e.g., a pump house, it would be possible to adjust the tool orientation so that it is orthogonal to the surface. For that, a better normal vector estimation is essential.

In the second example application, a match is successfully inserted into a hole, with match and hole being misaligned at experiment start. The difference of the diameters of hole and match leave around 1 mm space.

The tool trajectory in Figure 8.8 varies more than this 1 mm from a straight line. The reason for this is that the match is compliant and is deformed, while the tool trajectory is calculated by the robot kinematics from the measured joint positions.

The flexibility of the match increases the difficulty of the task, as the contact forces created are smaller than if a metal drill is used. The smaller forces lead to a lower signal-to-noise-ratio, which limits the control bandwidth.

A simple solution was chosen for the control of the orientational DOFs leading to one orientational DOF being undefined. This was not posing any problem in the experiment and can be solved by the remedies proposed in Section 8.4.

The advantage of lead-through programming compared to other robot assembly concepts is the human operator. He/she is the controller in the outermost control loop and has easy visual and sensual feedback and experience that cannot be put into a program or controller.

In the chosen examples, very simple contact scenarios were tested. Using e.g., a disc grinder instead of a rolling tool causes different geometric constraints. It is possible to demonstrate the task using a dummy tool. In case of demonstration with a real tool, the process forces need to be considered as well.

To accomplish a good 6 DOF lead-through control, accurate calibration of the force sensors and a high signal-to-noise-ratio of the force signals is important. These factors limit the bandwidth of the lead-through control, i.e., how fast the robot can be moved by the operator.

Other sensors for lead-through control, e.g., joysticks, have also been suggested in the literature, see e.g., [Hirzinger and Landzettel, 1985].

8.6 Conclusion

This chapter presented a lead-through programming concept using two force sensors and successfully demonstrated the control principle in practise by experiments for two different use cases. The control principle is to perform sensor fusion with the two force signals and to control, similar as in hybrid position/force control, different subspaces of *SE(3)* by projecting the sensor data to different subspaces.

9

Conclusions

9.1 Summary

The overall objective of this thesis is the development of robot concepts that fit the needs of SMEs. With the Gantry-Tau robot, a novel parallel kinematic robot was presented, which has the potential to fulfill the requirements of accuracy, mechanical stiffness and conceptual flexibility. A major part of this thesis deals with modeling and identification of the Gantry-Tau's kinematic and dynamic properties. Concepts that aid the SMEs to achieve the required accuracy and a more intuitive robot operation were developed and experimentally evaluated. The SMErobot project, an initiative joining several European robot manufacturers, research institutes and universities, aimed at developing such robot concepts and additional aspects were presented within the project [SMErobot, 2009]. The modularity aspects and the integration of software and hardware of the Gantry-Tau concept were then further studied in the MONROE project within the ECHORD project [MONROE, 2012; ECHORD, 2012].

Modeling

Both kinematic and dynamic modeling of the Gantry-Tau robot were presented. A previously known nominal kinematic model was generalized and developed into a kinematic error model to achieve improved accuracy. The static tool positioning accuracy of the D1 prototype could thus be improved from 140 µm to 90 µm. A rigid-body model including friction in the active joints was presented together with identification results. The inverse dynamic model gives satisfactory results for slow motions and can thus be used for model-based feedforward compensation at low velocities. However, notable compliant behaviour was observed at higher frequencies. Modal and frequency analysis of the identified compliance revealed

resonance frequencies between 10 Hz and 16 Hz. A possible explanation for the source of these elasticities might be that the framework is lacking stiffness, but also that the mechanical design of the arm cluster mounting on the carts was shown to be too compliant. This has been important information for the development of newer, improved prototype versions. The black box models were sucessfully used for model-based ILC.

A library of Gantry-Tau models was also implemented in the Modelica language. Several application examples demonstrate how easily these Modelica models can be further used for simulation, real-time control, or optimization.

Calibration

Work on kinematic calibration was carried out with two main goals: To improve the static positioning accuracy, and to enable non-expert SME staff to execute kinematic calibration after a robot reconfiguration. The kinematic modeling assumptions were verified by measurements and the results were incorporated in the development of the kinematic error model. A method for automated kinematic calibration has been implemented and was sucessfully tested. This method facilitates a geometric reconfiguration of robots at SMEs, as it enables kinematic calibration to be executed by the non-expert SME staff. In the search for affordable sensors, calibration based on camera vision was assessed. A stereo vision system including automated image processing was shown to improve the measurement accuracy from 2.3 mm for a single camera to 1.7 mm for stereo vision. With better, but still affordable equipment and an optimization of the method, the accuracy can be further improved.

Iterative Learning Control (ILC)

ILC based on an estimate of the tool position was evaluated and applied to the Gantry-Tau robot. Three different approaches were tested: ILC based on motor angle measurements, on tool position measurements and on the estimated tool position. It was shown that the dynamic positioning performace of the end-effector can be improved by using tool position estimates instead of motor angle measurements. Direct measurements of the tool position are however usually not available in an industrial setting. Therefore, measurements from an accelerometer mounted at the end-effector plate were successfully used to estimate the tool motion. ILC based on direct measurements of the tool motion was carried out to evaluate the maximum achievable accuracy using the method and proved a good performance of the estimation based ILC with only slightly larger remaining errors. For estimating the tool position based on motor angle and accelerometer measurements, a complementary filter gave similar results as a Kalman filter. With enhanced dynamic modeling, the performance

of the Kalman filter and observer-based ILC can be further improved. Learning was applied up to and above the robot's resonance frequency.

Lead-through Programming

A new force control concept was presented and tested, a concept that enables fast and intuitive robot programming in contact situations. Usually, lead-through programming relies on one force sensor, which the operator uses to control the robot motion. A lead-through control concept was presented which uses a second force sensor to give tool force feedback. This protects the tool and/or the workpiece from deformation or damage, makes it easier to keep continuous contact and helps estimating the geometric contact constraint, thus leading the tool along the desired trajectory. The lead-through control was successfully tested in two example applications; surface tracking and a peg-in-hole application.

9.2 Future Work

The modeling presented in this thesis identified the kinematic and dynamic properties of the Gantry-Tau robot in general and specifically for some of the prototypes. Although the results seem promising and the Gantry-Tau robot seems to be a suitable robot for the usage in SMEs, the modeling revealed sources of geometric errors and elasticities not previously known. This gained knowledge should be used, both to develop better prototypes, but also to derive and identify better kinematic and dynamic models. Some of the results were already used in the ongoing development of the T2 and L2 prototypes.

