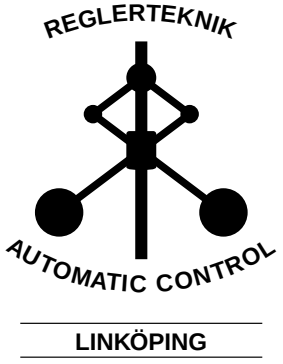
Linköping studies in science and technology. Thesis.
No. 1336

On Modeling and Control of Flexible Manipulators

Stig Moberg



Division of Automatic Control
Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
http://www.control.isy.liu.se
stig@isy.liu.se

Linköping 2007

This is a Swedish Licentiate's Thesis.

Swedish postgraduate education leads to a Doctor's degree and/or a Licentiate's degree.
A Doctor's Degree comprises 240 ECTS credits (4 years of full-time studies).
A Licentiate's degree comprises 120 ECTS credits,
of which at least 60 ECTS credits constitute a Licentiate's thesis.

Linköping studies in science and technology. Thesis.
No. 1336

On Modeling and Control of Flexible Manipulators

Stig Moberg

stig@isy.liu.se
www.control.isy.liu.se
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping
Sweden

ISBN 978-91-85895-29-8 ISSN 0280-7971 LiU-TEK-LIC-2007:45

Printed by LiU-Tryck, Linköping, Sweden 2007

To Karin and John

Abstract

Industrial robot manipulators are general-purpose machines used for industrial automation in order to increase productivity, flexibility, and quality. Other reasons for using industrial robots are cost saving, and elimination of heavy and health-hazardous work. Robot motion control is a key competence for robot manufacturers, and the current development is focused on increasing the robot performance, reducing the robot cost, improving safety, and introducing new functionalities. Therefore, there is a need to continuously improve the models and control methods in order to fulfil all conflicting requirements, such as increased performance for a robot with lower weight, and thus lower mechanical stiffness and more complicated vibration modes. One reason for this development of the robot mechanical structure is of course cost-reduction, but other benefits are lower power consumption, improved dexterity, safety issues, and low environmental impact.

This thesis deals with three different aspects of modeling and control of flexible, i.e., elastic, manipulators. For an accurate description of a modern industrial manipulator, the traditional flexible joint model, described in literature, is not sufficient. An improved model where the elasticity is described by a number of localized multidimensional spring-damper pairs is therefore proposed. This model is called the extended flexible joint model. This work describes identification, feedforward control, and feedback control, using this model.

The proposed identification method is a frequency-domain non-linear gray-box method, which is evaluated by the identification of a modern six-axes robot manipulator. The identified model gives a good description of the global behavior of this robot.

The inverse dynamics control problem is discussed, and a solution methodology is proposed. This methodology is based on a differential algebraic equation (DAE) formulation of the problem. Feedforward control of a two-axes manipulator is then studied using this DAE approach.

Finally, a benchmark problem for robust feedback control of a single-axis extended flexible joint model is presented and some proposed solutions are analyzed.

Acknowledgments

First of all I would like to thank my supervisor Professor Svante Gunnarsson for helping me in my research, and for always finding time for a meeting in his busy schedule.

This work has been carried out in the Automatic Control Group at Linköping University and I am very thankful to Professor Lennart Ljung for letting me join the group. I thank everyone in the group for the inspiring and friendly atmosphere they are creating. I especially thank Professor Torkel Glad for sharing his extensive knowledge and for his excellent graduate courses, Johan Sjöberg for all the knowledge and inspiration he has given me, Gustav Hendeby for always helping me out when having LaTeXproblems, and Ulla Salaneck for her help with many practical issues. I am also thankful to the Automatic Control Robotics Group, consisting of Johanna Wallén, Professor Svante Gunnarsson, Dr. Mikael Norrlöf, and Dr. Erik Wernholt, for their support, and also for their first class research cooperation with ABB Robotics.

This work was supported by ABB Robotics and the Swedish Research Council (VR) which are gratefully acknowledged. At ABB Robotics, I would first of all like to thank the head of controller development, Jesper Bergsjö, for supporting my research, and I hope that he is satisfied with the result thus far. The support from Henrik Jerregård, Wilhelm Jacobsson, and Staffan Elfving is also thankfully acknowledged. I am also greatly indepted to Dr. Torgny Brogård at ABB Robotics for the support and guidance he is giving me. Furthermore, I would like to thank all my other friends and colleagues at ABB Robotics for creating an atmosphere filled with great knowledge, but also of fun, which both provide a constant inspiration to my work. Among present and former colleagues, I would especially like to mention, in order of appearance, Ingvar Jonsson, Mats Myhr, Henrik Knobel, Lars Andersson, Sören Quick, Dr. Steve Murphy, Mats Isaksson, Professor Geir Hovland, Sven Hanssen and Hans Andersson. I would also like to thank all master thesis students whom I have had the privilege to supervise and learn from.

My coauthors are also greatly acknowledged, Sven Hanssen for his complete devotion to mechatronics and, as expert in modeling, being absolutely invaluable for my work, Dr. Jonas Öhr who inspired me to take up my graduate studies, and for inspiring discussions about automatic control, Dr. Erik Wernholt for equally inspiring discussions about identification, and Professor Svante Gunnarsson for guiding me in my work and keeping me on the right track. I am also thankful to Professor Per-Olof Gutman at Israel Institute of Technology, for teaching me QFT in an excellent graduate course, as well as being inspirational in many ways.

I am also very grateful to Dr. Torgny Brogårdh, Professor Svante Gunnarsson, Dr. Erik Wernholt, Dr. Mikael Norrlöf, Sven Hanssen, Johanna Wallén, Johan Sjöberg, and Dr. Jonas Öhr, for reading different versions of this thesis, or parts thereof, and giving me valuable comments and suggestions.

Finally, I would like to thank my son John for excellent web design and computer support, and my wife Karin for, among many things, helping me with the English language. And to the both of you, as well as the rest of my immediate family, thanks for the love, patience, and support that you are constantly giving me.

Stig Moberg
Linköping, December 2007

Contents

1

Introduction

Models of robot manipulators and their accompanying identification or verification procedures are two cornerstones of a robot motion control system. The control algorithms and the trajectory generation algorithms are two equally important cornerstones of such a system. This thesis deals with some aspects of modeling, identification, and control of flexible manipulators, i.e., with three of the mentioned cornerstones. Here, flexible should be interpreted as elastic, and a manipulator should be interpreted as an industrial robot although the results to some extent can be applied on other types of manipulators and mechanical systems.

1.1 Motivation and Problem Statement

Robot motion control is a key competence for robot manufacturers, and current development is focused on increasing the robot performance, reducing the robot cost, improving safety, and introducing new functionalities as described in Brogårdh (2007). There is a need to continuously improve the models and control methods in order to fulfil all conflicting requirements, e.g., increased performance for a robot with lower weight, and thus lower mechanical stiffness and more complicated vibration modes. One reason for this development of the robot mechanical structure is of course cost reduction but other benefits are lower power consumption, as well as improved dexterity, safety issues, and lower environmental impact. The need of cost reduction results in the use of optimized robot components, which usually have larger individual variation, e.g., variation of gearbox stiffness or in the parameters describing the mechanical arm. Cost reduction sometimes also results in higher level of disturbances and nonlinearities in some of the components, e.g., in the sensors or in the actuators. An example of a modern, weight optimized robot manipulator is shown in Figure 1.1.

An industrial robot is a general purpose machine for industrial automation, and even though the requirements of a certain application can be precisely formulated, there are

Figure 1.1: *The IRB6620 from ABB.*

no limits in what the robot users want with respect to the desirable performance and functionality of the motion control of a robot. The required motion control performance depends on the application. The better performance, the more applications can be subject to automation by a specific robot model, meaning higher manufacturing volumes and lower cost. Some examples are:

- High path tracking accuracy in a continuous application, e.g., water-jet cutting.
- High speed accuracy in a continuous application, e.g., painting.
- Low cycle time, i.e., high speed and acceleration, in a discrete application, e.g., material handling.
- Small overshoots and a short settling time in a discrete process application, e.g., spot welding.
- High control stiffness in a contact application, e.g., machining.

For weight- and cost optimized industrial manipulators, the requirements above can only be handled by increased computational intelligence, i.e., improved motion control. Motion control of industrial robot manipulators is a challenging task, which has been studied by academic and industrial researchers for more than three decades. Some results from the academic research have been successfully implemented in real industrial applications, while other results are far away from being relevant to the industrial reality. To some extent, the development of motion control algorithms has followed two separate routes, one by academic researchers and one by robot manufacturers, unfortunately with only minor interaction.

The situation can partly be explained by the fact that the motion control algorithms used in the industry sometimes are regarded as trade secrets. Due to the tough competitive situation among robot manufacturers, the algorithms are seldom published. Another explanation is that the academic robot control researchers often apply advanced mathematics on a few selected aspects of relatively small systems, whereas the industrial robot researchers and developers must deal with all significant aspects of a complex system where the proposed advanced mathematics often cannot be applied. Furthermore, the

problems that the robot industry sometimes present, might include too much engineering aspects to be attractive for the academic community. Industrial robot research and development must balance short term against long term activities. Typical time constants from start of research to the final product, in the area of the motion control technology discussed in this work, can be between 5 and 10 years, and sometimes even longer. Thus, long-term research collaborations between industry and academia should be possible, given that the intellectual property aspects can be handled satisfactorily.

The problems of how to get industrial-relevant academic research results, and of how to obtain a close collaboration between universities and industry, are not unique for robotics motion control. The existence of a gap between the academic research and industrial practise in the area of automatic control in general, is often discussed. One balanced description on the subject can be found in Bernstein (1999), where it is pointed out that the control practitioners must articulate their needs to the research community, and that motivating the researchers with problems from real applications "can have a significant impact on increasing the relevance of academic research to engineering practise". Another quote from the same article is "I personally believe that the gap on the whole is large and warrants serious introspection by the research community". The problem is somewhat provocatively described in Ridgely and McFarland (1999) as, freely quoted, "what the industry in most cases do not want is stability proofs, guarantees of convergence and other purely analytical developments based on idealized and unrealistic assumptions". Another view on the subject is "the much debated theory-applications gap is a misleading term that overlooks the complex interplay between physics, invention and implementation, on the one side, and theoretical abstractions, models, and analytical designs, on the other side" (Kokotovic and Arcak, 2001). The need for a balance between theory and practise is expressed in, e.g., Åström (1994), and finally a quote from Brogårdh (2007): "industrial robot development has for sure not reached its limits, and there is still a lot of work to be done to bridge the gap between the academic research and industrial development".

It is certainly true that, as in the science of physics, research on both "theoretical control" and "experimental control" is needed. The question is whether the balance between these two sides needs adjustment. This subject is certainly an important one, as automatic control can have considerable impact on many industrial processes as well as on other areas, affecting both environmental and economical aspects. It is my hope that this work can help "bridging the gap", as well as moving the frontiers somewhat in the area of robotics motion control.

Most publications concerning flexible (elastic) industrial robot manipulators only consider elasticity in the rotational direction. If the gear elasticity is considered we get the flexible joint model, and if link deformation restricted to a plane perpendicular to the preceding joint is included in the model we get the flexible link model. These restricted models simplify the control design but limit the attainable performance. Motivated by the trend of developing light-weight robots, a new model, here called the *extended flexible joint model*, is proposed for use in motion control systems as well as in design and performance simulation. The following aspects of this model are treated in this work:

- Multivariable identification of the unknown and uncertain elastic model parameters, applied to a real six-axes industrial robot.

- Multivariable feedforward control for trajectory tracking.

- Feedback control of a one-axis extended flexible joint model.

1.2 Outline

Part I contains an overview of robotics, modeling, identification, and control. Part II consists of a collection of edited papers.

1.2.1 Outline of Part I

Chapter 2 gives an introduction to robotics in general, and the motion control problem in particular. Modeling of robot manipulators is described in Chapter 3, and some system identification methods that are relevant for this thesis are described in Chapter 4. A survey on control methods used in robotics can be found in Chapter 5. Finally, Chapter 6 provides conclusions and some ideas for future research.

1.2.2 Outline of Part II

This part consists of a collection of edited papers, introduced below. A summary of each paper is given, together with a short paragraph describing the background to the paper and the contribution of the author of this thesis.

Paper A: Frequency-Domain Gray-Box Identification of Industrial Robots

> Wernholt, E. and Moberg, S. (2007b). Frequency-domain gray-box identification of industrial robots. Technical Report LiTH-ISY-R-2826, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to the 17th IFAC World Congress*, Seoul, Korea.

Summary: This paper describes the proposed identification procedure, where unknown parameters (mainly spring-damper pairs) in a physically parameterized nonlinear dynamic model are identified in the frequency domain, using estimates of the nonparametric frequency response function (FRF) in different robot configurations. In order to accurately estimate the nonparametric FRF, the experiments must be carefully designed. The selection of optimal robot positions for the experiments is a part of this design. Two different parameter estimators are compared, and experimental results show that the proposed method can generate accurate parameter estimations in an industrial environment, and in a short time.

Background and contribution: The basic idea of using the nonparametric FRF for the estimation of the parametric robot model is described in Öhr et al. (2006), where the author of this thesis has made much of the initial work on the identification procedure. Erik Wernholt has continued to analyze various aspects of the identification procedure such as the nonparametric FRF estimation, the selection of optimal experiment positions, and the choice of parameter estimator. In Paper A, the author of this thesis has served as a discussion partner, performed most of the experiments, and helped out with the experimental evaluation.

Paper B: A DAE approach to Feedforward Control of Flexible Manipulators

> Moberg, S. and Hanssen, S. (2007). A DAE approach to feedforward control of flexible manipulators. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3439–3444, Roma, Italy.

Summary: This paper investigates feedforward control of elastic robot structures. A general serial link elastic robot model which can describe a modern industrial robot in a realistic way is presented. This model is denoted the *extended flexible joint model*. The elasticity is modeled as localized 3 or 6 degrees-of-freedom spring-damper pairs. The feedforward control problem for this model is discussed and a solution method for the inverse dynamics problem is proposed. This method involves solving a differential algebraic equation (DAE). A simulation example for an elastic two-axes planar robot is also included, showing that it is possible to obtain accurate path tracking by using this method.

Background and contribution: The DAE formulation of the inverse dynamics problem was the result of a discussion between the author of this thesis and Sven Hanssen. Sven Hanssen derived the simulation model, and the author of this thesis has made the control research and implemented the DAE solver that is used.

Paper C: A Benchmark Problem for Robust Feedback Control of a Flexible Manipulator

> Moberg, S., Öhr, J., and Gunnarsson, S. (2007). A benchmark problem for robust feedback control of a flexible manipulator. Technical Report LiTH-ISY-R-2820, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to IEEE Transactions on Control Systems Technology*.

Summary: This paper describes a benchmark problem for robust feedback control of a flexible manipulator together with some proposed and tested solutions. The system to be controlled is a four-mass system subject to input saturation, nonlinear gear elasticity, model uncertainties, and load disturbances affecting both the motor and the arm. The system should be controlled by a discrete-time controller that optimizes the performance for given robustness requirements.

Background and contribution: The benchmark problem was first presented as *Swedish Open Championships in Robot Control* (Moberg and Öhr, 2004, 2005) where the author of this thesis formulated the problem together with Jonas Öhr. The analysis of the solutions as well as the experimental validation of the benchmark model were performed mainly by the author of this thesis. The final paper as presented in this thesis also includes Svante Gunnarsson as a valuable discussion partner and coauthor.

Related Publications

Publications of related interest not included in this thesis, where the author of this thesis has contributed:

Wernholt, E. and Moberg, S. (2007a). Experimental comparison of methods for multivariable frequency response function estimation. Technical Report LiTH-ISY-R-2827, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to the 17th IFAC World Congress*, Seoul, Korea.

Öhr, J., Moberg, S., Wernholt, E., Hanssen, S., Pettersson, J., Persson, S., and Sander-Tavallaey, S. (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. In *Proc. ISMA2006 International Conference on Noise and Vibration Engineering*, pages 3305–3314, Leuven, Belgium.

Hovland, G. E., Hanssen, S., Gallestey, E., Moberg, S., Brogårdh, T., Gunnarsson, S., and Isaksson, M. (2002). Nonlinear identification of backlash in robot transmissions. In *Proc. 33rd ISR (International Symposium on Robotics)*, Stockholm, Sweden.

1.3 Contributions

The main contributions of the thesis are:

- The initial work on the identification procedure introduced in Öhr et al. (2006) and further described in Paper A. The procedure has been successfully applied, by the author of this thesis, to experimental data from a six-axes industrial robot.
- The DAE formulation of the inverse dynamics problem for the extended flexible joint model as described in Paper B.
- The solution method for the inverse dynamics problem of the extended flexible joint model, and its application on a small but realistic robot model. This is also described in Paper B.
- The formulation and evaluation of a relevant industrial benchmark problem as described in Paper C.

Part I

Overview

2

Robotics

Robotics involves many technical and scientific disciplines, e.g., sensor- and vision technologies, computer architecture, drive system and motor technologies, real time systems, automatic control, modeling, mechanical design, applied mathematics, man-machine interaction, system communication and computer languages. A comprehensible introduction can be found in Snyder (1985). This chapter gives a short introduction describing the parts that are relevant for this work, i.e., modeling and motion control.

2.1 Introduction

Throughout this work, the term robot is used to denote an industrial robot, i.e., a manipulator arm, mainly used for manufacturing in industry. Some examples of such robots are shown in Figure 2.1. The first industrial robots were installed in the 1960's. Today there are about 1 million operational industrial robots installed worldwide (IFR, 2005). Robots are used for a variety of tasks, e.g., welding, painting, cutting, gluing, material handling, machine tending, machining, and assembly. There are many types of mechanical robot structures such as the parallel arm robot and the articulated robot, which can be of elbow or parallel linkage type. Examples of these three robot structures are shown in Figure 2.2. Further examples of mechanical robot structures (also called kinematic structures) can, e.g., be found in Spong et al. (2006). In the following, the writing will be restricted to the, at present, most common type of industrial robot which is the articulated robot of elbow type. This robot, or manipulator arm, has typically six serially mounted bodies connected by revolute joints. The bodies are called links, and the joints can also be called axes. The links are actuated by electrical motors via gear transmissions, i.e., gearboxes, also named speed reducers. The motor positions are measured by sensors. The first link is connected to the base, and the last link is connected to an end effector, i.e., a tool. With six actuated links, both the position and the orientation of the end effector can be controlled. The first three joints and links are often denoted main axes, and the last three are denoted wrist

Figure 2.1: *ABB robot family and the IRC5 controller.*

Figure 2.2: *Three examples of robot structures from ABB. The parallel arm robot IRB340 (left), the parallel linkage robot IRB4400 (middle), and the elbow robot IRB6600 (right).*

axes.

2.2 Models

A description of the most important models used in robotics can be found in, e.g., Craig (1989), Spong et al. (2006), and Sciavicco and Siciliano (2000). Here follows a short overview.

2.2.1 Kinematic Models

The kinematic models describe the robot motions without regard to the forces that cause the motions, i.e, all time-based and geometrical properties of the motion. The kinematics relate the joint angular position vector[1] $q \in R^6$ to the position $p \in R^3$ and orientation $\phi \in R^3$ of the tool frame attached to the tool and positioned in the tool center point (TCP). One example of a kinematic relation is the forward kinematics where the tool frame position and orientation are described as a function of the joint angular position vector as

$$X = \Gamma(q),$$

where X is the tool frame position and orientation, also named pose, defined as

$$X = \begin{bmatrix} p \\ \phi \end{bmatrix},$$

and $\Gamma(\cdot)$ is a nonlinear function. The tool frame is described in a reference frame, i.e., a coordinate system, attached to the base of the robot, called base frame. The orientation has many possible representations. Euler angles (e.g., roll-pitch-yaw) is a minimal representation but has singularities which can be avoided with a four component unit quaternion representation. Describing the manipulator pose by the joint angles is often denoted a joint space representation of the robot state while describing it by the tool position and orientation is denoted a task space representation which is usually implemented in Cartesian coordinates[2]. The described frames and the joint positions are illustrated in Figure 2.3.

2.2.2 Dynamic Models

Dynamic robot models describe the relations between the motions of the robot and the forces that cause the motions. The models are most often formulated in joint space. One example of a dynamic model is the model of a rigid manipulator which can be expressed as

$$\tau = M(q)\ddot{q} + c(q, \dot{q}) + g(q) + f(\dot{q}), \tag{2.1}$$

where τ is the actuator torque vector, $M(q)$ is the inertia matrix, $c(q, \dot{q})$ is a vector of Coriolis and centripetal torques, $g(q)$ is the gravity torque vector, and $f(\dot{q})$ is the vector

[1] For simplicity, a robot with six joints is considered here.

[2] Task space is sometimes also called operational space or Cartesian space. Joint space is also called configuration space.

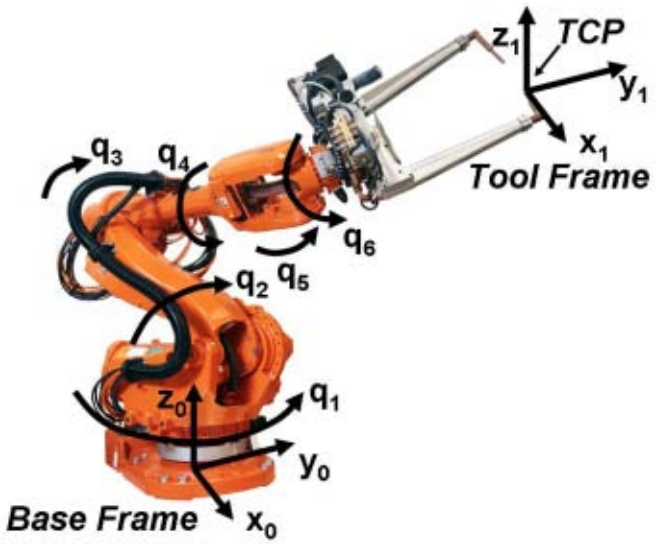


Figure 2.3: *Base frame, tool frame, and joint positions illustrated on a robot equipped with a spotwelding gun.*

of, possibly nonlinear, joint friction torques. The rigid body inertial parameters for each link are the mass, the center of mass, and the inertia. The actuator inertia and mass are added to the corresponding link parameters.

The inverse dynamics problem is useful for control, and consists of computing the required actuator torques as a function of the joint position vector q and its time derivatives, $\dot{q}$ and $\ddot{q}$. For the rigid model (2.1), this involves algebraic computations only. For simulation of the manipulator movement, the direct dynamics problem must be solved. The differential equation (2.1) is then solved with the actuator torques as input.

2.3 Motion Control

The motion control of a modern industrial manipulator is a complex task. A description of the current status of industrial robot motion control can be found in Brogårdh (2007) and references therein.

2.3.1 A General Motion Control System

A general robot motion control system, capable of synchronously controlling n robot manipulators, is illustrated in Figure 2.4. The system consists of the following components:

Robot 1, Robot 2, ..., Robot n The robot manipulators with actuators and sensors included. The manipulators can be in contact with the environment, e.g., in assembly tasks, or operate in free space without contact with the environment, e.g., in

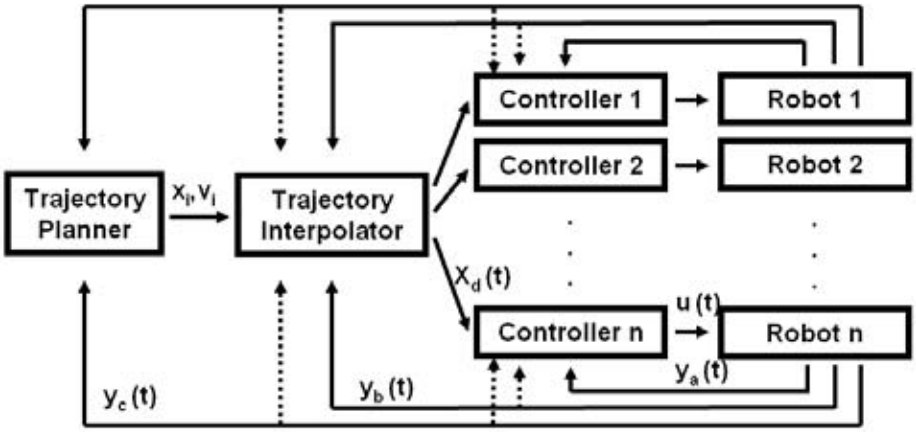


Figure 2.4: *A general robotics motion control system.*

laser cutting. The sensors can be of different types. A first type of sensors generates sensor readings y_a which are used by the feedback controller. Examples of such sensors are encoders or resolvers measuring actuator positions, accelerometers measuring link and tool acceleration, force sensors[3] measuring the contact forces acting on the end effector, torque sensors measuring the joint torques on the link side of the gearbox, and joint encoders measuring the joint positions. The second type of sensor information, y_b, can be exemplified by conveyor positions measured by encoders or the position of an arc-welding gun relative to the desired welding path as measured by a tracking sensor. The second type of sensors are used by the trajectory interpolator and the controller to adapt the robot motion to the measured path. A third type of sensors are primarily used by the trajectory planner, e.g., in the case of a vision system specifying the position of an object to be gripped by the robot. Some of the robots in Figure 2.4 could also be replaced by multi- or single-axis positioners used for, e.g., rotating the object in an arc-welding application.

Controller 1, Controller 2, ..., Controller n Generate control signals $u(t)$ for the actuators with the references $X_d(t)$ and the sensor readings $y_a(t)$ as input. The controller can operate in position control mode for point-to-point motion in, e.g., spot-welding, and continuous path tracking mode in, e.g., dispensing applications. When the robot is in contact with the environment, the controller can still be in position control mode, e.g., in pre-machining applications, where the stiffness of the mechanical arm and the torque disturbance rejection of the controller are important requirements. Some contact applications require a compliant behavior of the robot due to uncertain geometry or process requirements. This requires a controller in compliance control mode, defined as impedance or admittance control mode, for some directions and position control mode for other directions in task space. Examples of these applications, using compliance control, are assembly, machine

[3]The sensor is called force sensor for simplicity, even though both forces and torques normally are measured.

tending, and product testing. In other applications as friction stir welding, grinding, and polishing, the contact force must be controlled in a specific direction while position or speed control are made in other directions. Compliance and force control can be accomplished with or without the use of force sensors, dependent on the performance requirements.

Trajectory Interpolator The task of the trajectory interpolator is to compute controller references $X_d(t)$ that follow the programmed trajectory and which simultaneously are adapted to the dynamic performance of the robot. The input from the trajectory planner is the motion specification, e.g., motion commands specifying a series of end effector positions x_i along with desired end effector speeds v_i. Sensor readings $y_b(t)$ and $y_c(t)$ can also be used by the trajectory interpolator. The trajectory $X_d(t)$ contains positional information for all n robots, and can be expressed in Cartesian or joint space.

Trajectory Planner Specifies the desired motion of the robot end effector. This can be done manually by a robot programmer who specifies the motion in a robot programming language with a series of motion commands. The program can be taught by moving the robot to the desired positions and command the robot to read the actual positions. The motion commands can also be generated by off-line programming where the positions are defined in a CAD system. The desired motion can also be expressed on a higher level by task programming as, e.g., by an instruction as picking all objects on a moving conveyor, and placing the objects in desired locations. The positions for picking the objects can then be specified by a vision system.

Besides the robot motion execution described in connection to Figure 2.4, the robot motion control is also involved in a lot of other activities, as for example:

- Identification of the link parameters for an extended kinematic model in order to obtain high volumetric accuracy. This is useful for off-line programming and for fast robot replacement.
- Iterative learning control for improved path accuracy, used, e.g., for high precision laser cutting (Gunnarsson et al., 2006; Norrlöf, 2000).
- Identification of user tool and load to improve the dynamic model accuracy for high control performance, and to avoid overload of the robot mechanics (Brogårdh and Moberg, 2002).
- Model based supervision, e.g., collision detection and jam detection to save equipment at accidental robot movements, for example during programming (Brogårdh et al., 2001).
- Diagnosis of, e.g., gearboxes and mechanical brakes.
- Specific motion control modes for, e.g., emergency stop.
- Supervision of, e.g., position and speed for safety reasons, and for saving equipment on and close to the robot.
- Calibration of, e.g., tool frame or actuators.

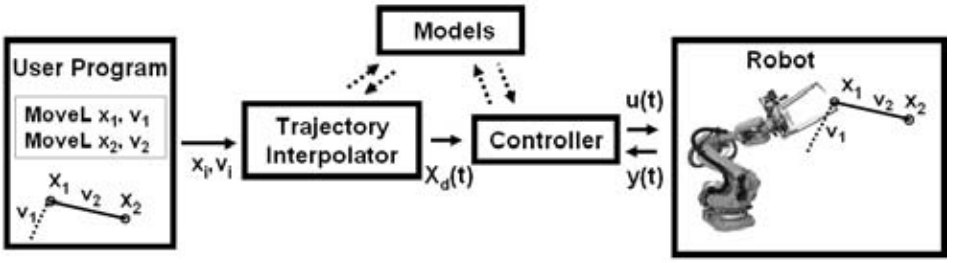


Figure 2.5: *A simplified robotics motion control system.*

2.3.2 A Model-Based Motion Control System for Position Control

A simplified outline of a motion control system is shown in Figure 2.5. The system controls one robot manipulator in position control mode. The trajectory interpolator and the controller make an extensive use of models, hence this type of control is denoted model-based control. The system has the following components:

Robot Manipulator The physical robot arm with actuators receiving the control signal $u(t)$ from the controller. The control signal can be, e.g., a torque reference to a torque controller, a velocity reference to a velocity controlled actuator or a three-phase current to an electrical motor. Throughout this work, the torque control is assumed to be ideal and a part of the actuator, which has been proven by experiments to be a reasonable assumption for most of the ABB robots. Hence, the control signal $u(t)$ will be a motor torque reference. The sensor readings $y(t)$ are normally the actuator positions only, but more sensors can be added as described in the previous section.

Models The models used by the motion control system, e.g., the kinematic and dynamic models described previously. The models and how to obtain the model parameters will be further described in Chapters 3 – 4.

Controller Generates control signals $u(t)$ for the actuators with the references $X_d(t)$ and the sensor readings $y(t)$ as input. The controller can be split into a feedback controller and a feedforward controller, and will be further described in Chapter 5.

Trajectory Interpolator Creates a reference $X_d(t)$ for the controller with the user program as input. The first step of the trajectory interpolation could be to compute the continuous geometric path $X_d(\gamma)$ where γ is a scalar path parameter, e.g., the distance along the path. The second step of the interpolation is to associate a timing law to γ and obtain $\gamma(t)$. In this way, the path $X_d(\gamma)$ is transformed to a time determined trajectory $X_d(\gamma(t)) = X_d(t)$. Note that the speed and acceleration as well as higher order derivatives of the path, are completely determined once the trajectory is computed. The requirement of path smoothness depends on the controller used and on the requested motion accuracy. One example of smoothness requirement is that the acceleration derivative, also called jerk, is continuous. The trajectory interpolator is responsible for limiting the speed and acceleration of the trajectory, to

make it possible for the robot to dynamically follow the reference without actuator saturations.

User Program The desired motion of the robot end effector is specified by a series of motion commands in the *user program*. In the example program, the command `MoveL x2, v2` specifies a linear movement of the end effector to the Cartesian position `x2` with velocity `v2`. The movement starts in the end position of the previous instruction, i.e. `x1`. Generally, the movement can be specified in joint space or in Cartesian space. In joint space the positions are described by the joint angular positions, and in Cartesian space by the Cartesian position and orientation of the end effector. The movement in Cartesian space can typically be specified as linear or circular. In reality, more arguments must be attached to the motion command to specify, e.g., behavior when the position is reached (stop or make a smooth direction change to the next specified position), acceleration (do not exceed $5\,\mathrm{m/s^2}$), events (set digital output $100\,\mathrm{ms}$ before the endpoint is reached).

The ultimate requirements on the described motion control system can be summarized as follows:

Optimal Time Requirement The user-specified path speed and possibly other user limitations regarding, e.g., acceleration, must be followed exactly, and may only be reduced if the robot movements are limited mechanically or electrically by the constraints from the robot components. Examples of component constraints are the maximum motor torque and speed. The speed and acceleration are always limited. Other examples are the allowed forces and torques acting on the manipulator links.

Optimal Path Requirement The user-specified path must be followed with specified precision even under the influence of different uncertainties. These uncertainties are disturbances acting on the robot and on the measurements, as well as uncertainties in the models used by the motion control system.

If these two requirements are fulfilled, the motion performance of the robot system depends entirely on the electromechanic components such as gearboxes, mechanical links, actuators, and power electronics. Generally, a given performance requirement can be fulfilled either by improving the electromechanics or by improving the computational intelligence of the software, i.e., improving the models, the control algorithms, and the trajectory optimization algorithms. The possibilities to use electromechanic or software solutions to fulfil performance requirements can be illustrated by two simple examples:

- **Requirement:** Path accuracy of 0.5 mm.
  - **Electromechanic solution:** Design the robot with a stiffer mechanical arm including bearings and gearboxes.
  - **Software solution:** Improve the models and control algorithms, i.e., a higher degree of fulfillment of the optimal path requirement.
- **Requirement:** The robot task must be accomplished in 10 s.
  - **Electromechanic solution:** Increase the power and torque of the drive-train, i.e., the motors, gearboxes, and power electronics.

– **Software solution:** Improve the trajectory optimization algorithms and the models used, i.e., a higher degree of fulfillment of the optimal time requirement.

A higher degree of fulfilment of the two requirements using software solutions means more complexity in the software and algorithms and, of course, more computational power. This means a more expensive computer in the controller, and initially, a longer development time. However, the trend of moving functionality from electromechanics to software will certainly continue due to the continuing development of low-cost computer hardware and efficient methods for developing more and more complex real-time software systems. However, in every product development project, there is an optimal trade-off between electromechanics cost, i.e., the cost of gearboxes, mechanical arm, actuators, and power electronics, and the software cost[4], i.e., the cost of computers, memory, sensors, and motion control development.

To make cost optimization of robot installations including robot electromechanics and controller software for a spectrum of different applications is a very difficult task, and how this is solved varies a lot from one robot manufacturer to another but to move as much as possible of the performance enhancement to the controller software must be regarded as the ultimate goal for any robot motion control system. Besides cost there is of course also a matter of usability which increases with the level of computational intelligence, e.g., easy programming and facilitation of advanced applications.

The *optimal time requirement* is mainly a requirement on the trajectory interpolator and the *optimal path requirement* is mainly a requirement on the controller. Accurate models are necessary for the fulfilment of both requirements.

The fulfilment of the second requirement, i.e., the optimal path requirement, is the subject of this work, and the next two chapters will treat the modeling aspects of robotics.

[4] Software is not used in the normal sense here and the software cost could also be denoted the motion control cost. *Cost of brain* could also be used, and in that case, the electromechanics cost could be denoted the *cost of muscles*.

3

Modeling of Robot Manipulators

This chapter describes the models that are relevant for this work, namely the kinematic and dynamic models of a serial link robot of elbow type.

3.1 Kinematic Models

The kinematic models describe the motion without regard to the forces that cause it, i.e, all time-based and geometrical properties of the motion. The position, velocity, acceleration, and higher order derivatives are all described by the kinematics. The presentation in this section is based on Craig (1989), Spong et al. (2006), and Sciavicco and Siciliano (2000).

3.1.1 Position Kinematics and Frame Transformations

The serial link robot, or manipulator arm, has N serially mounted bodies connected by revolute joints. The bodies are called links, and the joints can also be called axes. The links are actuated by electrical motors via gear transmissions. The motor positions are measured by sensors[1]. The first link is connected to the base, and the last link is connected to an end effector, i.e., a tool. With $N \geq 6$ actuated links, both the position and the orientation of the end effector can be controlled. The joint angular position vector, or joint angles, $q \in R^N$, describe the configuration or position of the manipulator. The position of the tool is described by attaching a coordinate system, or frame, fixed to the tool. The tool pose is then described by the position $p \in R^3$ and orientation $\phi \in R^3$ of this tool frame. The origin of the tool frame is known as the tool center point (TCP). The tool frame position is described relative to the base frame, attached to the base of the robot.

Describing the manipulator position by the joint angles q is often denoted a joint space description, while describing it by the tool position p and orientation ϕ is denoted a

[1] The motor position sensor is usually an encoder or a resolver.

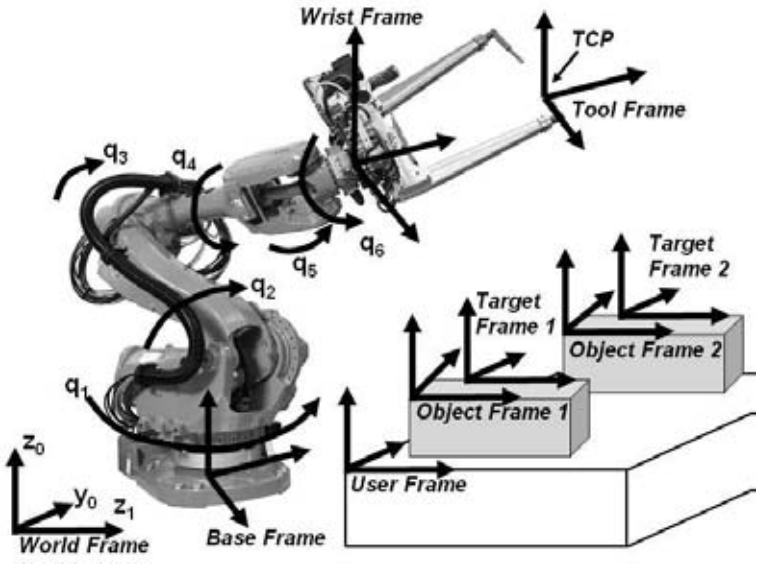


Figure 3.1: *A robot in a work-cell with the standard frames, as defined by ABB, used in cell modeling illustrated.*

Cartesian space description. The position kinematics relates the joint space to the Cartesian space. For a manipulator with gearboxes, the actuator space can also be defined. The relation between actuator space and joint space is fairly simple. The actuator position $\tilde{q}$ is related to the joint position q by

$$\tilde{q} = rq,$$

where r is a matrix of gear ratios. A robot with the described frames and the joint positions is illustrated in Figure 3.1. The other frames in the figure are typical frames used for work-cell modeling, to simplify the robot programming task[2]:

- *World frame* is the common reference frame in a work cell, used by all robots, positioners, conveyors, and other equipment.
- *Base frame* describes the position and orientation of the base of a robot.
- *Wrist frame* is attached to the mounting flange of the robot.
- *Tool frame* describes the tool position and orientation.
- *User frame* describes a task relevant location.
- *Object frame* describes an object relative to the task-relevant location.
- *Work object frame* is the object frame as seen from the world frame, i.e., defined by user frame and object frame.

[2]Different robot manufacturers have slightly different concepts and naming conventions for the frames used in cell modeling. In this text, the ABB frame concept and names are used.

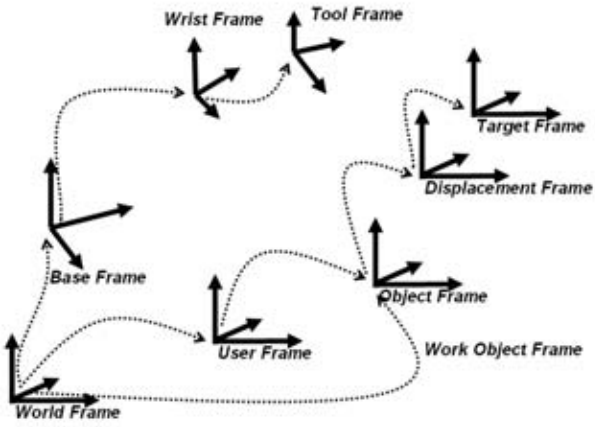


Figure 3.2: *Standard frames.*

- *Displacement frame* describes locations inside an object.
- *Target frame* (or programmed position) is where the tool frame eventually should be positioned.

