Data-driven robot motion control design

Citation for published version (APA):
Kostic, D. (2004). *Data-driven robot motion control design*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Mechanical Engineering]. Technische Universiteit Eindhoven. https://doi.org/10.6100/IR574861

DOI:
10.6100/IR574861

Document status and date:
Published: 01/01/2004

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

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

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

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 12. Jan. 2023

Data-driven Robot Motion Control Design

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Kostić, Dragan

Data-driven robot motion control design / by Dragan Kostić. - Eindhoven :
Technische Universiteit Eindhoven, 2004.
Proefschrift. - ISBN 90-386-2975-3
NUR 978
Subject headings: robotics / robot kinematics / robot dynamics / robot modelling / data-driven modelling / disturbance modelling / identification / friction modelling / friction compensation / Kalman filtering / direct-drive robots / motion control / loop-shaping control design / robust control / H-infinity control / mu-synthesis / flexibility / vibration isolation / disturbance-based control / performance robustness / discrete- time sliding mode control / data-based control / Iterative Learning Control / Iterative Feedback Tuning / Virtual Reference Feedback Tuning / data-based loop-shaping / application / handwriting

Printed by University Press Facilities, Eindhoven, The Netherlands.
Cover design by Paul Verspaget
Typeset by the author with the Microsoft® Office Word 2003.

Copyright © 2004 by D. Kostić
All rights reserved. This publication may not be translated or copied, in whole or in part, or used in connection with any form of information storage and retrieval, electronic adaptation, electronic or mechanical recording, including photocopying, or by any similar or dissimilar methodology now known or developed hereafter, without the permission of the copyright holder.

Data-driven Robot Motion Control Design

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. R.A. van Santen, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 28 april 2004 om 16.00 uur

door

Dragan Kostić

geboren te Priština, Servië en Montenegro

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. M. Steinbuch
en
prof.ir. O.H. Bosgra

Copromotor:
dr.ir. A.G. de Jager

To my parents

Table of contents

1 Introduction **5**

1.1. Robotics today 6
1.2. Experimental case studies with an RRR robot arm 7
1.3. Robot control design process 7
  1.3.1. Kinematic and dynamic modelling 8
  1.3.2. Estimation of model parameters 8
    1.3.2.1. Friction modelling and estimation 9
    1.3.2.2. Estimation of inertial parameters 9
  1.3.3. Experimental model validation 11
  1.3.4. Unmodelled dynamics 11
  1.3.5. Robot motion control 12
  1.3.6. Motion control approaches 13
1.4. An approach to improve control performance 15
  1.4.1. Model-based data-driven control design 16
  1.4.2. Data-based control 17
    1.4.2.1. Tuning of industrial controllers 17
    1.4.2.2. Knowledge-based disturbance suppression 18
    1.4.2.3. Iterative learning control 18
    1.4.2.4. Pulse response based control 19
    1.4.2.5. Simultaneous perturbation stochastic approximation algorithm 20
    1.4.2.6. Unfalsified control concept 21
    1.4.2.7. Iterative feedback tuning 22
    1.4.2.8. The virtual reference feedback tuning 23
    1.4.2.9. Stability issue 23
1.5. Contributions of the thesis 24
1.6. Outline of the thesis 26

2 Problem formulation 29

2.1. Robot motion control 29
  2.1.1. Motion control system 30
  2.1.2. Categorization of the motion control systems 32
2.2. Robot motion control strategies 34
  2.2.1. Decentralized motion control 35
  2.2.2. Nonlinear model-based compensation 36
  2.2.3. Inverse dynamics control with conventional feedback 36
  2.2.4. Computed-torque control 37
2.3. Research questions and problem statement 39
2.4. The solution strategy 43