The further development of robot concepts for SMEs is continued after the SMErobot project with a new European project, SMErobotics [SMErobotics, 2012].

A

A Method for Pose Accuracy Analysis

This chapter presents a geometric method for estimating the Cartesian positioning error of the Gantry-Tau robot due to actuator or link modeling errors based on the nominal kinematic model. The method is used to study the influence of actuator accuracy on the end-effector accuracy in Section 4.3.

A.1 Related Work

Solving the forward kinematics of the ideal Gantry-Tau robot consists of intersecting three spheres, a problem well known in many other domains as trilateration. Often, the application is to determine the position of an object like an airplane [Manolakis, 1996], mobile robot [Thomas and Ros, 2005] or wireless network node [Yang and Liu, 2010], knowing the distance of the object to three or more reference stations. Another well-known application is the Global Positioning System (GPS).

Numerous references treat the accuracy and error propagation of trilateration. Mostly errors in the range measurements (or link lengths of the robot), but also in the station (or cart) position are considered. Like described in [Coope, 2000], a problem that might occur in localization is that the spheres do not intersect at all. This is especially the case if the object is close to the plane defined by the three stations and the range measurements are noisy. In [Coope, 2000], it is suggested to solve this with a least square approximation of the best position estimate. However, for the Gantry-Tau robot, the spheres will usually intersect, as the ball joints are reaching their limit long before the carts would be far enough from each other to construct this case. In [Manolakis, 1996] it is shown how the range measurement errors with zero mean value and given co-

variance matrix propagate to the position estimate. Expressions for the position estimate bias and covariance matrix are developed by expanding the position estimate into a Taylor series around the solution for zero measurement errors. Similarly, the position estimate error due to errors in the station positions is derived in [Manolakis and Cox, 1998]. In [Thomas and Ros, 2005], both range measurement and station position errors are treated. Assuming only station location errors with zero-mean value and the same, uncorrelated variance for each coordinate direction, the variance for the position error is determined, which is shown to have zero mean value. The analysis of range measurement errors is equivalent to that in [Manolakis, 1996].

The geometric method presented here tries to give a more intuitive understanding on how the errors propagate and where and in what directions they affect the positioning accuracy the most.

A.2 Ellipsoidal Base Point Error

The base point position errors are often assumed equal in all three coordinate directions. However, for the Gantry-Tau robot, this assumption does not hold. The error in the actuator axis direction, i.e., the joint positioning error, has other causes than an error orthogonal to this direction. Depending on the implementation of the linear guideway and framework, there may as well be a weak direction orthogonal to the linear axis, along which the error might be larger. Therefore, an ellipsoid-formed position error is assumed. The main axes of the ellipsoid do not necessarily lie along the chosen coordinate axes.

For each arm $i = \{1,2,3\}$, the spherical joint center A_i on cart i is assumed to lie on or inside the ellipsoid E_i^A, centered around the nominal spherical joint center A_i^n and with axes $(r_{i,1} \cdot \mathbf{u}_{i,1}, r_{i,2} \cdot \mathbf{u}_{i,2}, r_{i,3} \cdot \mathbf{u}_{i,3})$. S_i^A denotes the set of all possible positions for A_i, i.e., the interior and surface of ellipsoid E_i^A.

The exact forward kinematics solution is the intersection of the three spheres with center A_i^n and radius L_i. The set S_i^B of all possible positions of the detached end-effector joint B_i of link i with length L_i and base point A_i can be described as

$$S_i^B = \{B_i \mid ||A_i - B_i|| = L_i,\, A_i \in S_i^A\} \tag{A.1}$$

The boundaries of the set S_i^B can be approximated by the two ellipsoids $E_i^{B,l}$ and $E_i^{B,u}$ with the center A_i^n and the axes $((L_i - r_{i,1}) \cdot \mathbf{u}_{i,1}, (L_i - r_{i,2}) \cdot \mathbf{u}_{i,2}, (L_i - r_{i,3}) \cdot \mathbf{u}_{i,3})$ and $((L_i + r_{i,1}) \cdot \mathbf{u}_{i,1}, (L_i + r_{i,2}) \cdot \mathbf{u}_{i,2}, (L_i + r_{i,3}) \cdot \mathbf{u}_{i,3})$, respectively. The exact boundaries coincide with the given ellipsoids for

the main axes directions. The exact S_i^B is slightly larger than the set limited by the ellipsoids, which is denoted S_i^B in the following.

The Cartesian positioning error is then defined by the intersection of S_1^B, S_2^B and S_3^B. The eight corner points T^j, $j = 1..8$ of the resulting shape S^T can be calculated by intersecting all possible combinations of the limiting ellipsoids. As the intersection point of three ellipsoids is difficult to calculate analytically, an iterative algorithm illustrated in Figure A.1 was implemented:

1. As a start value for the intersection point, the nominal end-effector position, T^n, is chosen.
2. Calculate the projections of the current intersection point estimate on each of the three ellipsoids, i.e., find the point where the half-line through the current intersection estimate starting in the ellipsoid's center intersects the ellipsoid.
3. If the distances between the three projection points are sufficiently small, stop, else continue.
4. Linearize the ellipsoids at the projection points calculated in Step 2.
5. The new intersection point estimate is the intersection of the linearizations calculated in Step 4.
6. Continue with Step 2.

The algorithm converges as illustrated in Figure A.1 in case of a sufficiently close start value, i.e., not a point on the opposite side of one of the ellipsoids.

Usually, $L_i \gg r_{i,j}, (j = 1,2,3)$, and the curved boundaries of S^T can be approximated by planes. Size and shape of S^T can then be determined by its eight corner points.

If the ellipsoidal modeling errors are dominant in the direction of the actuator axes directions, the resulting end-effector positioning error can as well be calculated with the kinematic Jacobian matrix.

A.3 Link Length Error

In case the joint position errors are equal in all directions, i.e., if $r_{i,1} = r_{i,2} = r_{i,3} = r_i$, the ellipsoids E_i^A are spheres. The eight corner points of the intersection shape S^T can now be calculated analytically using the forward kinematics with changed link length parameters $L_i \pm r_i$, $i = \{1,2,3\}$.

The same resulting geometry S^T of the end-effector error is obtained for errors $\pm r_i$ of the link length L_i.