The position and orientation of frame B can be described relative to frame A by a homogenous transformation ${}^{A}_{B}T$ describing the translation and rotation required to move from A to B. Homogenous transformations are described in the general references given, e.g., Craig (1989). Figure 3.2 shows the standard frames and the chains of transformations describing the frame positions. Note that, e.g., the user frame can be defined on a positioner, on another robot, or on a conveyor. This means that the transformation from world frame to user frame can be time-varying, defined by, e.g., the kinematics of the external positioner. The base frame can also be time-varying if, e.g., the robot base is moved by a track motion as illustrated in Figure 3.3. The desired position of the tool frame is described in the robot programs as a target frame. The target frame, which should be equal to the tool frame when the position is reached, can then be described in the base frame using the chain of transformations.

3.1.2 Forward Kinematics

The forward kinematics problem is, given the joint angles, to compute the position and orientation of the tool frame relative to the base frame, or

$$X = \Gamma(q, \theta_{kin}), \tag{3.1}$$

where X is the tool frame position and orientation defined as

$$X = \begin{bmatrix} p \\ \phi \end{bmatrix}, \tag{3.2}$$

Figure 3.3: *A robot moved by a track motion.*

and $\Gamma(\cdot)$ is a nonlinear function. θ_{kin} is a vector of fixed kinematic link parameters which, together with the joint positions, describe the relation between the base frame, frames attached to each link, and the tool frame. θ_{kin} consists of parameters[3] representing arm lengths and angles describing the rotation of the joint axes relative to the previous joint axis. The orientation has many possible representations. Euler angles (e.g., roll-pitch-yaw) gives a minimal representation but has mathematical singularities[4] which can be avoided with a four component unit quaternion representation.

3.1.3 Inverse Kinematics

The inverse kinematics problem is, given the position and orientation of the tool frame, to compute the corresponding joint angles. The inverse kinematics problem is considerably harder than the forward kinematics problem[5], where a unique closed form solution always exists. Some features of the inverse kinematics problem:

- A solution may not exist. The existence of a solution defines the workspace of a manipulator. The workspace is the volume which the end effector of the manipulator can reach[6].
- If a solution exists, it may not be unique. One example of multiple solutions is the elbow-up and elbow-down solutions for the elbow manipulator. There can even exist an infinite number of solutions, e.g., in the case of a kinematically redundant

[3] A well known parametrization of θ_{kin} are the so called Denavit-Hartenberg parameters.

[4] These type of singularities, caused by the representation of orientation, is not the same type as the singularities described in Section 3.1.4.

[5] This is true for the serial link manipulator considered here. For a parallel arm robot, the situation is the opposite.

[6] The volume which can be reached with arbitrary orientation is called dextrous workspace, and the volume that can be reached with at least one orientation is called reachable workspace.

manipulator. In this case the number of link degrees-of-freedom[7] is greater than the number of tool degrees-of-freedom, e.g., $N > 6$ for a tool which can be oriented and positioned arbitrarily. An example of a redundant system is the robot moved by a track motion in Figure 3.3, having totally 7 degrees-of-freedom. Methods for selecting one of many possible solutions are often needed, and each solution is then usually said to represent a robot configuration.

- The solution can be hard to obtain, even though it exists. Closed-form solutions are preferred, but for certain manipulator structures, only numerical iterative solutions are possible.

The inverse kinematics can be expressed as

$$q = \Gamma^{-1}(X, C, \theta_{kin}), \tag{3.3}$$

where C is some information used to select a feasible solution. One alternative is to let C be parameters describing the desired configuration. Another alternative is to let C be the previous solution, and to select the new solution as the, in some sense, closest solution.

3.1.4 Velocity Kinematics

The relation between the joint velocity $\dot{q}$ and the Cartesian velocity $\dot{X}$ is determined by the velocity Jacobian J of the forward kinematics relation

$$\dot{X} = \frac{\partial \Gamma(q)}{\partial q}\dot{q} = J(q)\dot{q}. \tag{3.4}$$

The relation between higher derivatives can be found by differentiation of the expression above, e.g., $\ddot{X} = J(q)\ddot{q} + \dot{J}(q)\dot{q}$. The velocity Jacobian, from now on called the Jacobian, is useful in many aspects of robotics. One example is the transformation of forces and torques, acting on the end effector, to the corresponding joint torques. This relation can be derived using the principle of virtual work, and is

$$\tau = J^T(q)F, \tag{3.5}$$

where $F \in R^6$ is the vector of end effector forces and torques, and $\tau \in R^N$ is a vector of joint torques.

The Jacobian is also useful for studying singularities. A singularity is a configuration where the Jacobian looses rank. Some facts about singularities:

- A Cartesian movement close to a singularity results in high joint velocities. This can be seen from the relation $\dot{q} = J^{-1}(q)\dot{X}$.
- Most singularities occur on the workspace boundary but can also occur inside the workspace, e.g., when two or more joint axes are lined up.
- Close to a singularity there may be no solutions or infinitely many solutions to the inverse kinematics problem.
- The ability to move in a certain direction is reduced close to a singularity.

[7] The number of independent coordinates necessary to specify the configuration of a certain system is called the number of degrees-of-freedom or number of DOF.

3.2 Dynamic Models

Dynamic models of the robot manipulator describe the relation between the motion of the robot, and the forces that cause the motion[8]. A dynamic model is useful for, e.g., simulation, control analysis, mechanical design, and real-time control. Some control algorithms require that the inverse dynamics problem is solved. This means that the required actuator torque is computed from the actuator position and its time derivatives. For simulation of the manipulator movement, the direct dynamic problem must be solved. This means that the dynamic model differential equations are solved with the actuator torques as input.

Depending on the intended use of the dynamic model, the manipulator can be modeled as rigid or elastic. A real flexible manipulator is a continuous nonlinear system, described by partial differential equations, PDEs, with infinite number of degrees-of-freedom. An infinite dimensional model is not realistic to use in real applications. Instead, finite dimensional models with the minimum number of parameters for the required accuracy level is preferred. The following three levels of elastic modeling are described in, e.g., Bascetta and Rocco (2002):

Finite Element Models These models are the most accurate models but normally not used for simulation and control due to their complexity. FEM models are widely used in the mechanical design of robot manipulators.

Assumed Modes Models These models are derived from the PDE formulation by modal truncation. Assumed modes models used for simulation and control design are frequently described in the literature.

Lumped Parameter Models The elasticity is modeled by discrete, localized springs. With this approach, a link can be divided into a number of rigid bodies connected by non-actuated joints. The gearbox elasticity can also be modeled with this approach.

In the following, both rigid and elastic dynamic models for robot manipulators will be described.

3.2.1 The Rigid Dynamic Model

There are several methods for obtaining a rigid dynamic model. The two most common approaches are the Lagrange formulation (Spong et al., 2006) and the Newton-Euler formulation (Craig, 1989). A third method is Kane's method (Kane and Levinson, 1983, 1985; Lesser, 2000). All methods are based on classical mechanics, see Goldstein (1980). All methods produce the same result even though the equations may differ in computational efficiency and structure. A detailed comparison of these methods is outside the scope of this work. In the following, only the Lagrange formulation will be described in some detail.

The Lagrangian method is based on describing scalar energy functions of the system, i.e., the kinetic energy $K(q, \dot{q})$ and the potential energy $V(q)$. These energy functions are

[8] A somewhat different definition of dynamics is usually adopted in general multibody dynamics (Shabana, 1998), where kinetics deals with motion and the forces that produce it, and kinematics deals with the geometric aspects of motion regardless of the forces that cause it. Dynamics then includes both kinematics and kinetics.

expressed as functions of some suitable generalized coordinates, q, defined by

$$q = \begin{bmatrix} q_1 & q_2 & \dots & q_N \end{bmatrix}^T. \tag{3.6}$$

For the manipulator considered here, the generalized coordinates can be chosen as the joint angles. It can be shown that the kinetic energy can be expressed as

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}, \tag{3.7}$$

where $M(\cdot)$ is the inertia matrix. The inertia matrix is positive definite and symmetric. More properties of Lagrangian dynamics are described in, e.g., Sciavicco and Siciliano (2000).

The next step is to compute the Lagrangian L as $L = K - V$. By applying the Lagrange equation

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_j} - \frac{\partial L}{\partial q_j} = \tau_j, \tag{3.8}$$

the equations of motion, i.e., the dynamic model, can be derived. In the equation above, τ_j is called a generalized force, in our case the actuator torque.

In a rigid dynamic model, the links and gearboxes are assumed to be rigid. The mass and inertia of the actuators and gearboxes are added to the corresponding link parameters. The model consists of a serial kinematic chain of N links modeled as rigid bodies as illustrated in Figure 3.4. One rigid body rb^i is illustrated in Figure 3.5, and is described by its mass m^i, center of mass ξ^i, and inertia tensor with respect to center of mass J^i. Due to the symmetrical inertia tensor, only six components of J^i need to be defined. For simplicity, it is assumed that the structure of the serial manipulator, i.e., the orientations of the rotational joint axes are given. The kinematics is then described by the length l^i. All parameters are described in a coordinate system a^i, fixed in rb^i, and are defined as follows

$$\xi^i = \begin{bmatrix} \xi^i_x & \xi^i_y & \xi^i_z \end{bmatrix}, \quad J^i = \begin{bmatrix} J^i_{xx} & J^i_{xy} & J^i_{xz} \\ J^i_{xy} & J^i_{yy} & J^i_{yz} \\ J^i_{xz} & J^i_{yz} & J^i_{zz} \end{bmatrix}, \quad l^i = \begin{bmatrix} l^i_x & l^i_y & l^i_z \end{bmatrix}.$$

The model can be derived by using, e.g., the Lagrange formulation which yields a system of second order ordinary differential equations or ODEs

$$M(q, \theta_{rb}, \theta_{kin})\ddot{q} + c(q, \dot{q}, \theta_{rb}, \theta_{kin}) + g(q, \theta_{rb}, \theta_{kin}) = \tau, \tag{3.9}$$

where $\dot{x}$ denotes dx/dt and the time dependence is omitted in the expressions. $M(\cdot) \in R^{N \times N}$ is the inertia matrix computed as $M(\cdot) = M_a(\cdot) + M_m$, where $M_a(\cdot)$ is the configuration dependent inertia matrix of the robot arm and, M_m is the inertia matrix of the rotating actuators expressed on the link side of the gearbox. The inertia matrix is symmetric and positive definite. The Coriolis, centrifugal, and gravity torques are described by $c(\cdot) \in R^N$ and $g(\cdot) \in R^N$, respectively. The vector of joint angles is denoted $q \in R^N$, and the actuator torque vector is denoted $\tau \in R^N$. Note that the equations are described on the link side, i.e., $\tau = \tau^a_m = r^T \tau^m_m$ and $M_m = M^a_m = r^T M^m_m r$ where r is

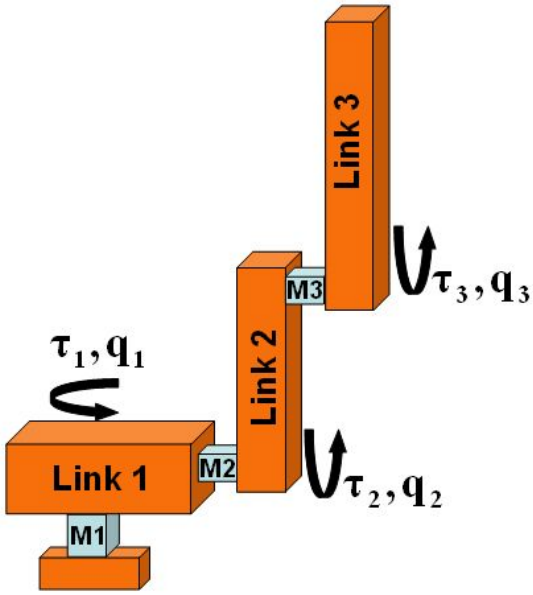


Figure 3.4: *A rigid dynamic model with 3 DOF.*

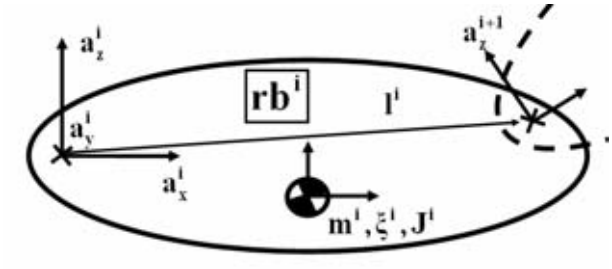


Figure 3.5: *A rigid body and its attributes.*

the gear ratio matrix. The notation X_m^a should be interpreted[9] *quantity X for the motor expressed on the link side of the gearbox.* The rigid body and kinematic parameters described previously are gathered in θ_{rb} and θ_{kin} respectively, i.e., for each link i

$$\theta_{rb}^i = \begin{bmatrix} m_i & \xi_x^i & \xi_y^i & \xi_z^i & J_{xx}^i & J_{yy}^i & J_{zz}^i & J_{xy}^i & J_{xz}^i & J_{yz}^i \end{bmatrix}, \tag{3.10a}$$

$$\theta_{kin}^i = \begin{bmatrix} l_x^i & l_y^i & l_z^i \end{bmatrix}. \tag{3.10b}$$

For this model to be complete, the friction torque and the torque from gravity-compensating springs, if present, must be added to (3.9).

3.2.2 The Flexible Joint Dynamic Model

This model is an elastic, lumped parameter model. Consider the robot described in Section 3.2.1 with elastic gearboxes, i.e., elastic joints. This robot can be modeled by the so called flexible joint model which is illustrated in Figure 3.6. The rigid body rb^i is then connected to rb^{i-1} by a torsional spring-damper pair. The motors are placed on the preceding body. If the inertial couplings between the motors and the rigid links are neglected we get the simplified flexible joint model[10]. If the gear ratio is high, this is a reasonable approximation as described in, e.g., Spong (1987). The motor mass and inertia are added to the corresponding rigid body. The total system has $2N$ DOF. The model equations of the simplified flexible joint model are

$$M_a(q_a)\ddot{q}_a + c(q_a, \dot{q}_a) + g(q_a) = \tau_a, \tag{3.11a}$$

$$\tau_a = K(q_m - q_a) + D(\dot{q}_m - \dot{q}_a), \tag{3.11b}$$

$$\tau_m - \tau_a = M_m\ddot{q}_m + f(\dot{q}_m), \tag{3.11c}$$

where joint and motor angular positions are denoted $q_a \in R^N$ and $q_m \in R^N$, respectively. τ_m is the motor torque and τ_a is the gearbox output torque. $K \in R^{N\times N}$ is the diagonal stiffness matrix and $D \in R^{N\times N}$ is the diagonal matrix of dampers. The dynamic and kinematic parameters are still described by θ_{rb} and θ_{kin} but are for simplicity omitted in the equations. A vector of friction torques is introduced for this model, and described by $f(\dot{q}_m) \in R^N$. The friction torque is here approximated as acting on the motor side only. The convention of describing the equations on the link side is used, i.e., $q_m = q_m^a = r^{-1}q_m^m$ and $f(\cdot) = f_m^a(\cdot) = r^T f_m^m(\cdot)$

If the couplings between the links and the motors are included we get the complete flexible joint model (Tomei, 1991)

$$\tau_a = M_a(q_a)\ddot{q}_a + S(q_a)\ddot{q}_m + c_1(q_a, \dot{q}_a, \dot{q}_m) + g(q_a), \tag{3.12a}$$

$$\tau_m - \tau_a = M_m\ddot{q}_m + S^T(q_a)\ddot{q}_a + c_2(q_a, \dot{q}_a) + f(\dot{q}_m), \tag{3.12b}$$

$$\tau_a = K(q_m - q_a) + D(\dot{q}_m - \dot{q}_a), \tag{3.12c}$$

where $S \in R^{N\times N}$ is a strictly upper triangular matrix of coupled inertia between links and motors. The structure of S depends on how the motors are positioned and oriented relative to the joint axis directions.

[9] The *a* is explained by the fact that the link side can also be denoted the arm side and sometimes also the low-speed side. The motor side can also be denoted the high-speed side.

[10] Sometimes, the viscous damping is also neglected in the simplified model.

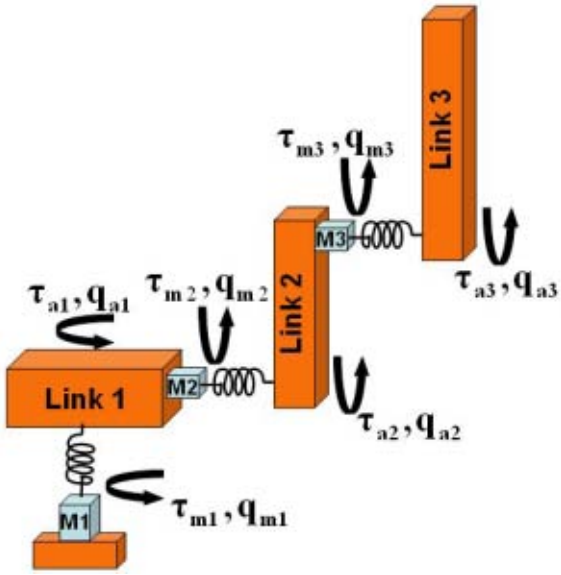


Figure 3.6: *A flexible joint dynamic model with 6 DOF.*

The flexible joint models can formally be derived in the same way as the rigid model, e.g., by Lagrange equations. The potential energy of the springs must then be added to the potential energy expressions as

$$V_s(q_a, q_m) = \frac{1}{2}(q_a - q_m)^T K(q_a - q_m), \tag{3.13}$$

and the kinetic energy of the rotating actuators must be added as well.

3.2.3 Nonlinear Gear Transmissions

The nonlinear characteristics of the gear transmission can have a large impact on the behavior of a robot manipulator, for example at low speed when the friction parameters with nonlinear speed dependency are dominant. A classical friction model[11] includes the viscous and Coulomb friction, f_v and f_c respectively, and is given by

$$f(v) = f_v v + f_c \,\mathrm{sign}(v), \tag{3.14}$$

where $v = \dot{q}_m$. More advanced friction models are described in, e.g., Armstrong-Hélouvry (1991). One model taking many phenomenon into account is the so called LuGre model, described in Canudas de Wit et al. (1995) and Olsson (1996). The LuGre model is a nonlinear differential equation modeling both static and dynamic behavior and

[11] All friction models described here are scalar models that can be used for each gear transmission.

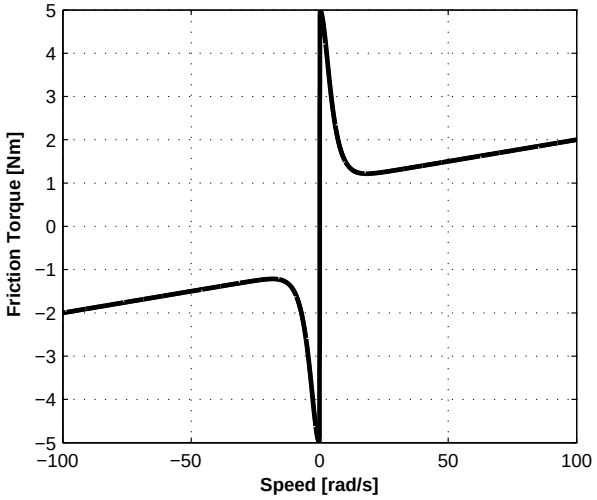


Figure 3.7: *Typical friction characteristics of a compact gearbox as described by* (3.16).

is described by

$$\frac{dz}{dt} = v - \frac{|v|}{g(v)}z, \tag{3.15a}$$

$$\sigma_0 g(v) = f_c + (f_s - f_c)e^{-(v/v_s)^2}, \tag{3.15b}$$

$$\sigma_1(v) = \sigma_1 e^{-(v/v_d)^2}, \tag{3.15c}$$

$$f(v) = \sigma_0 z + \sigma_1 \frac{dz}{dt} + f_v v, \tag{3.15d}$$

which in its simplified form has $\sigma_1(v) = \sigma_1$. The Stribeck friction is modeled by f_s. A smooth static friction law is suggested in Feeny and Moon (1994). This model avoids discontinuities to simplify numerical integration and is given by

$$f(v) = f_v v + f_c(\mu_k + (1 - \mu_k)\cosh^{-1}(\beta v))\tanh(\alpha v), \tag{3.16}$$

and is illustrated in Figure 3.7. Figure 3.8 shows the friction measured on one axis of an industrial robot under steady-state conditions, i.e., constant speed, in one configuration. In the same figure, a fitting of the steady-state LuGre model and the Feeny-Moon model to experimental data is shown. Both models describe the static behavior in a good way. However, the friction of a real robot shows large variations for different configurations in the workspace, for different tool loads, and for different temperatures. This means that some kind of off-line or on-line adaption of the model is necessary. An open question is whether the existing friction models can capture the dynamic effects of the compact gear boxes, together with motor bearings, typically used in industrial robots of today.

Another important nonlinear gearbox characteristics is the nonlinear stiffness. The nonlinear stiffness can also be included in the flexible joint model by replacing $K(q_m - q_a)$

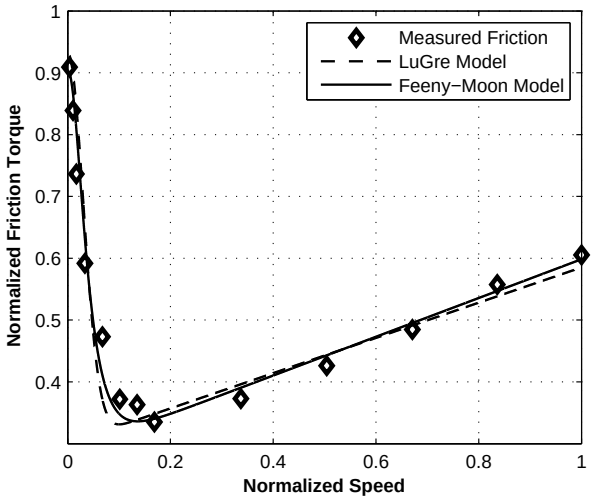


Figure 3.8: *One example of the friction characteristics for one axis of an industrial robot.*

in (3.11) by

$$\tau_s = \tau_s(q_m - q_a), \tag{3.17}$$

where $\tau_s(\cdot)$ is a nonlinear function describing a nonlinear spring. Three typical spring characteristics are shown in Figure 3.9. The first has a smooth nonlinear characteristics. This is typical for the compact gearboxes used by modern industrial robots. The second is an ideal linear spring and the third has a backlash behavior. Industrial gearboxes are designed for low backlash, i.e., a backlash that can be accepted with respect to the accuracy required. The backlash is, of course, not zero. More nonlinearities could be added to the model of the gear transmission, e.g., hysteresis and nonlinear damping. A smooth nonlinear stiffness is an important component of the benchmark problem described in Paper C, where the stiffness is modeled as a piecewise linear function.

3.2.4 The Extended Flexible Joint Dynamic Model

Most publications concerning flexible industrial robot manipulators only consider elastic deformation in the rotational direction. If only the gear elasticity is considered we get the flexible joint model, and if link deformation restricted to a plane perpendicular to the preceding joint is included in the model we get the flexible link model. One example of a flexible link model is described in, e.g., De Luca (2000). The article also describes the complete flexible joint model where inertial couplings between the motors and the links are taken into account. These restricted models are useful for many purposes and simplify the control design but limits, of course, the accuracy if used for simulation, and the performance if used for control. In Paper A and references therein it is shown that the flexible joint model must be extended in order to describe a modern flexible robot

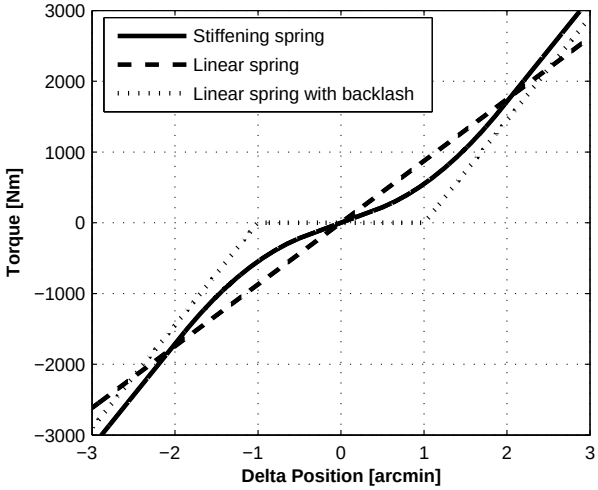


Figure 3.9: *Three typical gearbox stiffness characteristics.*

in a realistic way. The added elasticity can be of many different types, e.g., elasticity of bearings, foundation, tool, and load as well as bending and torsion of the links.

The extension of the flexible joint model is straightforward. First divide each link into two or more rigid bodies at proper locations, and connect the rigid bodies with multi-dimensional spring-damper pairs. The spring-damper pairs can have up to six degrees-of-freedom, and include both translational and rotational deflection. Then replace the one-dimensional spring-damper pairs in the actuated joints with the same type of multidimensional spring-damper pairs. In this way, non-actuated joints or pseudo-joints are added to the model. The principle is illustrated in Figure 3.10 where one extra torsional spring is added in the actuated third joint to model torsion of link three. Link three is then divided in two rigid bodies, and one more torsional spring is added to allow bending out of the plane of rotation. In this way two non-actuated joints are created. The model thus has 8 degrees-of-freedom, i.e., three motor coordinates q_{m1}–q_{m3}, three actuated joint coordinates q_{a1}–q_{a3}, and two non-actuated joints coordinates q_{a4}–q_{a5}. The number of non-actuated joints coordinates and their locations are, of course, not obvious. This model structure selection is therefore a crucial part of the modeling and identification procedure.

Generally, if the number of added non-actuated joints is M and the number of actuated joints is N, the system has $2N + M$ degrees-of-freedom. The model equations can then

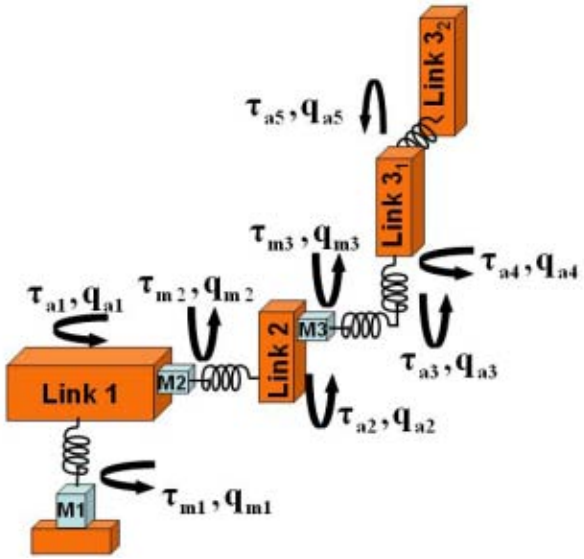


Figure 3.10: *An extended flexible joint dynamic model with 8 degrees-of-freedom.*

be described as

$$M_a(q_a)\ddot{q}_a + c(q_a, \dot{q}_a) + g(q_a) = \tau_a, \tag{3.18a}$$

$$\tau_a = \begin{bmatrix} \tau_g \\ \tau_e \end{bmatrix}, \tag{3.18b}$$

$$q_a = \begin{bmatrix} q_g \\ q_e \end{bmatrix}, \tag{3.18c}$$

$$\tau_g = K_g(q_m - q_g) + D_g(\dot{q}_m - \dot{q}_g), \tag{3.18d}$$

$$\tau_e = -K_e q_e - D_e \dot{q}_e, \tag{3.18e}$$

$$\tau_m - \tau_g = M_m \ddot{q}_m + f(\dot{q}_m), \tag{3.18f}$$

where $q_g \in R^N$ is the actuated joint angular position, $q_e \in R^M$ is the non-actuated joint angular position, and $q_m \in R^N$ is the motor angular position. $M_m \in R^{N\times N}$ is the inertia matrix of the motors and $M_a(q_a) \in R^{(N+M)\times(N+M)}$ is the inertia matrix for the joints. The Coriolis and centrifugal torques are described by $c(q_a, \dot{q}_a) \in R^{N+M}$, and $g(q_a) \in R^{N+M}$ is the gravity torque. $\tau_m \in R^N$ is the actuator torque, $\tau_g \in R^N$ is the actuated joint torque, and $\tau_e \in R^M$ is the non-actuated joint torque, i.e., the constraint torque. K_g, K_e, D_g, and D_e are the stiffness- and damping matrices for the actuated and non-actuated directions, with obvious dimensions. This model is from now on called the **extended flexible joint model**. The nonlinearities of the gear transmission described previously can also be added to this model. The non-actuated joint stiffness can also be modeled as nonlinear if required.

For a complete model including the position and orientation of the tool, X, the forward kinematic model of the robot must be added. The kinematic model is a mapping of $q_a \in$

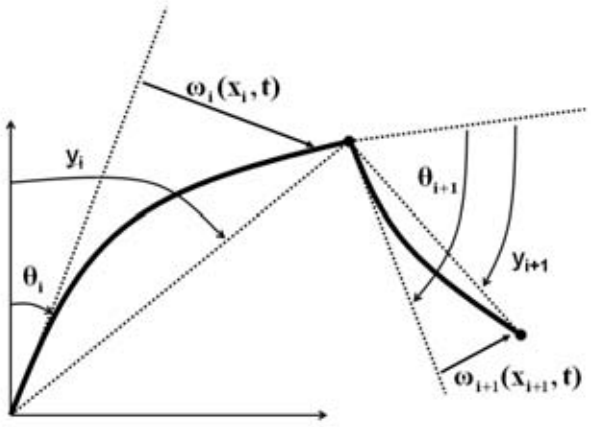


Figure 3.11: *The flexible link model.*

R^{N+M} to $X \in R^N$. The complete model of the robot is then described by (3.18) and

$$X = \Gamma(q_a). \tag{3.19}$$

Note that no inverse kinematics exists. This is a fact that makes the control problem considerably harder.

The identification of the extended flexible joint model is the subject of Paper A, and the inverse dynamics problem when using this model is treated in Paper B

3.2.5 Flexible Link Models

If the elasticity of the links cannot be neglected nor described by the joint elasticity approaches described in Sections 3.2.2 and 3.2.4, a distributed elasticity model can be used to increase the accuracy of the model. These models are described by partial differential equations with infinite dimension. One way to reduce the model to a finite-dimensional model is to use the assumed modes method. The link deflections are then described as an infinite series of separable modes that is truncated to a finite number of modes.

A model of this type is described in De Luca et al. (1998) and illustrated in Figure 3.11. The model consists of a kinematic chain of N flexible links directly actuated by N motors. It is assumed that the link deformations are small and that each link only can bend in one direction, i.e., in a plane perpendicular to the previous joint axis. The link deflection $w_i(x_i, t)$ at a point x_i along link i of length l_i is described by

$$w_i(x_i, t) = \sum_{j=1}^{N_{ei}} \phi_{ij}(x_i)\delta_{ij}(t), \quad i = 1 \dots N, \tag{3.20}$$

where link i has N_{ei} assumed modes $\phi_{ij}(x_i)$, and $\delta_{ij}(t)$ are the generalized coordinates.

By use of Lagrange equations the dynamic model for the system can be expressed as

$$\begin{bmatrix} B_{\theta\theta}(\theta) & B_{\theta\delta}(\theta) \\ B_{\theta\delta}^T(\theta) & B_{\delta\delta} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} c_\theta(\theta, \dot{\theta}, \dot{\delta}) \\ c_\delta(\theta, \dot{\theta}) \end{bmatrix} + \begin{bmatrix} 0 \\ D\dot{\delta} + K\delta \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \tag{3.21}$$

where $\theta \in R^N$ is a vector of joint angles, $\delta \in R^{N_e}$ is a vector of link deformations, and $N_e = \sum_{i=1}^{N} N_{ei}$. B is a partitioned inertia matrix, c a vector of Coriolis and centripetal forces, and K and D are stiffness and damping matrices, respectively, all with obvious dimensions. The actuator torque is denoted τ. The reference frame where the deflection is described is clamped to the base of each link. The end effector position can be approximated by the angles y_i as illustrated in Figure 3.11 and described by

$$y_i = \theta_i + \Phi_{ei}\delta_i, \tag{3.22}$$

where

$$\Phi_{ei} = \left[\phi_{i,1}(l_i)/l_i, \ldots, \phi_{i,N_{ei}}(l_i)/l_i\right], \tag{3.23}$$

and

$$\delta_i = \begin{bmatrix} \delta_{i,1} \\ \vdots \\ \delta_{i,N_{ei}} \end{bmatrix}. \tag{3.24}$$

Note that y can be regarded as output variable of this system, and that the same direct and inverse kinematic models as for the rigid or flexible joint can be used. This fact simplifies the inverse dynamics control problem. A fact that makes the inverse dynamics problem hard, however, is that the system from the actuator torque τ to the controlled variable y can have unstable zero dynamics[12], i.e., the system has non-minimum phase behavior (Isidori, 1995; Slotine and Li, 1991) which means that trajectory tracking is considerably harder to achieve.

The structure of the equations of motion and the assumed modes depend on the boundary conditions used, which is a critical choice for these models. Assumed mode models are further described in, e.g., Hastings and Book (1987), Book (1993), Book and Obergfell (2000), and Bascetta and Rocco (2002).

3.3 The Kinematics and Dynamics of a Two-Link Elbow Manipulator

In this section, the kinematic and dynamic models of a rigid two link elbow manipulator will be derived as a small but illustrative example of how the models used in this work can be derived. The manipulator is planar and constrained to movements in the x-y plane. By inspection of Figure 3.12, the forward kinematics is

$$\begin{aligned} X = \Gamma(q) = \begin{bmatrix} p_x \\ p_y \end{bmatrix} &= \begin{bmatrix} l_1 \sin(q_1) + l_2 \sin(\frac{\pi}{2} + q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(\frac{\pi}{2} + q_1 + q_2) \end{bmatrix} \\ &= \begin{bmatrix} l_1 \sin(q_1) + l_2 \cos(q_1 + q_2) \\ l_1 \cos(q_1) - l_2 \sin(q_1 + q_2) \end{bmatrix}. \end{aligned} \tag{3.25a}$$

The inverse kinematics can here be derived in closed form by either algebraic or geomet-

[12] For uniform mass distributions, the system is always minimum-phase.

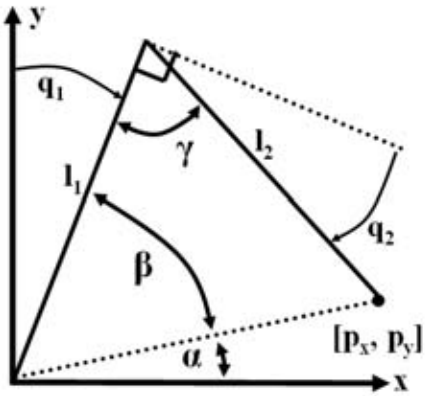


Figure 3.12: *Two link elbow manipulator kinematics.*

ric methods. Using the geometric approach and the law of cosine

$$\cos(\gamma) = \frac{l_1^2 + l_2^2 - p_x^2 - p_y^2}{2l_1 l_2} = \sin(q_2) \triangleq s_2, \tag{3.26}$$

and finally obtain the expression for q_2 by use of the $\mathrm{atan2}$ function

$$q_2 = \mathrm{atan2}(s_2, \pm\sqrt{1 - s_2^2}), \tag{3.27}$$

where the function $\mathrm{atan2}$ is preferred for numerical reasons. Continuing with the solution for q_2, in the same way

$$\cos(\beta) = \frac{l_1^2 + p_x^2 + p_y^2 - l_2^2}{2l_1\sqrt{p_x^2 + p_y^2}} = c_\beta, \tag{3.28a}$$

$$\beta = \mathrm{atan2}(\pm\sqrt{1 - c_\beta^2}, c_\beta), \tag{3.28b}$$

$$\alpha = \mathrm{atan2}(p_y, p_x), \tag{3.28c}$$

$$q_1 = \frac{\pi}{2} - \alpha - \beta. \tag{3.28d}$$

The inverse kinematics is given by (3.28d) and (3.27). The alternative signs in (3.27) and (3.28b) should be chosen as both plus or both minus corresponding to the two solutions, elbow-up and elbow-down.

The Jacobian of the velocity kinematics is obtained by differentiation of the forward kinematics (3.25a), i.e.,

$$J(q) = \frac{\partial \Gamma(q)}{\partial q} = \begin{bmatrix} l_1 c_1 - l_2 s_{12} & -l_2 s_{12} \\ -l_1 s_1 - l_2 c_{12} & -l_2 c_{12} \end{bmatrix}, \tag{3.29}$$

where the notations $\sin(q_1) \triangleq s_1$, $\cos(q_1) \triangleq c_1$, and $\cos(q_1 + q_2) \triangleq c_{12}$ etc. are used.

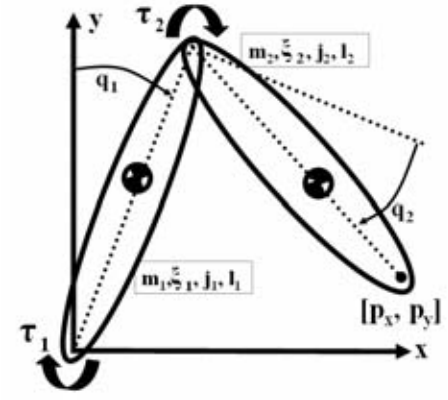


Figure 3.13: *Simplified two link elbow manipulator dynamic model.*

The dynamics is computed based on the simplified model in Figure 3.13. The dynamic and kinematic parameters are defined according to Section 3.2.1. The actuator inertial parameters are here included in the link parameters. The first step is to derive the kinematics for each link center of mass, i.e.,

$$X_1 = \Gamma_1(q) = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \xi_1 s_1 \\ \xi_1 c_1 \end{bmatrix}, \tag{3.30a}$$

$$X_2 = \Gamma_2(q) = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} l_1 s_1 + \xi_2 c_{12} \\ l_1 c_1 - \xi_2 s_{12} \end{bmatrix}, \tag{3.30b}$$

and the corresponding Jacobians

$$J_1(q) = \begin{bmatrix} \xi_1 c_1 & 0 \\ -\xi_1 s_1 & 0 \end{bmatrix}, \tag{3.31a}$$

$$J_2(q) = \begin{bmatrix} l_1 c_1 - \xi_2 s_{12} & -\xi_2 s_{12} \\ -l_1 s_1 - \xi_2 c_{12} & -\xi_2 c_{12} \end{bmatrix}. \tag{3.31b}$$

The rotational kinetic energy in this simple case is

$$K_{rot} = \frac{1}{2} j_1 \dot{q}_1^2 + \frac{1}{2} j_2 (\dot{q}_1 + \dot{q}_2)^2, \tag{3.32}$$

and the translational kinetic energy is generally

$$\begin{aligned} K_{trans} &= \frac{1}{2} \sum_{i=1}^{N} (m_i \dot{X}_i^T \dot{X}_i) \\ &= \frac{1}{2} \sum_{i=1}^{N} (m_i (J_i(q)\dot{q})^T J_i(q)\dot{q}) = \frac{1}{2} \dot{q}^T (\sum_{i=1}^{N} m_i J_i^T(q) J_i(q)) \dot{q}. \end{aligned} \tag{3.33}$$

Thus, the total kinetic energy is then

$$K = K_{trans} + K_{rot}. \tag{3.34}$$

The potential energy is given from the y-coordinate of each mass as

$$V = \sum_{i=1}^{N} (m_i g y_i), \tag{3.35}$$

where g is the gravitational constant. The Lagrange function is then

$$L = K - V. \tag{3.36}$$

Applying the Lagrange equation (3.8) and using some symbolic mathematical software, e.g., Matlab™ Symbolic Toolbox, gives the equations of motion as

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q), \tag{3.37}$$

where

$$M(q) = \begin{bmatrix} J_{11}(q) & J_{12}(q) \\ J_{21}(q) & J_{22}(q) \end{bmatrix}, \tag{3.38a}$$

$$J_{11}(q) = j_1 + m_1\xi_1^2 + j_2 + m_2(l_1^2 + \xi_2^2 - 2l_1\xi_2 s_2), \tag{3.38b}$$

$$J_{12}(q) = J_{21}(q) = j_2 + m_2(\xi_2^2 - l_1\xi_2 s_2), \tag{3.38c}$$

$$J_{22}(q) = j_2 + m_2\xi_2^2, \tag{3.38d}$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_2 l_1 \xi_2 c_2 (2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ m_2 l_1 \xi_2 c_2 \dot{q}_1^2 \end{bmatrix}, \tag{3.38e}$$

$$G(q) = \begin{bmatrix} -g(m_1\xi_1 s_1 + m_2(l_1 s_1 + \xi_2 c_{12})) \\ -m_2\xi_2 g c_{12} \end{bmatrix}. \tag{3.38f}$$

$$\tag{3.38g}$$

Note that the kinetic energy has the form

$$K = \frac{1}{2}\dot{q}^T M(q)\dot{q}, \tag{3.39}$$

as described in Section 3.2.1. If the rotational kinetic energy is included in (3.33), the inertia matrix of the system, $M(q)$, can be derived without applying the Lagrange equation. Deriving dynamic models is clearly a matter of formulating the kinematics, and then use a powerful tool for symbolic mathematics. Of course, the models can be much more complex than shown in this small example. Describing the problems and solutions for those cases is outside the scope of this work.