3 Robot modelling and identification 45

3.1. Kinematic modelling 45
3.2. Dynamic rigid-body modelling 47
3.3. Friction modelling 48
3.4. Estimation of model parameters 50
  3.4.1. Friction parameters 50
  3.4.2. Base parameter set 52
    3.4.2.1. Batch least-squares estimation 52
    3.4.2.2. Online estimation using a batch adaptive control 53
  3.4.3. Excitation trajectory 55
3.5. Online reconstruction of motion coordinates 56
3.6. Model validation 58
3.7. Dynamics not covered with a rigid-body dynamic model with friction 58
3.8. Summary 62

4 The RRR robot arm 63

4.1. Experimental set-up 63
4.2. Robot kinematics and dynamics in closed-form 65
  4.2.1. Kinematics 65
  4.2.2. Dynamics 66
4.3. Evaluating the correctness of the kinematic and dynamic models 66
4.4. Friction modelling and estimation 67
4.5. Estimation of the base parameter set 70
  4.5.1. Batch least-squares estimation of the base parameter set 71
  4.5.2. Online estimation of the base parameter set using the batch adaptive control 74
  4.5.3. Experimental model validation 75
4.6. Parasitic linear dynamics 78
4.7. Summary 82

5 Motion control in the robot joint space 83

5.1. Problem description 83
5.2. Requirements from feedback control designs 84
5.3. Conventional feedback control design 86
5.4. H_∞ feedback control design 89
5.5. μ-synthesis feedback control design 95
5.6. Discrete-time sliding-mode feedback control design 101
5.7. Summary 108

6 Vibration isolation in the robot task space 111

6.1. Problem description 111
6.2. Mathematical formulation 112
6.3. Vibrations at the tip of the RRR robot 113
6.4. Design of a vibration compensator 117
6.5. Experimental evaluation 120
6.6. Summary 122

7 Data-based approaches to motion control 123

7.1. Problem description 123
7.2. Iterative learning control 124
  7.2.1. Learning algorithm 124
  7.2.2. Control design for the RRR robot 126
  7.2.3. Experimental results 127
7.3. Iterative feedback tuning 130
  7.3.1. An iterative feedback tuning algorithm 131
  7.3.2. Control design for the RRR robot via iterative feedback tuning 135
7.4. Data-based loop-shaping control design 145
  7.4.1. Problem formulation 145
  7.4.2. Data-based controller design 147
  7.4.3. Relation with the Virtual reference feedback tuning 150
  7.4.4. Effect of Disturbances 150
  7.4.5. Data-based design for the RRR robot 151
  7.4.6. Experimental results 153
7.5. Summary and conclusions 155

8 Conclusions and recommendations 157

Appendix: RRR robot kinematics and dynamics 163

A.1. Kinematic models in closed-form 163
A.2 Rigid-body dynamics in closed-form 164
A.3 Estimates of friction parameters 166
A.4 Estimates of the base parameter set 167

Bibliography 169

Summary 177

Samenvatting 179

Резиме 181

Acknowledgments 183

Curriculum Vitae 185

Chapter 1

Introduction

Robotics is a large and interdisciplinary field which covers several fundamental aspects, e.g., kinematics, statics and dynamics, identification, trajectory planning, motion control, force control, artificial intelligence, etc. Technological aspects include actuators, sensors, and hardware/software control architectures. A functional combination of the fundamental and technological aspects is needed to obtain a machine which can replace a human being in the execution of a task, regarding the human contribution both in physical activity and in decision making.

This thesis deals with the motion control aspect of robotics, which can be formulated in the form of a problem to be solved: given a desired motion task for the robot, determine suitable control inputs to the robot actuators that would steer the robot motions as close as possible to the desired ones. To be “as close as possible” is typically defined relative to a specific norm or some other measure of quality, and it represents the performance aspect of the motion control problem. This aspect is of particular concern in the thesis.

Traditionally, a solution to the motion control problem requires the knowledge of mathematical models representing the kinematics and dynamics of the robot. Kinematics is modelled by applying the well-known physical principles describing the motion of a body in the multidimensional working space. A dynamic model is typically obtained from the basic physical laws governing the robot dynamics. When the models are available, some strategy is needed to act on them, in order to realize the desired motion task with sufficient quality in respect of accuracy, dynamic response, sensitivity to disturbances, etc. This strategy, which requires sensory capabilities and means for acting and reacting to the sensed variables, is called control design.