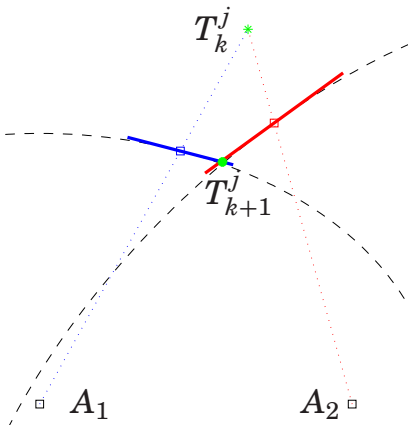


Figure A.1 Planar illustration of algorithm: To calculate the next estimate T_{k+1}^j of the intersection point of the ellipses with centers in A_1 and A_2, first the current estimate T_k^j is projected on the ellipses. T_{k+1}^j is then found by intersecting the linearizations in the projection points.

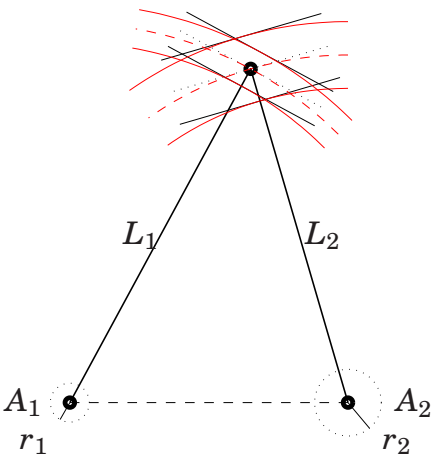


Figure A.2 Planar illustration of the end-effector accuracy's dependence on actuator accuracy: The spherical joint position A_i on cart i is known to be within a circle with radius r_i; the position of the other link-end is then limited by the circle lines indicated in red (nominal position on dashed red lines). The exact end-effector limits obtained by intersecting the respective limits for link 1 and 2 can be approximated by the parallelogram indicated in black.

A.4 Parallelepiped Approximation

As mentioned in Sect. A.2, the curved boundaries of S^T can be approximated by planes for $L_i \gg r_{i,j}$, which is usually the case. For joint position errors which are equal in all direction, i.e., A_i lie inside a sphere with radius r_i, planes of S^T which lie opposite to each other are parallel, and S^T is a parallelepiped. In the following, the case of spherically shaped errors is treated.

The shape of the parallelepiped approximation gives information about the maximum Cartesian positioning error and its direction. The maximum error lies in the direction of one of the 4 parallelepiped diagonals.

Figure A.3 illustrates the problem's geometry. The nominal base joint centers A_i^n form a pyramid with the top T^n. Each of the three parallel parallelogram pairs $F_i = \{F_{i1}, F_{i2}\}$, which are the faces of the parallelepiped, is orthogonal to one of the links L_i. The distance between the planes F_{i1} and F_{i2} is $h_i = 2r_i$. The angle between two faces F_i and F_j is equal to the top angle α_k of pyramid side S_k, $i \neq j \neq k$. The angle β_k, that the pyramid sides S_i and S_j enclose, is equal to one of the angles in the parallelograms F_{k1} or F_{k1}. With the aid of these relations, the diagonals can be calculated.

For given r_i, the influences of the kinematic parameters and the end-effector pose on the worst case error can now be derived. If the links L_i are orthogonal to each other, the parallelepiped's faces are rectangles and the diagonals equal. In consequence, the worst case error is minimal in this case. Changing the angles α_i in either direction increases the length of the longest diagonal and thus the worst case positioning error.

For ellipsoidally shaped joint position errors, the distance between the upper and the lower limit for the position of a decoupled link end varies, and the tangential planes of the limits are not perfectly parallel to each other. However, as $L_i \gg r_{i,j}$, the S_i^B in (A.1) look very much like those for the spherically shaped errors, and the tangential planes can approximately be considered as parallel.

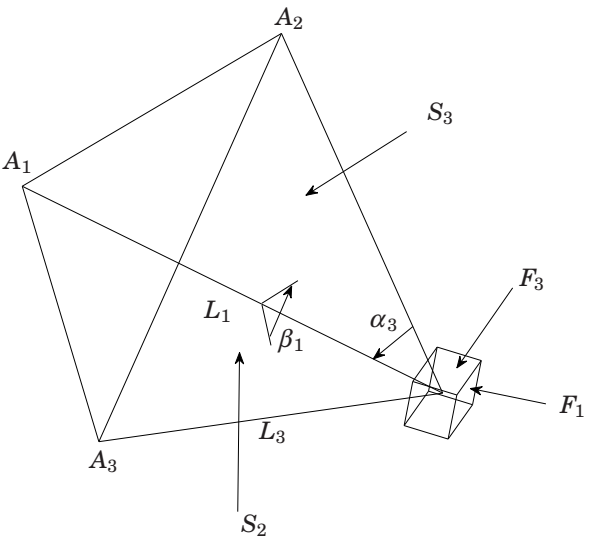


Figure A.3 Illustration of parallelepiped approximation: The spherical joint positions A_i on the carts form a pyramid with the TCP. The surface S_i with top angle α_i is the pyramid side opposite to link i. The angle β_i is the angle between the two pyramid sides intersecting in link i. The pair of parallelepiped faces F_i is orthogonal to link i.

References

ABB (2009): `http://www.abb.com`.

Abdellatif, H., M. Feldt, and B. Heimann (2006): "Application study on iterative learning control of high speed motions for parallel robotic manipulator." In *Proc of IEEE Int. Conf. on Control Applications (CCA'06)*, pp. 2528–2533. Munich, Germany.

Abdellatif, H., M. Grotjahn, and B. Heimann (2007): "Independent identification of friction characteristics for parallel manipulators." *Journal of Dynamic Systems, Measurement, and Control*, **129:3**, pp. 294–302.

Abdellatif, H. and B. Heimann (2010): "Advanced model-based control of a 6-DOF Hexapod robot: A case study." *IEEE/ASME Transactions on Mechatronics*, **15:2**, pp. 269–279.

Ahn, H.-S., Y. Chen, and K. Moore (2007): "Iterative learning control: Brief survey and categorization." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **37:6**, pp. 1099–1121.

Åkesson, J. (2008): "Optimica—an extension of Modelica supporting dynamic optimization." In *Proc. 6th Int. Modelica Conference 2008*. Modelica Association, Bielefeld, Germany.