4

Identification of Robot Manipulators

This chapter gives a short introduction to system identification in general, and to the identification of robot manipulators in particular.

4.1 System Identification

System identification is the art of estimating a model of a system from measurement data. For a thorough treatment of system identification, the reader is referred to, e.g., Ljung (1999), Söderström and Stoica (1989), or Johansson (1993). An in-depth treatment of frequency-domain identification can be found in Pintelon and Schoukens (2001). Physical modeling and identification is treated in Ljung and Glad (1994). In the area of automatic control, the estimated models are often used for controller design, simulation, and prediction.

4.1.1 Introduction

The identification experiment can be performed in **open loop** or **closed loop**. Identification of a system not subject to feedback control, i.e, an open-loop system, is illustrated in Figure 4.1. This system has input u, output y, and is affected by a disturbance v. The disturbance can include measurement noise as well as other system inputs not included in u. An identification experiment on a system subject to feedback control, i.e., a closed-loop system, is shown in Figure 4.2 where r is the reference signal for the system. The reason for performing a closed-loop experiment could be, e.g., that the system is unstable, and must be controlled in order to remain stable. This is typically the case for a robot manipulator. If we take a robot manipulator as an example system, u is the actuator torque, y is the actuator position, and r is the position reference. The disturbance v includes both measurement noise and internally generated disturbances, e.g., torque ripple generated by

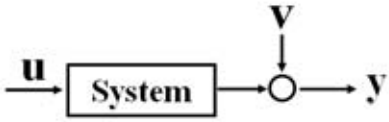


Figure 4.1: *An open-loop system.*

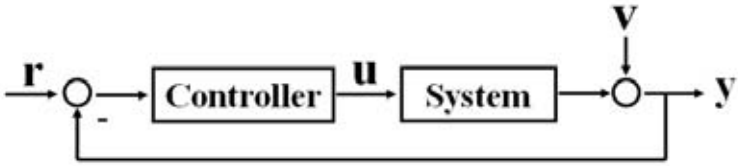


Figure 4.2: *A closed-loop system.*

the actuator[1].

Models can be divided into **nonlinear models** and **linear models**. A real world system is in general continuous and nonlinear. However, linear time-invariant approximations are often used for modeling the nonlinear reality. Therefore, in this section, we will only discuss identification of linear time-invariant systems.

Moreover, models can be described as **continuous-time models** or **discrete-time models** although the measurements, $u(t)$ and $y(t)$, normally are represented as sampled, discrete-time, data. Is is assumed that the reader has a basic knowledge of linear system theory for continuous-time and discrete-time systems. Some books treating this subject in-depth are, e.g., Kailath (1980) and Rugh (1996). Other recommended sources are Åström and Wittenmark (1996) and Ljung and Glad (2001).

The different types of models can further be divided into **non-parametric models** and **parametric models**. A non-parametric model is a vector of numbers or a graphical curve describing the model in the time-domain or frequency-domain. A parametric model is a model where the information obtained by the measurements has been condensed to a small number of parameters. The model is described as, e.g., a differential or difference equation or, in the case of a linear model, as a transfer function. In the next two sections, these two model types will be described.

4.1.2 Non-Parametric Models

Non-parametric models in the **time-domain** are, e.g., impulse responses or step responses. The model in these cases consists of vectors of system outputs and the corresponding time

[1]The torque ripple is not entering the system at the output which means that it must be filtered by some appropriate system dynamics before added to v.

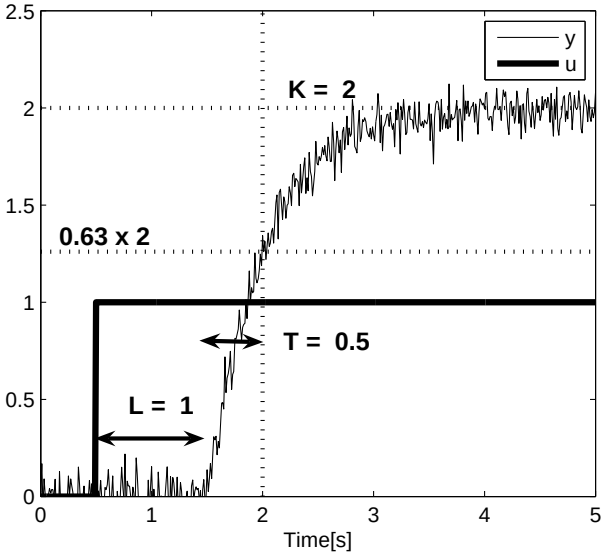


Figure 4.3: *Step response of a first order process with delay.*

stamps. One example of a step response is shown in Figure 4.3. The system is a first-order system with a time-delay, and the measured output is affected by measurement noise. The non-parametric model (the step response) can in this case be described by a parametric transfer function model

$$G(s) = \frac{K}{sT+1}e^{-Ls}. \tag{4.1}$$

This three-parameter model[2] is sometimes used to describe industrial processes. The parametric model (4.1) can be identified by inspection of the step-response according to Figure 4.3. This model can then be used for tuning of a PI-controller or a PID controller using, e.g., lambda tuning. Identification and control of industrial processes are treated in, e.g., Åström and Hägglund (2006). The described methodology of obtaining a parametric model from a non-parametric model is important, and the method considered in Paper A includes such a methodology, although based on a non-parametric frequency-domain method instead of a non-parametric time-domain method.

Methods for obtaining non-parametric **frequency-domain** models are, e.g., frequency-response analysis, Fourier analysis, and spectral analysis. The obtained model is a frequency response function (FRF) consisting of a vector of complex numbers and the corresponding frequency vector. The complex numbers represent the transfer function from input to output at different frequencies. The FRF can be illustrated in, e.g., a Bode diagram, a Nichols diagram, or a Nyquist diagram, and can be used directly for controller design with frequency domain methods, as loop-shaping using lead-lag compensation or QFT-design (Horowitz, 1991). The FRF can also be used for obtaining a parametric model as described in Paper A.

[2]Sometimes called the KLT model.

The method used in Paper A, for obtaining the FRF, is based on Fourier analysis and the discrete-time Fourier Transform. This transform, if the time-domain signal is $x(t)$, is defined by

$$X(e^{j\omega_k T_s}) = \sum_{n=-\infty}^{\infty} x(nT_s)e^{-j\omega_k nT_s}, \tag{4.2}$$

where T_s is the sample time and ω_k the frequency considered. As an approximation of this transform, when the measurement is a finite sequence of discrete-time data, i.e., sampled for $t = nT_s,\ n = 1, \ldots, N$, the discrete Fourier transform (DFT) is usually adopted. The DFT[3] is usually defined as

$$X_N(e^{j\omega_k T_s}) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} x(nT_s)e^{-j\omega_k nT_s}, \tag{4.3}$$

where

$$\omega_k = \frac{2\pi k}{NT_s},\ k = 1, \ldots, N. \tag{4.4}$$

For a thorough treatment of the discrete Fourier transforms, see, e.g., Oppenheim and Schafer (1975).

The DFT may contain errors, known as leakage errors, due to the fact that it is computed for a finite duration sequence of data. This error can be reduced by applying windowing functions before computing the DFT. The leakage error can be eliminated if the signals are of finite duration, and if the signals are sampled until the system is at rest. This type of excitation is called burst excitation. Another way of eliminating the leakage error is to use a periodic excitation, and to sample the signals for an integer number of periods, when a steady state is reached. For further discussions on these issues see Pintelon and Schoukens (2001).

The FRF[4] for a SISO system with the transfer function G, at frequency ω_k, can be estimated as

$$\hat{G}_N(\omega_k) = \frac{Y_N(\omega_k)}{U_N(\omega_k)}, \tag{4.5}$$

where $Y_N(\cdot)$ and $U_N(\cdot)$ are the DFTs of $y(\cdot)$ and $u(\cdot)$, respectively. In the following, we assume that no leakage errors are present because burst or periodic excitation has been used in the experiments. We further assume that a proper anti-alias[5] filtering is performed so that no alias errors are present. For notational simplicity, the arguments of $\hat{G}_N(\cdot)$, $Y_N(\cdot)$ and $U_N(\cdot)$ are written as ω_k, although $e^{j\omega_k T_s}$ would be more correct. If $\hat{G}_N(\cdot)$ is considered to be an estimate of a continuous-time model, $j\omega_k$ is the proper argument.

For a MIMO system described by the $n \times n$ transfer function matrix G with inputs u and outputs y, the following relation between the FRF of G, and the Fourier transforms of the input and output, U and Y, for frequency ω_k, holds

$$Y(\omega_k) = G(\omega_k)U(\omega_k). \tag{4.6}$$

[3]The fast Fourier transform (FFT) is an efficient way of computing the DFT.

[4]This estimated FRF is sometimes called the empirical transfer function estimate (ETFE).

[5]Alias or frequency folding occurs when sampling a continuous-time signal with a frequency content above half the sample frequency. The alias effect is further described in, e.g., Åström and Wittenmark (1996).

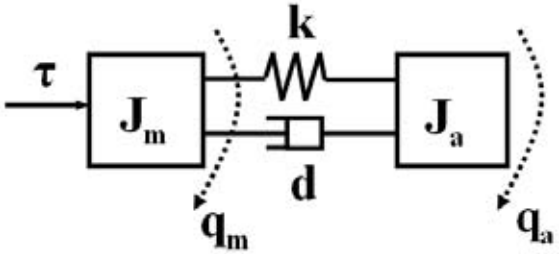


Figure 4.4: *A linear two-mass flexible joint model.*

If n independent experiments of length N are performed, the multivariable FRF can be estimated as

$$\hat{G}_N(\omega_k) = \bar{Y}_N(\omega_k)\bar{U}_N^{-1}(\omega_k), \tag{4.7}$$

where $\bar{U}_N(\omega_k)$ and $\bar{Y}_N(\omega_k)$ have n columns from the n experiments. If more than n experiments are performed, other FRF estimators can be applied, see, e.g., Pintelon and Schoukens (2001) or Wernholt (2007).

4.1.3 A Robot Example

A linear one-axis flexible joint model (see Section 3.2.2) will now be used as an example of how to obtain a FRF by closed-loop identification. The model, illustrated in Figure 4.4, can be described by the differential equations

$$0 = J_a\ddot{q}_a + k(q_a - q_m) + d(\dot{q}_a - \dot{q}_m), \tag{4.8a}$$
$$\tau = J_m\ddot{q}_m - k(q_a - q_m) - d(\dot{q}_a - \dot{q}_m), \tag{4.8b}$$

or by the transfer function

$$G(s) = \frac{s^2/\omega_z^2 + 2\zeta_z s/\omega_z + 1}{s^2(J_a + J_m)(s^2/\omega_p^2 + 2\zeta_p s/\omega_p + 1)}, \tag{4.9}$$

where

$$\omega_z = \sqrt{\frac{k}{J_a}}, \tag{4.10a}$$
$$\omega_p = \sqrt{\frac{k(J_a + J_m)}{J_a J_m}}, \tag{4.10b}$$
$$\zeta_z = \frac{d}{2}\sqrt{\frac{1}{kJ_a}}, \tag{4.10c}$$
$$\zeta_p = \frac{d}{2}\sqrt{\frac{J_a + J_m}{kJ_a J_m}}. \tag{4.10d}$$

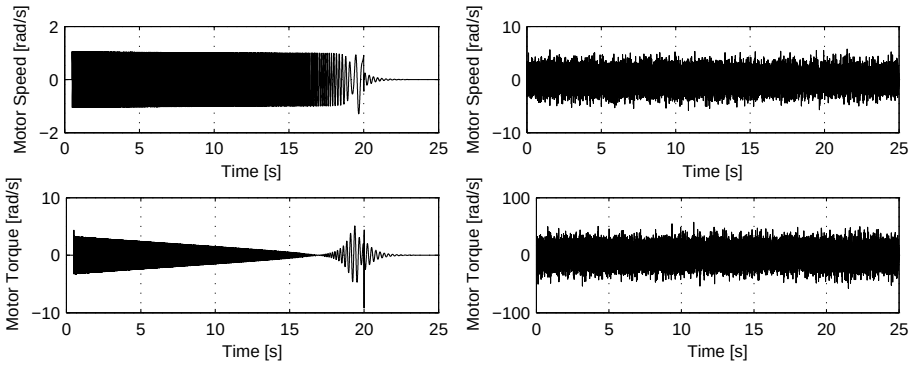


Figure 4.5: *Motor signals with chirp excitation. Left: no measurement noise, right: with measurement noise.*

The input u is the motor torque τ and the output y is the motor position q_m. J_a and J_m are the inertias of the arm and motor respectively, k is the joint stiffness, d is the joint viscous damping, and finally, q_a is the arm position. The measurement y is affected by measurement noise. The system is controlled by a speed controller of P-type with sample time 1 ms. The motor speed is obtained by differentiation of the measured position, and the excitation, i.e., the motor speed reference, is a swept sinusoid (chirp), starting at 50 Hz and ending at 1 Hz. The motor speed and torque signals, without and with measurement noise, are shown in Figure 4.5. Clearly, the noise level is very high compared to the excitation signal. Note that the closed-loop identification to be described, is performed with this high noise level.

Figure 4.6 shows the magnitude of the true system FRF and the estimated FRF obtained by applying (4.5). The FRF is computed from the torque input to the differentiated output, i.e., the motor speed. To reduce the influence of measurement noise, filtering in the frequency domain is applied. A triangular window of length 5, is applied in order to smooth the FRF. A negative effect of this can be seen if inspecting the anti-resonance, where the filtering increases the estimated damping.

In Figure 4.7, the measurement is stopped at $t = 21$ s, i.e., before the system has come to rest. However, the excitation is finished at $t = 20$ s. The leakage errors due to this clearly affects the identification.

Figure 4.8 shows a case where the chirp reference is turned off, i.e., the system is only excited by measurement noise. In this case the identified model has large errors and approaches a constant gain, -20 dB. This is the inverse controller gain as expected (see, e.g., Söderström and Stoica, 1989). A Bartlett window is used to reduce the leakage error in this case.

Finally, in Figure 4.9, a comparison is made between the model FRF of the continuous-time model, and of the discrete-time, zero-order hold sampled model. The FRFs are shown up to the Nyquist frequency 500 Hz. The FRFs are almost identical for lower frequencies. Some conclusions can be drawn from these examples:

- FRFs can be estimated even if the system runs in closed loop. To reduce bias and

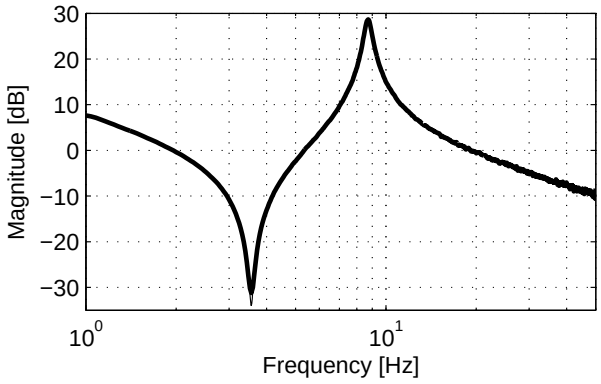


Figure 4.6: *Real FRF (thin line) and estimated FRF (thick line) of the two-mass model.*

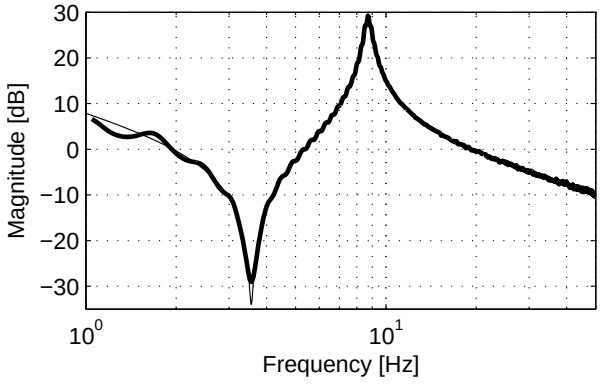


Figure 4.7: *Real FRF (thin line) and estimated FRF (thick line) of the two-mass model when the measurement time is too short.*

variance errors, the excitation level must be reasonably high compared to the level of disturbances.

- A burst excitation works fine if the measurements contain the whole system response, i.e., the system is at rest when measurements start and end.
- It might be necessary to smooth the estimated FRF if the disturbance level is high.
- If the sample frequency is well above the frequencies of the interesting process dynamics, continuous-time model FRFs can be directly compared to discrete-time estimated FRFs, e.g., if a parametric model is to be estimated from the FRF. If this is not the case, the discrete-time FRF of the model must be computed.

For further discussions on closed-loop versus open-loop identification, see, e.g, Ljung (1999). Periodic excitation is often recommended as an alternative way of obtaining a

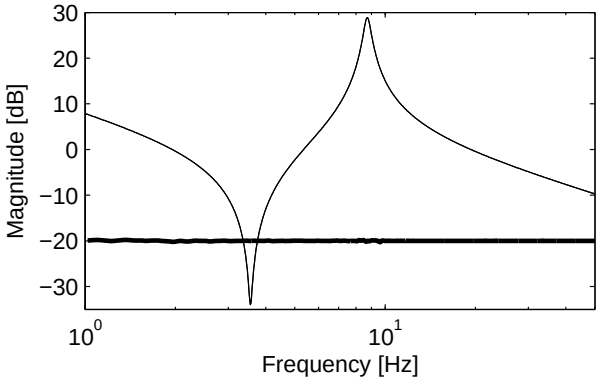


Figure 4.8: *Real FRF (thin line) and estimated FRF (thick line) of the two-mass model when the system is excited only by the measurement noise.*

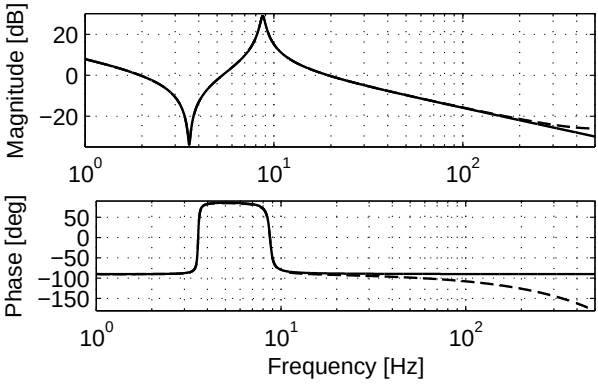


Figure 4.9: *FRF of the continuous (solid line) and discrete (dashed line) two-mass model.*

leakage-free FRF, see, e.g., Pintelon and Schoukens (2001). This is the adopted solution described in Paper A, where the excitation signal used is a sum of sinusoids, called multisine. The multisine excitation has, among many things, the advantage that the excitation energy is concentrated at selected frequencies, thus improving the signal-to-noise ratio at the frequencies where the DFT is evaluated.

4.1.4 Parametric Models

A parametric model is a model described as differential or difference equations. System identification is one route of obtaining a parametric model of a system. Another route is physical modeling, i.e., deriving a mathematical model from the basic laws of physics.

If the parameters of a physical model are known with sufficient accuracy, we get a **white-box model**. An example of such a model is a rigid-body model, as described in Section 3.2.1, where the kinematic and inertial parameters are known from the CAD models.

A **gray-box model** is a physical model where the model structure is known but where the physical parameters are unknown or only partly known. Identification of parameters in this case is called gray-box identification. One example of a gray-box model is a flexible joint model, as described in Section 3.2.2, where the rigid-body parameters are known, and the elastic parameters, consisting of springs and dampers, are unknown.

A third type of parametric model is the so called **black-box model**. In this case, the model structure is not known, and therefore, a model structure without any direct physical interpretation is used for the identification. The black-box model parameters can, e.g., be the coefficients of a standard linear difference equation.

The gray-box and black-box models can be identified directly in the **time domain**. There are a number of standard black-box model structures for discrete-time identification, where a noise model also is included. Examples of such standard structures are the output error structure, the ARX structure, and the Box-Jenkins structure. One way of estimating the parameters is by the use of a so-called prediction error method. When adopting this method, e.g., the least squares error between the predicted model output and the measured output is minimized. For some model structures the model can be estimated using linear regression but for most structures a numerical search procedure is necessary. For a thorough description of time-domain methods see, e.g., Ljung (1999).

The gray-box and black-box models can also be identified in the **frequency domain**. One way of doing this is to minimize the least squares error between the estimated nonparametric FRF and the parametric model FRF. The frequency-domain methods are described in, e.g., Pintelon and Schoukens (2001). Paper A treats frequency-domain gray-box identification of the unknown parameters in a continuous-time nonlinear system.

4.1.5 Summary

In system identification there are alternative choices concerning the experimental setup, excitation signals, model types, and identification methods, e.g.,

- Experimental setup
  - Open loop

  - Closed loop
- Excitation signal
  - Periodic, e.g., multisine, chirp or pseudo-random sequence
  - Non-periodic, e.g., chirp, random sequence, step, or impulse
- Model type
  - Non-parametric, e.g., step response or frequency response function (FRF)
  - Parametric, e.g., continuous-time linear gray-box, discrete-time linear black-box, or continuous-time nonlinear gray-box
- Identification method
  - Time-domain methods
  - Frequency-domain methods

There are many aspects of system identification not mentioned in this brief description, e.g., validation, experimental design, noise models, and model quality measures like bias and variance. Identification of nonlinear models is another subject not treated. The interested reader is referred to the literature cited in this section.

4.2 Identification of Robot Manipulators

This section briefly describes the identification of some models needed for the control and simulation of robot manipulators. The most important models for this work are the

- Kinematic model
- Rigid dynamic model
- Elastic dynamic model

Other important models subject to identification but not further described are, e.g., friction models, backlash and hysteresis models, thermal models, fatigue models, actuator models, and sensor models. For a more thorough survey of identification methods for robot manipulators, see, e.g., Kozlowski (1998) and Wernholt (2007).

4.2.1 Identification of Kinematic Models and Rigid Dynamic Models

The described models depend on a number of parameters. The kinematic link parameters consist of lengths and angles. The dynamic model also depends on the kinematic link parameters as well as the inertial link parameters. The accuracy of these models depends on the accuracy of these parameters. The parameter values could in some cases be obtained from the CAD models of the robot manipulator, and in other cases from measurements of the individual parts of the robot. If these methods are not accurate enough or simply not

possible to perform, then identification of the unknown parameters can be used to obtain the unknown parameter values. This type of identification is usually denoted gray-box identification as described in Section 4.1.4.

The nominal kinematic model of a large industrial robot typically gives a volumetric accuracy of 2–15 mm due to the tolerances of components and variations in the assembly procedure. This does not fulfill the accuracy requirement for off-line programming of, e.g., a spot-welding application. By identification of the kinematic parameters of the individual manipulator, as well as the elastostatic model, used for compensating the deflection due to gravity, a volumetric accuracy of ± 0.5 mm can be obtained.

It is also interesting to be able to identify the rigid dynamic model (2.1). This can be performed as a verification of the CAD model parameters when a new robot is developed or for direct use in the controller. The rigid dynamic model can be expressed as (Sciavicco and Siciliano, 2000)

$$\tau = H(q, \dot{q}, \ddot{q})\pi, \tag{4.11}$$

where π is a vector of the unknown dynamic parameters. The model thus has the property of linearity in the parameters. The parameters are not the same as the original link inertial parameters which, in general, cannot all be identified. The parameters in π are a set of uniquely identifiable parameters, and consist of different combinations of the physical parameters described in Section 3.2.1, e.g., $J_{yy} + m(\xi_x^2 + \xi_z^2)$. If measurements of torques and positions are performed along an exciting trajectory, the identification problem can be formulated as a linear regression

$$\overline{\tau} = \begin{bmatrix} \tau(t_1) \\ \vdots \\ \tau(t_N) \end{bmatrix} = \begin{bmatrix} H(t_1) \\ \vdots \\ H(t_N) \end{bmatrix} \pi = \overline{H}\pi, \tag{4.12}$$

and the solution is obtained as, e.g., the least-squares solution

$$\pi = (\overline{H}^T \overline{H})^{-1} \overline{H}^T \overline{\tau}. \tag{4.13}$$

Note that the speed and acceleration, if not measured, must be estimated from the measured positions. For the identification to be possible, the movements of the manipulator during the identification experiment must reveal information of all unknown parameters, i.e., the excitation must be rich enough. Furthermore, the movements should not excite the mechanical resonances of the manipulator. Identification of dynamic parameters is further described in, e.g., Swevers et al. (2007).

4.2.2 Identification of Flexible Dynamic Models

This section gives some examples of proposed identification methods for flexible robot manipulators.

Identification with No Additional Sensors

This group of methods uses the actuator torque and position only. Paper A in this thesis presents a method that belongs to this group.

Time-domain gray-box closed-loop identification of a linear SISO model of an industrial robot is described in Östring et al. (2003). The excitation used is multisine and chirp signals. Both inertial, elasticity and friction parameters are identified. A three-mass model is proposed for describing the elasticity.

A three-mass SISO model is also proposed and identified for an industrial robot in Berglund and Hovland (2000). The inertial and elasticity model parameters are computed using a general method, based on first estimating the frequency response function, and then solving an inverse eigenvalue problem. This method is extended to MIMO systems in Hovland et al. (2001).

A three-step procedure for identification of a SISO model is proposed in Wernholt and Gunnarsson (2006). First, the rigid body model and a nonlinear friction model are identified by use of a least-squares method. In the second step, the parameters of a linear three-mass model are identified by use of a frequency domain inverse eigenvalue method. The third step is a time-domain gray-box nonlinear prediction error method, and uses the parameters from the previous steps as initial values. In this last step, the parameters of a three-mass model with nonlinear gear-box elasticity are identified.

A MIMO model of an industrial robot describing two axes is identified in Johansson et al. (2000), using a chirp signal as excitation. The identification is performed in the time-domain using a subspace algorithm. A linear black box model combined with a nonlinear friction model is proposed.

Identification with additional sensors

In this case, additional sensors, e.g., force sensors or accelerometers are attached to the robot when performing the identification. In Pfeiffer and Hölzl (1995) the joint stiffness and damping as well as motor inertia are identified for a PUMA robot. In this case the links are fixed, and the torques between the links and the link fixations are measured with a force sensor. The motor is used for excitation.

In Behi and Tesar (1991), experimental modal analysis is used for parametric identification of an industrial robot. The elasticity model of the robot consists of four spring-damper pairs. In modal analysis, it is common to use an impact hammer as excitation. The vibrations are measured using accelerometers attached to the robot links. An introduction to modal analysis can be found in Avitabile (2001).

4.2.3 Identification of the Extended Flexible Joint Dynamic Model

In Paper A, an identification procedure for the unknown elastic parameters of the extended flexible joint model, described in Section 3.2.4, is proposed. The model is global and nonlinear. The identification procedure can be summarized as:

1. Local non-parametric models are estimated in a number of configurations. The models are frequency response functions, FRFs.
2. The nonlinear parametric robot model is linearized in each of these configurations.
3. The parametric FRFs of these linearized models are obtained for a value of the unknown parameter vector.

4. The model FRFs and the estimated non-parametric FRFs are compared and an error computed. The parameter vector is adjusted to minimize the error.

5. Repeat from 3 until some criteria is fulfilled.

The linearization procedure is further described in Wernholt (2007).

5

Control of Robot Manipulators

5.1 Introduction

Advanced motion control of robot manipulators has been studied by academic and industrial researchers since the beginning of the 1970's. A historical summary with many early references is given in Craig (1988).

The plant to be controlled, i.e., the manipulator, is an elastic multibody system. The system is multivariable and strongly coupled, and its highly nonlinear dynamics changes rapidly as the manipulator moves within its working range. Moreover, for a robot with gear transmissions, the gears have nonlinearities such as hysteresis, backlash, friction, and nonlinear elasticity. The actuators have non-ideal characteristics with internally generated disturbances, e.g., torque ripple disturbances. For a typical industrial robot, the position of the controlled variable, i.e., the tool, is not measured, and the only measured variable is the actuator position, i.e., the motor position. This position measurement could be impaired with a high level of measurement noise as well as deterministic disturbances. Summing up, robot control is a very difficult task with a lot of nonlinearities, noise, disturbances, and with no measurements of the controlled variable.

The dynamic position accuracy requirements can for some applications, e.g., laser cutting, be less than 0.1 mm at low speed, e.g., 20 mm/s. For high-speed applications, such as dispensing, the maximum allowed error can be 1 mm at 1000 mm/s, and no visible vibrations are allowed. In parallel with these requirements, there is also a need for high acceleration and speed. Thus, the control problem can, in general, not be solved by applying smooth trajectories to avoid exciting the mechanical resonances.

The conclusion of this is that the industrial manipulator control problem is a challenging task. The control of such manipulators can be described and classified in many ways, according to, e.g.,

- type of mechanical arm to control, e.g., serial link or parallel link robot.
- type of drive system, e.g., direct drive or gear transmission.

- type of model used for the (model-based) control, e.g., rigid models, flexible joint models, or flexible link models.
- controlled variable, e.g., position, speed, compliance, or force.
- motion type considered, e.g., high-speed continuous path tracking in open space, low-speed continuous path tracking in open space, point-to-point movement, tracking in contact with the environment, or regulation control.
- type of control law used, e.g.,
  - linear or nonlinear
  - feedback dominant or feedforward dominant
  - static or dynamic
  - discrete-time or continuous-time
  - robust or adaptive
  - diagonal or full-matrix
- type of measurements, e.g., actuator position, actuator speed, link position, link speed, link acceleration, link torque, tool position, tool speed, or tool acceleration.

This chapter is a survey of position control methods for articulated robot manipulators[1] described and suggested in the literature. The main focus will be on methods applicable to a typical industrial robot, i.e., a elastic manipulator with gear transmissions, where the only measured variables are the actuator positions. Many of the described methods assume that more variables are measured but can in many cases be modified such that actuator position only is sufficient. The actuator is assumed to be an electrical motor. The actuator dynamics as well as the current and torque control will not be treated. It is assumed that the torque control is ideal, and that the control signal is the motor torque. Furthermore, even though friction is the dominating source of error in some cases, e.g., at low speed, control methods specially designed for dealing with friction will not be covered. The emphasis will be on control methods for handling the elasticity of the manipulator.

A general controller structure is illustrated in Figure 5.1. Z_d is the desired tool trajectory described in Cartesian coordinates, and Z is the actual trajectory[2]. The reference and feedforward generation block (FFW) computes the feedforward torque u_{ffw} and the state references $\bar{x}_d$ used by the feedback controller (FDB). The manipulator has uncertain parameters, illustrated as a feedback with unknown parameters Δ, and is exposed to disturbances d and measurement noise e. The measured signals are denoted y_m. Note that the dimension of $\bar{x}_d$ and y_m may differ if some states are reconstructed by FDB. The purpose of the FFW is to generate model-based references for perfect tracking (if possible). The purpose of the FDB is, under the influence of measurement noise, to stabilize the system, reject disturbances, and to compensate for errors in the FFW.

[1] Results for manipulators of parallel linkage type are also applicable to a large extent for serial link manipulators and vice versa.

[2] Z is here used instead of X to denote the Cartesian position and orientation to avoid confusion with the states x of the system.

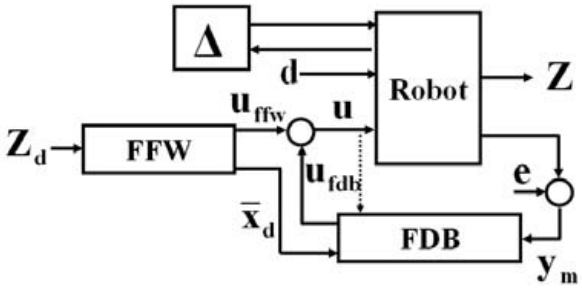


Figure 5.1: *Robot controller structure.*

5.2 Control of Rigid Manipulators

Although this work primarily is focused on flexible manipulators, the control of rigid manipulators is a good starting point. The main approaches used for flexible manipulator control are the same as for rigid manipulators. A rigid manipulator should here, from a control point of view, be interpreted as a manipulator with the lowest mechanical resonances well above the bandwidth of the control. A direct-drive manipulator with stiff links is an example of a rigid manipulator. In direct-drive manipulators, the motor axes are directly coupled to the links, and the negative effects of gear transmissions are eliminated, i.e., friction, backlash, and elasticity. The N-link manipulator has N degrees-of-freedom, and the Cartesian position Z, or the joint configuration q, can be regarded as output variable. Hence, the reference to the controller, i.e., the desired trajectory, is computed by the trajectory generator as joint angles q_d.

5.2.1 Feedback Linearization and Feedforward Control

A summary of control methods and experimental results for rigid direct-drive robots are, e.g., described in An et al. (1988). In the model-based approaches, the rigid dynamic model (3.9) is used.

The first method described is called independent joint PD control, and is illustrated in Figure 5.2. The controller for the direct drive manipulator is given by

$$u = K_p(q_d - q) + K_v(\dot{q}_d - \dot{q}), \tag{5.1}$$

where the measured position and speed are q and $\dot{q}$, respectively. The desired position and speed are q_d and $\dot{q}_d$, respectively. The control signal is the reference torque u. Finally, K_v and K_p are diagonal gain matrices with obvious dimensions. In this method, disturbances due to the multivariable coupling between the axes are regarded as unmodeled disturbances.

The second method shown in Figure 5.3 is called feedforward control, and is an extension of the PD controller with a feedforward torque u_{ffw} according to

$$u_{ffw} = M(q_d)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d), \tag{5.2a}$$

$$u = u_{ffw} + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q}). \tag{5.2b}$$

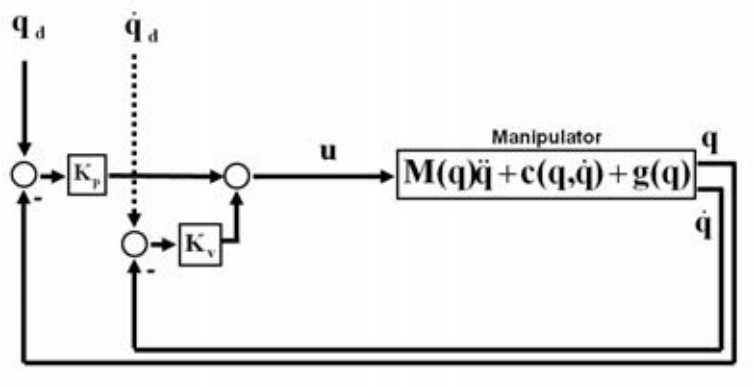


Figure 5.2: *Diagonal PD control (independent joint PD control).*

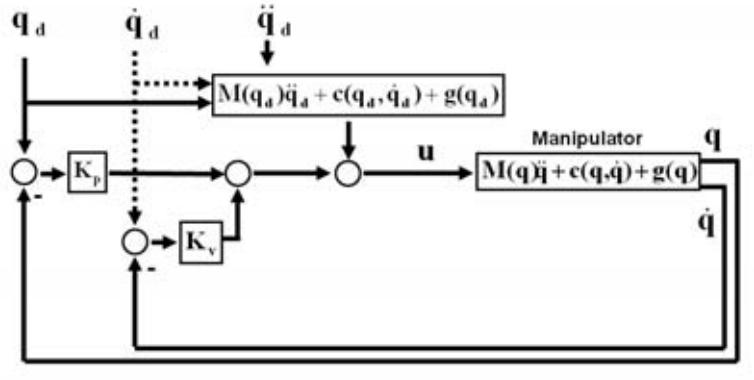


Figure 5.3: *Feedforward control.*

Note that the reference q_d must at least be twice differentiable.

The third method described is called computed torque control, and is illustrated in Figure 5.4. In this approach the system is first partly linearized by canceling the nonlinear dynamics $c(\cdot) + g(\cdot)$ with a feedback term. Then the system is finally linearized and decoupled by multiplying the controller output with the inverse system, i.e., the mass matrix. Ideally, the resulting system is a system of decoupled double integrators, i.e., $v = \ddot{q}$. The natural choice of controller, with output v, is again a PD controller. The first approaches of linearizing a nonlinear system by nonlinear feedback can be found in the robotics literature from the 1970's. The computed torque controller is

$$v = \ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q}), \tag{5.3a}$$

$$u = M(q)v + c(q, \dot{q}) + g(q). \tag{5.3b}$$

The terminology in this field is somewhat confusing. Computed torque can sometimes denote the feedforward control law, and sometimes the linearizing and decoupling

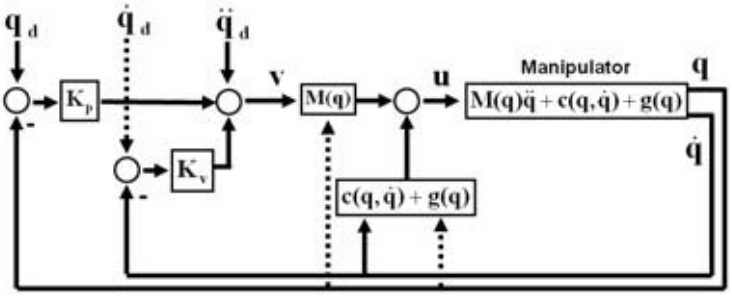


Figure 5.4: *Feedback linearization (computed torque control).*

control law. With standard control terminology, the control methods can also be described as *diagonal PD control*, *feedforward control*, and *feedback linearization control*. These terms will be used from now on. In both of these model-based methods it is necessary to solve the *inverse dynamics problem*, i.e., compute the torque from the desired trajectory. Also note that one part of the feedback linearization controller output is computed by the feedforward acceleration $\ddot{q}$.

The following results, concerning tracking errors, are reported (An et al., 1988) for the diagonal PD, feedforward, and feedback linearization control methods, when applied to the main axes of a direct-drive manipulator (the MIT serial-link direct-drive arm). A smooth fifth order polynomial trajectory with high speed and acceleration (360 deg/s and 850 deg/s^2) was used in the experiments.

- For two axes, the model-based controllers reduced the tracking error to 2 deg compared to 4 deg with PD control. No significant difference between feedforward and feedback linearization was noticed.

- The third axis had the same tracking error, 4 deg, for all three methods. This was explained by unmodeled motor dynamics and bearing friction in combination with low inertia.

- The sampling time was critical for the model-based methods. It affected the digitization error but also the possible level of feedback gain.