If the models are accurate representations of the robot kinematics and dynamics, then the control designs utilizing these models can deliver satisfactory control performance. However, discrepancies between the models and reality can seriously degrade control quality and, if this is the case, one should look for alternative approaches to maintain the desired performance. Information about the model mismatches is contained within sensory information obtained from the controlled robot. By suitable processing of experimental data delivered by the sensors, one can extract information about the modelling errors and adjust the control design accordingly. This thesis investigates possibilities to enhance the existing model-based control strategies with control approaches that closely utilize experimental data. Therefore, the thesis offers a data-driven approach to realize robot control of desired performance.

The purpose of this chapter is to establish a link between model-based control strategies and the data-driven approach. Section 1.1 gives a brief insight into the present status of robotics. Section 1.2 introduces an experimental robotic set-up used in the thesis for practical demonstration of the considered concepts and for their experimental evaluation. Section 1.3 puts forward issues of significance to the robot motion control problem, related with the aspects of modelling, identification, and control design. Data-driven control strategies are addressed in section 1.4. The contributions of the thesis are summarized in section 1.5. Finally, an overview of the thesis is given in section 1.6.

1.1 Robotics today

Robotics is a product of everlasting ambition of humans to create substitutes that would be able to mimic their behaviours in various aspects of interaction with the environment. Boosted by such ambition, robotics has profound cultural roots that date back from ancient mythology which tells stories about titans and giants, across the tale of Frankenstein, up to science fiction literature and movies. The prophetic play of Karel Čapek “Rossum’s Universal Robots” introduced the concept of human-like machines that substitute live beings in subordinate labour duties. The term *robot* is coined from the Slav word *rabota*, which means executive labour. The writer of science fiction literature Isaac Asimov, introduced the term *robotics* to describe the science devoted to the study of robots that represent automation of human appearance but have no feelings.

Today’s robotics covers many aspects of human living, although it is still far from the models anticipated by human imagination and science fiction. Some of the areas where robotics is present are: industrial manufacturing, military, sea and space exploration, medicine, rescue services, entertainment. A robot represents a flexibly programmable manipulation system with multiple degrees of freedom, able to perform different tasks. Regardless of particular tasks, any robot should be capable to perform motions in a multidimensional working space such as to realize the desired position and orientation either of its complete structure (e.g. mobile robots) or of a part of its structure which is called the terminal device or the end-effector. One robot may contain more than one terminal device. Its multiple degrees of freedom must be controlled in a coordinated way in order to achieve a functional movement. The robot’s capacity for movement is provided by a mechanical system which in general consists of the locomotion and manipulation subsystems. The purpose of the locomotion subsystem is to move the robot within the environment, while the manipulation subsystem is used to operate neighbouring objects. The robot’s capacity for perception is provided by a sensory system which can acquire data on the internal status of the mechanical system, as well as on the external status of the environment. The robot’s capacity for connecting perception to movement in a functional way is provided by a control system which can decide the execution of the action in respect of the constraints imposed by the mechanical system and the environment.

Although highly desirable, adaptability, autonomy, dexterity, and versatility are the features that are not yet implemented in today’s robots to the extent that would make these machines universal substitutes for humans. Rather than adaptable to arbitrary working environments, most of the robots are well suited only to strictly arranged artificial surroundings. Robots admit a far limited number of external stimuli that cause functional robot reactions. By increasing robot autonomy and dexterity, one enhances the abilities of the robot to recognize various situations that emerge within its environment and to produce functional motions consistent with these situations. Robots are not versatile enough, since without significant human interference in the instances of path planning, trajectory generation, control retuning, and programming, robots are not yet able to accomplish essentially different duties. Basic reasons for the limited capabilities of robots are the actual mechanical designs (e.g., low number of degrees of freedom and parasitic effects, such as flexibilities and friction) and the challenges of intelligent connection of robot perception with robot action (needs for arrays of sensors, efficient processing of sensory information, large computational power of control systems, effective but safe control strategies, etc.).