Åkesson, J., K.-E. Årzén, M. Gäfvert, T. Bergdahl, and H. Tummescheit (2010): "Modeling and optimization with Optimica and JModelica.org—languages and tools for solving large-scale dynamic optimization problem." *Computers and Chemical Engineering*, **34:11**, pp. 1737–1749.

Andreff, N. and I. Dressler (2008): "Closed-form calibration of the Gantry-Tau parallel robot." In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2008)*, pp. 993–998. Nice, France.

Arimoto, S., S. Kawamura, and F. Miyazaki (1984): "Bettering operation of robots by learning." *Journal of Robotic Systems*, **1:2**, pp. 123–140.

Asada, H. and Y. Asari (1988): "The direct teaching of tool manipulation skills via the impedance identification of human motions." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'88)*, vol. 2, pp. 1269–1274. Philadelphia, PA, USA.

Asada, H. and H. Izumi (1989): "Automatic program generation from teaching data for the hybrid control of robots." *Robotics and Automation, IEEE Transactions on*, **5:2**, pp. 166–173.

Åström, K. and A. Heyden (1999): "Stochastic analysis of image acquisition, interpolation and scale-space smoothing." *Advances in Applied Probability*, **31:4**, pp. pp. 855–894.

Åström, K. J. and B. Wittenmark (1997): *Computer–Controlled Systems: Theory and Design*, 3rd edition. Prentice-Hall.

ATI Industrial Automation (2012): "Ati force/ torque sensor: Nano25." `http://www.ati-ia.com`.

Automatica (2012): "Automatica – innovations and solutions. 5th international trade fair for automation and mechatronics." `http://www.automatica-munich.com`.

Bai, S. and M. Y. Teo (2002): "Kinematic calibration and pose measurement of a medical parallel manipulator by optical position sensors." In *Proc. 7th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV 2002)*, vol. 1, pp. 419–424. Singapore.

Baron, L. and J. Angeles (2000): "The direct kinematics of parallel manipulators under joint-sensor redundancy." *Robotics and Automation, IEEE Transactions on*, **16:1**, pp. 12–19.

Besnard, S. and W. Khalil (2001): "Identifiable parameters for parallel robots kinematic calibration." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'01)*, vol. 3, pp. 2859–2866. Seoul, Korea.

Blomdell, A., G. Bolmsjö, T. Brogårdh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, A. Robertsson, and J.Wang (2005): "Extending an industrial robot controller – implementation and applications of a fast open sensor interface." *IEEE Robotics and Automation Magazine*, **12:3**, pp. 85–94.

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. ICRA 2010 Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications*, pp. 62–66. Anchorage, AK.

Bouguet, J.-Y. (2007): "A camera calibration toolbox for Matlab." `http://robots.stanford.edu/cs223b04/JeanYvesCalib/index.html`.

BoxJoint (2011): `http://www.boxjoint.se`.

Bristow, D., M. Tharayil, and A. Alleyne (2006): "A survey of iterative learning control." *IEEE Control Systems Magazine*, **26:3**, pp. 96–114.

Brogårdh, T. (1996): "A device for relative movement of two elements." Patent WO 97/33726.

Brogårdh, T. (2000): "Design of high performance parallel arm robots for industrial applications." In *A Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball Upon the 100th Anniversary of A Treatise on the Theory of Screws*. Cambridge, UK.

Brogårdh, T. (2002): "PKM research-important issues, as seen from a product development perspective at ABB Robotics." In *Proc. of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*. Quebec, Canada.

Brogårdh, T., S. Hanssen, and G. Hovland (2005): "Application-oriented development of parallel kinmeatic manipulators with large workspace." In *Proc. 2nd International Colloquium of the Collaborative Research Center 562: Robotc Systems for Handling and Assembly*, pp. 153–170. Braunschweig, Germany.

Budde, C., M. Rose, J. Maass, and A. Raatz (2008): "Automatic detection of assembly mode for a triglide-robot." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'08)*, pp. 1568–1575. Pasadena, CA, USA.

Burdet, E., L. Rey, and A. Codourey (2001): "A trivial and efficient learning method for motion and force control." *Engineering Applications of Artificial Intelligence*, **14:4**, pp. 487–496.

Byröd, M., K. Josephson, and K. Åström (2009): "Fast and stable polynomial equation solving and its application to computer vision." *Int. Journal of Computer Vision*, **84:3**, pp. 237–255.

Cagneau, B., G. Morel, D. Bellot, N. Zemiti, and G. d'Agostino (2008): "A passive force amplifier." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'08)*, pp. 2079–2084. Pasadena, CA, USA.

Casalino, G. and G. Bartolini (1984): "A learning procedure for the control of movements of robotic manipulators." In *Proc. IASTED symposium on Robotics and Automation*, pp. 108–111. San Francisco, CA, USA.

Cescon, M. (2008): "Subspace-based identification of a parallel kinematic manipulator dynamics.". Master's thesis ISRN LUTFD2/TFRT--5814--SE, Department of Automatic Control, Lund University, Sweden.

Cescon, M., I. Dressler, R. Johansson, and A. Robertsson (2009): "Subspace-based identification of compliance dynamics of parallel kinematic manipulator." In *Proc. 2009 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM'2009)*, pp. 1028–1033. Singapore.

Cheung, J. and Y. Hung (2009): "Robust learning control of a high precision planar parallel manipulator." *Mechatronics*, **19:1**, pp. 42–55.

Choset, H., W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun (2005): *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press.

Chuang, H.-Y. and Y.-C. Chang (2001): "Dynamics analysis and learning control for 3-PRPS platform." *Int. J. of Computer Applications in Technology*, **14:4–6**, pp. 204–214.

Clavel, R. (1988): "DELTA, a fast robot with parallel geometry." In *Proc. 18th International Symposium on Industrial Robots*. Lausanne.

Clavel, R. (1991): *Conception d'un robot parallèle rapide à 4 degrés de liberté*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Switzerland.

Coope, I. (2000): "Reliable computation of the points of intersection of n spheres in R^n." *ANZIAM Journal*, **42**, pp. C461–C477.

Craig, J. and M. Raibert (1979): "A systematic method of hybrid position/force control of a manipulator." In *Proc. IEEE Comp. Society's 3rd Int. Computer Software and Applications Conference (COMPSAC 79)*, pp. 446–451. Chicago, IL, USA.

Craig, J. J. (1984): "Adaptive control of manipulators through repeated trials." In *Proc. American Control Conference (ACC'84)*, vol. 21, pp. 1566–1573. San Diego, CA, USA.