- The feedforward control method was believed to be the best choice for free space movements. For cases with large disturbances, the feedback linearization controller was believed to yield better results.

One reflection concerning these results is that the PD controller performance was surprisingly good. It should also be noted that a feedforward controller has the advantage that it, in principle, can even be computed off-line in order to save on-line computer load. Even if the feedforward is computed on-line, the sample rate of the feedforward computations does not affect the stability of the system. Thus, it might be possible to save computer load by using a lower sample rate since it will not reduce the stability. On the other hand, the

feedback linearization controller is a part of the feedback loop, and could cause instability if the model errors are large.

A more recent publication on the same topic is Santibanez and Kelly (2001). The conclusion in this and a number of other articles is the same as in the previously referenced book. Feedforward control gives the same tracking performance as feedback linearization and is the preferred choice.

Feedback linearization control and diagonal PD control is also evaluated w.r.t. tracking performance of a direct-drive manipulator in Khosla and Kanade (1989). The conclusion is that feedback linearization control gives better performance than diagonal PD control, and that it is important to include the centripetal and Coriolis terms in the linearization.

A final reflection on this topic is that the test cases studied in the referred articles are typically a few test trajectories in each article, and only with the best possible model. Feedback linearization and feedforward based on high-accuracy models should give the same tracking performance if the sample rate is chosen such that the discretization effects are negligible. It would be very interesting to see a comparative study concerning robustness to model errors and disturbance rejection, for the different published methods.

5.2.2 Other Control Methods for Direct Drive Manipulators

In this section, some examples of other control methods that have been considered for the control of direct-drive manipulators, are given.

Adaptive Control Adaptive control of direct-drive manipulators is studied in, e.g., Craig (1988). The model parameters of the feedback linearization controller described in the previous section are adapted on-line. An experimental study on two direct-drive axes of the Adept One robot is also included. It is shown that the constant gain default controller yields a better result than the adaptive controller. However, it is believed that more fine tuning and a new implementation of the adaptive controller concept could improve the result.

Nonlinear Robust Control A nonlinear outer-loop controller, replacing the PD controller, can be used to robustify the feedback linearization controller. One example of such a controller, based on *Lyapunov's Second Method*, is described in Spong et al. (2006). Another proposed controller is the *Sliding Mode Controller*. One example of this controller type used for robustification of a feedback linearization controller is described in Bellini et al. (1989).

A good summary of control methods for rigid manipulators is given in Spong (1996).

5.3 Control of Flexible Joint Manipulators

The flexible joint model, as described in Section 3.2.2, is a more realistic description of an industrial robot with gear transmissions. This model has elastic gear transmissions and rigid links. The N-link manipulator has $2N$ degrees-of-freedom and, as in the case of the rigid manipulator, the Cartesian position Z, or the joint configuration q, can be regarded

as the output variable. Hence, the reference to the controller, i.e., the desired trajectory, is assumed to be computed by the trajectory generator as joint angles q_d.

5.3.1 Feedback Linearization and Feedforward Control

Feedback linearization and feedforward control can be regarded as the main approaches for the control of flexible joint manipulators, and will therefore be treated in some detail.

Simplified Flexible Joint Model

In Spong (1987) and Spong et al. (2006), control methods for the simplified flexible joint model are discussed. In the simplified model (3.11), the inertial couplings between the links and the motors are neglected. Furthermore, the viscous damping is also neglected in order to simplify the controller design. In Spong (1987) it is shown that a manipulator described by this model can be linearized and decoupled by static feedback linearization. As for rigid manipulators described in the previous section, the flexible joint model can be used for feedforward control or feedback linearization. The feedforward approach is described in, e.g, De Luca (2000). The model is here given by

$$M_a(q_a)\ddot{q}_a + n(q_a, \dot{q}_a) + K(q_a - q_m) = 0, \tag{5.4a}$$

$$M_m\ddot{q}_m + K(q_m - q_a) = u, \tag{5.4b}$$

$$c(q_a, \dot{q}_a) + g(q_a) = n(q_a, \dot{q}_a), \tag{5.4c}$$

where the same notations as in Section 3.2.2 are used, except that the control signal, the motor torque, is denoted u.

The flexible joint manipulator is an example of a differentially flat system (Rouchon et al., 1993). Such a system can be defined as a system where all state variables and control inputs can be expressed as an algebraic function of the desired trajectory for a flat output, and its derivatives, up to a certain order. The flat output is the selected output variable of the system. Feedback linearization by static or dynamic state feedback is equivalent to differential flatness (Nieuwstadt and Murray, 1998). By solving (5.4a) for q_m, and differentiating twice, we get an expression for $\ddot{q}_m$, and adding (5.4a) to (5.4b) yields

$$u = \tau(q_a, \dot{q}_a, \ddot{q}_a, q_a^{[3]}, q_a^{[4]}) = M_a(q_a)\ddot{q}_a + n(q_a, \dot{q}_a) + M_m\ddot{q}_m, \tag{5.5a}$$

$$\ddot{q}_m = \ddot{q}_a + K^{-1}[M_a(q_a)q_a^{[4]} + 2\dot{M}_a(q_a, \dot{q}_a)q_a^{[3]} + \ddot{M}_a(q_a, \dot{q}_a, \ddot{q}_a)\ddot{q}_a + \ddot{n}(q_a, \dot{q}_a, \ddot{q}_a, q_a^{[3]})], \tag{5.5b}$$

where $x^{[i]}$ denotes $d^i x/dt^i$. These expressions fulfill the requirements for a differentially flat system with the flat output q_a.

By choosing the states

$$x = \begin{bmatrix} q_a \\ \dot{q}_a \\ \ddot{q}_a \\ q_a^{[3]} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \tag{5.6}$$

the system can be expressed in the following state-space form by the use of (5.5)

$$\dot{x}_1 = x_2, \tag{5.7a}$$
$$\dot{x}_2 = x_3, \tag{5.7b}$$
$$\dot{x}_3 = x_4, \tag{5.7c}$$
$$\dot{x}_4 = f(x) + g(x)u, \tag{5.7d}$$
$$y = x_1, \tag{5.7e}$$

where

$$\begin{aligned} f(x) = &-M_a^{-1}(x_1)KM_m^{-1}(M_a(x_1)x_3 + n(x_1, x_2)) \\ &- M_a^{-1}(x_1)(K + \ddot{M}_a(x_1, x_2, x_3))x_3 - \\ &2M_a^{-1}(x_1)\dot{M}_a(x_1, x_2)x_4 - M_a^{-1}(x_1)\ddot{n}(x_1, x_2, x_3, x_4), \end{aligned} \tag{5.8a}$$
$$g(x) = M_a^{-1}(x_1)KM_m^{-1}. \tag{5.8b}$$

It is clear that 4 differentiations of each component of the output y are needed in order for $y_i^{[4]}$ to depend directly on u, i.e., the relative degree $\nu_i = 4$. Now, $\Sigma\nu_i = 4N$ so the system has full relative degree and no zero dynamics associated with the output y (Isidori, 1995; Slotine and Li, 1991). Thus the system is fully linearizable by a static feedback control law that can be derived from the controller canonical form (5.7) as

$$u = g^{-1}(x_m)(v - f(x_m)), \tag{5.9}$$

where v is a new control signal for the linearized and decoupled system $q_a^{[4]} = v$ consisting of N independent chains of 4 integrators. For tracking control, v can be chosen as

$$v = q_{a,r}^{[4]} + L(x_r - x_m), \tag{5.10}$$

where $L \in R^{N\times 4N}$ is a linear feedback gain matrix, and x_r, x_m are the reference states and the measured states respectively. The fourth derivative of the reference trajectory must be defined, and is denoted $q_{a,r}^{[4]}$. This control law can also be derived by inserting the measured states x_m and v in (5.5) to yield

$$u = \tau(x_{1m}, x_{2m}, x_{3m}, x_{4m}, q_{a,r}^{[4]} + L_1(x_r - x_m)). \tag{5.11}$$

The derived control law is a combination of feedback and feedforward where the feedback part is dominating. The feedback gain matrix can be computed, e.g., by using LQ optimal control (Anderson and Moore, 1990).

It is also possible to find a feedforward-dominant control law according to

$$u = \tau(x_{1r}, x_{2r}, x_{3r}, x_{4r}, q_{a,r}^{[4]}) + L_2(x_r - x_m), \tag{5.12}$$

where, ideally in the case of a perfect model, all torques needed for the desired trajectory are computed by feedforward calculations, i.e., based on the reference states x_r. Note that the the desired trajectory q_d must be at least four times differentiable for both the feedback linearization and the feedforward control laws.

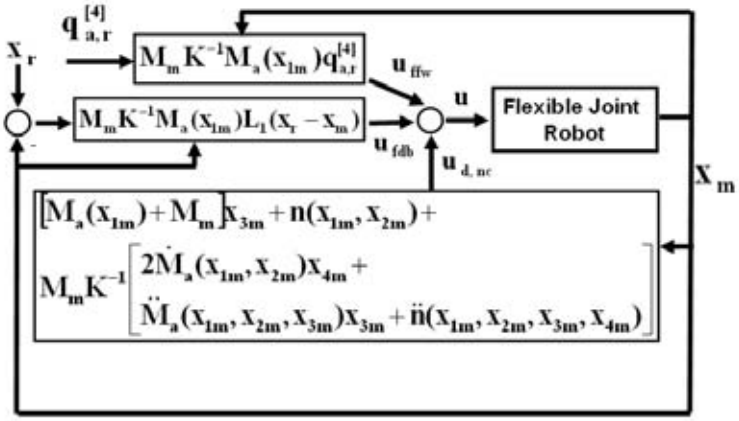


Figure 5.5: *Feedback linearization control law.*

The feedback linearization control law (5.11) gives constant bandwidth of the feedback controller for all robot configurations. For constant bandwidth there would be no need for gain scheduling. Due to the varying manipulator dynamics, gain scheduling, or some other model-based feedback gain computation is probably needed in the feedforward control law (5.12).

The two control laws are illustrated in Figures 5.5 –5.6. The control signal u can be described as

$$u = u_{d,nc} + u_{ffw} + u_{fdb}, \tag{5.13}$$

where $u_{d,nc}$ is the torque for decoupling and nonlinear cancelation, u_{ffw} is the feedforward torque, and u_{fdb} is the torque from the linear feedback controller. For feedforward control, $u_{d,nc} = 0$.

For feedback linearization control, u_{ffw} is probably considerably smaller than $u_{d,nc}$ that depends on the fast varying acceleration and jerk[3] as well as on the slower varying gravity and speed-dependent terms. This means that even in the case of a perfect model, only a small part of the torque is generated by feedforward. In feedback linearization of a rigid manipulator, as described in Section 5.2, $u_{d,nc}$ consists of the slowly varying gravity torque and the speed-dependent torques. The fast varying acceleration torque is is included in u_{ffw}. Thus, it can be expected that u_{ffw} is smaller part of the total torque u for a flexible manipulator than for a rigid manipulator[4]. This means that the tracking errors, due to the inevitable time delay caused by the discrete implementation, are larger for a flexible manipulator than for a rigid manipulator. Time delays causes the feedback linearization to be only partial.

[3] Jerk is a common name for the time derivative of the acceleration.

[4] To understand this, consider the feedback linearization control law applied on a rigid one-axis manipulator with no friction or gravity, i.e., on a double integrator. If the model inertia is correct, all required torque will be included in u_{ffw}. $u_{d,nc}$ will, of course, be zero.

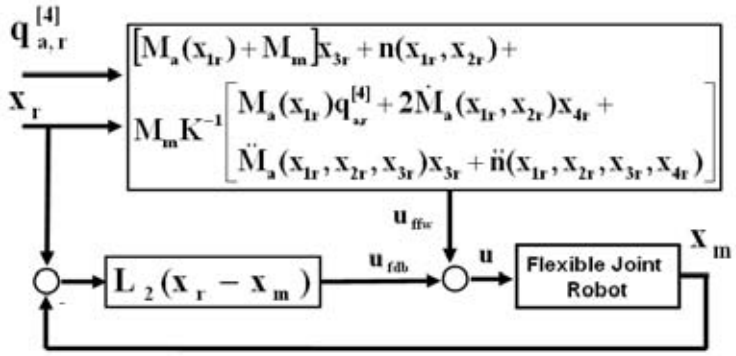


Figure 5.6: *Feedforward control law.*

Complete Flexible Joint Model

The complete flexible joint model (3.12) cannot be linearized by static state feedback. However, if the viscous damping is excluded, any flexible link model can be linearized and decoupled by dynamic state feedback as shown in (De Luca, 1988; De Luca and Lanari, 1995; De Luca and Lucibello, 1998; Isidori and De Luca, 1986). The static feedback controller described in the previous section has the form

$$u = \alpha(x) + \beta(x)v, \tag{5.14}$$

where x are the states, v are the new control signals for the linearized system, and $\alpha(\cdot)$, $\beta(\cdot)$ are nonlinear functions of appropriate dimensions. A dynamic feedback controller has the form

$$\dot{\xi} = \alpha(x, \xi) + \beta(x, \xi)v, \tag{5.15a}$$
$$u = \gamma(x, \xi) + \delta(x, \xi)v, \tag{5.15b}$$

where ξ are the internal states of a feedback compensator.

The proof that the complete model is feedback linearizable, is based on the fact that the system is invertible with no zero dynamics. Given a desired trajectory, q_d, and its derivatives up to a certain order, the motor states and the required torques can be computed recursively. The solution is based on the fact that S in (3.12) is upper triangular. This result can also be used for feedforward control based on the complete model (De Luca, 2000). Static or dynamic feedback linearization can also be performed when the damping term is included but in this case only input-output linearization is possible (De Luca et al., 2005). This is possible since the zero dynamics is stable.

The complete model increases the complexity of the linearization procedure considerably. The requirement on the smoothness of q_d is high using these control laws. If the manipulator has N links, q_d must be $2(N + 1)$ times differentiable compared to 4 times for the simplified flexible joint model as described in Section 5.3.1.

State Estimation

All states must be available for feedback in order to use the control laws (5.11) and (5.12). However, the feedforward control law (5.12) can, of course, be combined with, e.g., a PD controller for the actuator position, which only requires the motor states to be available (De Luca, 2000). The motor position is available for all industrial manipulators considered, and the motor speed can be estimated by, e.g., differentiation of the motor position. This means that a simplified version of the feedforward control law is possible to evaluate on a standard industrial robot, if the required computer capacity is available.

The feedback linearization control law (5.11) requires the link position, speed, acceleration, and jerk for each link to be measured. This means that some of the states, e.g., the jerk, must be estimated from the available measurements. The estimated states will then depend on the model, and this will certainly reduce the performance and robustness of the control law, compared to full-state measurements. Even in the case of measurements of motor position, motor speed, link position, and link speed, model parameters are needed in order to compute the four link states required by the control law.

Estimation of states can be performed by observers[5]. Nonlinear observers are treated in, e.g., Isidori (1995) and Robertsson (1999).

A nonlinear observer for the motor position, motor speed, link position, and link speed is suggested in De Luca et al. (2007). The observer works for both the simplified and the complete flexible joint model, and requires measurements of motor position and link acceleration. The observer does not depend on the inertial link parameters but on the motor inertias, the joint spring-damper pairs, and the kinematic link parameters describing the accelerometer locations. For an N-link manipulator, M accelerometers are needed, with $M \geq N$. The observer is characterized by a linear and decoupled error dynamics. The observer is evaluated on the three main axes of an industrial robot, as well as in closed loop with the feedforward control law decribed in Section 5.3.1. The result is significantly improved with respect to damping and overshoot, compared to a control law using motor states only. More references on nonlinear observers for flexible joint robots can be found in De Luca et al. (2007).

5.3.2 Linear Feedback Control

Some examples of linear controllers suggested for flexible joint manipulators when only the motor position is measured are

PD A PD controller for the motor position is suggested in Tomei (1991). Here it is also proved that the suggested controller globally stabilizes the manipulator around a reference position if gravity compensation is used.

LQG LQG control is a natural way of approaching the control problem in the case where the actuator positions are measured but the control objective is to control

[5] One well-known type of observer for linear systems is the Kalman filter where the observer gain is computed based on a stochastic description of the measurement- and system-disturbances. For nonlinear systems, the extended Kalman filter, based on a linearized system, can be used. The gain of an observer with the same structure as the Kalman filter can also be determined by pole placement of the observer error dynamics. Another type of observer is the reduced observer, sometimes called Luenberger observer.

the tool. This approach is described in, e.g., Elmaraghy et al. (2002) and Ferretti et al. (1998).

5.3.3 Experimental Evaluations

This section presents some reported experimental evaluations of control methods for flexible joint manipulators.

In Swevers et al. (1991), an industrial robot from KUKA is used for evaluation of an improved trajectory generation, and a model-based control concept. The robot was equipped with three extra encoders in order to measure the link positions of the main axes. The motor positions were measured by the standard controller. A state feedback controller combined with feedforward control based on a flexible joint model was evaluated and compared with the standard controller for this robot. The trajectory generation was also modified to yield a trajectory based on a 9th order polynomial. The performance of the standard- and model-based controllers were evaluated using some test cases[6] defined in the industrial robot test standard ISO 9283 (ISO, 1998). The tests showed significant improvements compared to the standard KUKA industrial controller implementation. The conclusion was that the smooth trajectory from the new trajectory generation contributed most, but also that the new flexible controller improved the performance at very high velocities and accelerations.

Some problems with feedback linearization are mentioned in Jankowski and Van Brussel (1992a), such as the complexity of the control laws, the need for measurement or estimation of link acceleration and jerk, and the need for high sampling frequencies. The suggested solution is a discrete-time formulation of the inverse dynamics which requires the solution of an index 3 differential algebraic equation. Some experimental results are presented in Jankowski and Van Brussel (1992b).

In Caccavale and Chiacchio (1994), an experimental evaluation on an industrial robot with gear transmission is reported. The robot is the SMART-3 6.12R robot by COMAU. A feedforward torque was added to the conventional PID controller output. Sample times were 1 ms and 10 ms for the PID controller and the feedforward computation, respectively. The feedforward torque calculations are based on a rigid dynamic model in identifiable form, as described in Section 4.2.1. The path error was decreased from 3.4 mm to 1.1 mm at a speed of $2\,\mathrm{rad/s}$ and an acceleration of $6\,\mathrm{rad/s^2}$. It was concluded that the diagonal inertia terms are the most important terms for use in feedforward.

In Grotjahn and Heimann (2002), model-based control of a KUKA KR15 manipulator is described. It is concluded that the nonlinear multibody dynamics and the nonlinear friction are the dominating reasons for path deviation, and that the elasticity does not need to be considered. No torque interface to the controller was available. Instead a path correction interface is used together with a feedforward control method called nonlinear precorrection based on an identified rigid body model including friction. Improvement by learning control and training of the feedforward controller is discussed and evaluated. Learning control can compensate for all deviations whereas the feedforward training only compensates for the modeled effects. Thus, learning gives better performance but is very sensitive to path changes after the learning where the feedforward training is more

[6] Settling time, overshoot, and path-following error according to the ISO standard were evaluated.

roboust. The different algorithms improve the path-following in a path defined by the ISO 9283 standard.

A full-state feedback controller is presented in Albu-Schäffer and Hirzinger (2000). The motor position q_m and the joint output torque τ are measured for each joint. By numerical differentiation, the state vector $x = \begin{bmatrix} q_m^T & \dot{q}_m^T & \tau^T & \dot{\tau}^T \end{bmatrix}^T$ is obtained. The state feedback controller is diagonal, i.e., based on an independent joint approach that neglects the disturbance due to the strong coupling. Gravity and friction compensation is added to the controller, and a gain scheduling based on the diagonal inertial terms is also suggested. A Lyaponov-based proof of global stability is given. The experimental evaluation is performed on a DLR light-weight robot and shows that the bandwidth of the proposed controller is twice the bandwidth of a motor PD controller, given the same damping requirement. This type of controller is further developed and analyzed in Le Tien et al. (2007) and Albu-Schäffer et al. (2007).

5.4 Control of Flexible Link Manipulators

This section describes some proposed control methods for the flexible link model described in Section 3.2.5. The choice of coordinates and reference frames is not unique and there is more than one possible approximate description of the same system (Book, 1993). If the actuator torque is applied at one point of the distributed structure, and the response is measured at another point, the system is said to be noncollocated[7]. If the finite dimensional approximation of a beam-like noncollocated system has a sufficiently large number of assumed modes, the system description will be non-minimum phase. This means that if the Cartesian end effector position is to be controlled, the system can be non-minimum phase, and the inverse dynamics is then hard to obtain. The control methods described in this section are in general feedforward control methods combined with a feedback controller of, e.g., PD-type.

If the desired Cartesian trajectory Z_d is known for $t \in \begin{bmatrix} 0, & T \end{bmatrix}$, the desired beam tip angles y_d, in Figure 3.11, can be computed by the inverse kinematics. In De Luca et al. (1998) it is shown that the state trajectories and the control torque can be obtained by solving an ordinary differential equation (ODE). The problem is to find a bounded solution to this ODE as the system normally is non-minimum phase, i.e., no bounded causal solutions to the inverse dynamics exist. Three different methods for solving the problem are suggested, and one is experimentally evaluated. This method finds a non-causal solution by applying iterative learning control for time $t \in \begin{bmatrix} -\Delta, & T + \Delta \end{bmatrix}$.

A different approach for the problem of point-to-point motion is described for a one-link manipulator in De Luca and Di Giovanni (2001a). Here, an auxiliary output is designed and used as output variable of the system. In this case the angle y_d points to a location where the system is minimum phase, but close to non-minimum phase. In De Luca and Di Giovanni (2001b) the same problem is solved for a two-link manipulator of which one link is flexible.

A method based on dividing the inverse system in a causal and an anti-causal part is presented in Kwon and Book (1990). The method is limited to linear systems, i.e., one-

[7] A collocated system measures the response at the same location as the actuator torque is applied.

link manipulators. In De Luca and Siciliano (1993) it is shown that a PD controller with gravity feedforward is global asymptotically stable for the flexible link manipulator.

The robust control problem of a four-link flexible manipulator is treated in Wang et al. (2002). An H_∞ controller for regional pole-placement is designed for the uncertain linearized system. The controller is evaluated by simulation, and tests on an experimental manipulator show that the proposed controller has better performance than an LQR controller.

To summarize, the control of the flexible link manipulator is complicated by the fact that the system can be non-minimum phase. There are several alternatives for solving the problem, e.g.,

- Find a stable non-causal solution of the inverse dynamics as described in the references of this section.
- Choose a new output so that the system becomes minimum phase, e.g., the joint angle for a flexible link robot. The new output should be a reasonable approximation of the original output.
- For a linear discrete-time system, the Zero Phase Tracking Controller (Torfs et al., 1991; Tung and Tomizuka, 1993) can be used.
- In Aguiar et al. (2005) a reformulation from tracking to path-following[8] is suggested. By parameterizing the geometrical path X by a path variable θ, and then selecting a timing law for θ, the inverse dynamics for the non-minimum phase system can be stabilized.

5.5 Industrial Robot Control

As described in Section 1.1, the control algorithms used by robot manufacturers are seldom published. This section gives a few examples of known facts about the control of commercial robots[9], taken from public information sources.

In Grotjahn and Heimann (2002) it is claimed that the feedback controller of a KUKA robot is a cascaded controller with an inner speed controller of PI type with sample time 0.5 ms. The outer loop is a position controller with 2 ms sample time. Only the position is measured, and the speed is estimated by differentiation and low-pass filtering. The controller can thus be described as a diagonal PID controller.

A sliding mode controller based on an elastic model of two-mass type is suggested in Nihei and Kato (1993) by FANUC. The motivation is the reduction of vibrations in short point-to-point movements in, e.g., spotwelding applications.

A recent patent by Fanuc (Nihei et al., 2007) describes a method for reduction of vibrations in robot manipulators. The method is based on observer estimation of the link position. The estimated variable is then used in a controller of internal model control (IMC) type. The controller is based on a linear elastic SISO model of two-mass type.

[8] The path-following error is in fact a more relevant performance measure for industrial applications as reflected by the ISO 9283 standard.

[9] The examples are restricted to the four major robot manufacturers, i.e., Fanuc, Motoman, ABB, and KUKA.

On their web-site (Fanuc, 2007), the new Fanuc controller R-J3iC offers *enhanced vibration control* as a feature.

The motion control of ABB robots is described as model-based in Madesäter (1995). The accuracy in continuous path tracking is claimed to be very high. The model-based controller is implemented in a controller functionality denoted *TrueMove*. A newly released robot has an improved version of this functionality called *the second generation of TrueMove* (ABB, 2007).

Finally, Motoman has a control concept called *Advanced Robot Motion (ARM) control* for high-performance path accuracy and vibration control (Motoman, 2007).

Clearly, high performance motion control is important for the industrial robot manufacturers. The actual algorithms used are hard to reveal, and an article or a patent does not mean for sure that the described technology is actually used in the product.

It is also clear that the performance of industrial robots can be very high. This indicates that advanced concepts for motion control are used. Some examples of obtained performance for the "best in class" commercial robots during optimal-time movements:

- Path-following error of large robots with considerable link- and joint flexibilities and a payload of more than 200 kg can be in the order of 2 mm at 1.5 m/s according to the ISO 9283 standard, i.e., when the maximum path-following error is considered. The mean error is considerably smaller, about 0.5 mm.
- The corresponding errors for a medium-sized robot with a payload of 20 kg could be maximum 0.5 mm and mean 0.1 mm.
- A positioning time of 0.25 s in 50 mm point-to-point movements for the large 200 kg payload robot described above. Time is measured from start of movement until the tool is inside an error band of 0.3 mm. This means that there are almost no vibrations or overshoots when reaching the final position.
- Dynamic position accuracy better than 0.15 mm considering maximum error, and with a mean value of 0.05 mm at 50 mm/s in a laser-cutting applications when using iterative learning control.

5.6 Conclusion

Control of flexible joint and flexible link manipulators is a large research area with numerous publications, and almost every possible control method ever invented has been suggested for dealing with these systems. This survey has described only the main approaches.

Some examples of methods not treated in this thesis are iterative learning control, adaptive control, backstepping, sliding mode control, neural networks, singular perturbations, composite control, pole placement, input shaping, passivity-based control, and robustification by Lyaponov's second method. Two survey articles with many references are Sage et al. (1999) and Benosman and Le Vey (2004). A good description of the most important control methods considered can be found in Spong et al. (2006).

It is clear that the theoretical foundation of the control methods, and the ability to prove stability, is in focus for many academic robot control researchers. Evaluation of

nominal and robust performance of the proposed methods, is often neglected. From the reported simulation studies and experimental evaluations, it is in fact quite hard to judge what the attainable performance is for the different methods. It is true that it can be hard to use commercial robots for control evaluations, and that some methods require a larger computer capacity than currently available in present robot controllers. However, simulation studies are always possible to perform. For example, to the author's knowledge, no rigorous simulation or experimental study of feedback linearization for the simplified flexible joint model has been published. A comparative study, with strict industrial requirements, of at least the main approaches described in this section, i.e., feedback linearization and feedforward control, would be very valuable.

One further comment is that an integral term is most certainly needed in order to handle model errors and disturbances in a real application. The reason for avoiding the integral term in many publications is probably motivated mainly by the need to prove stability.

The following facts regarding the tracking and point-to-point performance is at least indicated in the experimental results presented in this survey:

1. Model-based control improves the performance of an industrial-type robot.

2. A rigid model can improve the performance although the robot has elastic gear transmissions.

3. A flexible joint model improves the performance even more. It is not clear whether the nonlinear model should be used in the feedback loop, i.e., feedback linearization, or in the feedforward part of the controller.

4. More measurements improve the result, e.g., measurement of link position or acceleration.

6

Concluding Remarks

This first part of the thesis has served as an introduction to modeling and control of robot manipulators. The aim has been to show how the included papers in Part II relate to the existing methods and to motivate the need for the research presented. In Section 6.1 the conclusions are given. A number of areas are still subject to future research and some ideas are discussed in Section 6.2.

6.1 Conclusion

This thesis has investigated some aspects of modeling and control of elastic manipulators. The work is motivated by the industrial trend of developing weight- and cost-optimized robot manipulators. A large amount of applied research is needed in order to maintain and improve the motion performance of such manipulators. The approach adopted in this thesis is to improve the model-based control by developing more accurate elastic models. A model, called the *extended flexible joint model*, is suggested for use in motion control systems as well as in design and performance simulation. Different aspects of this model are treated in this thesis.

Paper A: A procedure for multivariable identification of the unknown elastic model parameters in a gray-box model is proposed. The unknown parameters describe a number of spring-damper pairs. The same procedure is applied for identifying these parameters of a real six-axes industrial robot. The result is surprisingly good although the uncertainties in the dampers are large.

Paper B: The inverse dynamics of the model is studied, and a DAE formulation is proposed. A method for solving the DAE is presented, and the feasibility of the method is demonstrated in a simulation study. The control application that is studied is trajectory tracking, and the proposed control method is multivariable feedforward

control based on the inverse dynamics method. The model used is a small but realistic two-link manipulator. Many problems need to be solved before the method can be applied to a general large manipulator model of this type.

Paper C: Robust feedback control of a one-axis four-mass model is studied. This model can be seen as a one-axis version of the extended flexible joint model. The problem studied is a somewhat neglected problem in the academic robot research. It concerns disturbance rejection for an uncertain elastic robot manipulator using a discrete-time controller. A benchmark problem is described, and some proposed solutions are presented and analyzed. The conclusion is that it is hard to improve the result of a PID-controller, and that the QFT design methodology is a powerful tool for robust control design of, at least, SISO systems. Methods optimizing some performance criteria for fixed-structure controllers are also useful for solving this problem.

6.2 Future Research

Some ideas for future research in the areas covered by this thesis are the following:

- **Identification**
  - Identification methods with minimum energy, i.e., minimum time and minimum amplitude, are interesting to study.
  - Identification of nonlinearities requires more research. Typical nonlinearities are friction, nonlinear stiffness, and hysteresis.
  - Improvement of the described identification method by the use of additional sensors, e.g., accelerometers.
  - Design and identification of a friction model that can describe the static and dynamic behavior of the compact gearboxes typically used in the robotics industry today.
- **Feedforward Control**
  - More efficient DAE solvers for the inverse dynamics problem presented in this thesis, i.e., the inverse dynamics for the extended flexible joint model.
  - Bounded approximate solutions when solving the DAE problem for non-minimum phase systems.
  - Experimental evaluation of feedforward control based on the extended flexible joint model.
  - Analysis of solvability and uniqueness of the DAE system.
- **Feedback Control**
  - Evaluation of feedback controllers for a MIMO benchmark system with parametric uncertainty.

- Experimental evaluation of some MIMO feedback controllers, suggested for the benchmark system.
- Comparative study of feedforward vs. feedback linearization w.r.t. robustness in tracking applications and in control tasks where disturbance rejection is important.

Bibliography

ABB (2007). IRB 6640 - ABB announce a new stronger robot - the next generation. http://www.abb.com/cawp/seitp202/ce7d8060f91e36e4c125736a0024afb5.aspx. 2007-12-03.

Aguiar, A., Hespanha, J., and Kokotivic, P. (2005). Path-following for nonminimum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, 50(2):234–239.

Albu-Schäffer, A. and Hirzinger, G. (2000). State feedback controller for flexible joint robots: A globally stable approach implemented on DLR's light-weight robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1087–1093, Takamatsu, Japan.

Albu-Schäffer, A., Ott, C., and Hirzinger, G. (2007). A unified passivity-based control framework for position, torque and impedence control of flexible joint robots. *The International Journal of Robotics Research*, 26(1):23–39.

An, C., Atkeson, C., and Hollerbach, J. (1988). *Model-Based Control of a Robot Manipulator*. The MIT press, Cambridge, Massachusetts.

Anderson, B. and Moore, J. (1990). *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Armstrong-Hélouvry, B. (1991). *Control of Machines with Friction*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.

Åström, K. (1994). The future of control. *Modeling, Identification and Control*, 15(3):127–134.

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, Research Triangle Park, NC, USA.

Åström, K. and Wittenmark, B. (1996). *Computer-Controlled Systems: Theory and Design*. Prentice Hall, Englewood Cliffs, New Jersey, USA.

Avitabile, P. (2001). Experimental modal analysis - a simple non-mathematical presentation. *Sound and vibration*, 35(1):20–31.

Bascetta, L. and Rocco, P. (2002). Modelling flexible manipulators with motors at the joints. *Mathematical and Computer Modelling of Dynamical Systems*, 8(2):157–183.

Behi, F. and Tesar, D. (1991). Parametric identification for industrial manipulators using experimental modal analysis. *IEEE Transactions on Robotics and Automation*, 7(5):642–652.

Bellini, A., Figalli, G., and Ulivi, G. (1989). Sliding mode control of a direct drive robot. In *Conference Record of the IEEE Industry Application Society Annual Meeting, vol.2*, pages 1685–1692, San Diego, CA, USA.

Benosman, M. and Le Vey, G. (2004). Control of flexible manipulators: A survey. *Robotica*, 22:533–545.

Berglund, A. and Hovland, G. (2000). Automatic elasticity tuning of industrial robot manipulators. In *39th IEEE Conference on Decision and Control*, pages 5091–5096, Sydney, Australia.

Bernstein, D. (1999). On bridging the theory/practise gap. *IEEE Control Systems Magazine*, 19(6):64–70.

Book, W. (1993). Controlled motion in an elastic world. *Journal of Dynamic Systems Measurement and Control, Transactions of the ASME*, 115:252–261.

Book, W. and Obergfell, K. (2000). Practical models for practical flexible arms. In *Proc. 2000 IEEE International Conference on Robotics and Automation*, pages 835–842, San Fransisco, CA.

Brogårdh, T. (2007). Present and future robot control development—an industrial perspective. *Annual Reviews in Control*, 31(1):69–79.

Brogårdh, T. and Moberg, S. (2002). Method for determining load parameters for a manipulator. US Patent 6343243.

Brogårdh, T., Moberg, S., Elfving, S., Jonsson, I., and Skantze, F. (2001). Method for supervision of the movement control of a manipulator. US Patent 6218801.

Caccavale, F. and Chiacchio, P. (1994). Identification of dynamic parameters and feedforward control for a conventional industrial manipulator. *Control Eng. Practice*, 2(6):1039–1050.

Canudas de Wit, C., Olsson, H., Åström, K., and Lischinsky, P. (1995). A new model for control of systems with friction. *IEEE Transactions on Automatic Control*, 40(3):419–425.

Craig, J. (1988). *Adaptive Control of Mechanical Manipulators*. Addison-Wesley Publishing Company, Menlo Park, California, USA.

Craig, J. J. (1989). *Introduction to Robotics Mechanics and Control*. Addison Wesley, Menlo Park, California, USA.

De Luca, A. (1988). Dynamic control of robots with joint elasticity. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 152–158, Philadelphia, PA.

De Luca, A. (2000). Feedforward/feedback laws for the control of flexible robots. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 233–240, San Francisco, CA.

De Luca, A. and Di Giovanni, G. (2001a). Rest-to-rest motion of a one-link flexible arm. In *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings*, pages 923–928, Como, Italy.

De Luca, A. and Di Giovanni, G. (2001b). Rest-to-rest motion of a two-link robot with a flexible forearm. In *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Proceedings*, pages 929–935, Como, Italy.

De Luca, A., Farina, R., and Lucibello, P. (2005). On the control of robots with viscoelastic joints. In *Proc. 2005 IEEE International Conference on Robotics and Automation*, pages 4297–4302, Barcelona, Spain.

De Luca, A. and Lanari, L. (1995). Robots with elastic joints are linearizable via dynamic feedback. In *34th IEEE Conference on Decision and Control*, pages 3895–3897, New Orleans, LA.

De Luca, A. and Lucibello, P. (1998). A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 504–510, Leuven, Belgium.

De Luca, A., Panzieri, S., and Ulivi, G. (1998). Stable inversion control for flexible link manipulators. In *Proc. 1998 IEEE International Conference on Robotics and Automation*, pages 799–804, Leuven, Belgium.

De Luca, A., Schröder, D., and Thummel, M. (2007). An acceleration-based state observer for robot manipulators with elastic joints. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3817–3823, Roma, Italy.

De Luca, A. and Siciliano, B. (1993). Regulation of flexible arms under gravity. *IEEE Transactions on Robotics and Automation*, 9(4):463–467.

Elmaraghy, H., Lahdhiri, T., and Ciuca, F. (2002). Robust linear control of flexible joint robot systems. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 34(4):335–356.

Fanuc (2007). R-30iA - Product detail. `http://www.fanucrobotics.cz/products/robots/controller.asp?idp=317&id=13`. 2007-12-03.

Feeny, B. and Moon, F. (1994). Chaos in a forced dry-friction oscillator: Experiments and numerical modelling. *Journal of Sound and Vibration*, 170(3):303–323.

Ferretti, G., Magnani, G., and Rocco, P. (1998). Lqg control of elastic servomechanisms based on motor position measurements. In *AMC'98 - Coimbra. 1998 5th International Workshop on Advanced Motion Control. Proceedings*, pages 617–622, Coimbra, Portugal.

Goldstein, H. (1980). *Classical Mechanics*. Addison-Wesley Publishing Company, Reading, Massachusetts, USA.

Grotjahn, M. and Heimann, B. (2002). Model-based feedforward control in industrial robots. *The International Journal of Robotics Research*, 21(1):45–60.

Gunnarsson, S., Norlöf, M., Hovland, G., Carlsson, U., Brogårdh, T., Svensson, T., and Moberg, S. (2006). Pathcorrection for an industrial robot. US Patent 7130718.

Hastings, G. and Book, W. (1987). A linear dynamic model for flexible robotic manipulators. *IEEE Control Systems Magazine*, 7(1):61–64.

Horowitz, I. (1991). Survey of quantitative feedback theory (QFT). *International Journal of Control*, 53(2):255–291.

Hovland, G., Berglund, A., and Hanssen, S. (2001). Identification of coupled elastic dynamics using inverse eigenvalue theory. In *Proceedings of the 32nd ISR(International Symposium on Robotics)*, pages 1392–1397, Seoul, Korea.

Hovland, G. E., Hanssen, S., Gallestey, E., Moberg, S., Brogårdh, T., Gunnarsson, S., and Isaksson, M. (2002). Nonlinear identification of backlash in robot transmissions. In *Proc. 33rd ISR (International Symposium on Robotics)*, Stockholm, Sweden.

IFR (2005). International federation of robotics - statistics 2005. `http://www.ifr.org/statistics/keyData2005.htm`.

Isidori, A. (1995). *Nonlinear Control Systems*. Springer-Verlag, London, Great Britain.

Isidori, A., M. C. and De Luca, A. (1986). A sufficient condition for full linearization via dynamic state feedback. In *46th IEEE Conference on Decision and Control*, pages 203–208, Athens, Greece.

ISO (1998). ISO 9283:1998, manipulating industrial robots - performance criteria and related test methods. `www.iso.org`.

Jankowski, K. and Van Brussel, H. (1992a). An approach to discrete inverse dynamics control of flexible-joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 114:229–233.

Jankowski, K. and Van Brussel, H. (1992b). An approach to discrete inverse dynamics control of flexible-joint robots. *IEEE Transactions on Robotics and Automation*, 8(5):651–658.

Johansson, R. (1993). *System Modeling and Identification*. Prentice Hall.

Johansson, R., Robertsson, A., Nilsson, K., and Verhaegen, M. (2000). State-space system identification of robot manipulator dynamics. *Mechatronics*, 10(3):403–418.

Kailath, T. (1980). *Linear Systems*. Prentice Hall, Englewood Cliffs, New Jersey, USA.

Kane, T. R. and Levinson, D. A. (1983). The use of Kane's dynamical equations in robotics. *International Journal of Robotics Research*, 2(3).

Kane, T. R. and Levinson, D. A. (1985). *Dynamics: Theory and Applications*. McGraw-Hill Publishing Company.

Khosla, P. and Kanade, T. (1989). Real-time implementation and evaluation of computed-torque scheme. *IEEE Transactions on Robotics and Automation*, 5(2):245–253.

Kokotovic, P. and Arcak, M. (2001). Constructive nonlinear control: a historical perspective. *Automatica*, 37:637–662.

Kozlowski, K. (1998). *Modelling and identification in robotics*. Advances in Industrial Control. Springer, London.

Kwon, D.-S. and Book, W. (1990). An inverse dynamic method yielding flexible manipulator state trajectories. In *Proceedings of the 1990 American Control Conference, vol 1*, pages 186–193, San Diego, CA, USA.

Le Tien, L., Albu-Schäffer, A., and Hirzinger, G. (2007). Mimo state feedback controller for a flexible joint robot with strong joint coupling. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3824–3830, Roma, Italy.

Lesser, M. (2000). *The Analysis of Complex Nonlinear Mechanical Systems: A Computer Algebra assisted approach*. World Scientific Publishing Co Pte Ltd, Singapore.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition.

Ljung, L. and Glad, T. (1994). *Modeling of Dynamic Systems*. Prentice Hall.

Ljung, L. and Glad, T. (2001). *Control Theory*. CRC Press.

Madesäter, Å. (1995). Faster and more accurate industrial robots. *Industrial Robot*, 22(2):14–15.

Moberg, S. and Hanssen, S. (2007). A DAE approach to feedforward control of flexible manipulators. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3439–3444, Roma, Italy.

Moberg, S. and Öhr, J. (2004). Svenskt mästerskap i robotreglering. In *Swedish Control Meeting 2004, Göteborg, Sweden*.

Moberg, S. and Öhr, J. (2005). Robust control of a flexible manipulator arm: A benchmark problem. Prague, Czech Republic. 16th IFAC World Congress.

Moberg, S., Öhr, J., and Gunnarsson, S. (2007). A benchmark problem for robust feedback control of a flexible manipulator. Technical Report LiTH-ISY-R-2820, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to IEEE Transactions on Control Systems Technology*.

Motoman (2007). Introducing NX100 - the Next Generation of Robot Controller. `http://www.motoman.co.uk/NX100.htm`. 2007-12-03.

Nieuwstadt, M. V. and Murray, R. (1998). Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8(11):995–1020.

Nihei, R. and Kato, T. (1993). Servo control for robot. In *24th International Symposium on Industrial Robots (ISIR), Tokyo*, Tokyo, Japan.

Nihei, R., Kato, T., and Arita, S. (2007). Vibration control device. US Patent 7181294.

Norrlöf, M. (2000). *Iterative Learning Control: Analysis, Design, and Experiments*. PhD thesis, Linköping University, SE-581 83 Linköping, Sweden.

Öhr, J., Moberg, S., Wernholt, E., Hanssen, S., Pettersson, J., Persson, S., and Sander-Tavallaey, S. (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. In *Proc. ISMA2006 International Conference on Noise and Vibration Engineering*, pages 3305–3314, Leuven, Belgium.

Olsson, H. (1996). *Control Systems with Friction*. PhD thesis, Lund Institue of Technology, SE-221 00 Lund, Sweden.

Oppenheim, A. and Schafer, R. (1975). *Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, USA.

Östring, M., Gunnarsson, S., and Norrlöf, M. (2003). Closed-loop identification of an industrial robot containing flexibilities. *Control Engineering Practice*, 11:291–300.

Pfeiffer, F. and Hölzl, J. (1995). Parameter identification for industrial robots. In *Proc. 1995 IEEE International Conference on Robotics and Automation*, pages 1468–1476, Nagoya, Japan.

Pintelon, R. and Schoukens, J. (2001). *System identification: a frequency domain approach*. IEEE Press, New York.

Ridgely, D. and McFarland, M. (1999). Tailoring theory to practise in tactical missile control. *IEEE Control Systems Magazine*, 19(6):49–55.

Robertsson, A. (1999). *On Observer-Based Control of Nonlinear Systems*. PhD thesis, Dept. of Automatic Control, Lund Institue of Technology, SE-221 00 Lund, Sweden.

Rouchon, P., Fliess, M., Lévine, J., and Martin, P. (1993). Flatness, motion planning and trailer systems. In *Proceedings of the 32nd Conference on Decision and Control*, pages 2700–2705, San Antonio, Texas.

Rugh, W. (1996). *Linear System Theory*. Prentice Hall, Upper Saddle River, New Jersey, USA.

Sage, H., De Mathelin, M., and Ostertag, E. (1999). Robust control of robot manipulators: A survey. *International Journal of Control*, 72(16):1498–1522.

Santibanez, V. and Kelly, R. (2001). PD control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, 19:11–19.

Sciavicco, L. and Siciliano, B. (2000). *Modeling and Control of Robotic Manipulators*. Springer, London, Great Britain.

Shabana, A. (1998). *Dynamics of Multibody Systems*. Cambridge University Press, Cambridge, United Kingdom.

Slotine, J.-J. and Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall, Englewood Cliffs, New Jersey, USA.

Snyder, W. E. (1985). *Industrial Robots: Computer Interfacing and Control*. Prentice-Hall, Englewood Cliffs, New Jersey, USA.

Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice-Hall Int., London, Great Britain.

Spong, M. (1996). *Handbook of Control*. CRC Press. Chapter, Motion Control of Robot Manipulators, page 1339–1350.

Spong, M., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control*. Wiley.

Spong, M. W. (1987). Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 109:310–319.

Swevers, J., Torfs, D., Adams, M., De Schutter, J., and Van Brussel, H. (1991). Comparison of control algorithms for flexible joint robots implemented on a KUKA ir 161/60 industrial robot. In *91 ICAR. Fifth Conference on Advanced Robotics. Robots in Unstructured Environments.*, Pisa, Italy.

Swevers, J., Verdonck, W., and De Schutter, J. (2007). Dynamic model identification for industrial robots. *IEEE Control Systems Magazine*, pages 58–71.

Tomei, P. (1991). A simple PD controller for robots with elastic joints. *IEEE Transactions on Automatic Control*, 36(10):1208–1213.

Torfs, D., Swevers, J., and De Schutter, J. (1991). Quasi-perfect tracking control of non-minimal phase systems. In *46th IEEE Conference on Decision and Control*, pages 241–244, Brighton, England.

Tung, E. and Tomizuka, M. (1993). Feedforward tracking controller design based on the identification of low frequency dynamics. *Journal of Dynamic Systems Measurement and Control, Transactions of the ASME*, 115:348–356.

Wang, Z., Zeng, H., Ho, D., and Unbehauen, H. (2002). Multiobjective control of a four-link flexible manipulator: A robust H-infinity approach. *IEEE Transactions on Control Systems Technology*, 10(6):866–875.

Wernholt, E. (2007). *Multivariable Frequency-Domain Identification of Industrial Robots*. PhD thesis, Linköping University, SE-581 83 Linköping, Sweden.

Wernholt, E. and Gunnarsson, S. (2006). Nonlinear identification of a physically parameterized robot model. In *Proc. 14th IFAC Symposium on System Identification*, pages 143–148, Newcastle, Australia.

Wernholt, E. and Moberg, S. (2007a). Experimental comparison of methods for multivariable frequency response function estimation. Technical Report LiTH-ISY-R-2827, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to the 17th IFAC World Congress*, Seoul, Korea.

Wernholt, E. and Moberg, S. (2007b). Frequency-domain gray-box identification of industrial robots. Technical Report LiTH-ISY-R-2826, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to the 17th IFAC World Congress*, Seoul, Korea.

Part II

Publications

Paper A

Frequency-Domain Gray-Box Identification of Industrial Robots

Edited version of the paper:

> Wernholt, E. and Moberg, S. (2007b). Frequency-domain gray-box identification of industrial robots. Technical Report LiTH-ISY-R-2826, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to the 17th IFAC World Congress*, Seoul, Korea.

Frequency-Domain Gray-Box Identification of Industrial Robots

Erik Wernholt[1] and Stig Moberg[1,2]

[1]Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden.
E-mail: {erikw,stig}@isy.liu.se.

[2]ABB AB – Robotics,
SE–721 68 Västerås, Sweden.

Abstract

This paper considers identification of unknown parameters in elastic dynamic models of industrial robots. Identifying such models is a challenging task since an industrial robot is a multivariable, nonlinear, resonant, and unstable system. Unknown parameters (mainly spring-damper pairs) in a physically parameterized nonlinear dynamic model are identified in the frequency domain, using estimates of the nonparametric frequency response function (FRF) in different robot configurations/positions. The nonlinear parametric robot model is linearized in the same positions and the optimal parameters are obtained by minimizing the discrepancy between the nonparametric FRFs and the parametric FRFs (the FRFs of the linearized parametric robot model). In order to accurately estimate the nonparametric FRFs, the experiments must be carefully designed. The selection of optimal robot configurations for the experiments is also part of the design. Different parameter estimators are compared and experimental results show the usefulness of the proposed identification procedure. The weighted logarithmic least squares estimator achieves the best result and the identified model gives a good global description of the dynamics in the frequency range of interest.

Keywords: System identification, multivariable systems, nonlinear systems, closed-loop identification, frequency response methods, industrial robots

1 Introduction

Accurate dynamic models of industrial robots are needed for mechanical design, performance simulation, control, supervision, diagnosis, and so on. The industrial robot poses a

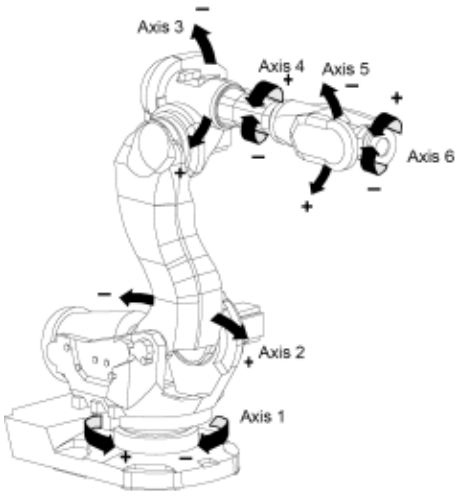


Figure 1: *The ABB manipulator IRB6600.*

challenging modeling problem both due to the system complexity and the required model accuracy. Usually a robot has six joints (also called axes), see Figure 1, with coupled dynamics, giving a truly multivariable system. The dynamics is nonlinear, both with respect to the rigid body dynamics and other things such as non-ideal motors and sensors, and a transmission with friction, backlash, hysteresis, and nonlinear stiffness. The system is resonant due to elastic effects and, in addition, experimental data must usually be collected while the robot controller is operating in closed loop since the system is unstable.

Historically, the dynamic models used for control are either entirely rigid (An et al., 1988), or only flexible joint models are considered, i.e., elastic gear transmission and rigid links (Albu-Schäffer and Hirzinger, 2000; Spong, 1987). The trend in industrial robots is toward lightweight robot structures with a reduced mass but with preserved payload capabilities. This is motivated by cost reduction as well as safety issues, but results in lower mechanical resonance frequencies inside the controller bandwidth. The sources of elasticity in such a manipulator are, e.g., gearboxes, bearings, elastic foundations, elastic payloads, as well as bending and torsion of the links. In Öhr et al. (2006) it is shown that there are cases when these other sources of flexibilities can be of the same order as the gearbox flexibilities for a modern industrial robot. Accurate dynamic models that also describe these elastic effects are therefore needed in order to obtain high performance. These models are, however, very difficult to use for robot control, where, e.g., feedforward control involves solving a DAE, but could in the future improve the performance (Moberg and Hanssen, 2007).

2 Problem Description

The main problem considered in this paper is about identification of unknown parameters in a nonlinear dynamic model of an industrial robot. The model must be *global*, i.e., valid

throughout the whole workspace (all robot configurations/positions), as well as *elastic*, which here means that resonances due to elastic effects are captured by the model. The elastic effects are modeled through a lumped parameter approach (Khalil and Gautier, 2000) where each rigid body is connected by spring-damper pairs (see also Section 3). The model is of gray-box type (a physically parameterized model) and the rigid body parameters of the model are usually assumed to be known from a CAD model or prior rigid body identification. The main objective is identification of elasticity parameters (spring-damper pairs) but other parameters can be added, such as the location in the robot structure of the spring-damper pairs and a few unknown rigid body parameters. It is also possible to include nonlinear descriptions of selected quantities (e.g., the gearbox stiffness) and identify those by a linearization for each position (the terms *robot configuration* and *position* are used interchangeably in this paper).

The real challenge for system identification methods is that the industrial robot is multivariable, nonlinear, unstable, and resonant at the same time. Usually, in the literature, at least one of the first three topics is left out. Identification of such a complex system is therefore a huge task, both in finding suitable model structures and efficient identification methods.

One solution could be to apply a nonlinear prediction error method (Ljung, 1999, pp. 146–147), where measured input-output data are fed to the model and the predicted output from the model is compared with the measured output. This has been treated in Wernholt and Gunnarsson (2006b) for axis one of the industrial robot, which means a stable scalar system (axis one is not affected by gravity). Extending these results to a multivariable and unstable system would involve, for example: finding a stable predictor, numerical problems, and handling large data sets. The last two problems stem from the fact that the system is resonant and numerically stiff, as well as large in dimension both with respect to the number of states and parameters. In addition comes also the choice of model structure (parameters) and handling local minima in the optimization. Apart from all these problems, such a solution would really tackle our main problem.

Due to the complexity of the industrial robot, it is common practice to estimate approximate models for various purposes. By, for example, using a low-frequency excitation, elastic effects have a minor influence and a nonlinear model of the rigid body dynamics can be estimated using least squares techniques. This is a much studied problem in the literature, see, e.g., Kozlowski (1998) for an overview. Taking elastic effects into account makes the identification problem much harder. The main reason is that only a subset of the state variables now are measured such that linear regression cannot be used. One option could be to add sensors during the data collection to measure all states, even though accurate measurements of all states are not at all easy to obtain (if even possible) and such sensors are probably very expensive (for example laser trackers).

It is common to study the local dynamic behavior around certain operating points (also called positions in the paper) and there estimate parametric or nonparametric linear models (see, e.g., Albu-Schäffer and Hirzinger, 2001; Behi and Tesar, 1991; Johansson et al., 2000; Öhr et al., 2006). One application area for these linear models is control design, where a global (feedback) controller is achieved through gain scheduling. The linear models can also be used for the tuning of elastic parameters in a global nonlinear robot model, which is the adopted solution in this paper:

- The local behavior is considered by estimating the nonparametric frequency re-

sponse function (FRF) of the system in a number of positions.

- Next, the nonlinear parametric robot model is linearized in each of these positions.
- Finally, the parameters are optimized such that the parametric FRFs (the FRFs of the linearized parametric robot model) match the estimated nonparametric FRFs.

This identification procedure, first suggested in Öhr et al. (2006), will here be described in more detail. Various aspects of the procedure are also treated in Wernholt and Gunnarsson (2006a), Wernholt and Löfberg (2007), Wernholt and Gunnarsson (2007) and Wernholt and Moberg (2007). Using an FRF-based procedure allows for data compression, unstable systems are handled without problems, it is easy to validate the model such that all important resonances are captured, and model requirements in the frequency domain are also easily handled.

The proposed procedure also has some possible problems. The choice of model structure (parameters) and handling local minima in the optimization are problems here as well. In addition comes some difficulties with biased nonparametric FRF estimates due to closed-loop data and nonlinearities. There are also cases when even a small perturbation around a working point can give large variations due to the nonlinearities, which makes a linear approximation inaccurate, e.g., passing through Coulomb friction, backlash, or different parts of a nonlinear stiffness. This can be partly handled by the choice of excitation (e.g., avoid zero velocity to reduce Coulomb friction). Using multiple positions is good for the parameter accuracy as well as for identifiability issues. It will, however, make it harder to use a linear approximation of certain quantities. Consider, for example, the problem of nonlinear stiffness, where a linear approximation will vary between different positions due to gravity and the amplitude of the excitation. It is then impossible to find a linear stiffness that perfectly matches the resonances for all positions. Still, if the nonlinearity can be parameterized and properly linearized in the different positions, those parameters could possibly be identified as well.

The procedure will now be described, starting with the robot model in Section 3, carrying on by describing the nonparametric FRF estimation and the parameter estimation in Sections 4 and 5, respectively. Experimental results are shown in Section 6, and finally some conclusions are drawn in Section 7.

3 Robot Model

The robot model described in this section comes from Moberg and Hanssen (2007). A general serial link industrial robot, as in Figure 1, is then modeled by a kinematic chain of rigid bodies, where each rigid body is connected to the preceding body by three torsional spring-damper pairs, giving three degrees-of-freedom (DOF) to each rigid body. At most one of these DOFs can be actuated, corresponding to a connection of the two rigid bodies by a motor and a gearbox. In this representation, a robot link (always actuated) can consist of one or more rigid bodies. The model equations, described in Moberg and Hanssen (2007), can be written as a nonlinear gray-box model

$$\dot{x}(t) = f(x(t), u(t), \theta), \tag{1a}$$

$$y(t) = h(x(t), u(t), \theta), \tag{1b}$$

with state vector $x(t)$, input vector $u(t)$, output vector $y(t)$, and nonlinear functions $f(\cdot)$ and $h(\cdot)$ that describe the dynamics. The rigid body parameters are assumed to be known and θ is a vector of unknown parameters for (mainly) springs and dampers. See the previous section for examples of other unknown parameters to include.

4 FRF Estimation

As a first step toward the parameter identification, estimates of the nonparametric FRF in a number of positions are needed. These are obtained by performing experiments where the robot is moved into a position and a speed reference signal is fed to the robot controller. The resulting motor torques (actually the torque reference to the torque controller) and angular positions are sampled and stored. The measured angular positions are then filtered and differentiated to obtain estimates of the motor angular speeds, which are here considered as the output signals.

The open-loop system to be identified is unstable, which makes it necessary to collect data while the robot controller is running in closed loop. Consider therefore the setup in Figure 2, where the controller takes as input the difference between the reference signal r and the measured and sampled output y, and u is the input. The disturbance, v, contains various sources of noise and disturbances. An experimental control system is used, which enables the use of off-line computed reference signals for each motor controller.

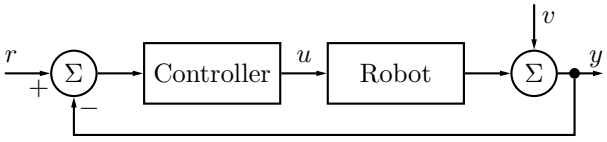


Figure 2: *Closed-loop measurement setup.*

To avoid leakage effects in the discrete Fourier transform (DFT), which is used by the estimation method, the excitation signal, r, is assumed to be periodic, with N_P samples in each period, and an integer number of periods of the steady-state response are collected. The nonparametric FRF estimate $\widehat{G}(\omega_k) \in \mathbb{C}^{n\times n}$ (assuming n inputs and outputs) is calculated from a block of n experiments like (Pintelon and Schoukens, 2001, p. 61)

$$\widehat{G}(\omega_k) = \mathbf{Y}(\omega_k)\mathbf{U}^{-1}(\omega_k), \tag{2}$$

where the n columns of $\mathbf{Y}(\omega_k)$ and $\mathbf{U}(\omega_k)$ contain the DFT of the sampled data from the n experiments. See also, e.g., Wernholt and Gunnarsson (2007) and Wernholt and Moberg (2007) for other FRF estimators for multivariable systems.

As excitation, an *orthogonal random phase multisine* signal (Dobrowiecki and Schoukens, 2007) is used, which here gives

$$\mathbf{R}(\omega_k) = \mathbf{R}_{\text{diag}}(\omega_k)\mathbf{T},$$

where $\mathbf{R}_{\text{diag}}(\omega_k)$ is a diagonal matrix

$$\mathbf{R}_{\text{diag}}(\omega_k) = \text{diag}\left\{R_1(\omega_k), \ldots, R_n(\omega_k)\right\},$$

and $\mathbf{T}$ is an orthogonal matrix

$$\mathbf{T}_{il} = e^{\frac{2\pi j}{n}(i-1)(l-1)},$$

with $\mathbf{T}\mathbf{T}^H = nI$. Each $R_l(\omega_k)$ is the DFT of a random phase multisine signal, which in the time domain can be written as

$$r(t) = \sum_{k=1}^{N_f} A_k \cos(\omega_k t + \phi_k), \tag{3}$$

with amplitudes A_k, frequencies ω_k chosen from the grid $\{\omega_k = \frac{2\pi k}{N_P T_s},\ k = 1, \ldots, \frac{N_p}{2} - 1\}$ (N_P even) with T_s the sampling period, and random phases ϕ_k uniformly distributed on the interval $[0, 2\pi)$. Using the orthogonal multisine signal in closed loop corresponds to an optimal experiment design given output amplitude constraints.

The selection of frequencies as well as the amplitude spectrum will affect the parameter estimation in the next step. Using too many frequencies will give a low signal-to-noise ratio, which increases both the bias and the variance in the nonparametric FRF estimate. The amplitude spectrum should also reflect the sensitivity for the unknown parameters (cf. $\Psi_0^{(i)}(k)$ in (5)), at least such that the unknown parameters influence the parametric FRF for the selected frequencies.

The nonparametric FRF estimate can be improved by averaging over multiple blocks and/or periods. The covariance matrix can then also be estimated. For a linear system, averaging over different periods is sufficient, whereas for a nonlinear system, it is essential to average over blocks where $\mathbf{R}_{\text{diag}}$ in each block should have different realizations of the random phases. The reason is that nonlinearities otherwise will distort the estimate and give a too low uncertainty estimate, see Pintelon and Schoukens (2001, Chap. 3) and Schoukens et al. (2005).

For the industrial robot, the nonlinearities cause large distortions and averaging over multiple blocks is therefore important (Wernholt and Gunnarsson, 2006a). For the same reason, one should only excite odd frequencies (only $\omega_k = 2\pi(2k+1)/(N_P T_s)$ in (3)), see Schoukens et al. (2005).

5 Parameter Estimation

When the FRFs have been estimated from data, the next step is to linearize the nonlinear model (1) in the same positions and calculate the parametric FRFs, $G^{(i)}(\omega_k, \theta)$, $i = 1, \ldots, Q$. A cost function $V(\theta)$ is then formed, measuring the (weighted) discrepancy between the parametric FRF and the estimated nonparametric FRF for all the Q positions. This cost function is finally minimized to identify the unknown parameters.

First, two different parameter estimators will be analyzed and compared. Next, the selection of optimal positions for the experiments is treated, and finally, the solution of the optimization problem is discussed.

Remark A.1. Note that the parametric FRF, $G^{(i)}(\omega_k, \theta)$, is a function of the nonlinear gray-box model (1) such that for each parameter vector θ during the minimization, (1) is linearized in $(x_0^{(i)}, u_0^{(i)})$ before calculating the FRF.

5.1 Estimators

Weighted Nonlinear Least Squares (NLS) Estimator

The NLS estimator is given by

$$\hat{\theta}_{N_f}^{\text{NLS}} = \arg\min_{\theta\in\Theta} V_{N_f}^{\text{NLS}}(\theta), \tag{4a}$$

$$V_{N_f}^{\text{NLS}}(\theta) = \sum_{i=1}^{Q}\sum_{k=1}^{N_f} [\mathcal{E}^{(i)}(k,\theta)]^H [\Lambda^{(i)}(k)]^{-1} \mathcal{E}^{(i)}(k,\theta), \tag{4b}$$

$$\mathcal{E}^{(i)}(k,\theta) = \text{vec}(\widehat{G}^{(i)}(\omega_k)) - \text{vec}(G^{(i)}(\omega_k,\theta)), \tag{4c}$$

with $\Lambda^{(i)}(k)$ a Hermitian ($\Lambda = \Lambda^H$) weighting matrix, and $(\cdot)^H$ denoting complex conjugate transpose. The asymptotic properties ($N_f \to \infty$) of this estimator will be derived in the following theorem.

Theorem A.1
Consider the NLS estimator (4) *and assume that:*

1. $\widehat{G}^{(i)}(\omega_k) = G^{(i)}(\omega_k,\theta_0) + \eta^{(i)}(\omega_k)$ *with* $\text{vec}(\eta^{(i)}(\omega_k))$ *a zero mean circular complex random vector, independent over* i *and* ω_k*, with covariance matrix* $\Lambda_0^{(i)}(\omega_k)$*.*
2. Θ *is a compact set where* $V_{N_f}^{NLS}(\theta)$ *and its first- and second-order derivatives are continuous for any value of* N_f*.*
3. *For* N_f *large enough, the expected value of* $V_{N_f}^{NLS}(\theta)$ *has a unique global minimum in* Θ*.*

The estimator $\hat{\theta}_{N_f}^{NLS}$ *will then converge to* θ_0 *as* $N_f \to \infty$ *and* $\sqrt{N_f}(\hat{\theta}_{N_f}^{NLS} - \theta_0)$ *is asymptotically Normal distributed with covariance matrix* P_θ*,*

$$\begin{aligned} P_\theta = \frac{1}{2}\Bigg[\frac{1}{N_f}\sum_{i=1}^{Q}\sum_{k=1}^{N_f} \Re\left\{\overline{\Psi_0^{(i)}(k)}\Xi^{(i)}(k)[\Psi_0^{(i)}(k)]^T\right\}\Bigg]^{-1} \\ \times \Bigg[\frac{1}{N_f}\sum_{i=1}^{Q}\sum_{k=1}^{N_f} \Re\left\{\overline{\Psi_0^{(i)}(k)}\Sigma^{(i)}(k)[\Psi_0^{(i)}(k)]^T\right\}\Bigg] \\ \times \Bigg[\frac{1}{N_f}\sum_{i=1}^{Q}\sum_{k=1}^{N_f} \Re\left\{\overline{\Psi_0^{(i)}(k)}\Xi^{(i)}(k)[\Psi_0^{(i)}(k)]^T\right\}\Bigg]^{-1}, \end{aligned} \tag{5}$$

with the Jacobian matrix $[\Psi_0^{(i)}(k)]^T = \left.\frac{\partial\,\text{vec}(G^{(i)}(\omega_k,\theta))}{\partial\theta}\right|_{\theta=\theta_0}$*,* $\overline{(\cdot)}$ *denoting complex conjugate, and*

$$\begin{aligned} \Xi^{(i)}(k) &= [\Lambda^{(i)}(\omega_k)]^{-1}, \\ \Sigma^{(i)}(k) &= [\Lambda^{(i)}(\omega_k)]^{-1}\Lambda_0^{(i)}(\omega_k)[\Lambda^{(i)}(\omega_k)]^{-1}. \end{aligned}$$

The covariance is minimized by using the optimal weights

$$\Lambda^{(i)}(\omega_k) = \Lambda_0^{(i)}(\omega_k), \tag{6}$$

which also simplifies (5) *to*

$$P_\theta = \frac{1}{2}\left[\frac{1}{N_f}\sum_{i=1}^{Q}\sum_{k=1}^{N_f} \Re\left\{\overline{\Psi_0^{(i)}(k)}[\Lambda^{(i)}(\omega_k)]^{-1}[\Psi_0^{(i)}(k)]^T\right\}\right]^{-1}.$$

Proof: Follows from fairly straightforward calculations using Theorem 7.21 in Pintelon and Schoukens (2001). □

Note that in addition to the mentioned assumptions, there are some technical details for the asymptotic normality that $\eta^{(i)}(\omega_k)$ has uniformly bounded absolute moments of order $4+\epsilon$ with $\epsilon > 0$. See Pintelon and Schoukens (2001, Theorem 7.21) for details.

Weighted Logarithmic Least Squares (LLS) Estimator

For systems with a large dynamic range, the NLS estimator may become ill-conditioned. The weighted logarithmic least squares (LLS) estimator has been suggested as an alternative (Pintelon and Schoukens, 2001, pp. 206–207)

$$\hat{\theta}_{N_f}^{\text{LLS}} = \arg\min_\theta V_{N_f}^{\text{LLS}}(\theta), \tag{7a}$$

$$V_{N_f}^{\text{LLS}}(\theta) = \sum_{i=1}^{Q}\sum_{k=1}^{N_f}[\mathcal{E}^{(i)}(k,\theta)]^H[\Lambda^{(i)}(k)]^{-1}\mathcal{E}^{(i)}(k,\theta), \tag{7b}$$

$$\mathcal{E}^{(i)}(k,\theta) = \log \operatorname{vec}(\widehat{G}^{(i)}(\omega_k)) - \log \operatorname{vec}(G^{(i)}(\omega_k,\theta)), \tag{7c}$$

where $\log G = \log|G| + j\arg G$. This estimator has improved numerical stability and is particularly robust to outliers in the measurements. However, from a theoretical point of view, the estimator is inconsistent ($\lim_{N_f\to\infty}\hat{\theta}_{N_f}^{\text{LLS}} \neq \theta_0$). The bias can be neglected if the signal-to-noise ratio ($\operatorname{vec}(\widehat{G})$ vs. $\sqrt{\operatorname{diag}\{\Lambda_0\}}$) is large enough (at least 10 dB according to Pintelon and Schoukens, 2001, p. 207).

Similarly to Theorem A.1, one can show that the covariance matrix, using the LLS estimator (7), is approximately given by (5) with

$$\Xi^{(i)}(k) = \left[G_d^{(i)}(\omega_k,\theta_0)\Lambda^{(i)}(\omega_k)[G_d^{(i)}(\omega_k,\theta_0)]^H\right]^{-1},$$
$$\Sigma^{(i)}(k) = \Xi^{(i)}(k)\Lambda_0^{(i)}(\omega_k)\Xi^{(i)}(k),$$

and $G_d^{(i)}(\omega_k,\theta_0) = \operatorname{diag}\left\{\operatorname{vec}(G^{(i)}(\omega_k,\theta_0))\right\}$. Using the optimal weights

$$\Lambda^{(i)}(\omega_k) = \left[G_d^{(i)}(\omega_k,\theta_0)\right]^{-1}\Lambda_0^{(i)}(\omega_k)\left[G_d^{(i)}(\omega_k,\theta_0)\right]^{-H}, \tag{8}$$

gives approximately the same covariance as for the NLS estimator.

Selection of Weights

Even if the covariance is minimized by using the optimal weights, the choice of weights will in general deviate from the optimal ones for a number of reasons. Firstly, the true covariance matrix $\Lambda_0^{(i)}(\omega_k)$ is usually not known so the user must instead be content with an estimated covariance matrix $\hat{\Lambda}_0^{(i)}(\omega_k)$. Secondly, the weights also reflect where the user requires the best model fit. This is important in case the model is unable to describe every detail in the measurements. The bias-inclination will then be small for frequencies, elements, and positions where the weights $[\Lambda^{(i)}(\omega_k)]^{-1}$ are large.

For a resonant system, it is often easier to use the LLS estimator in the way that even constant weights will make sure that both resonances and anti-resonances are matched by the model. This is due to the fact that the logarithm in the LLS estimator inherently gives the relative error, compared to the absolute error when using the NLS estimator. With the NLS estimator, the anti-resonances are easily missed if not choosing large weights at those frequencies.

5.2 Optimal Positions

Given a nonlinear gray-box model (1), the information about the unknown parameters will differ between nonparametric FRF estimates in different positions. Therefore, given a limited total measurement time, one should perform experiments in the position(s) that contribute the most to the information about the unknown parameters. In Wernholt and Löfberg (2007), this problem is formulated as follows: Assume a set of Q_c candidate positions. Determine the number of experiments to be performed in each position (m_i experiments in position i) such that the parameter uncertainty is minimized, given a total of $M = \sum_{i=1}^{Q_c} m_i$ experiments. Determining the values m_i, $i = 1, \ldots, Q_c$, is a combinatorial experiment design problem which relatively quickly will become intractable when Q_c is large. If M is not too small, a good approximate solution can be found by relaxing the constraint that each m_i should be an integer. This relaxed problem is convex, which enables the global optimum to be found. In the paper Wernholt and Löfberg (2007), it is also shown that the experiment design is efficiently solved by considering the dual problem. The candidate positions are obtained by griding the workspace. Given thousands of candidate positions, only a few positions typically have a nonzero m_i in the optimum. See Wernholt and Löfberg (2007) for details and examples.

5.3 Solving the Optimization Problem

The minimization problem to be solved, (4) or (7), is unfortunately non-convex. Here, the problem is solved using *fminunc* in MATLAB, which is a gradient-based method which only returns a local optimum. Due to the existence of local minima, a good initial parameter vector, θ_{init}, is important. The problem is solved for a number of random perturbations around θ_{init} in order to avoid local minima. Or, alternatively stated, to obtain a local minimum which is good enough for the purpose of the model. The quality of the resulting model, as well as problems with local minima and identifiability properties, depend on the choices of estimator, weights, and position(s) for the experiments. This will be illustrated in the next section.

6 Experimental Results

The described identification procedure from the previous sections will here be used for the identification of an industrial robot from the ABB IRB6600 series using an experimental controller. A nonlinear gray-box model with 26 unknown parameters is used. The nonparametric FRFs are estimated in the 15 optimal positions from Wernholt and Löfberg (2007) by using an odd orthogonal random phase multisine signal with a flat amplitude spectrum as excitation and averaging over a number of blocks. The parameters are then estimated using the following estimators:

`LLSM15U`: LLS estimator, $Q = 15$, only magnitude ($\log|G|$), user-defined weights.

`LLS15U`: LLS estimator, $Q = 15$, user-defined weights.

`LLS15O`: LLS estimator, $Q = 15$, optimal weights.

`NLS15U`: NLS estimator, $Q = 15$, user-defined weights.

`NLS15O`: NLS estimator, $Q = 15$, optimal weights.

`LLS1U`: LLS estimator, $Q = 1$, user-defined weights.

For simplicity only diagonal weights $[\Lambda^{(i)}(\omega_k)]^{-1}$ are considered[1]. The user-defined weights are constant for each element in the FRF, the same for all positions, zero for low frequencies where the nonparametric FRF is uncertain, and lower for the non-diagonal elements in the FRF. The optimal weights are calculated from (6) and (8), using the estimated covariance $\hat{\Lambda}_0^{(i)}(\omega_k)$ and $G^{(i)}(\omega_k, \theta_0) \approx \widehat{G}^{(i)}(\omega_k)$. The optimal weights often turn out to be small at the resonances and anti-resonances due to a larger relative error in the nonparametric FRF estimate at those frequencies. For the `LLS1U` estimator, the single position with the smallest theoretical parameter covariance is used.

To assess the sensitivity to the initial parameter vector, θ_{init}, 100 optimizations are performed for each of the 6 estimators, using randomly perturbed initial parameters, $\theta_{init}^{[l]},\ l = 1, \ldots, 100$ (the same for all estimators). Each element in $\theta_{init}^{[l]}$ is obtained by multiplying the corresponding element in θ_{init} by 10^{φ}, where φ is a random number from a uniform distribution on the interval $[-1,\ 1]$.

To evaluate the resulting 600 models, the same cost function is used for all models. The LLS cost, $V^{\text{LLS}}(\theta)$, is calculated with optimal weights and user-defined weights, which can be seen in Figures 3 and 4, respectively. The cost varies quite much between the different estimators. What is more important is the trend over the different optimizations. The first three estimators tend to be much more robust to varying initial parameters.

To compare the number of reasonable models, some measure is needed. Since the optimal weights are small at the resonances and anti-resonances, Figure 3 is not so well suited for judging if resonances (and anti-resonances) are accurately modeled or not. Consider therefore Figure 4. When comparing the FRFs of the parametric models with the estimated nonparametric FRFs, the models usually miss important resonances when the

[1]This means that, e.g., (4b) can be rewritten as
$V_{N_f}^{\text{NLS}}(\theta) = \sum_{i=1}^{Q}\sum_{k=1}^{N_f}\sum_{m=1}^{n_y}\sum_{n=1}^{n_u} |\widehat{G}_{mn}^{(i)}(\omega_k) - G_{mn}^{(i)}(\omega_k, \theta)|^2 W_{m+(n-1)n_y}^{(i)}(\omega_k)$,
where $W^{(i)}(\omega_k)$ is the diagonal of $[\Lambda^{(i)}(\omega_k)]^{-1}$.

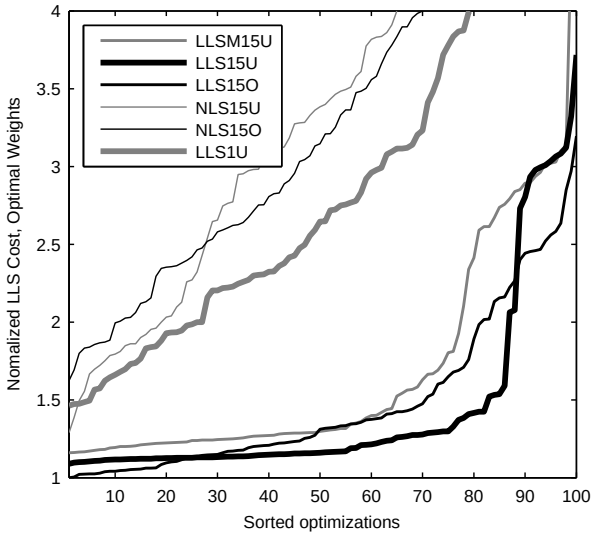


Figure 3: *Normalized LLS cost with optimal weights for all initial parameters and all estimators.*

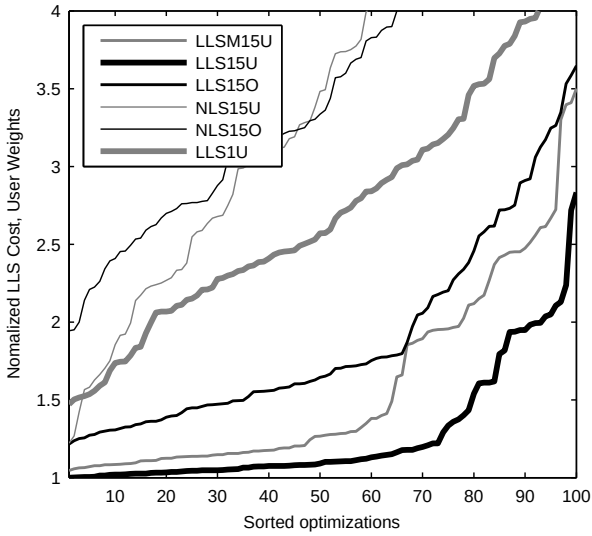


Figure 4: *Normalized LLS cost with user-defined weights for all initial parameters and all estimators.*

Table 1: *Statistics for the* `LLS15U` *estimator, where the first five columns are obtained using the* 31 *best models in Figure 4, and the last column is calculated for the best model, using* (5).

θ_i	min	median	max	mean	std	std_{th}
k_1	0.97	1.01	1.05	1.01	0.020	0.0069
k_2	0.99	1.02	1.04	1.02	0.012	0.0081
k_3	0.061	1.00	3.67	1.10	0.66	0.103
k_4	0.72	0.99	1.26	0.97	0.13	0.042
c_1	0.84	0.99	1.45	1.00	0.12	0.037
c_2	0.24	1.04	1.51	0.99	0.30	0.056
c_3	0.16	0.55	7.05	1.57	1.93	0.865
c_4	0.048	0.44	34.1	2.27	6.14	0.186

normalized cost in Figure 4 exceeds approximately 1.5. That gives the following percentage of reasonable models (out of the 100 models): `LLS15U`, 80 %, `LLSM15U`, 64 %, `LLS15O`, 35 %, `NLS15U`, 3 %, `LLS1U`, 2 %, and `NLS15O`, 0 %. These numbers are only approximate since the same cost can be achieved if one resonance is missed completely but all others are accurate, and if many resonances are only modeled with moderate accuracy. The latter is often the case for the `LLS15O` estimator since the exact location of the resonances, as well as their damping, are not so important when using the optimal weights.