Crothers, P., P. Freeman, T. Brogårdh, I. Dressler, K. Nilsson, A. Robertsson, W. Zulauf, B. Felber, R. Loser, and K. Siercks (2010): "Characterisation of the Tau parallel kinematic machine for aerospace application." *SAE Int. J. Aerosp.*, pp. 205–213. DOI:10.4271/2009-01-3222SAE.

Cui, H., Z. Zhu, Z. Gan, and T. Brogårdh (2005): "Kinematic analysis and error modeling of TAU parallel robot." *Robotics and Computer-Integrated Manufacturing*, **21:6**, pp. 497–505.

Dahl, O. (1992): *Path Constrained Robot Control*. PhD thesis ISRN LUTFD2/TFRT--1038--SE, Department of Automatic Control, Lund University, Sweden.

Daney, D. (1999): "Self calibration of Gough platform using leg mobility constraints." In *Proc. World Congress on the Theory of Machine and Mechanisms*, pp. 104–109. Oulou, Finland.

Dietmaier, P. (1998): "The Stewart-Gough platform of general geometry can have 40 real postures." *Advances in Robot Kinematics: Analysis and Control*, pp. 7–16.

Dressler, I., M. Haage, K. Nilsson, R. Johansson, A. Robertsson, and T. Brogårdh (2007): "Configuration support and kinematics for a reconfigurable Gantry-Tau manipulator." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, pp. 2957–2962. Rome, Italy.

Dressler, I., A. Robertsson, and R. Johansson (2008): "Automatic kinematic calibration of a modular Gantry-Tau parallel robot from a kinematics point of view." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'08)*, pp. 1282–1287. Pasadena, CA, USA.

Dressler, I., J. Schiffer, and A. Robertsson (2009): "Modeling and control of a parallel robot using Modelica." In *Proc. 7th International Modelica Conference*, pp. 261–269. Como, Italy.

Du, Y., P. Xie, and B. Liu (2010): "Error analysis in parallel robot using edgeworth series and information entropy." In *Proc. Int. Conf. on Computer Design and Applications (ICCDA 2010)*, vol. 2, pp. V2–350–V2–354. Qinhuangdao, China.

Dynasim (2009): `http://www.Dynasim.se/`.

ECHORD (2012): `http://www.echord.info/wikis/website/home`.

Eriksson, F. and M. Welander (2009): "Haptic interface for a contact force controlled robot.". Master's thesis ISRN LUTFD2/TFRT--5837--SE, Department of Automatic Control, Lund University, Sweden.

Everett, L. J. (1989): "Forward calibration of closed-loop jointed manipulators." *The Int. Journal of Robotics Research*, **8:4**, pp. 85–91.

Farhat, N., V. Mata, A. Page, and F. Valero (2008): "Identification of dynamic parameters of a 3-DOF RPS parallel manipulator." *Mechanism and Machine Theory*, **43:1**, pp. 1–17.

Freescale (2010): "MMA7361L." `http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=KIT3376MMA73x1L`.

Frenkel, J., C. Schubert, G. Kunze, and K. Jankov (2009): "Using Modelica for interactive simulations of technical systems in a virtual reality environment." In *Proc. 7th Int. Modelica Conference*. Como, Italy.

Friman, J. (2010): "Kinematic and force control for a Gantry-Tau robot.". Master's thesis ISRN LUTFD2/TFRT--5850--SE, Department of Automatic Control, Lund University, Sweden.

Ge, J., F. Tan, and H. Zhang (2008): "Robotic machining: A force-control-based fast programming method." In *Proc. IEEE Int. Conf. on Robotics, Automation and Mechatronics (RAM 2008)*, pp. 730–735. Chengdu, China.

Grunwald, G., G. Schreiber, A. Albu-Schäffer, and G. Hirzinger (2003): "Programming by touch: The different way of human-robot interaction." *IEEE Transactions on Industrial Electronics*, **50:4**, pp. 659–666.

Gunnarsson, S., M. Norrlöf, E. Rahic, and M. Özbek (2007): "On the use of accelerometers in iterative learning control of a flexible robot arm." *Int. J. of Control*, **80:3**, pp. 363–373.

Gustafsson, E. (2008): "Controlling a robot with two force sensors in a lead-through programming scenario.". Master's thesis ISRN LUTFD2/TFRT--5818--SE, Department of Automatic Control, Lund University, Sweden.

Hanselman, D. (1991): "Techniques for improving resolver-to-digital conversion accuracy." *IEEE Transactions on Industrial Electronics*, **38:6**, pp. 501–504.

Harris, F. (1978): "On the use of windows for harmonic analysis with the discrete fourier transform." *Proceedings of the IEEE*, **66:1**, pp. 51–83.

Hartenberg, R. S. and J. Denavit (1955): "A kinematic notation for lower pair mechanisms based on matrices." *Journal of Applied Mechanics*, **77:2**, pp. 215–221.

Hast, M., J. Åkesson, and A. Robertsson (2009): "Optimal robot control using Modelica and Optimica." In *Proc. of the 7th Int. Modelica Conference 2009*. Modelica Association, Como, Italy.

Heidenhain (2010): "Length gauge ST 3078." `http://www.heidenhain.com`.

Heindl, J. and G. Hirzinger (1983): "Device for programming movements of a robot." U.S. Patent 4,589,810.

Hexagon Metrology (2012): `http://www.hexagonmetrology.ch/en/index.htm`.

Higgins, W. (1975): "A comparison of complementary and Kalman filtering." *IEEE Transactions on Aerospace and Electronic Systems*, **AES-11:3**, pp. 321–325.

Hirzinger, G. and K. Landzettel (1985): "Sensory feedback structures for robots with supervised learning." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'85)*, vol. 2, pp. 627–635. St Louis, MO, USA.

Hogan, N. (1984): "Impedance control of industrial robots." *Robotics and Computer Integrated Manufacturing*, **1:1**, pp. 97–113.

Honegger, M., A. Codourey, and E. Burdet (1997): "Adaptive control of the hexaglide, a 6 dof parallel manipulator." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'97)*, vol. 1, pp. 543–548. Albuquesrque, NM, USA.

Hosoda, K., K. Igarashi, and M. Asada (1998): "Adaptive hybrid control for visual and force servoing in an unknown environment." *IEEE Robotics & Automation Magazine*, **5:4**, pp. 39–43.