To further evaluate the estimated models, the parameter variation is studied. The spring parameters of the gearboxes and the arm structure as a function of the sorted optimizations can be seen in Figures 5 and 6, respectively. These parameters are normalized by the best `LLS15U` model. One immediately notes that the arm structure springs are harder to estimate, in particular one of them. This parameter does not influence the FRF that much in the selected frequency interval and is therefore hard to estimate. Considering the number of optimizations with estimated parameters inside the interval $[0.5,\ 2]$ (black lines in the figures) gives the following percentage of reasonable models (out of the 100 models): `LLS15U`, 62 %, `LLSM15U`, 43 %, `LLS15O`, 21 %, `NLS15U`, 2 %, `LLS1U`, 2 %, and `NLS15O`, 0 %. One arm structure parameter is excluded in these numbers, but the `LLS15U` estimator actually manages to accurately estimate the 12 spring parameters in 12 % of the optimizations. The dampers are unfortunately much harder to accurately estimate, as can be seen in Figure 7. Some of the damping parameters fluctuate quite much even among the best models.

The `LLS15U` estimator is further analyzed by computing the theoretical parameter uncertainty from (5) for the model with the lowest V^{LLS} cost, as well as the statistics for the best 31 models, i.e., all models with a cost less than 1.05 in Figure 4. Statistics for 8 representative springs k_i and dampers d_i are shown in Table 1, where the parameters with both the smallest and the largest uncertainties are included. The conclusions are that the springs are more accurately estimated than the dampers and that the theoretical uncertainty gives a good indication of the quality of the estimated parameters.

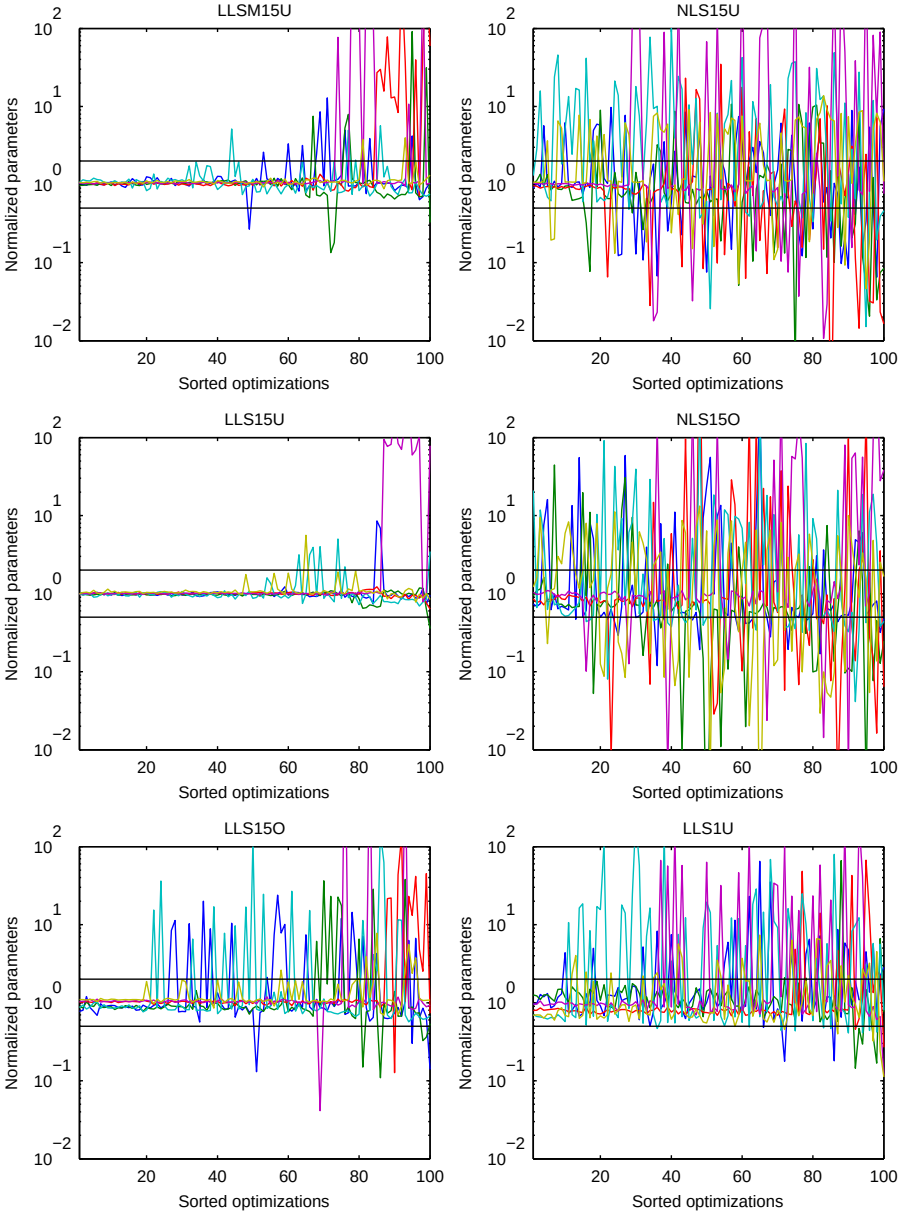


Figure 5: *Normalized spring parameters of the gearboxes for the 6 different estimators, sorted according to Figure 4.*

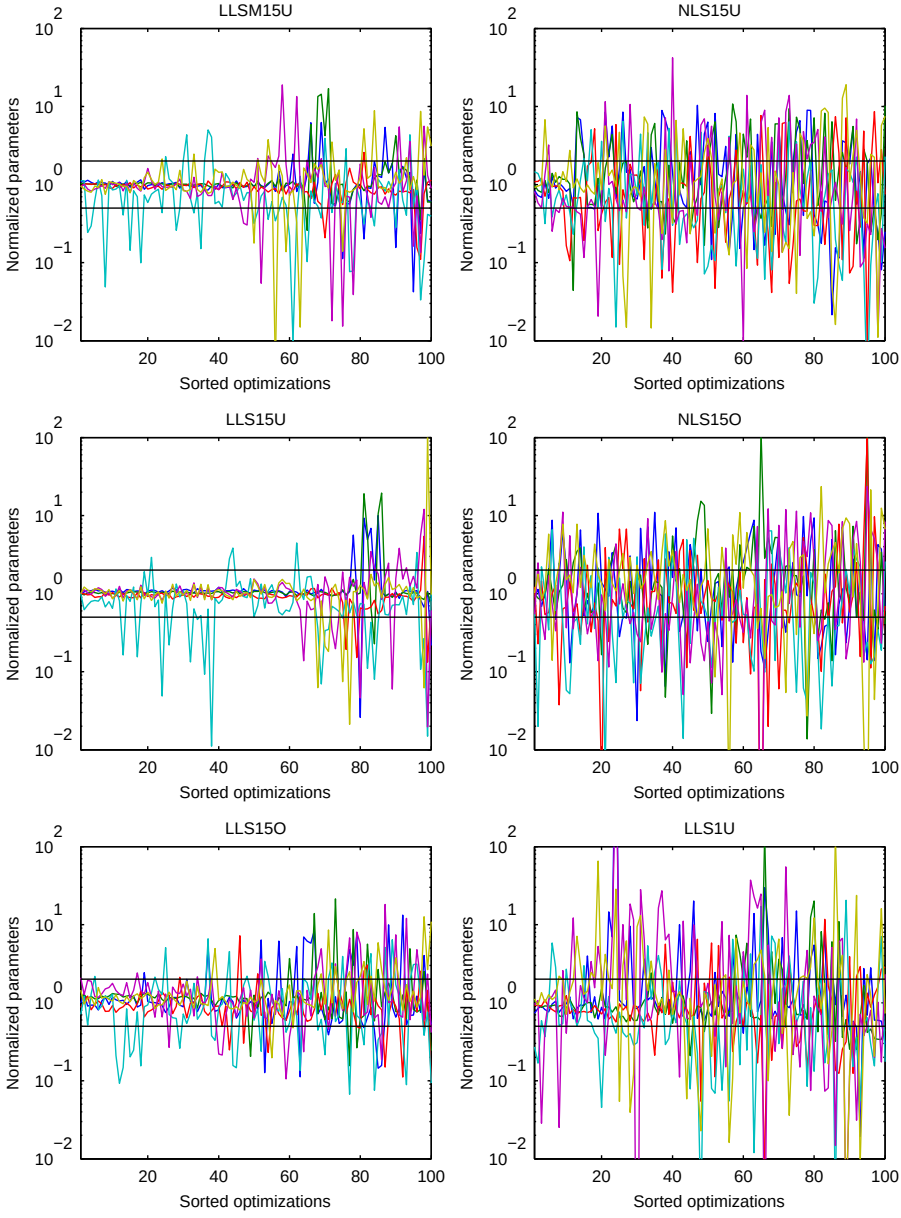


Figure 6: *Normalized spring parameters of the arm structure for the 6 different estimators, sorted according to Figure 4.*

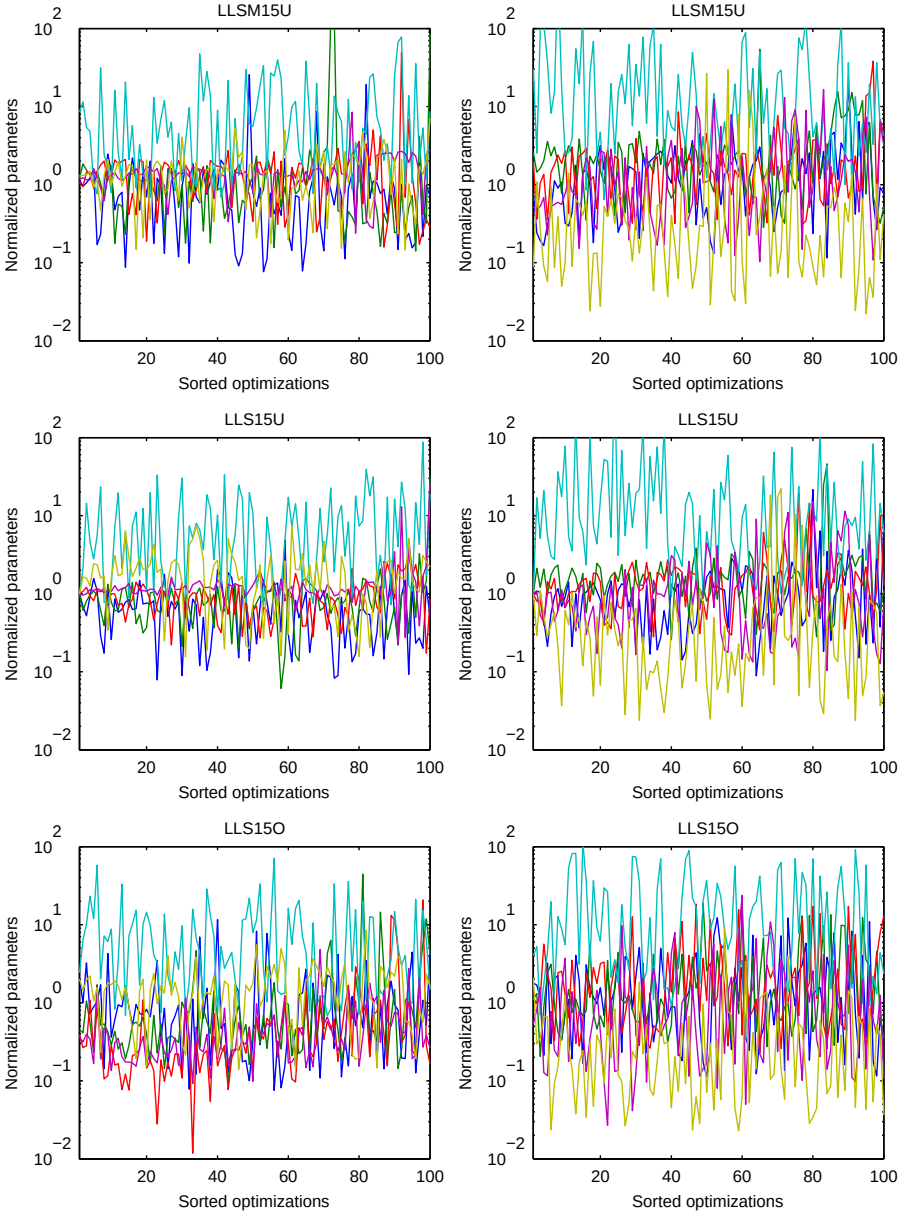


Figure 7: *Normalized damping parameters of the gearboxes (left column) and the arm structure (right column) for the estimators* `LLSM15U`, `LLS15U` *and* `LLS15O`, *sorted according to Figure 4.*

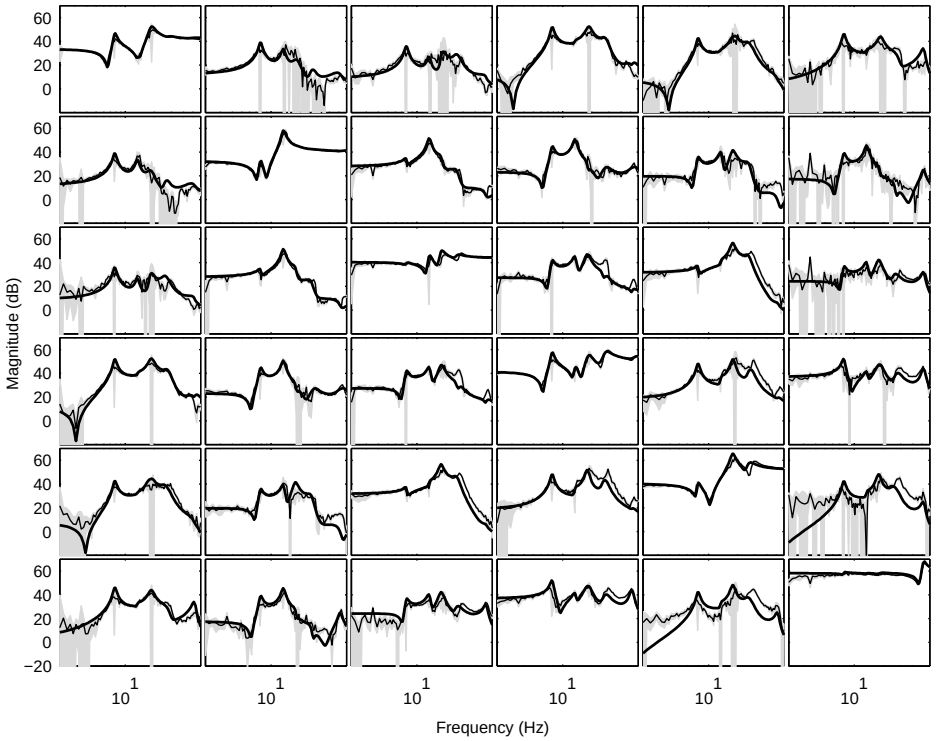


Figure 8: *Estimated nonparametric FRF $\widehat{G}^{(i)}(\omega_k)$ and parametric FRF $G^{(i)}(\omega_k, \theta)$ (the best* `LLS15U` *model) in one of the positions. Input: 6 motor torques, output: 6 motor accelerations. Thin line: $\widehat{G}^{(i)}(\omega_k)$, shaded: one standard deviation for $\widehat{G}^{(i)}(\omega_k)$, and thick line: $G^{(i)}(\omega_k, \theta)$.*

Figure 8 finally shows the magnitude of the estimated nonparametric FRF and the best parametric model for one of the positions. The identified model gives a good global description of the dynamics in the frequency range of interest.

7 Concluding Discussion

This paper has dealt with the problem of estimating unknown elasticity parameters in a nonlinear gray-box model of an industrial robot. An identification procedure has been proposed where the parameters are identified in the frequency domain, using estimates of the nonparametric FRFs for a number of robot configurations/positions. The nonlinear parametric gray-box model is linearized in the same positions and the optimal parameters are obtained by minimizing the discrepancy between the nonparametric and the parametric FRFs. Two different parameter estimators (NLS and LLS) have been analyzed. The estimators, as well as the selection of weights in the estimators, have been evaluated in an

experimental study. The conclusions of the experimental study are that:

- the LLS estimator is superior to the NLS estimator for this type of system,
- more than one position is needed in order to get a reasonable estimate,
- using phase information improves the estimate,
- rough user-defined weights work much better than the theoretically optimal weights,
- gearbox parameters are easier to estimate than arm structure parameters,
- spring parameters are easier to identify than damping parameters, and
- the theoretical uncertainties for the estimated parameters in Table 1 give a good indication of the quality of the estimated parameters.

The uncertainties in the dampers and in some of the springs in the best identified model are quite large, but the resulting global model should anyway be useful for many purposes.

An explanation to the fourth point is that the assumptions in Theorem A.1 are violated since the nonparametric FRF estimate has bias errors due to nonlinearities and closed-loop data, and the model is unable to describe every detail of the true system. The weights should, for such a case, primarily be selected to distribute the bias, not to get minimum variance. The theoretically optimal weights are low at the resonance and anti-resonance frequencies (due to uncertainties in the FRF), which in turn give large model errors there.

The fifth point comes as no surprise. The model structure, where all elastic effects in the arm structure are lumped into a few spring-damper pairs, can of course be modified and refined. Both regarding the location of these spring-damper pairs, as well as how many that are needed in order to properly model the system. Identifiability of these added parameters can also be discussed. Maybe additional sensors are needed, e.g., accelerometers attached to the structure, as is the case in experimental modal analysis (Behi and Tesar, 1991; Verboven, 2002).

The main reasons for the large uncertainties in the damping parameters probably are that the system is poorly damped and that the nonparametric FRFs contain errors at the resonances and anti-resonances such that unique damping parameters are hard to find.

A number of areas are still subject to future work. The selection of weights can certainly be improved by combining the user choices and the estimated FRF uncertainty. The selection of frequencies, as well as the amplitude spectrum for the nonparametric FRF estimation can be further improved. An experimental verification of the optimal positions for identification is still interesting to perform. The parameter accuracy and problems with local minima, versus measurement time and excitation energy are also interesting problems to study. Using a frequency-domain method for identification of a nonlinear system has some problems, as was pointed out in Section 2. Therefore, it would be interesting to apply time-domain prediction error methods as a comparison, even though that involves a number of hard problems to tackle, also mentioned in Section 2. A simulation-based study, using a realistic nonlinear model, could also be enlightening.

Finally, to conclude this paper: Identification of industrial robots is a challenging task. Using a general purpose method by pressing a button will almost surely fail. The problem instead requires a combination of tailored identification methods, experiment design, and a skilled user, using all available knowledge about the system.

References

Albu-Schäffer, A. and Hirzinger, G. (2000). State feedback controller for flexible joint robots: A globally stable approach implemented on DLR's light-weight robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1087–1093, Takamatsu, Japan.

Albu-Schäffer, A. and Hirzinger, G. (2001). Parameter identification and passivity based joint control for a 7DOF torque controlled light weight robot. In *Proc. 2001 IEEE International Conference on Robotics and Automation*, pages 2852–2858, Seoul, Korea.

An, C., Atkeson, C., and Hollerbach, J. (1988). *Model-Based Control of a Robot Manipulator*. The MIT press, Cambridge, Massachusetts.

Behi, F. and Tesar, D. (1991). Parametric identification for industrial manipulators using experimental modal analysis. *IEEE Transactions on Robotics and Automation*, 7(5):642–652.

Dobrowiecki, T. and Schoukens, J. (2007). Measuring a linear approximation to weakly nonlinear MIMO systems. *Automatica*, 43(10):1737–1751.

Johansson, R., Robertsson, A., Nilsson, K., and Verhaegen, M. (2000). State-space system identification of robot manipulator dynamics. *Mechatronics*, 10(3):403–418.

Khalil, W. and Gautier, M. (2000). Modeling of mechanical systems with lumped elasticity. In *Proc. 2000 IEEE International Conference on Robotics and Automation*, pages 3964–3969, San Francisco, CA.

Kozlowski, K. (1998). *Modelling and identification in robotics*. Advances in Industrial Control. Springer, London.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition.

Moberg, S. and Hanssen, S. (2007). A DAE approach to feedforward control of flexible manipulators. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3439–3444, Roma, Italy.

Öhr, J., Moberg, S., Wernholt, E., Hanssen, S., Pettersson, J., Persson, S., and Sander-Tavallaey, S. (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. In *Proc. ISMA2006 International Conference on Noise and Vibration Engineering*, pages 3305–3314, Leuven, Belgium.

Pintelon, R. and Schoukens, J. (2001). *System identification: a frequency domain approach*. IEEE Press, New York.

Schoukens, J., Pintelon, R., Dobrowiecki, T., and Rolain, Y. (2005). Identification of linear systems with nonlinear distortions. *Automatica*, 41(3):491–504.

Spong, M. W. (1987). Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 109:310–319.

Verboven, P. (2002). *Frequency-domain system identification for modal analysis*. PhD thesis, Vrije Universiteit Brussel, Belgium.

Wernholt, E. and Gunnarsson, S. (2006a). Detection and estimation of nonlinear distortions in industrial robots. In *Proc. 23rd IEEE Instumentation and Measurement Technology Conference*, pages 1913–1918, Sorrento, Italy.

Wernholt, E. and Gunnarsson, S. (2006b). Nonlinear identification of a physically parameterized robot model. In *Proc. 14th IFAC Symposium on System Identification*, pages 143–148, Newcastle, Australia.

Wernholt, E. and Gunnarsson, S. (2007). Analysis of methods for multivariable frequency response function estimation in closed loop. In *46th IEEE Conference on Decision and Control*, New Orleans, Louisiana. Accepted for publication.

Wernholt, E. and Löfberg, J. (2007). Experiment design for identification of nonlinear gray-box models with application to industrial robots. In *46th IEEE Conference on Decision and Control*, New Orleans, Louisiana. Accepted for publication.

Wernholt, E. and Moberg, S. (2007). Experimental comparison of methods for multivariable frequency response function estimation. Technical Report LiTH-ISY-R-2827, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to the 17th IFAC World Congress*, Seoul, Korea.

Paper B

A DAE Approach to Feedforward Control of Flexible Manipulators

Edited version of the paper:

Moberg, S. and Hanssen, S. (2007). A DAE approach to feedforward control of flexible manipulators. In *Proc. 2007 IEEE International Conference on Robotics and Automation*, pages 3439–3444, Roma, Italy.

A DAE Approach to Feedforward Control of Flexible Manipulators

Stig Moberg[1,3] and Sven Hanssen[2,3]

[1]Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden.
E-mail: stig@isy.liu.se.

[2]Dept. of Solid Mechanics,
Royal Institute of Technology,
SE–10044 Stockholm, Sweden.
E-mail: soha@kth.se.

[3]ABB AB – Robotics,
SE–721 68 Västerås, Sweden.

Abstract

This work investigates feedforward control of elastic robot structures. A general serial link elastic robot model which can describe a modern industrial robot in a realistic way is presented. The feedforward control problem is discussed, and a solution method for the inverse dynamics problem is proposed. This method involves solving a differential algebraic equation (DAE). A simulation example for an elastic two axis planar robot is also included and shows promising results.

Keywords: Robotics, manipulators, control, feedforward, flexible arms, differential-algebraic equations

1 Introduction

High accuracy control of industrial robot manipulators is a challenging task which has been studied by academic and industrial researchers since the 1970's. Control methods for rigid direct drive robots are, e.g., described in An et al. (1988). The two main approaches are feedforward control and computed torque control (i.e., feedback linearization and decoupling) respectively. Both are based on a rigid dynamic model, and combined with

a diagonal PD or PID controller. The methods show similar results as described, e.g., in Santibanez and Kelly (2001).

Control methods for flexible joint robots (i.e., elastic gear transmissions and rigid links) can, e.g., be found in De Luca and Lucibello (1998) and De Luca (2000). Experiments on industrial robots are described in Caccavale and Chiacchio (1994) and Albu-Schäffer and Hirzinger (2000). The main approaches are the same as for direct drive robots.

The trend in industrial robots is towards lightweight robot structures with a higher degree of elasticity but with preserved payload capabilities. This results in lower mechanical resonance frequencies inside the control bandwidth. The sources of elasticity in such a manipulator are, e.g., gearboxes, bearings, elastic foundations, elastic payloads as well as bending and torsion of the links. In Öhr et al. (2006), it is shown that there are cases when the total elasticity in a plane perpendicular to the preceding joint, and the total elasticity out of this plane (bending and torsion), are of the same order.

In most publications concerning industrial robots, only gear elasticity in the rotational direction or link deformation restricted to a plane perpendicular to the preceding joint, are included in the model. These restricted models simplify the control design but limit the attainable performance.

This work presents a general serial link elastic model that includes joint elasticity in all directions, and thus describes a modern industrial robot in a reasonable way. Furthermore, a feedforward approach based on the solution of a differential algebraic equation (DAE) is proposed. The DAE formulation of the robot feedforward problem has been described previously by others but to the author's knowledge never been implemented in a simulation environment for such a complex and realistic model structure as mentioned above. Furthermore, a way of reducing the DAE complexity is also proposed. Finally, the model structure and the feedforward method are illustrated in a simulation example with a two axis robot model.

2 An Industrial Manipulator

The most common type of industrial manipulator has six serially mounted links, all controlled by electrical motors via gears. An example of a serial industrial manipulator is shown in Figure 1. The dynamics of the manipulator change rapidly as the robot links move fast within its working range, and the dynamic couplings between the links are strong. Moreover, the robot system is elastic as described in Section 1, and the gears have nonlinearities such as backlash, friction and nonlinear elasticity. From a control engineering perspective a manipulator can be described as a nonlinear multivariable dynamical system having the six motor currents as the inputs and the six measurable motor angles as outputs. The goal of the motion control is to control the *orientation* and the *position* of the *tool* along a certain desired path.

3 Robot Model

In this section, a general serial link robot model capable of adequately describing the different sources of flexibility, as described in Section 1, is proposed. This model structure

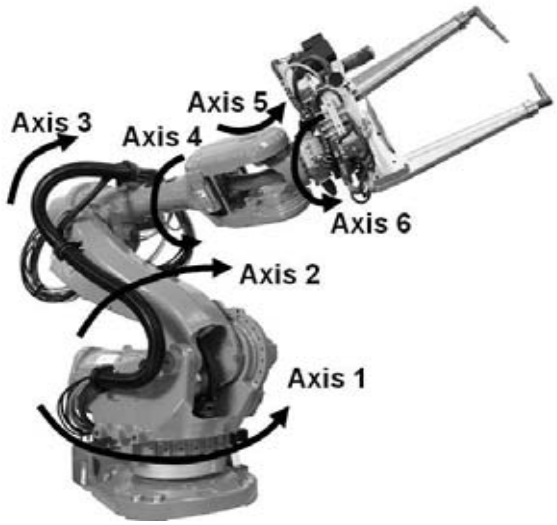


Figure 1: *IRB6600 from ABB equipped with a spotwelding gun.*

will later be used for deriving a feedforward control law. The identification of an industrial robot using a model derived from this general model class is described in Öhr et al. (2006).

3.1 General Description

The model consists of a serial kinematic chain of R rigid bodies. One rigid body rb^i is illustrated in Figure 2, and is described by its mass m^i, center of mass ξ^i, inertia tensor w.r.t. center of mass J^i and length l^i. Due to the symmetrical inertia tensor, only six components of J^i need to be defined. All parameters are described in a coordinate system a^i, fixed in rb^i, and are defined as follows:

$$\xi^i = \begin{bmatrix} \xi_x^i & \xi_y^i & \xi_z^i \end{bmatrix}, \tag{1a}$$

$$J^i = \begin{bmatrix} J_{xx}^i & J_{xy}^i & J_{xz}^i \\ J_{xy}^i & J_{yy}^i & J_{yz}^i \\ J_{xz}^i & J_{yz}^i & J_{zz}^i \end{bmatrix}, \tag{1b}$$

$$l^i = \begin{bmatrix} l_x^i & l_y^i & l_z^i \end{bmatrix}. \tag{1c}$$

The rigid body rb^i is connected to rb^{i-1} by three torsional spring-damper pairs and adds 3 DOF to the model as its configuration can be described by three angular positions, and the given position and orientation of a^{i-1}. The position of a^{i-1} is determined by the angular positions of all rb^k, $k < i$. Thus, the arm system has $3R$ DOF (i.e., the number of independent coordinates necessary to specify its configuration). M of the DOFs (maximum one per rigid body) are actuated, and correspond to a connection of two rigid bodies by a motor and a gearbox. Therefore, the total system has $3R + M$ DOF. Note that one link (always actuated) can consist of one or more rigid bodies.

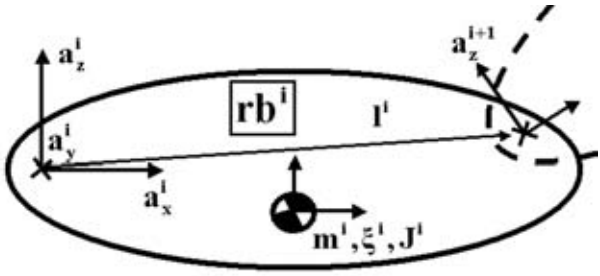


Figure 2: *Definition of rigid body.*

The angular position and inertia for motor i are denoted q_m^i and J_m^i respectively. The generalized coordinates q_d^i (arm angular positions) defines rotations of the coordinate systems a^i. Index d denotes the direction (x, y, or z). The coordinate systems a^i are defined to have the same orientation in zero position, i.e., $q_d^i = q_m^i = 0$. The generalized speeds are defined as $v_d^i = \dot{q}_d^i$ and $v_m^i = \dot{q}_m^i$. The motors are placed on the preceding body, and the inertial couplings between the motors and the rigid bodies are neglected under the assumption of high gear ratio, see, e.g., Spong (1987), but the motor mass and inertia are added to the corresponding rigid body.

The springs and dampers are generally nonlinear functions expressing the gearbox torque $\tau_d^i = \tau_d^i(q_d^i, q_m^i, v_d^i, v_m^i)$ for an actuated DOF, or the constraint torque $\tau_d^i = \tau_d^i(q_d^i, v_d^i)$ for an unactuated DOF. For a linear spring, $\tau_d^i = k_d^i(q_d^i - q_m^i) + d_d^i(v_d^i - v_m^i)$ or $\tau_d^i = k_d^i q_d^i + d_d^i v_d^i$.

The torque control of the motor is assumed to be ideal so the M input signals of the system, which are the motor torque references u equal the motor torques τ. The system has M controlled output variables, typically the position and orientation of the robot tool.

An extension of the model is to define the number of arm DOF as N, which can be maximum $6R$, by adding three linear springs for translational deformation. Furthermore, if some of the springs are defined as rigid, N can be reduced. As the purpose of this work is to study and control effects caused by elasticity, friction is omitted.

The equations of motion are derived by computing the linear and angular momentum. By using Kane's method (Kane and Levinson, 1985) the projected equations of motion are derived to yield a system of ordinary differential equations (ODE) with minimum number of DOFs, see also Lesser (2000).

The model equations can be described as a system of first order ODEs

$$M_a(q_a)\dot{v}_a = c(q_a, v_a) + g(q_a) + \tau_a(q_a, q_m, v_a, v_m), \tag{2a}$$

$$M_m\dot{v}_m = \tau_m(q_a, q_m, v_a, v_m) + u, \tag{2b}$$

$$\dot{q}_a = v_a, \tag{2c}$$

$$\dot{q}_m = v_m, \tag{2d}$$

where $q_a \in R^N$ are the arm angular positions (q_d^i) and $q_m \in R^M$ are the motor angular positions. The corresponding speeds are v_a and v_m. The vector of spring torques acting

on the arm system is described by $\tau_a \in R^N$, and of the spring torques on the motors by $\tau_m \in R^M$. $M_a(q_a) \in R^{N\times N}$ is the inertia matrix for the arms, and $M_m \in R^{M\times M}$ is the diagonal inertia matrix of the motors. The Coriolis and centrifugal torques are described by $c(q_a, \dot{q}_a) \in R^N$, and $g(q_a) \in R^N$ is the gravity torque. The time t is omitted in the expressions.

For a complete model including the position and orientation of the tool, Z, the forward kinematic model of the robot must be added. The kinematic model is a mapping of $q_a \in R^N$ to $Z \in R^M$. The complete model of the robot is then described by (2) and

$$Z = \Gamma(q_a). \tag{3}$$

3.2 A Robot Model with 5 DOF: Description and Analysis

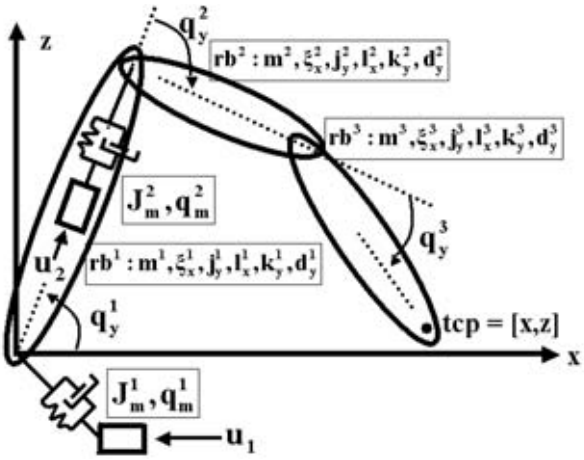


Figure 3: *The 5 DOF model (k_y^3 and d_y^3 not shown).*

The general robot model described in Section 3.1 is here used to derive a specific model with $R = 3$, i.e., three rigid bodies. Generally, there are in total 21 parameters associated to one rigid body, so the model has in total 63 possible parameters as shown in Table 1. The derived model has $N = 3$ and $M = 2$, i.e., three arm DOF and two actuated DOF, as indicated in the Table 1 by "ActuatedDirection" and by declaring parameters as "Rigid". This information together with parameters declared as "Zero" is used to simplify the equations. All links are aligned along the x axis at zero position. All parameters are defined in SI units with the motor inertia transformed by the square of the gear ratio. The model is illustrated in Figure 3 and is a planar model with linear elasticity, constrained to work in the xz-plane with the gravitational constant g set to zero. Note that this model has its elasticity in a plane perpendicular to the preceding joint and do not demonstrate the type of elasticity described in Section 1. However, it is a simple model for a first verification of the new feedforward control algorithm. All model equations can be found in the Appendix.

Table 1: *Parameters of the model E12*

	rb^1	rb^2	rb^3
m^i	100	100	200
ξ_x^i	0.5	0.5	0.1
ξ_y^i	Zero	Zero	Zero
ξ_z^i	Zero	Zero	Zero
l_x^i	1	1	0.2
l_y^i	Zero	Zero	Zero
l_z^i	Zero	Zero	Zero
j_{xx}^i	Zero	Zero	Zero
j_{yy}^i	5	5	50
j_{zz}^i	Zero	Zero	Zero
j_{xy}^i	Zero	Zero	Zero
j_{xz}^i	Zero	Zero	Zero
j_{yz}^i	Zero	Zero	Zero
k_x^i	Rigid	Rigid	Rigid
k_y^i	1E5	1E5	1E5
k_z^i	Rigid	Rigid	Rigid
d_x^i	Rigid	Rigid	Rigid
d_y^i	50	50	50
d_z^i	Rigid	Rigid	Rigid
J_m^i	100	100	NA
$ActuatedDirection$	Y	Y	None

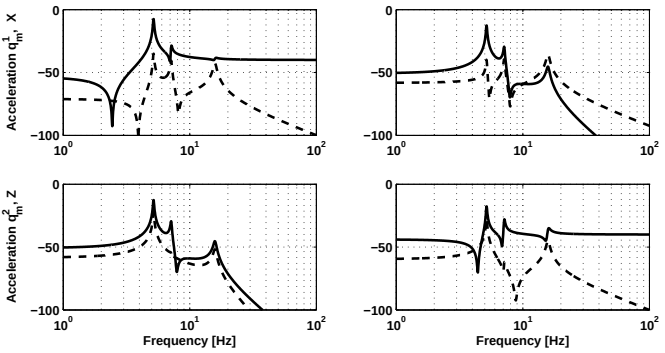


Figure 4: *Transfer Function Magnitude from u to motor acceleration (solid) and cartesian acceleration (dashed).*

One important restriction for the type of control considered in this work, i.e., perfect causal feedforward control, is that the system must be minimum phase. A linear analysis of the model for $q_y^1 = q_m^1 = 0$, $q_y^2 = q_m^2 = 0.3$, and $q_y^3 = 0$ results in a controllable minimum phase system. A nonlinear analysis of controllability and minimum phase be-

havior is outside the scope of this article. The transfer functions of the linearized system are shown in Figure 4.

4 Feedforward Control of a Flexible Manipulator

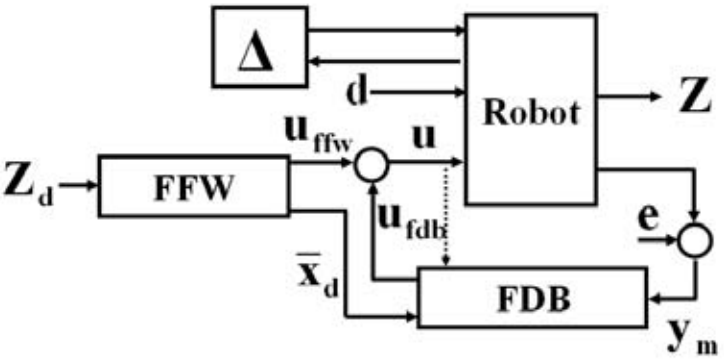


Figure 5: *Robot Controller Structure.*

The controller structure considered is illustrated in Figure 5. Z_d is the desired tool trajectory described in Cartesian coordinates, and Z is the actual trajectory. The Reference and Feedforward Generation Block (FFW) computes the feedforward torque u_{ffw} and the state references $\bar{x}_d$ that are used by the feedback controller (FDB). The robot has uncertain parameters illustrated as a feedback with unknown parameters Δ, and is exposed to disturbances d and measurement noise e. The measured signals are denoted y_m. Note that the dimension of $\bar{x}_d$ and y_m may differ if some states are reconstructed by FDB. The purpose of the FFW is to generate model based references for perfect tracking (if possible). The purpose of the FDB is, under the influence of measurement noise, to stabilize the system, reject disturbances, and to compensate for errors in the FFW.

In this section, we treat the design of the FFW block. The proposed feedback controller is a diagonal controller of PID type which often proves to be a suitable choice for realistic industrial systems, see, e.g., Moberg and Öhr (2005). Only the motor position is measured by the PID-controller (i.e., $y_m = q_{m,measured}$, $\bar{x}_d = q_{m,desired}$).

The forward dynamics problem, i.e., solving (2) for the state variables with the motor torque as input involves solving an ODE. The tool position can then be computed from the states according to (3). The inverse dynamics problem, i.e., solving for the states and the motor torque, with the desired tool position as input to the system, is generally much harder.

The inverse dynamics solution for flexible joint robots without damping and with linear elasticity is described in, e.g., De Luca (2000). Flexible joint robots (gearbox elasticity only) have $R = M = N$ in the model structure described in Section 3. This is an example of a so-called differentially flat system (defined, e.g., in Rouchon et al. (1993)) which can be defined as a system where all state variables and control inputs can

be expressed as an algebraic function of the desired trajectory and its derivatives up to a certain order. In this case the desired trajectory must be four times differentiable.

If damping is introduced, an ODE must be solved, and if the elasticity is allowed to be nonlinear, the ODE becomes nonlinear. This is observed in, e.g., Thümmel et al. (2001), where a DAE formulation of the problem is suggested.

Generally, the inverse dynamics problem can be formulated as a differential algebraic equation (DAE). This is formulated and illustrated by some linear spring-mass systems in Blajer (1997) and Blajer and Kolodziejczyk (2004). In Blajer and Kolodziejczyk (2004), a solution based on flatness and a solution based on DAE yield the same result. However, it is also concluded that solutions based on flatness are not realistic for more complicated systems. The same conclusion is also presented in Jankowski and Van Brussel (1992) for the case of feedback linearization for a flexible joint robot.

Another situation occurs when the number of arm DOF is greater than the number of controlled outputs ($N > M$) which is the case for the model structure described in Section 3.1. The kinematic relation (3) is then non-invertible, and solving the DAE, (2) and (3), is the natural solution. One approximation would be to invert the linearized system, and to use a gain-scheduled feedforward controller. However, for nonlinear inverse dynamics of the system, the DAE approach should be considered.

5 DAE Background

The presentation in this section is primarily based on Brenan et al. (1996) and Kumar and Daoutidis (1999). Solving the inverse dynamics problem for states and control signals, given the desired output, generally involves solving the DAE described by (2) and (3). As the name implies, a DAE consists of differential and algebraic equations. A DAE can generally be expressed by the fully-implicit description

$$F(\dot{x}, x, u, t) = 0, \tag{4}$$

where $x \in R^n$ is the state vector, $u \in R^p$ is the control input and $F : R^{2n+p+1} \rightarrow R^m$. If $F_{\dot{x}} = \partial F / \partial \dot{x}$ is nonsingular, (4) represents an implicit ODE. Otherwise it represents a DAE which in general is considerably harder to solve than an ODE. The (differential) index, ν of a DAE provides a measure of the "singularity" of the DAE. Generally, the higher the index, the harder the DAE is to solve. An ODE has $\nu = 0$ and a DAE with $\nu > 1$ is denoted a high-index DAE. The index can somewhat simplified be defined as *the minimum number of times that all or part of* (4) *must be differentiated to determine* $\dot{x}$ *as a function of* t, x, u, *and higher derivatives of* u. A semi-explicit DAE is a special case of (4) described as

$$\dot{x} = F(x, y), \tag{5a}$$

$$0 = G(x, y), \tag{5b}$$

where u and t are omitted. Differentiation of (5b) w.r.t. time yields

$$0 = G_y(x, y)\dot{y} + G_x(x, y)\dot{x}, \tag{6}$$

and if G_y is nonsingular it is possible to solve for $\dot{y}$ and the index is equal to 1. Further on, by the implicit function theorem, it is also possible (at least numerically) to solve for

$y = \phi(x)$. This suggests a straightforward method for solving an Index-1 DAE, which in its simplest Forward Euler form yields (h is the step length):

1. Compute $x_{t+1} = x_t + hF(x_t, y_t)$
2. Solve y_{t+1} from $G(x_{t+1}, y_{t+1}) = 0$

For arbitrary initial conditions, the DAE solution exhibits an impulse behavior while an ODE solution is well-defined for any initial conditions. The reason is that the solutions are restricted to a space with dimension less than n by the algebraic equation (5b) and its derivatives up to order $\nu - 1$. These differentiated constraints are often denoted implicit constraints. Another difference from an ODE is that the DAE solution may depend on the derivatives of the input u.

The main method for solving DAEs is to reduce the index by some sort of repeated differentiation until Index-1 or Index-0 form is reached. Many index reduction techniques exist. In the numerical solution following the index reduction, a problem denoted as *drift off* can occur. This means that the solution diverges from the algebraic constraint as it is replaced by a differentiated constraint. In these cases a method for keeping the solution in the allowed solution space must be used. Several methods exists, see, e.g., Mattson and Söderlind (1993) and references therein.

One common software for solving Index-1 or Index-0 DAEs is called DASSL. The basic principle in DASSL is to replace the derivatives in (4) with a backwards differentiation formula (BDF) of order k. Some higher index DAEs can be solved directly by using the simple 1-step BDF (Euler Backwards). The *drift off* problem then disappears if the original algebraic constraint is kept.

6 Inverse Dynamics Solution by Index Reduction

The previously described model, (2) and (3), can in principle be written in semi-explicit form since the mass matrices M_a and M_m are nonsingular. Repeated differentiation of the algebraic constraint (3), and substitution of differentiated states, gives the following system where $x^{[i]}$ denotes $d^i x/dt^i$:

$$\dot{v}_a = \eta_a(q_a, q_m, v_a, v_m), \tag{7a}$$
$$\dot{v}_m = \eta_m(q_a, q_m, v_a, v_m) + M_m^{-1}u, \tag{7b}$$
$$\dot{q}_a = v_a, \tag{7c}$$
$$\dot{q}_m = v_m, \tag{7d}$$
$$0 = \Gamma(q_a) - Z, \tag{7e}$$
$$0 = \dot{\Gamma}(q_a, v_a) - \dot{Z}, \tag{7f}$$
$$0 = \ddot{\Gamma}(q_a, v_a, q_m, v_m) - \ddot{Z}, \tag{7g}$$
$$0 = \Gamma^{[3]}(q_a, v_a, q_m, v_m, u) - Z^{[3]}, \tag{7h}$$
$$0 = \Gamma^{[4]}(q_a, v_a, q_m, v_m, u, \dot{u}) - Z^{[4]}. \tag{7i}$$

Note that the position and speed dependent terms from (2) multiplied by the inverse mass matrices are denoted η_a and η_m. The control signal u is here regarded as a state, and the

full state vector is $x = [q_a^T, q_m^T, v_a^T, v_m^T, u^T]^T$. If $\dot{u}$ can be solved from (7i) we have an Index-4 DAE, and the system consisting of (7a) - (7d) and (7h) (which is Index-1) can be solved with, e.g., the DASSL software. One problem remains: the *drift off* problem described in Section 5 must be handled in some way.

7 Inverse Dynamics Solution by 1-step BDF

The proposed solution method is the constant step size 1-step BDF applied on the original system where the DAE-index is reduced from 4 to 3 by discarding the motor torque equation (2b) from the system. When the remaining DAE is solved, the control signal u can be computed from the states. Note that a friction compensation, assuming friction in the gearbox and motor, can be added to the motor torque without increasing the complexity of the solution. The DAE system to solve is then reduced to

$$M_a(q_a)\dot{v}_a = c(q_a, v_a) + g(q_a) + \tau_a(q_a, q_m, v_a, v_m), \tag{8a}$$

$$\dot{q}_a = v_a, \tag{8b}$$

$$\dot{q}_m = v_m, \tag{8c}$$

$$Z = \Gamma(q_a), \tag{8d}$$

with $q_a \in R^N$, $q_m \in R^M$, $v_a \in R^N$, and $v_m \in R^M$, which gives $2(N + M)$ states and $2(N + M)$ equations, and thus we have a determined system. Note that the term state is somewhat misused here as all variables in a DAE do not hold a system memory. The *states* of the system are $x = [q_a^T, q_m^T, v_a^T, v_m^T]^T$, and with reference $Z_d = Z_d(t)$ implicit in the equation the system can then be described as

$$F(\dot{x}, x, t) = 0. \tag{9}$$

Consistent initial conditions w.r.t. all explicit and implicit constraints must be given to avoid initial transients. Moreover, the trajectory reference must be sufficiently smooth as it must be (implicitly) differentiated four times for the (hidden) Index-4 system. This is accomplished by using a reference Z_d with $Z_d^{[4]}$, i.e., jerk derivative, well defined. The final algorithm can be described as follows:

1. Consistent $x(0)$ must be given
2. Solve $F(\frac{x(t+h)-x(t)}{h}, x_{t+h}, t + h) = 0$
3. Compute control signal $u_{ffw}(t + h)$
4. Repeat from 2 with $t = t + h$

A nonlinear equation solver from Matlab (fsolve) is used in step 2. Experiments show that the solvability of the DAE using this method depends on the size of the system (method tested with some positive result for 12 DOF) as well as on the step size selection (short step size is hard), and that the described index reduction increases the solvability.

Some commercially available software packages (Dymola, Maple) were also tried on the same systems, but none was able to solve these equations if the number of DOF

exceeded 5. Numerical problems could also be seen for the cases where a solution was found.

Thus, the suggested method works better than the commercial software packages tested, and is adequate for a preliminary evaluation of the feedforward control method.

8 Performance Requirement Specification

The problem specification illustrates a typical requirement for a dispensing application (e.g., gluing inside a car body) and is stated as follows:

- The programmed path should be followed by an accuracy of $2\,\text{mm}$ (maximum deviation) at an acceleration of $15\,\text{m}/s^2$ and a speed of $0.5\,\text{m/s}$.
- The specification above must be fulfilled for model errors in the tool load by ± 10 kg (i.e., $\pm 5\,\%$ of mass 3).
- The test path is a circular path with radius $25\,\text{mm}$.

9 Simulation Example

The robot model described in Section 3.2 is simulated with the controller structure from Section 4. The FFW block is implemented using the DAE solver from Section 7, and the FDB block is a diagonal PID controller. The implementation is discrete time with the step size of DAE solver equal to the sample time of the feedback controller.

No measurement noise or disturbances are used in the simulation as the purpose is to verify the feedforward algorithm. The Cartesian Trajectory Reference Z_d is a circle computed in polar coordinates [radius r, angle Q] by integration of a desired jerk derivative $Q^{[4]}(t)$ shown in Figure 6. The initial position of the robot is according to Section 3.2.

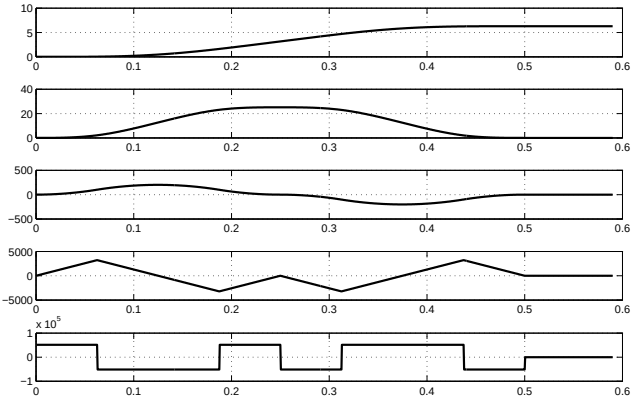


Figure 6: *The Circular Angle Q and its derivatives.*

The proposed algorithm (DAEFFW) is compared to the standard *flexible joint* feedforward (FJFFW) approach described previously. The non-actuated DOF is then regarded as rigid by the controller. The remaining stiffness parameters k_y^1 and k_y^2 are computed to include the k_y^3 in the best possible way.

The result of the simulations (control signals DAEFFW and path error for both methods) with nominal parameter values and sample time 0.5 ms can be seen in Figure 7 and 8. The circular path is shown in Figure 9. As an illustration of the need for high bandwidth modelbased feedforward at high speed path tracking, the result for the PID controller with no feedforward is included in the last figure.

The maximum error for DAEFFW is 0.32 mm (nominal parameters) and 1.39 mm (tool load ± 10 kg). The corresponding values for FJFFW is 1.74 mm and 2.82 mm. The result confirms that feedforward is naturally sensitive to model errors but the result also shows that the specification could be fulfilled, and that a more advanced feedforward still can yield better result than a less complex one. There are many ways to handle the robustness problem of robot feedforward control. Some suggestions:

- Reducing uncertainty by identification of uncertain parameters (e.g., tool load identification).
- Improved feedback control with arm side sensors added.
- Smoothing the trajectory on the expense of cycle time performance.

Increasing the sample time to 1 ms increases the nominal error for DAEFFW to 0.64 mm. This sample time sensitivity could also be expected since we are using an Euler Backwards approximation of the derivatives and this could certainly be improved.

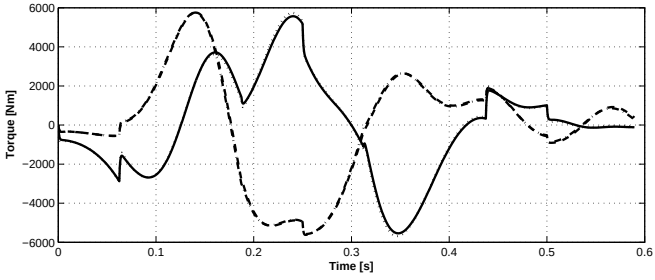


Figure 7: *DAE FFW: Torque feedforward (1-solid, 2-dashed) and simulated robot torque (1-dotted, 2-dashdot).*

10 Conclusions and Future Work

The proposed feedforward method shows promising results. The method is sensitive to model errors as can be expected for a feedforward method. The sampling time selection is critical for good performance as well as for solvability of the DAE. The limitation of this method is that the system must be minimum phase.

Future work will include testing the method on a more complex robot model, e.g., by increasing the number of DOF and by introducing non linear elasticities, as well as testing the method on real robot structures. Alternative DAE solvers and robustness issues should also be addressed in the future. Since the model structure presented can become non-minimum phase, methods for dealing with this is of greatest importance.

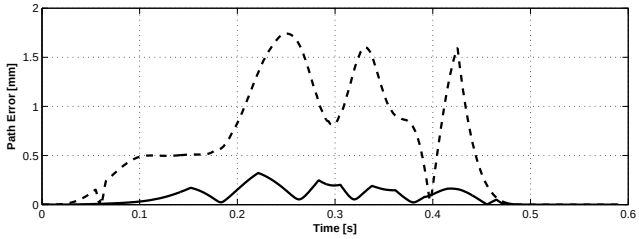


Figure 8: *Path Error for DAE FFW (solid) and Flexible Joint FFW (dashed).*

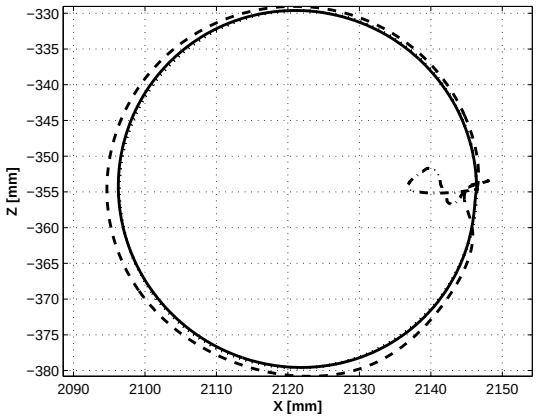


Figure 9: *Cartesian Path: Reference (dotted), DAE FFW (solid), Flexible Joint FFW (dashed) and PID Control (dashdot).*

References

Albu-Schäffer, A. and Hirzinger, G. (2000). State feedback controller for flexible joint robots: A globally stable approach implemented on DLR's light-weight robots. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1087–1093, Takamatsu, Japan.

An, C., Atkeson, C., and Hollerbach, J. (1988). *Model-Based Control of a Robot Manipulator.* The MIT press, Cambridge, Massachusetts.

Blajer, W. (1997). Dynamics and control of mechanical systems in partly specified motion. *Journal of the Franklin Institute*, 334B(3):407–426.

Blajer, W. and Kolodziejczyk, K. (2004). A geometric approach to solving problems of control constraints: Theory and a dae framework. *Multibody System Dynamics*, 11(4):341–350.

Brenan, K. E., Campbell, S. L., and Petzold, L. (1996). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Caccavale, F. and Chiacchio, P. (1994). Identification of dynamic parameters and feedforward control for a conventional industrial manipulator. *Control Eng. Practice*, 2(6):1039–1050.

De Luca, A. (2000). Feedforward/feedback laws for the control of flexible robots. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 233–240, San Francisco, CA.

De Luca, A. and Lucibello, P. (1998). A general algorithm for dynamic feedback linearization of robots with elastic joints. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 504–510, Leuven, Belgium.

Jankowski, K. and Van Brussel, H. (1992). An approach to discrete inverse dynamics control of flexible-joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 114:229–233.

Kane, T. R. and Levinson, D. A. (1985). *Dynamics: Theory and Applications*. McGraw-Hill Publishing Company.

Kumar, A. and Daoutidis, P. (1999). *Control of Nonlinear Differential Algebraic Equation Systems with Applications to Chemical Processes*. Taylor & Francis Ltd.

Lesser, M. (2000). *The Analysis of Complex Nonlinear Mechanical Systems: A Computer Algebra assisted approach*. World Scientific Publishing Co Pte Ltd, Singapore.

Mattson, S.-E. and Söderlind, G. (1993). Index reduction in differential-algebraic equations using dummy derivatives. *SIAM Journal of Scientific Computing*, 14:677–692.

Moberg, S. and Öhr, J. (2005). Robust control of a flexible manipulator arm: A benchmark problem. Prague, Czech Republic. 16th IFAC World Congress.

Öhr, J., Moberg, S., Wernholt, E., Hanssen, S., Pettersson, J., Persson, S., and Sander-Tavallaey, S. (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. In *Proc. ISMA2006 International Conference on Noise and Vibration Engineering*, pages 3305–3314, Leuven, Belgium.

Rouchon, P., Fliess, M., Lévine, J., and Martin, P. (1993). Flatness, motion planning and trailer systems. In *Proceedings of the 32nd Conference on Decision and Control*, pages 2700–2705, San Antonio, Texas.

Santibanez, V. and Kelly, R. (2001). PD control with feedforward compensation for robot manipulators: analysis and experimentation. *Robotica*, 19:11–19.

Spong, M. W. (1987). Modeling and control of elastic joint robots. *Journal of Dynamic Systems, Measurement, and Control*, 109:310–319.

Thümmel, M., Otter, M., and Bals, J. (2001). Control of robots with elastic joints based on automatic generation of inverse dynamic models. In *Proceedings of the 2001 IEEE/RSJ International Conference in Intelligent Robots and Systems*, pages 925–930, Maui, Hawaii, USA.

Appendix

A The Complete Model Equations

The 5 DOF model from Section 3.2 can be described by the following equations where the position and speed dependent terms from (2) are denoted γ_a and γ_m:

$$\begin{aligned} M_a(q_a)\dot{v}_a &= \gamma_a(q_a, q_m, v_a, v_m), \\ M_m\dot{v}_m &= \gamma_m(q_a, q_m, v_a, v_m) + u, \\ \dot{q}_a &= v_a, \\ \dot{q}_m &= v_m, \\ Z &= \Gamma(q_a). \end{aligned}$$

The following shorthand notation is used:

$$s_y^1 = \sin q_y^1, \quad c_y^1 = \cos q_y^1 \quad etc.$$

The inertia matrices M_a and M_m are defined and computed as:

$$
\begin{aligned}
M_m &= \begin{bmatrix} J_m^1 & 0 \\ 0 & J_m^2 \end{bmatrix}, \\
M_a &= \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}, \\
M_{11} &= (\xi_x^1)^2 m_1 + (s_y^2)^2 (l_x^1)^2 m_2 - (-c_y^2 l_x^1 - \xi_x^2) m_2 (c_y^2 l_x^1 \\
&\quad + \xi_x^2) + j_{yy}^1 - (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 + c_y^3 s_y^2 l_x^1) m_3 (-s_y^3 c_y^2 \\
&\quad l_x^1 - c_y^3 s_y^2 l_x^1 - s_y^3 l_x^2) + j_{yy}^2 - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 \\
&\quad - c_y^3 l_x^2) m_3 (c_y^3 c_y^2 l_x^1 + c_y^3 l_x^2 - s_y^3 s_y^2 l_x^1 + \xi_x^3) + j_{yy}^3, \\
M_{12} &= j_{yy}^3 + j_{yy}^2 - (-c_y^2 l_x^1 - \xi_x^2) m_2 \xi_x^2 + (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 \\
&\quad + c_y^3 s_y^2 l_x^1) m_3 s_y^3 l_x^2 - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 - \\
&\quad c_y^3 l_x^2) m_3 (c_y^3 l_x^2 + \xi_x^3), \\
M_{13} &= j_{yy}^3 - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 - c_y^3 l_x^2) m_3 \xi_x^3, \\
M_{21} &= \xi_x^2 m_2 (c_y^2 l_x^1 + \xi_x^2) - s_y^3 l_x^2 m_3 (-s_y^3 c_y^2 l_x^1 - c_y^3 s_y^2 l_x^1 \\
&\quad - s_y^3 l_x^2) - (-\xi_x^3 - c_y^3 l_x^2) m_3 (c_y^3 c_y^2 l_x^1 + c_y^3 l_x^2 - s_y^3 s_y^2 l_x^1 \\
&\quad + \xi_x^3) + j_{yy}^2 + j_{yy}^3, \\
M_{22} &= (\xi_x^2)^2 m_2 - (-\xi_x^3 - c_y^3 l_x^2) m_3 (c_y^3 l_x^2 + \xi_x^3) + j_{yy}^2 \\
&\quad + (s_y^3)^2 (l_x^2)^2 m_3 + j_{yy}^3, \\
M_{23} &= j_{yy}^3 - (-\xi_x^3 - c_y^3 l_x^2) m_3 \xi_x^3, \\
M_{31} &= \xi_x^3 m_3 (c_y^3 c_y^2 l_x^1 + c_y^3 l_x^2 - s_y^3 s_y^2 l_x^1 + \xi_x^3) + j_{yy}^3, \\
M_{32} &= \xi_x^3 m_3 (c_y^3 l_x^2 + \xi_x^3) + j_{yy}^3, \\
M_{33} &= (\xi_x^3)^2 m_3 + j_{yy}^3.
\end{aligned}
$$

The kinematics is computed as:

$$
\begin{aligned}
\Gamma &= \begin{bmatrix} x \\ z \end{bmatrix}, \\
x &= c_y^3 c_y^1 c_y^2 l_x^3 + c_y^1 l_x^1 - s_y^3 s_y^1 c_y^2 l_x^3 - s_y^3 c_y^1 s_y^2 l_x^3 - c_y^3 s_y^1 s_y^2 l_x^3 \\
&\quad - s_y^1 s_y^2 l_x^2 + c_y^1 c_y^2 l_x^2, \\
z &= -c_y^3 c_y^1 s_y^2 l_x^3 - c_y^3 s_y^1 c_y^2 l_x^3 - s_y^1 l_x^1 - s_y^3 c_y^1 c_y^2 l_x^3 + s_y^3 s_y^1 s_y^2 l_x^3 \\
&\quad - c_y^1 s_y^2 l_x^2 - s_y^1 c_y^2 l_x^2.
\end{aligned}
$$

Finally, the position and speed dependent terms are computed as:

$$
\begin{aligned}
\gamma_{a1} = {} & s_y^2 l_x^1 m_2((v_y^2)^2\xi_x^2 + c_y^2(v_y^1)^2 l_x^1 + 2v_y^1 v_y^2 \xi_x^2 + (v_y^1)^2\xi_x^2) \\
& + (-c_y^2 l_x^1 - \xi_x^2)m_2 s_y^2 (v_y^1)^2 l_x^1 + (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 \\
& + c_y^3 s_y^2 l_x^1)m_3(2v_y^1 v_y^2 \xi_x^3 + 2v_y^3 v_y^1 \xi_x^3 + (v_y^1)^2\xi_x^3 + (v_y^3)^2\xi_x^3 \\
& + 2v_y^3 v_y^2 \xi_x^3 - s_y^3 s_y^2 (v_y^1)^2 l_x^1 + 2c_y^3 v_y^1 v_y^2 l_x^2 + c_y^3 (v_y^2)^2 l_x^2 \\
& + (v_y^2)^2\xi_x^3 + c_y^3 (v_y^1)^2 l_x^2 + c_y^3 c_y^2 (v_y^1)^2 l_x^1) + (-\xi_x^3 \\
& + s_y^3 s_y^2 l_x^1 - c_y^3 c_y^2 l_x^1 - c_y^3 l_x^2)m_3(c_y^3 s_y^2 (v_y^1)^2 l_x^1 \\
& + s_y^3 c_y^2 (v_y^1)^2 l_x^1 + 2s_y^3 v_y^1 v_y^2 l_x^2 + s_y^3 (v_y^2)^2 l_x^2 + s_y^3 (v_y^1)^2 l_x^2) \\
& + \xi_x^1 c_y^1 m_1 g - s_y^2 l_x^1(-s_y^1 c_y^2 - c_y^1 s_y^2)m_2 g - (-c_y^2 l_x^1 - \xi_x^2) \\
& (c_y^1 c_y^2 - s_y^1 s_y^2)m_2 g - (s_y^3 c_y^2 l_x^1 + s_y^3 l_x^2 + c_y^3 s_y^2 l_x^1)((-s_y^1 c_y^2 \\
& - c_y^1 s_y^2)c_y^3 - (c_y^1 c_y^2 - s_y^1 s_y^2)s_y^3)m_3 g - (-\xi_x^3 + s_y^3 s_y^2 l_x^1 \\
& - c_y^3 c_y^2 l_x^1 - c_y^3 l_x^2)((-s_y^1 c_y^2 - c_y^1 s_y^2)s_y^3 + (c_y^1 c_y^2 - s_y^1 s_y^2)c_y^3) \\
& m_3 g - k_y^1(q_y^1 - q_m^1) - d_y^1(v_y^1 - v_m^1), \\
\gamma_{a2} = {} & -\xi_x^2 m_2 s_y^2 (v_y^1)^2 l_x^1 + s_y^3 l_x^2 m_3(2v_y^1 v_y^2 \xi_x^3 + 2v_y^3 v_y^1 \xi_x^3 \\
& + (v_y^1)^2\xi_x^3 + (v_y^3)^2\xi_x^3 + 2v_y^3 v_y^2 \xi_x^3 - s_y^3 s_y^2 (v_y^1)^2 \\
& l_x^1 + 2c_y^3 v_y^1 v_y^2 l_x^2 + c_y^3 (v_y^2)^2 l_x^2 + (v_y^2)^2\xi_x^3 + c_y^3 (v_y^1)^2 \\
& l_x^2 + c_y^3 c_y^2 (v_y^1)^2 l_x^1) + (-\xi_x^3 - c_y^3 l_x^2)m_3(c_y^3 s_y^2 (v_y^1)^2 l_x^1 \\
& + s_y^3 c_y^2 (v_y^1)^2 l_x^1 + 2s_y^3 v_y^1 v_y^2 l_x^2 + s_y^3 (v_y^2)^2 l_x^2 + s_y^3 (v_y^1)^2 l_x^2) \\
& + \xi_x^2(c_y^1 c_y^2 - s_y^1 s_y^2)m_2 g - s_y^3 l_x^2((-s_y^1 c_y^2 - c_y^1 s_y^2)c_y^3 \\
& - (c_y^1 c_y^2 - s_y^1 s_y^2)s_y^3)m_3 g - (-\xi_x^3 - c_y^3 l_x^2)((-s_y^1 c_y^2 \\
& - c_y^1 s_y^2)s_y^3 + (c_y^1 c_y^2 - s_y^1 s_y^2)c_y^3)m_3 g \\
& - k_y^2(q_y^2 - q_m^2) - d_y^2(v_y^2 - v_m^2), \\
\gamma_{a3} = {} & -\xi_x^3 m_3(c_y^3 s_y^2 (v_y^1)^2 l_x^1 + s_y^3 c_y^2 (v_y^1)^2 l_x^1 + 2s_y^3 v_y^1 v_y^2 l_x^2 \\
& + s_y^3 (v_y^2)^2 l_x^2 + s_y^3 (v_y^1)^2 l_x^2) + \xi_x^3((-s_y^1 c_y^2 - c_y^1 s_y^2)s_y^3 \\
& + (c_y^1 c_y^2 - s_y^1 s_y^2)c_y^3)m_3 g - k_y^3 q_y^3 - d_y^3 v_y^3, \\
\gamma_{m1} = {} & k_y^1(q_y^1 - q_m^1) + d_y^1(v_y^1 - v_m^1), \\
\gamma_{m2} = {} & k_y^2(q_y^2 - q_m^2) + d_y^2(v_y^2 - v_m^2).
\end{aligned}
$$

Paper C

A Benchmark Problem for Robust Feedback Control of a Flexible Manipulator

Edited version of the paper:

Moberg, S., Öhr, J., and Gunnarsson, S. (2007). A benchmark problem for robust feedback control of a flexible manipulator. Technical Report LiTH-ISY-R-2820, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden. *Submitted to IEEE Transactions on Control Systems Technology*.

Parts of the paper in:

Moberg, S. and Öhr, J. (2005). Robust control of a flexible manipulator arm: A benchmark problem. Prague, Czech Republic. 16th IFAC World Congress.

A Benchmark Problem for Robust Feedback Control of a Flexible Manipulator

Stig Moberg[1,2] and Jonas Öhr[3] and Svante Gunnarsson[1]

[1]Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden.
E-mail: {stig,svante}@isy.liu.se.

[2]ABB AB – Robotics,
SE–721 68 Västerås, Sweden.

[3]ABB AB – Corporate Research,
SE–721 78 Västerås, Sweden.

Abstract

A benchmark problem for robust feedback control of a flexible manipulator is presented together with some suggested solutions. The system to be controlled is a four-mass system subject to input saturation, nonlinear gear elasticity, model uncertainties, and load disturbances affecting both the motor and the arm. The system should be controlled by a discrete-time controller that optimizes performance for given robustness requirements.

Keywords: Robots, manipulators, flexible structures, robustness, position control

1 Introduction

Experiments are essential in control technology research. A method that has been developed by means of realistic experiments has a larger potential to work in reality compared to methods that have not. Benchmark problems, often given in the form of mathematical equations embodied as software simulators together with performance specifications, can serve as substitute for real control experiments. A benchmark problem should be sufficiently realistic and complete, but also avoid unmotivated complexity. This paper presents an industrial benchmark problem with the intention to stimulate research in the area of robust control of flexible industrial manipulators (robots). Some proposed solutions to this benchmark problem will also be presented and discussed. The authors hope

that the degree of authenticity is just right, and that researchers will take on the problem and eventually propose solutions and methods. When evaluating strengths and drawbacks of a certain method in the light of realistic examples, control engineers and researchers will find it easier to interpret results, and the chance for the proposed method to be used in a real application increases. Thus, it is believed that the benchmark problem presented below can help increasing the ratio of control methods used outside the academic world to control methods proposed in literature. An example of a similar (at least in some parts) benchmark problem is the flexible transmission system presented by Landau et al. (1995). Another benchmark problem for controller design is Graebe (1994) where the participants did not know the true system, which was supplied in the form of scrambled simulation code. A third example is the Grumman F-14 Benchmark Control Problem described in Rimer and Frederick (1987). However, in the area of robot manipulator control it is believed that a realistic and relevant industrial benchmark problem is needed for reasons stated above. The paper is organized as follows. Section 2 presents the original control problem and discusses some of the main aspects of the problem, and then in Section 3 the nonlinear simulation model as well as a linearized model are presented. An experimental model validation is presented in Section 4. The control design task is described in Section 5, and some suggested solutions are presented in Section 6.

2 Problem Description

The most common type of industrial manipulator has six serially mounted links, all controlled by electrical motors via gears. An example of a serial industrial manipulator is shown in Figure 1.

The dynamics of the manipulator change rapidly as the robot links move fast within its working range, and the dynamic couplings between the links are strong. Moreover, the structure is elastic and the gears have nonlinearities such as backlash, friction, and nonlinear elasticity. From a control engineering perspective a manipulator can be described as a nonlinear multivariable dynamical system having the six motor currents as the inputs, and the six measurable motor angles as outputs. The goal of the motion control is to control the *orientation* and the *position* of the *tool* when moving the tool along a certain desired path.

The benchmark problem described in this paper concerns only the so-called regulator problem, where a feedback controller should be designed such that the actual tool position is close to the desired reference, in the presence of motor torque disturbances, e.g., motor torque ripple, and tool disturbances acting on the tool, e.g., under material processing. For simplicity only the first axis of a horizontally mounted manipulator will be considered here. The remaining axes are positioned in a fixed configuration. In this way the influence of the nonlinear rigid body dynamics associated with the change of configuration (operating point) as well as gravity, centripetal, and Coriolis torques can be neglected. Moreover, the remaining axes are positioned to minimize the couplings to the first axis. In this way, the control problem concerning the first axis can be approximated as a SISO control problem.

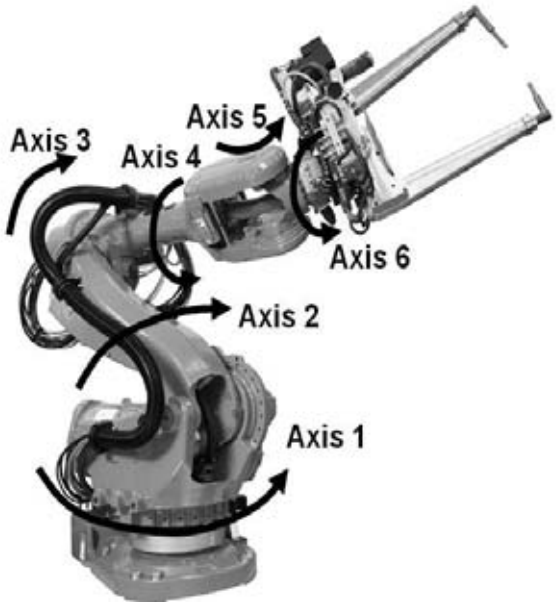


Figure 1: *IRB6600ID from ABB equipped with a spot welding gun.*

3 Mathematical Models

3.1 Nonlinear Simulation Model

The simulation model to be used is a four-mass model having nonlinear gear elasticity. The motor current- and torque control is assumed to be ideal so that the motor torque becomes the model input. The gearbox and motor friction effects are approximately assumed to be linear. In reality the friction normally exhibits nonlinear behavior, but, as illustrated in the model validation, the model gives a realistic description of the real system even though the friction is assumed to be linear. The model is illustrated in Figure 2. Experiments have shown that a two-mass model is not sufficient in order to describe the flexibilities. The results in Östring et al. (2003) show that at least a three-mass model is needed in order to model the dynamics of the first axis of a moderate size robot. In Öhr et al. (2006) it can be seen than even higher model order can be necessary for some axes. Based on the authors experience, the suggested model structure is adequate for the regulator problem considered in this paper. The model structure is further justified in Section 4.

The rotating masses are connected via spring-damper pairs. The first spring-damper pair, corresponding to the gear, has linear damping d_1 but nonlinear elasticity k_1. A typical relationship between deflection and torque is illustrated in Figure 3. In the simulation model the nonlinear gear elasticity is approximated by a piecewise linear function having five segments. The second and third spring-damper pair are both assumed to be linear and represented by d_2, k_2, d_3 and k_3.

The moment of inertia of the arm is here split-up into the three components J_{a1}, J_{a2},

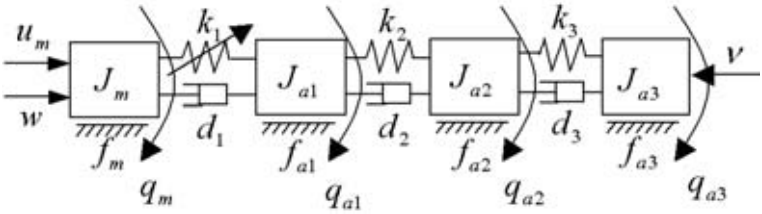


Figure 2: *Simulation model of the robot arm.*

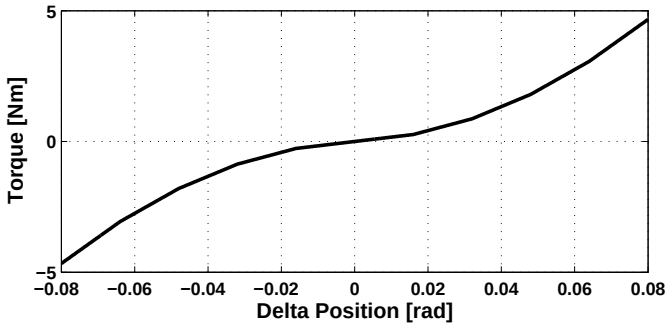


Figure 3: *Nonlinear gear elasticity: Torque as function of deflection.*

and J_{a3}. The moment of inertia of the motor is J_m. The parameters f_m, f_{a1}, f_{a2}, and f_{a3} represent viscous friction in the motor and in the arm structure respectively. The motor torque u_m, which is the manipulated input of the system, is limited to ± 20 Nm. The disturbance torque acting on the motor and tool are denoted w and v respectively. The only measured output signal is the motor angle q_m, and this signal is subject to a measurement disturbance and a time delay. The variables q_{a1}, q_{a2}, and q_{a3} are arm angles of the three masses, and together they define the position of the tool. The angles in this model are, however, expressed on the high-speed side of the gear, so in order to get the real arm angles one must divide the model angles by the gear-ratio. Details concerning the implementation of the simulation model are given in Section 5.6.

3.2 Linearized Model

For control design purpose there is also a linearized, with respect to the nonlinear elasticity, version of the simulation model available. The linearized model is given by

$$J\ddot{q}(t) + (D + F)\dot{q}(t) + Kq(t) = u(t), \tag{1}$$

where

$$q(t) = \begin{bmatrix} q_m(t) & q_{a1}(t) & q_{a2}(t) & q_{a3}(t) \end{bmatrix}^T, \tag{2}$$

and

$$u(t) = \begin{bmatrix} u_m(t) + w(t) & 0 & 0 & v(t) \end{bmatrix}^T. \tag{3}$$

Furthermore

$$J = \mathrm{diag}(J_m, J_{a1}, J_{a2}, J_{a3}), \tag{4}$$

$$D = \begin{bmatrix} d_1 & -d_1 & 0 & 0 \\ -d_1 & d_1 + d_2 & -d_2 & 0 \\ 0 & -d_2 & d_2 + d_3 & -d_3 \\ 0 & 0 & -d_3 & d_3 \end{bmatrix}, \tag{5}$$

$$F = \mathrm{diag}(f_m, f_{a1}, f_{a2}, f_{a3}), \tag{6}$$

and

$$K = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 + k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2 + k_3 & -k_3 \\ 0 & 0 & -k_3 & k_3 \end{bmatrix}. \tag{7}$$

The tool position $z(t)$ (which is the controlled variable) can for small variations around a given working point be calculated as

$$z(t) = \frac{l_1 q_{a1}(t) + l_2 q_{a2}(t) + l_3 q_{a3}(t)}{n}, \tag{8}$$

where n is the gear-ratio and l_1, l_2, l_3 are distances between the (fictive) masses and the tool. Using state-space formulation the linearized system can now be described by

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{9}$$

$$y(t) = Cx(t) + n(t), \tag{10}$$

$$z(t) = Ex(t), \tag{11}$$

where $y(t)$ is the measured motor angle, $n(t)$ is measurement noise, and $z(t)$ the controlled variable. Selecting the states

$$x(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix}, \tag{12}$$

yields

$$A = \begin{bmatrix} 0 & I \\ -J^{-1}K & -J^{-1}(D+F) \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ J^{-1} \end{bmatrix}, \tag{13}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{14}$$

and

$$E = \begin{bmatrix} 0 & \frac{l_1}{n} & \frac{l_2}{n} & \frac{l_3}{n} & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{15}$$

The parameter values of the nominal model, which will be denoted by M_{nom}, are defined in Table 3 of the Appendix. For the piecewise linear spring elasticity only the first segment, $k_{1,low}$, the last segment, $k_{1,high}$, and the position difference where the last segment begins, $k_{1,pos}$, are given.

4 Model Validation

The model proposed in Section 3.1 is a simplification of the real problem. In order to illustrate that it, despite simplifications, is a realistic and relevant description of the problem for the purpose in this paper, some validation experiments will be presented. The model has been validated by identification and measurements on the first axis of a robot from the ABB IRB6600 series (see Figure 1) using an experimental controller. The model is semi physical, i.e., partly physical and partly gray-box with fictive physical elements. Some of the parameters were known in advance, e.g., motor inertia, total axis inertia, and the nonlinear gear box elasticity. Other parameters were identified by comparing the measured frequency response of the first robot axis with the frequency response of the linear model while adjusting the parameters. Example of frequency domain identified parameters are the dampers, remaining springs, and the distribution of link inertia. Finally, the distribution of the axis length parameters were adjusted to yield a similar time domain response for the tool position when step disturbances were applied to the system during closed loop control. The resulting parameters of the validation model are close to the parameters of the benchmark model in Table 3. The positions of the second and third robot links were chosen to place the tool in the middle of the working area, and the positions of the last three axes were chosen to minimize the coupling with the first axis. The frequency response was obtained by applying a multi-sine reference to a speed controller of PI type for the first axis, and measuring the motor position and motor torque. The excitation energy is distributed from 3 to 30 Hz. The frequency response function (FRF) was then computed, see, e.g., computation of ETFE in Ljung (1999). The FRF's for the real robot and the linear model are shown in Figure 4, and the agreement is good.