Hovland, G., M. Choux, M. Murray, and T. Brogårdh (2007a): "Benchmark of the 3-DOF Gantry-Tau parallel kinematic machine." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'07)*, pp. 535–542. Rome, Italy.

Hovland, G., M.Murray, and T. Brogårdh (2007b): "Experimental verification of friction and dynamic models of a parallel kinematic machine." In *Proc. 2007 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*. Zürich.

Huang, T., Z. Hong, J. Mei, and D. Chetwynd (2006): "Kinematic calibration of the 3-DOF module of a 5-DOF reconfigurable hybrid robot using a double-ball-bar system." In *Proc. 2006 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2006)*, pp. 508–512. Beijing, China.

Hui, Z. and Z. C. Feng (2009): "Accuracy analysis of a parallel robot with particle swarm optimization." In *Proc. 5th Int. Conf. on Computational Intelligence and Security (CIS'09)*, vol. 2, pp. 142–145. Beijing, China.

Husty, M. (1996): "An algorithm for solving the direct kinematics of general stewart-gough platforms." *Mechanism and Machine Theory*, **31:4**, pp. 365–379.

Innocenti, C. (2001): "Forward kinematics in polynomial form of the general stewart platform." *Journal of Mechanical Design*, **123:2**, pp. 254–260.

Isaksson, M., T. Brogårdh, I. Lundberg, and S. Nahavandi (2010): "Improving the kinematic performance of the SCARA-Tau PKM." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'10)*, pp. 4683–4690. Anchorage, AK, USA.

Isaksson, M., T. Brogårdh, M. Watson, S. Nahavandi, and P. Crothers (2012): "The octahedral hexarot a novel 6-DOF parallel manipulator." *Mechanism and Machine Theory*, **55:0**, pp. 91–102.

ISG (2012): `http://www.isg-stuttgart.de`.

Ji, Z. and Z. Li (1999): "Identification of placement parameters for modular platform manipulators." *Journal of Robotic Systems*, **16:4**, pp. 227–236.

JModelica (2010): `http://www.jmodelica.org`.

Johannesson, L., V. Berbyuk, and T. Brogårdh (2003): "Gantry-Tau – a new three degrees of freedom parallel kinematic robot." In *Proc. of the Mechatronics Meeting 2003*, pp. 1–6. Gothenburg, Sweden.

JR3 Inc (2012): "Multi-axis load cell technologies." `http://www.jr3.com`.

Khalil, W., S. Besnard, and P. Lemoine (2000): "Comparison study of the geometric parameter calibration methods." *Int. Journal of Robotics and Automation*, **15:2**, pp. 56–67.

Khalil, W. and O. Ibrahim (2007): "General solution for the dynamic modeling of parallel robots." *Journal of Intelligent and Robotic Systems*, **49:1**, pp. 19–37.

Kim, H. S. and Y. J. Choi (2000): "The kinematic error bound analysis of the stewart platform." *Journal of Robotic Systems*, **17:1**, pp. 63–73.

Krabbes, M. and C. Meißner (2006): "Dynamic modeling and control of a 6 DOF parallel kinematics." In *Proc. 5th International Modelica Conference*, pp. 385–390. Vienna, Austria.

Kumar, R., P. Berkelman, P. Gupta, A. Barnes, P. Jensen, L. Whitcomb, and R. Taylor (2000): "Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, vol. 1, pp. 610–617. San Francisco, CA, USA.

LabComm (2010): `http://wiki.cs.lth.se/moin/LabComm`.

Lamy, X., F. Colledani, F. Geffard, Y. Measson, and G. Morel (2010): "Human force amplification with industrial robot : Study of dynamic limitations." In *Proc. 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2010)*, pp. 2487–2494. Taipei, Taiwan.

Lee, J. and Z. Geng (1993): "A dynamic model of a flexible Stewart platform." *Computers & Structures*, **48:3**, pp. 367–374.

Leica Geosystems (2009): `http://www.leica-geosystems.com`.

Longuet-Higgins, H. (1981): "A computer program for reconstructing a scene from two projections." *Nature*, **239:11**.

Lyzell, C. and G. Hovland (2007): "Verification of the dynamics of the 5-DOF Gantry-Tau parallel kinematic machine." In *Proc. IASTED Robotics and Applications Conference*, pp. 445–450. Würzburg, Germany.

Manolakis, D. (1996): "Efficient solution and performance analysis of 3-d position estimation by trilateration." *Aerospace and Electronic Systems, IEEE Transactions on*, **32:4**, pp. 1239–1248.

Manolakis, D. and M. Cox (1998): "Effect in range difference position estimation due to stations' position errors." *Aerospace and Electronic Systems, IEEE Transactions on*, **34:1**, pp. 329–334.

Marton, L. and B. Lantos (2009): "Friction and backlash measurement and identification method for robotic arms." In *Proc. Int. Conf. on Advanced Robotics (ICAR 2009)*, pp. 1–6. Munich, Germany.

MathWorks (2012): "MathWorks, products and services." `http://www.mathworks.com/products/`.

Matlab (2011): "Zero-phase digital filtering, signal processing toolbox." `http://www.mathworks.com/help/toolbox/signal/ref/filtfilt.html`.

McGlone, C., E. Mikhail, and J. Bethel, Eds. (2004): *Manual of Photogrammetry, 5th Edition*. ASPRS.

Meeussen, W., J. Rutgeerts, K. Gadeyne, H. Bruyninckx, and J. D. Schutter (2007): "Contact state segmentation using particle filters for programming by human demonstration in compliant motion tasks." *Robotics, IEEE Transactions on*, **23:2**.

Merlet, J.-P. (1993): "Closed-form resolution of the direct kinematics of parallel manipulators using extra sensors data." In *Proc. IEEE Int. Conf on Robotics and Automation (ICRA'93)*, vol. 1, pp. 200–204.

Merlet, J.-P. (2000): *Parallel Robots*. Kluwer Academid Publishers, Norwell, MA.

Merlet, J.-P. and D. Daney (2005): "Dimensional synthesis of parallel robots with a guaranteed given accuracy over a specific workspace." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'05)*, pp. 942–947. Barcelona, Spain.

Miel, G. (1981): “An updated version of the Kantorovitch theorem for Newton’s method.” *Computing*, **27:3**, pp. 237–244.

Miermeister, P. and A. Pott (2012): “Auto calibration method for cable-driven parallel robots using force sensors.” In Lenarˇcič and Husty, Eds., *Latest Advances in Robot Kinematics*, pp. 269–276. Springer Netherlands.

Modelica (2011): `http://www.Modelica.org`.

MONROE (2012): `http://www.echord.info/wikis/website/monroe`.

Moore, K. (1998): “Multi-loop control approach to designing iterative learning controllers.” In *Proc. 37th IEEE Conf. on Decision and Control, (CDC’98)*, vol. 1, pp. 666–671. Tampa, FL, USA.

Motta, J., G. de Carvalho, and R. McMaster (2001): “Robot calibration using a 3d vision-based measurement system with a single camera.” *Robotics and Computer-Integrated Manufacturing*, **17:6**, pp. 487–497.

Murray, M., G. Hovland, and T. Brogårdh (2006): “Collision-free workspace design of the 5-axis Gantry-Tau parallel kinematic machine.” *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS’06)*, pp. 2150–2155.

Murray, M., G. Hovland, and T. Brogårdh (2008): “Singularity-free reconfiguration of the 5-DOF Gantry-Tau parallel kinematic machine.” In *Proc. 2nd Intl. Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*. Montpellier, France.

Murray, R., Z. Li, and S. Sastry (1994): *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Boca Raton, FL.

Nabat, V., O. Company, and F. Pierrot (2006): “Dynamic modeling and identification of Par4, a very high speed parallel manipulator.” In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS’06)*, pp. 496–501. Beijing, China.

Nahvi, A. and J. M. Hollerbach (1996): “The noise ampliciation index for optimal pose selection in robot calibration.” In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA’96)*, pp. 647–654.

Nilsson, A. (2011): “Parallel kinematic wrist.” Patent 11 405 333.3.

Nordström, A. (2006): “Identifiering och reglering av industrirobot med hjälp av accelerometer.”. LiTH-ISY-EX-06/3785, Dept. Electrical Engineering, Linköping University, Sweden.

Norrlöf, M. and S. Gunnarsson (2002): "Time and frequency domain convergence properties in iterative learning control." *Int. J. of Control*, **75**, pp. 1114–1126.

Norrlöf, M. and R. Karlsson (2005): "Position estimation and modeling of a flexible industrial robot." In *Proc. 16th IFAC World Congress*, vol. 16:1.

Olsson, T. (2007): *High-Speed Vision and Force Feedback for Motion-Controlled Industrial Manipulators*. PhD thesis ISRN LUTFD2/TFRT--1078--SE, Department of Automatic Control, Lund University, Sweden.

Otter, M., H. Elmqvist, and S.-E. Mattson (2003): "The new Modelica MultiBody library." In *Proc. 3rd International Modelica Conference*, pp. 311–330. Linköping, Sweden.

Pan, Z. and H. Zhang (2008): "Robotic machining from programming to process control." *Proc. 7th World Congress on Intelligent Control and Automation*, pp. 553–558.

Pedreira Carabel, C. J. and A. A. Zambrano García (2011): "Modeling, control and automatic code generation for a two-wheeled self-balancing vehicle using modelica.". Master's thesis ISRN LUTFD2/TFRT--5884--SE, Department of Automatic Control, Lund University, Sweden.

Pfeiffer, F. and R. Johanni (1987): "A concept for manipulator trajectory planning." *IEEE Journal on Robotics and Automation*, **3:2**, pp. 115–123.

Phan, M. and R. W. Longman (1988): "A mathematical theory of learning control of linear discrete multivariable systems." In *Proc. AIAA/AAS Astrodynamics Conference*, pp. 740–746.

Philip, J. (1998): *Critical point configurations of the 5-, 6-, 7-, and 8-point algorithms for relative orientation*. Trita-mat-1998-ma-13, Department of Mathematics, Royal Institute of Technology, Stockholm, Sweden. Technical Report.

Pott, A., A. Kecskeméthy, and M. Hiller (2007): "A simplified force-based method for the linearization and sensitivity analysis of complex manipulation systems." *Mechanism and Machine Theory*, **42:11**, pp. 1445–1461.

Python (2011): `http://www.python.org`.

Qi, L., D. Zhang, J. Zhang, and J. Li (2009): "A lead-through robot programming approach using a 6-dof wire-based motion tracking

device." In *Proc. 2009 IEEE Int. Conf. on Robotics and Biomimetics (ROBIO)*, pp. 1773–1777. Guilin, Guangxi, China.

Raghavan, M. (1993): "The stewart platform of general geometry has 40 configurations." *Journal of Mechanical Design*, **115:2**, pp. 277–282.

Ratcliffe, J. D., P. L. Lewin, E. Rogers, J. J. Hätönen, and D. H. Owens (2006): "Norm-optimal iterative learning control applied to Gantry robots for automation applications." *IEEE Transactions on Robotics*, **22:6**, pp. 1303–1307.

Renaud, P. (2003): *Apport de la vision pour l'identification géométrique de méchanismes parallèles*. PhD thesis, Univ. Blaise Pascal, Clermont-Ferrand, France.

Renaud, P., N. Andreff, G. Gogu, and M. Dhome (2003): "Optimal pose selection for vision-based kinematic calibration of parallel mechanisms." In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, vol. 3, pp. 2223–2228. Las Vegas, NV, USA.

Renaud, P., N. Andreff, J.-M. Lavest, and M. Dhome (2006): "Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology." *IEEE Transactions on Robotics*, **22:1**, pp. 12–22.

Renaud, P., N. Andreff, P. Martinet, and G. Gogu (2005): "Kinematic calibration of parallel mechanisms: A novel approach using legs observation." *Robotics, IEEE Transactions on*, **21:4**, pp. 529–538.

Renaud, P., N. Andreff, F. Pierrot, and P. Martinet (2004): "Combining end-effector and legs observation for kinematic calibration of parallel mechanisms." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'04)*, vol. 4, pp. 4116–4121. New Orleans, LA, USA.

Robertz, S., K. Nilsson, R. Henriksson, and A. Blomdell (2007): "Industrial robot motion control with real-time Java and EtherCAT." In *Proc. IEEE Conf. on Emerging Technologies & Factory Automation (EFTA 2007)*, pp. 1453–1456. Patras, Greece.

Schmidt, J. and H. Niemann (2001): "Using quaternions for parametrizing 3-d rotations in unconstrainded nonlinear optimization." In *Proc. Vision, Modeling, and Visualization (VMV 2001)*, pp. 399–406.