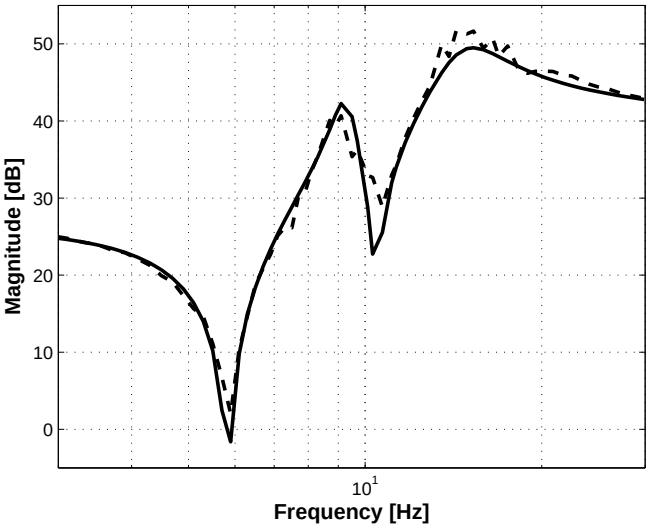


Figure 4: *The frequency response function of the linear model (solid) and of the real robot (dashed).*

The first robot axis was then controlled by using a reasonably tuned PID controller of the same type as the default controller in this benchmark problem. All links were controlled with similar controllers. Torque disturbances were applied, and the tool position was measured using a Leica laser measurement system LTD600 from Leica GeoSystems described in Leica (2007). The simulation model was then simulated in closed loop with the same controller and disturbance input as in the real robot system. Figure 5 shows the tool position when a constant torque disturbance acting on the tool is suddenly released.

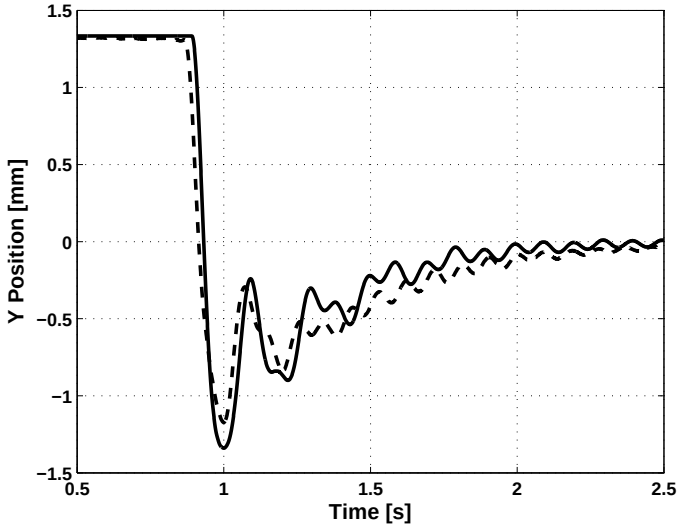


Figure 5: *The tool position for a tool step disturbance. Nonlinear simulation model (solid) and real robot (dashed).*

Figure 6 shows the tool position when applying a step in the motor torque. Note that some backlash for the real robot link is seen in the second figure.

Finally, the loop gain of the real system was increased until the stability limit was reached and the amplitude margin could be determined. The amplitude margin of the simulated system was in good correspondence with the real system. The identification and measurements show that the suggested model structure is valid for its use in this benchmark problem. The use of a SISO semi physical model, and thus neglecting the interaction with other links as well as neglecting the nonlinear friction, the motor dynamics and the motor torque control is reasonably well justified.

5 The Control Design Task

5.1 Introduction

The control problem can schematically be described as in Figure 7. The task can be expressed as a classical regulator problem where the aim is to reject the influence of load

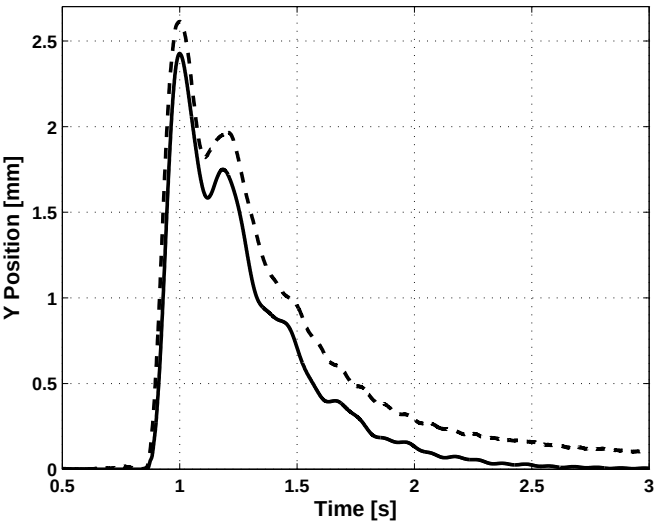


Figure 6: *The tool position for a motor step disturbance. Nonlinear simulation model (solid) and real robot (dashed).*

disturbances as much as possible, and at the same time avoid too large input signal, and be able to handle variations in the system dynamics.

5.2 Load and Measurement Disturbances

In reality an industrial robot is affected by various disturbances. One disturbance source is the electrical motor itself which generates torque ripple, and in the model this is modeled as a load disturbance w acting on the motor, i.e., the first mass in the model. Another disturbance source is the external forces that affect the tool during, e.g., material processing. This type of disturbance is modeled as a load disturbance v affecting the last mass in the model. In order to capture the various types of disturbances a specially designed sequence of disturbances will be used. It consists of torque disturbances acting on the motor and on the tool according to Figure 8, and it is a combination of steps, pulses, and sweeping sinusoids (chirps). The measurement disturbance n is modeled as a band limited random noise.

Figure 9 shows the tool position when the disturbance sequence in Figure 8 acts on the nominal system, and a PID-type controller is used.

Figure 10 shows the input signal (motor torque) under the same conditions, and here also the influence of the measurement disturbance is evident.

The various notations in Figures 9 and 10 will be used in Section 5.5, where a performance measure is formulated.

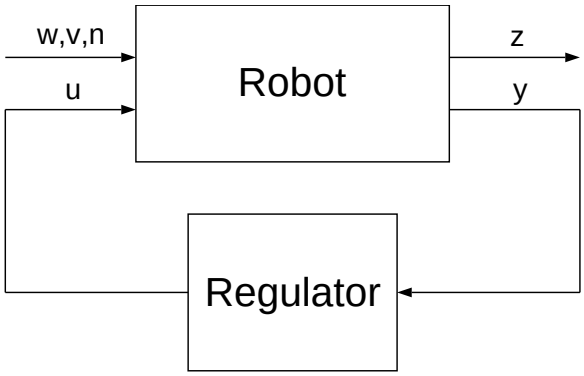


Figure 7: *Control system.*

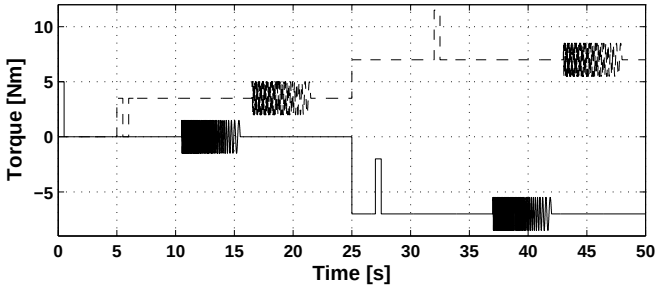


Figure 8: *Torque disturbances on motor (dashed) and tool (solid).*

5.3 Parameter Variations and Model Sets

The performance of the control systems will be evaluated for both the nominal model M_{nom} and for two sets of models which will be denoted by M_1 and M_2 respectively. In Figure 11 the frequency response function amplitude for of M_{nom} (torque to angular acceleration) is shown. The solid line corresponds to the stiffest region of the gear $(k_{1,high})$, and the dashed line corresponds to the least stiff region $(k_{1,low})$.

The sets M_1 and M_2 contain ten models each. The set M_1 represents relatively small variations in the physical parameters, and the set M_2 represents relatively large variations. The Figures 12 and 13 show the absolute value of the frequency responses of the models $m \in M_1$ and $m \in M_2$ respectively, for the stiffest region of the gear $(k_{1,high})$[1].

The uncertainty described by M_1 can be motivated by at least five sources of uncertainty:

I Model structure selection: The real robot is of infinite order, and the choice of model order always introduces errors. Non modeled or incompletely modeled nonlinearities such as friction and stiffness are other examples.

[1] In the simulation model, M_1 and M_2 also have the nonlinear gear elasticity described in Section 3.1.

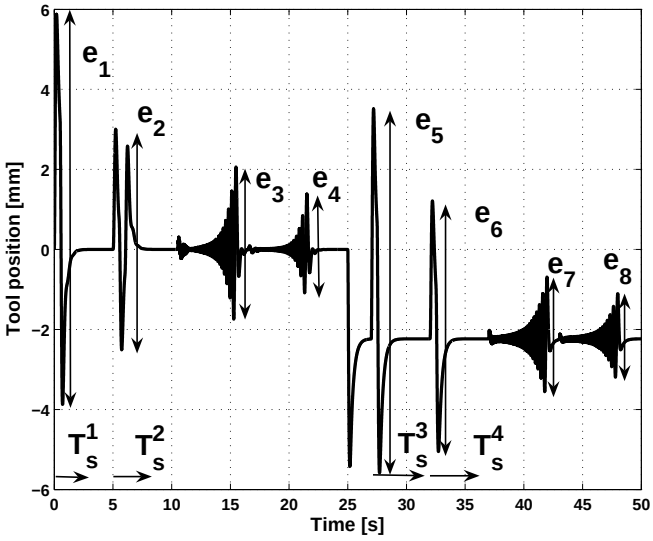


Figure 9: *Tool position when using PID-control.*

II Accuracy of nominal model parameters: Model parameters can be obtained by identification or by other types of measurements, and their values always have limited accuracy.

III Variation of model parameters for individual robots: Friction and stiffness are examples of parameters that can differ significantly from one robot individual to another. Temperature dependent parameters and aging also belong to this group.

IV Robot installation: The stiffness of the foundation where the robot is mounted, and the user definition of tool and payload (e.g., mass and center of mass), introduces uncertainty of this type. Elasticity in the tool or payload increases the uncertainty further.

V Controller implementation In a real implementation, the controller would probably be time varying by, e.g., gain scheduling. Errors due to gain scheduling of controllers for different operating points also adds to the total uncertainty.

The uncertainty described by M_2 can be motivated by the fact that a real control system must be stable even for relatively large deviations between the model and the real manipulator dynamics. It is important to understand that the uncertainty is partly a design choice, and depends of the actual implementation of the robot control system. One extreme is that the feedback controller has constant parameters for all configurations and all loads, and the other is that an extremely accurate model of the robot is implemented in the robot control system. This model can then be used for gain scheduling or directly used in the feedback controller. The first extreme would have a considerably larger model set M_1, and the second extreme would have a smaller set.

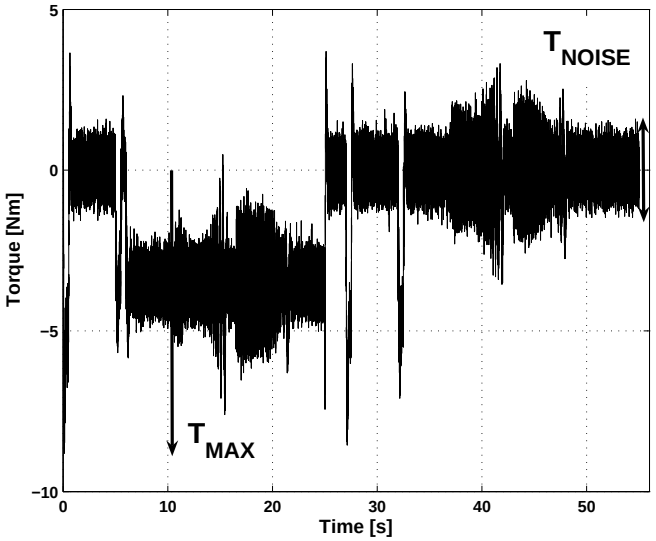


Figure 10: *Motor torque when using PID-control.*

5.4 The Design Task

The task of this benchmark problem is to minimize a performance measure by designing one or two discrete time controllers for the systems described above. The performance measure is described in Section 5.5, and the general requirements and some implementation constraints are described in Section 5.6. One of the controllers must be capable of controlling all the models $m \in M_{nom} \cup M_1 \cup M_2$ whereas the other controller should be able to control M_{nom} alone. The controllers can be linear or nonlinear. In order to investigate how well a controller can perform when really good models are available it is recommended to design two different controllers where one is optimized for the control of M_{nom} only. This controller will in the sequel be denoted by C_1 and the other by C_2. Note that C_1 and C_2 can be identical, can have the same structure and differ only by different tuning, or can have completely different structure and tuning parameters. The control requirements put on the systems in M_1 can be motivated by the fact that robust performance is required for this level of uncertainty. The use of the model set M_2 is motivated by robust stability as described above in the previous section.

5.5 Performance Measures

From an industrial viewpoint, time domain performance measures are to prefer when evaluating the control system. It is then a task for the control designer to translate these requirements to a form that suits the design method chosen, e.g., frequency domain norms for H_∞ design. Figures 9 and 10 show all the individual performance measures that will be weighted together into one cost function. The measures referring to the controlled

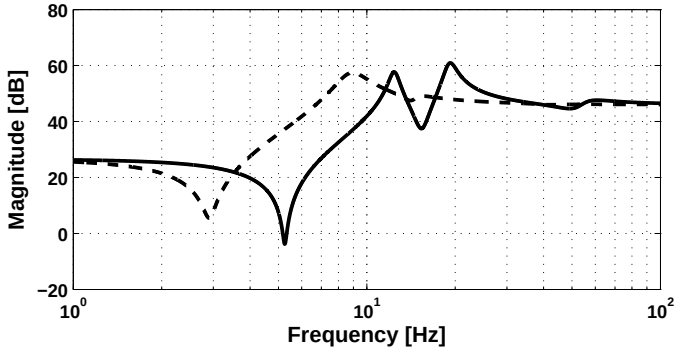


Figure 11: *Frequency response for M_{nom}. Stiff region (solid), least stiff region (dashed).*

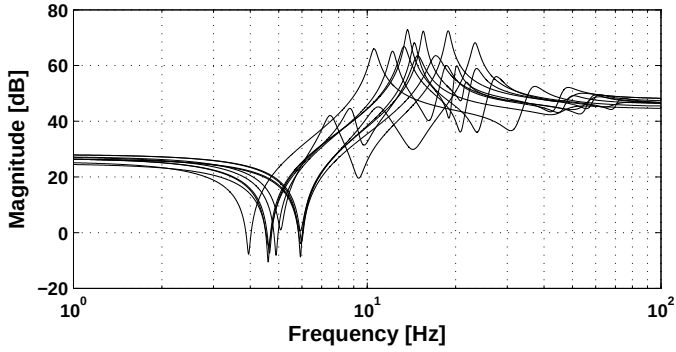


Figure 12: *Frequency response for all models in the set M_1.*

output variable, tool position, are

- Peak-to-peak error (e_1 - e_8).
- Settling times (Ts_1 - Ts_4),

and the measures related to the input signal, torque, are

- Maximum value T_{MAX}
- Adjusted rms value T_{RMS}
- Torque "noise" (peak-to-peak) T_{NOISE}.

Note that T_{NOISE}, which can be caused by measurement noise and/or chattering caused by a discontinuous controller, is measured by the simulation routines when the system is at rest but that a good controller would keep the chattering/noise on a decent level also when it operates actively.

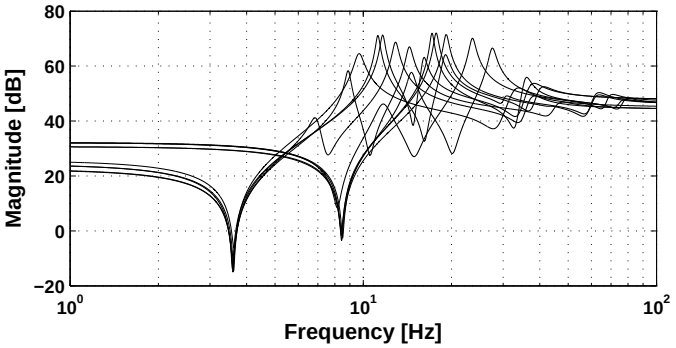


Figure 13: *Frequency response for all models in the set M_2.*

For the nominal system M_{nom} using controller C_1 the cost function V_{nom} is given by

$$V_{nom} = \gamma \sum_{i=1}^{15} \alpha_i e_i, \tag{16}$$

where e_i represents a generalized "error" (i.e., position error, settling time, or torque), γ and α_i are weights. For the set M_1 using controller C_2 the maximum error from the simulations are used and the cost functions V_1 is given by

$$V_1 = \gamma \sum_{i=1}^{15} \alpha_i \max_{m \in M_1} (e_i). \tag{17}$$

Similarly, for the set M_2 using controller C_2 the maximum error from the simulations are used and the cost functions V_2 is given by

$$V_2 = \gamma \sum_{i=1}^{15} \alpha_i \max_{m \in M_2} (e_i). \tag{18}$$

The total cost function V is given by

$$V = \beta_{nom} V_{nom} + \beta_1 V_1 + \beta_2 V_2. \tag{19}$$

It might then seem strange to weight performance for the set M_2 into the total cost function but it is motivated by a desire to reward the robustness of the proposed controller. It is unavoidable that this type of performance measures will be subjective, but they have been found to be relevant for a general purpose robot from an application viewpoint.

5.6 Implementation and Specifications

To evaluate a proposed control design a set of files has been developed. A control system in form of a Simulink^{TM}-model has been developed, and it is shown in Figure 14.

In the implementation the following conditions hold:

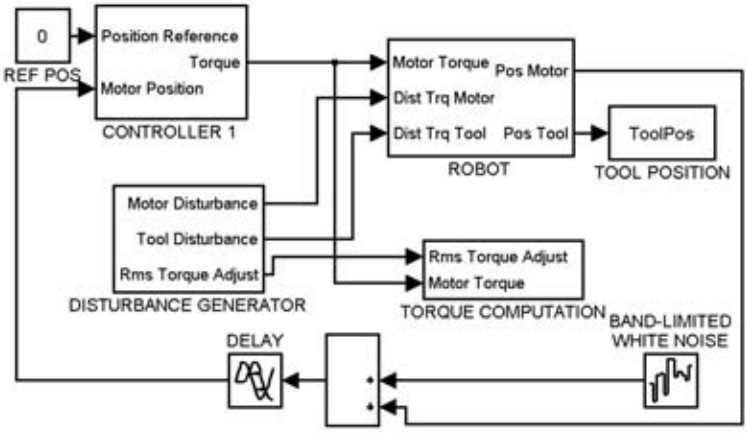


Figure 14: *The control system.*

- Sampling time 0.5 ms
- Time delay T_d 0.5 ms
- Torque saturation limits $\pm$20 Nm (the saturation function in the controller block should not be removed)

The control task is then formulated as minimizing the overall cost function V in (19) subject to the conditions

- Settling times for M_{nom} (C_1) and M_1 (C_2): $Ts_{1,2,3,4}$ < 3 s, error band $\pm$ 0.1 mm
- Settling times for M_2 (C_2): $Ts_{1,2,3,4}$ < 4 s, error band $\pm$ 0.3 mm
- Torque noise $T_{NOISE} < 5$ Nm for all systems
- Stability[2] for all systems
- Stability for M_{nom} when increasing the loop gain by a factor 2.5 for C_1 and C_2
- Stability for M_{nom} when increasing the delay T_d from 0.5 ms to 2 ms for C_1 and C_2

In addition some conditions concerning the implementation have to be considered:

- Only the blocks `Controller1` and `Controller2` and the files `Controller_1.m` and `Controller_2.m` are allowed to be changed.
- No continuous-time blocks are allowed be added.
- No knowledge of deterministic nature about the noise and disturbances is allowed to be used in the controller.

[2]The stability requirement also includes that limit cycles larger than 10 μm peak-to-peak are not allowed.

- The controller C_2 must have the same initial states and parameter values for all the simulations of $m \in M_{nom} \cup M_1 \cup M_2$

The control system and the models are described in detail by Matlab^{TM} and Simulink^{TM} files available for download at Moberg (2004). The system comes with a simple PID-controller. The Matlab^{TM} products used are described in, e.g., MathWorks (2003).

6 Suggested Solutions

This benchmark problem was first presented as "Swedish Open Championships in Robot Control". See Moberg and Öhr (2004) and Moberg and Öhr (2005). On request, the benchmark problem was spread outside Sweden. The four most interesting solutions were:

A A QFT controller proposed by P.-O. Gutman, Israel Institute of Technology, Israel.

B A QFT controller of order 13 proposed by O. Roberto, Uppsala University.

C A Polynomial Controller proposed by F. Sikström and A.-K. Christiansson, University of Trollhättan/Uddevalla, Sweden.

D A so called Linear Sliding Mode Controller proposed by W.-H. Zhu, Canadian Space Agency, Canada.

For the solutions A, C, and D the controllers are of order 3 to 7. The QFT approach is generally described in Nordin and Gutman (1995), and the linear sliding mode approach in Zhu et al. (1992) and Zhu et al. (2001). The polynomial controller is optimized for the given cost function and the optimization procedures used are described in MathWorks (1999). The frequency responses of the controllers are shown in Figure 15 and 16. The overall shape of the frequency functions are similar with a clear lead-lag character. Solution A differs due to the peak in the magnitude curve around 10 Hz, and the essentially higher high frequency (above 100 Hz) gain.

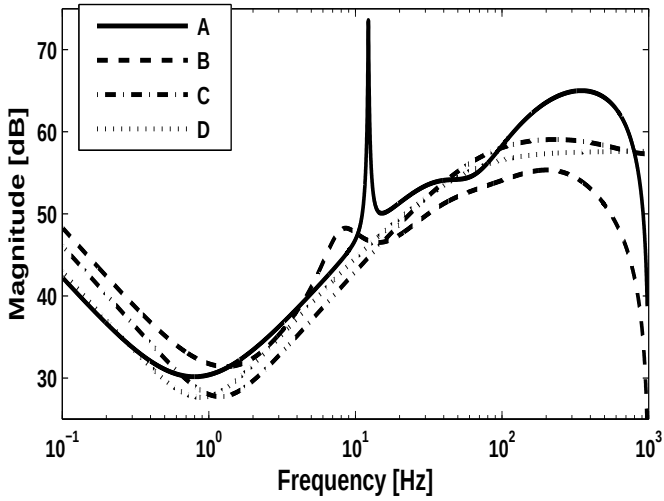


Figure 15: *The absolute value of the frequency response of the controllers C_2.*

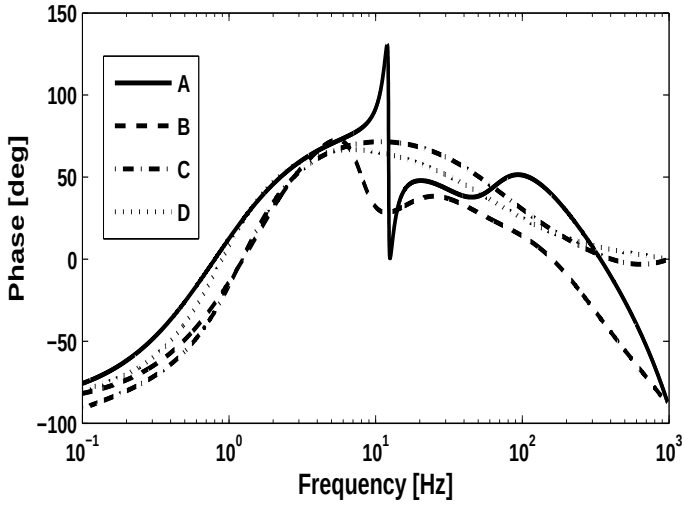


Figure 16: *The argument of the frequency response of the controllers C_2.*

The cost function V_1 and the generalized errors for model set M_1 are shown in Table 1. The table shows that no solution is in general better than all other solutions. For example, solution A gives better performance measured via the quantities e_1 to e_8 but higher values for the other measures.

Table 2 shows a summary of the results. The final performance measure, defined by (19), contains weights which allow some freedom in the interpretation of the results. The

weights used for computing the performance measure in Table 2 were selected in order to reflect good performance with respect to the original industrial problem.

Table 1: *Numerical result for model set* M_1

Solution	A	B	C	D
e_1 [mm]	8.22	8.57	9.75	9.11
e_2 [mm]	2.56	2.43	3.41	3.22
e_3 [mm]	5.39	5.56	5.34	5.28
e_4 [mm]	1.58	1.74	2.12	1.77
e_5 [mm]	7.78	8.22	9.37	8.64
e_6 [mm]	2.82	2.82	4.02	3.68
e_7 [mm]	4.88	5.13	4.20	4.59
e_8 [mm]	1.40	1.56	1.90	1.57
Ts_1 [s]	2.04	2.13	1.79	1.68
Ts_2 [s]	1.25	1.47	1.52	1.05
Ts_3 [s]	1.04	0.77	0.71	0.77
Ts_4 [s]	0.95	0.55	0.69	0.71
T_{NOISE} [Nm]	2.67	1.05	1.85	1.66
T_{MAX} [Nm]	12.1	12.0	11.0	11.3
T_{RMS} [Nm]	1.53	1.52	1.43	1.46
V_1	82.5	80.8	84.8	80.5

Table 2: *Summary of total result*

Solution	A	B	C	D
V_{nom}	64.6	58.8	64.8	62.0
V_1	82.5	80.8	84.8	80.5
V_2	82.6	84.2	84.1	81.6
V	146.0	141.4	148.9	142.2

Inspection of the frequency response of controller D suggests a realization by a PID controller with derivative filter, i.e., the same structure as the default controller of the benchmark problem. In Figure 17 the frequency response of a manually tuned PID controller is compared with controller D. The performance of the PID controller is 143.4. One interesting observation is that this PID controller has complex zeros, and must thus be realized as a parallel PID controller (non-interacting), see, e.g., Åström and Hägglund (2006).

It seems hard to improve the performance further, and reaching a performance index below 140, by using motor position feedback only. One possibility is to use additional or improved sensors on the motor side, e.g., speed sensors or position sensors with decreased measurement noise. Another possibility is to use additional sensors on the link side, e.g., acceleration or position sensors for the links or the tool. In Nordström (2006) a performance index of 105 was reached by using tool acceleration feedback.

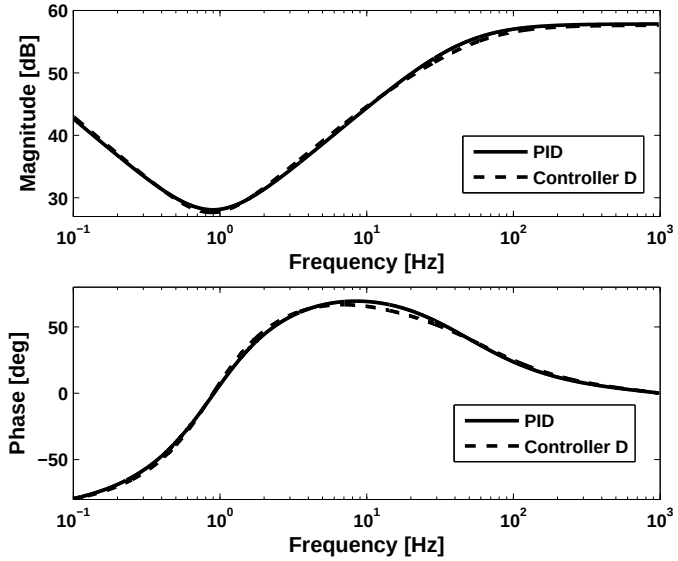


Figure 17: *The frequency response of the original controller D, and its PID realization.*

7 Conclusions and Future Work

A benchmark problem for robust control has been presented. The purpose of the problem is to formulate a problem that is industrially relevant in terms of both the system description and the performance requirements. Four proposed solutions, using different design methods, have been presented. Although the solutions use different approaches the resulting performance from all four solutions end up on the same level. For future work it would be interesting to extend the problem to a multivariable case since a real manipulator has multiple inputs and outputs. Another direction for future work could be to translate the time domain performance measures to the frequency domain, in order to enable the use of frequency domain methods for robust control design. Finally there is a potential to improve the performance by using additional sensors.

8 Acknowledgments

The authors would like to thank those who have contributed with solutions to the benchmark problem. Valuable help and support have also gratefully been received from T. Brogårdh, S. Elfving, S. Hanssen, and A. Isaksson.

References

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society, Research Triangle Park, NC, USA.

Graebe, S. (1994). Robust and adaptive control of an unknown plant: A benchmark of new format. *Automatica*, 30(4):567–575.

Landau, I., Ray, D., Karimi, A., Voda, A., and Franco, A. (1995). A flexible transmission system as a benchmark for robust digital control. *European Journal of Control*, 1(2):77–96.

Leica (2007). Leica geosystems laser trackers. `www.leica-geosystems.com`.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, New Jersey, USA, 2nd edition.

MathWorks (1999). *MATLAB Optimization Toolbox Users Guide*. The MathWorks Inc., Natic, Mass.

MathWorks (2003). *MATLAB and Simulink Users Guide*. The MathWorks Inc., Natic, Massachusetts, USA.

Moberg, S. (2004). Swedish open championships in robot control. `www.robustcontrol.org`.

Moberg, S. and Öhr, J. (2004). Svenskt mästerskap i robotreglering. In *Swedish Control Meeting 2004, Göteborg, Sweden*.

Moberg, S. and Öhr, J. (2005). Robust control of a flexible manipulator arm: A benchmark problem. Prague, Czech Republic. 16th IFAC World Congress.

Nordin, M. and Gutman, P.-O. (1995). Digital QFT design for the benchmark problem. *European Journal of Control*, 1(2):97–103.

Nordström, A. (2006). Identifiering och reglering av industrirobot med hjälp av accelerometer. Master thesis (in swedish), Linköping University.

Öhr, J., Moberg, S., Wernholt, E., Hanssen, S., Pettersson, J., Persson, S., and Sander-Tavallaey, S. (2006). Identification of flexibility parameters of 6-axis industrial manipulator models. In *Proc. ISMA2006 International Conference on Noise and Vibration Engineering*, pages 3305–3314, Leuven, Belgium.

Östring, M., Gunnarsson, S., and Norrlöf, M. (2003). Closed-loop identification of an industrial robot containing flexibilities. *Control Engineering Practice*, 11:291–300.

Rimer, M. and Frederick, D. (1987). Solutions of the Grumman F-14 benchmark control problem. *IEEE Control Systems Magazine*, 7(4):36–40.

Zhu, W.-H., Chen, H.-T., and Zhang, Z.-J. (1992). A variable structure robot control algorithm with an observer. *IEEE Transactions on Robotics and Automation*, 8(4):486–492.

Zhu, W.-H., Yun, M., and Altintas, Y. (2001). A fast tool servo design for precision turning of shafts on conventional CNC lathes. *International Journal of Machine Tools & Manufacture*, 41(7):953–965.

Appendix

A Nominal Parameters

Table 3: *Nominal parameter values*

Parameter	Value	Unit
J_m	$5 \cdot 10^{-3}$	$kg \cdot m^2$
J_{a1}	$2 \cdot 10^{-3}$	$kg \cdot m^2$
J_{a2}	0.02	$kg \cdot m^2$
J_{a3}	0.02	$kg \cdot m^2$
$k_{1,high}$	100	Nm/rad
$k_{1,low}$	16.7	Nm/rad
$k_{1,pos}$	0.064	rad
k_2	110	Nm/rad
k_3	80	Nm/rad
d_1	0.08	$Nm \cdot s/rad$
d_2	0.06	$Nm \cdot s/rad$
d_3	0.08	$Nm \cdot s/rad$
f_m	$6 \cdot 10^{-3}$	$Nm \cdot s/rad$
f_{a1}	$1 \cdot 10^{-3}$	$Nm \cdot s/rad$
f_{a2}	$1 \cdot 10^{-3}$	$Nm \cdot s/rad$
f_{a3}	$1 \cdot 10^{-3}$	$Nm \cdot s/rad$
n	220	
l_1	20	mm
l_2	600	mm
l_3	1530	mm
T_d	$0.5 \cdot 10^{-3}$	s

Licentiate Theses
Division of Automatic Control
Linköping University

P. Andersson: Adaptive Forgetting through Multiple Models and Adaptive Control of Car Dynamics. Thesis No. 15, 1983.
B. Wahlberg: On Model Simplification in System Identification. Thesis No. 47, 1985.
A. Isaksson: Identification of Time Varying Systems and Applications of System Identification to Signal Processing. Thesis No. 75, 1986.
G. Malmberg: A Study of Adaptive Control Missiles. Thesis No. 76, 1986.
S. Gunnarsson: On the Mean Square Error of Transfer Function Estimates with Applications to Control. Thesis No. 90, 1986.
M. Viberg: On the Adaptive Array Problem. Thesis No. 117, 1987.
K. Ståhl: On the Frequency Domain Analysis of Nonlinear Systems. Thesis No. 137, 1988.
A. Skeppstedt: Construction of Composite Models from Large Data-Sets. Thesis No. 149, 1988.
P. A. J. Nagy: MaMiS: A Programming Environment for Numeric/Symbolic Data Processing. Thesis No. 153, 1988.
K. Forsman: Applications of Constructive Algebra to Control Problems. Thesis No. 231, 1990.
I. Klein: Planning for a Class of Sequential Control Problems. Thesis No. 234, 1990.
F. Gustafsson: Optimal Segmentation of Linear Regression Parameters. Thesis No. 246, 1990.
H. Hjalmarsson: On Estimation of Model Quality in System Identification. Thesis No. 251, 1990.
S. Andersson: Sensor Array Processing; Application to Mobile Communication Systems and Dimension Reduction. Thesis No. 255, 1990.
K. Wang Chen: Observability and Invertibility of Nonlinear Systems: A Differential Algebraic Approach. Thesis No. 282, 1991.
J. Sjöberg: Regularization Issues in Neural Network Models of Dynamical Systems. Thesis No. 366, 1993.
P. Pucar: Segmentation of Laser Range Radar Images Using Hidden Markov Field Models. Thesis No. 403, 1993.
H. Fortell: Volterra and Algebraic Approaches to the Zero Dynamics. Thesis No. 438, 1994.
T. McKelvey: On State-Space Models in System Identification. Thesis No. 447, 1994.
T. Andersson: Concepts and Algorithms for Non-Linear System Identifiability. Thesis No. 448, 1994.
P. Lindskog: Algorithms and Tools for System Identification Using Prior Knowledge. Thesis No. 456, 1994.
J. Plantin: Algebraic Methods for Verification and Control of Discrete Event Dynamic Systems. Thesis No. 501, 1995.
J. Gunnarsson: On Modeling of Discrete Event Dynamic Systems, Using Symbolic Algebraic Methods. Thesis No. 502, 1995.
A. Ericsson: Fast Power Control to Counteract Rayleigh Fading in Cellular Radio Systems. Thesis No. 527, 1995.
M. Jirstrand: Algebraic Methods for Modeling and Design in Control. Thesis No. 540, 1996.
K. Edström: Simulation of Mode Switching Systems Using Switched Bond Graphs. Thesis No. 586, 1996.
J. Palmqvist: On Integrity Monitoring of Integrated Navigation Systems. Thesis No. 600, 1997.
A. Stenman: Just-in-Time Models with Applications to Dynamical Systems. Thesis No. 601, 1997.
M. Andersson: Experimental Design and Updating of Finite Element Models. Thesis No. 611, 1997.
U. Forssell: Properties and Usage of Closed-Loop Identification Methods. Thesis No. 641, 1997.

M. Larsson: On Modeling and Diagnosis of Discrete Event Dynamic systems. Thesis No. 648, 1997.
N. Bergman: Bayesian Inference in Terrain Navigation. Thesis No. 649, 1997.
V. Einarsson: On Verification of Switched Systems Using Abstractions. Thesis No. 705, 1998.
J. Blom, F. Gunnarsson: Power Control in Cellular Radio Systems. Thesis No. 706, 1998.
P. Spångéus: Hybrid Control using LP and LMI methods – Some Applications. Thesis No. 724, 1998.
M. Norrlöf: On Analysis and Implementation of Iterative Learning Control. Thesis No. 727, 1998.
A. Hagenblad: Aspects of the Identification of Wiener Models. Thesis No. 793, 1999.
F. Tjärnström: Quality Estimation of Approximate Models. Thesis No. 810, 2000.
C. Carlsson: Vehicle Size and Orientation Estimation Using Geometric Fitting. Thesis No. 840, 2000.
J. Löfberg: Linear Model Predictive Control: Stability and Robustness. Thesis No. 866, 2001.
O. Härkegård: Flight Control Design Using Backstepping. Thesis No. 875, 2001.
J. Elbornsson: Equalization of Distortion in A/D Converters. Thesis No. 883, 2001.
J. Roll: Robust Verification and Identification of Piecewise Affine Systems. Thesis No. 899, 2001.
I. Lind: Regressor Selection in System Identification using ANOVA. Thesis No. 921, 2001.
R. Karlsson: Simulation Based Methods for Target Tracking. Thesis No. 930, 2002.
P.-J. Nordlund: Sequential Monte Carlo Filters and Integrated Navigation. Thesis No. 945, 2002.
M. Östring: Identification, Diagnosis, and Control of a Flexible Robot Arm. Thesis No. 948, 2002.
C. Olsson: Active Engine Vibration Isolation using Feedback Control. Thesis No. 968, 2002.
J. Jansson: Tracking and Decision Making for Automotive Collision Avoidance. Thesis No. 965, 2002.
N. Persson: Event Based Sampling with Application to Spectral Estimation. Thesis No. 981, 2002.
D. Lindgren: Subspace Selection Techniques for Classification Problems. Thesis No. 995, 2002.
E. Geijer Lundin: Uplink Load in CDMA Cellular Systems. Thesis No. 1045, 2003.
M. Enqvist: Some Results on Linear Models of Nonlinear Systems. Thesis No. 1046, 2003.
T. Schön: On Computational Methods for Nonlinear Estimation. Thesis No. 1047, 2003.
F. Gunnarsson: On Modeling and Control of Network Queue Dynamics. Thesis No. 1048, 2003.
S. Björklund: A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems. Thesis No. 1061, 2003.
M. Gerdin: Parameter Estimation in Linear Descriptor Systems. Thesis No. 1085, 2004.
A. Eidehall: An Automotive Lane Guidance System. Thesis No. 1122, 2004.
E. Wernholt: On Multivariable and Nonlinear Identification of Industrial Robots. Thesis No. 1131, 2004.
J. Gillberg: Methods for Frequency Domain Estimation of Continuous-Time Models. Thesis No. 1133, 2004.
G. Hendeby: Fundamental Estimation and Detection Limits in Linear Non-Gaussian Systems. Thesis No. 1199, 2005.
D. Axehill: Applications of Integer Quadratic Programming in Control and Communication. Thesis No. 1218, 2005.
J. Sjöberg: Some Results On Optimal Control for Nonlinear Descriptor Systems. Thesis No. 1227, 2006.
D. Törnqvist: Statistical Fault Detection with Applications to IMU Disturbances. Thesis No. 1258, 2006.
H. Tidefelt: Structural algorithms and perturbations in differential-algebraic equations. Thesis No. 1318, 2007.