Schöllig, A. and R. D'Andrea (2009): "Optimization-based iterative learning control for trajectory tracking." In *Proc. of the European Control Conference (ECC'09)*, pp. 1505–1510.

Siciliano, B. (1999): "The Tricept robot: Inverse kinematics, manipulability analysis and closed-loop direct kinematics algorithm." *Robotica*, **17:4**.

Siciliano, B. and L. Villani (1999): *Robot Force Control*. Kluwer Academic Publishers.

S.L., P. T. (2012): `http://www.pkmtricept.com`.

SMErobot (2009): `http://www.smerobot.org`.

SMErobotics (2012): `http://www.smerobotics.org`.

Stewart, D. (1965): "A platform with 6 degrees of freedom." *Proc. of the Institution of Mechanical Engineers*, **180:1**, pp. 371–386.

Stewénius, H. (2005): *Gröbner Basis Methods for Minimal Problems in Computer Vision*. PhD thesis, Dept. of Mathematics, Lund University.

Stewénius, H., C. Engels, and D. Nistér (2006): "Recent developments on direct relative orientation." *ISPRS Journal of Photogrammetry and Remote Sensing*, **60**, June, pp. 284–294.

Symétrie (2012): "Positioning hexapods." `http://www.hexapod-system.com/hexapod_positioning/`.

Tanev, T. K. (2000): "Kinematics of a hybrid (parallel-serial) robot manipulator." *Mechanism and Machine Theory*, **35:9**, pp. 1183–1196.

Tavolieri, C., E. Ottaviano, and M. Ceccarelli (2002): "Pose determination for a rigid body by means of CaTraSys II (cassino tracking system)." In *Proc. 1st European Conf. on Mechanism Science (EuCoMeS)*. Obergurgl, Austria. ISBN 3-901249-85-0.

Tayebi, A. and J.-X. Xu (2003): "Observer-based iterative learning control for a class of time-varying nonlinear systems." *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, **50:3**, pp. 452–455.

Thomas, F. and L. Ros (2005): "Revisiting trilateration for robot localization." *Robotics, IEEE Transactions on*, **21:1**, pp. 93–101.

Thümmel, M., G. Looye, M. Kurze, M. Otter, and J. Bals (2005): "Nonlinear inverse models for control." In *Proc. 4th International Modelica Conference 2005*, pp. 267–279. Hamburg-Harburg, Germany.

Thümmel, M., M. Otter, and J. Bals (2001): "Control of robots with elastic joints based on automatic generation of inverse dynamics models." In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'01)*, vol. 2, pp. 925–930. Hawaii, USA.

Tyapin, I., G. Hovland, and T. Brogårdh (2007): "Workspace optimisation of a reconfigurable parallel kinematic manipulator." In *Proc. 2007 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pp. 1–6. Zürich, Switzerland.

Tyapin, I., G. Hovland, and T. Brogårdh (2008): "Kinematic and elastodynamic design optimisation of the 3-DOF Gantry-Tau parallel kinematic manipulator." In *Proc. 2nd International Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*. Montpellier, France.

Vaidyanathan, P. P. (1993): *Multirate systems and filter banks*. Prentice Hall, Englewood Cliffs, New Jersey.

Verscheure, D., B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl (2009): "Time-optimal path tracking for robots: A convex optimization approach." *IEEE Transactions on Automatic Control*, **54:10**, pp. 2318–2327.

Vivas, A., P. Poignet, and F. Pierrot (2005): "Predictive functional control of a parallel robot." *Control Engineering Practice*.

Wallén, J. (2011): *Estimation-based iterative learning control*. PhD thesis, Department of Automatic Control, Linköping University, Sweden. Dissertations. No. 1358.

Wallén, J., S. Gunnarsson, R. Henriksson, S. Moberg, and M. Norrlöf (2009): "ILC applied to a flexible two-link robot model using sensor-fusion-based estimates." In *Proc. 48th IEEE Conference on Decision and Control (CDC'09)*, pp. 458–463.

Wallén, J., S. Gunnarsson, and M. Norrlöf (2010): "Some implementation aspects of iterative learning control." Technical Report LiTH-ISY-R-2967. Dept. Electrical Engineering, Linköping University, Sweden.

Wallén, J., M. Norrlöf, and S. Gunnarsson (2011): "A framework for analysis of observer-based ILC." *Asian Journal of Control*, **13:1**, pp. 3–14.

Wang, J. and O. Masory (1993): "On the accuracy of a Stewart platform. i. the effect of manufacturing tolerances." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'93)*, vol. 1, pp. 114–120. Atlanta, Georgia.

Wang, J., H. Zhang, and G. Zhang (2008): "A force control assisted robot path generation system." In *Proc. IEEE Int. Conf. on Automation Science and Engineering (CASE 2008)*, pp. 528–533. Washington DC, USA.

Wang, S.-M. and K. F. Ehmann (2002): "Error model and accuracy analysis of a six-dof Stewart platform." *ASME Journal Of Manufacturing Science And Engineering*, **124:2**, pp. 286–295.

Williams, I., G. Hovland, and T. Brogårdh (2006): "Kinematic error calibration of the Gantry-Tau parallel manipulator." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, pp. 4199–4204. Orlando, FL.

Xenomai (2012): "Real-time framework for linux." `http://xenomai.org`.

Xiao, D., B. Ghosh, N. Xi, and T. Tarn (2000): "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment." *Control Systems Technology, IEEE Transactions on*, **8:4**, pp. 635–645.

Yang, Z. and Y. Liu (2010): "Quality of trilateration: Confidence-based iterative localization." *IEEE Transactions on Parallel and Distributed Systems*, **21:5**, pp. 631–640.

Yoshikawa, T. (2000): "Force control of robot manipulators." In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'00)*, vol. 1, pp. 220–226. San Francisco, CA, USA.

Yoshikawa, T. and A. Sudou (1993): "Dynamic hybrid position/force control of robot manipulators-on-line estimation of unknown constraint." *Robotics and Automation, IEEE Transactions on*, **9:2**, pp. 220–226.

Zhaocai, D. and Y. Yueqing (2008): "Dynamic modeling and inverse dynamic analysis of flexible parallel robots." *Int. Journal of Advanced Robotic Systems*.

Zhu, Z., J. Li, Z. Gan, and H. Zhang (2005): "Kinematic and dynamic modelling for real-time control of Tau parallel robot." *Mechanism and Machine Theory*, **40:9**, pp. 1051–1067.