Modern science and technology puts significant effort to overcome the observed limitations of robots. Various aspects of robotics are under permanent research and many contributions are appearing. The subject of the research reported in this thesis is motion control of robots. The objective is to develop control strategies that can improve quality in realizing the desired motions. Roughly speaking, better quality means higher position accuracy with low sensitivity to disturbances and parasitic effects. Another emphasis is on the efficiency of robot motions, i.e., the thesis considers quality improvement when the desired motions feature higher velocities. The motives for such considerations can be illustrated by an example from industrial manufacturing: if a robot more accurately accomplishes its task when manufacturing a product, then the steps on the product post processing are reduced; if the same robot operation is accomplished in shorter time (thus with higher velocity), then the throughput is increased; less post processing steps and faster production can drop the price per product. In many cases high position accuracy is directly imposed by the mechanism of the manufacturing process. For instance, robot manipulators for laser welding are supposed to execute highly accurate motions, since the laser welding process is characterized with small diameter of the laser spot and the small heat effected zone.

The next two sections summarize theoretical issues of interest for robot motion control and announce the concepts that will be used later on in the thesis.

1.2 Experimental case-studies with an RRR robot arm

The robot arm presented in Fig. 1.1 is the test bed for the research in robot modelling, identification, and control at the Department of Mechanical Engineering of the Technische Universiteit Eindhoven [176-179]. Its three revolute degrees-of-freedom (RRR kinematics) are implemented as waist, shoulder, and elbow. Similar kinematics is very common in industry: e.g., the first three degrees-of-freedom of the PUMA 560 robot [53] are implemented in a similar fashion. Therefore, the models and identification procedures worked-out for the RRR robot should also be relevant for industrial cases.

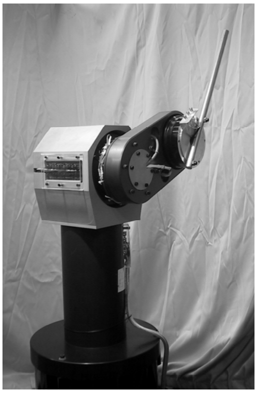


Figure 1.1: The RRR robot arm.

In this thesis, the RRR robot is the subject of all case studies related with robot modelling, identification, and control. Chapter 4 and Appendix A are completely devoted to this robot, presenting detailed models of its kinematics, dynamics, and friction. It is also used for practical demonstration and experimental evaluation of all motion control strategies discussed in Chapters 5, 6, and 7. High complexity of the robot dynamic behaviour ensures rigorous validation of the quality of all motion control results to be presented.

1.3 Robot control design process

There is a lasting challenge to increase the velocity of robot motions, while preserving a prescribed accuracy of robot-tip positioning and/or trajectory tracking. As the performance of modern robots is improved, robustness against uncertainty in the robot dynamics and against disturbances must be maintained. To achieve faster and more accurate motions, we need relevant models of both kinematics and dynamics. A kinematic model provides a relation between joint configuration and robot-tip position and orientation. It also reveals singular robot configurations that must be avoided. A dynamic model determines the control inputs needed to realize a reference motion. It also enables an analysis of how particular dynamic effects influence the overall robot behaviour.

This section discusses all the necessary steps a control engineer must take to enable high-performance model-based control of a robot. These steps are: (*i*) kinematic and dynamic modelling of the robot, (*ii*) obtaining model parameters via direct measurements and/or identification, (*iii*) establishing the correctness of the structure of the models and validating the estimated parameters, (*iv*) deducing to what extent the model covers the real robot dynamics, and, if needed, identification of the dynamics not covered with the derived model, (*v*) controller design, and (*vi*) controller evaluation. Better quality resulting from each of these steps contributes to better performance of robot control.

1.3.1 Kinematic and dynamic modelling

The robotics literature offers various tutorials on kinematic and dynamic modelling of robot manipulators [7,53,100,153,181]. A number of modelling methods are available, meeting various requirements. As for kinematics, the model is a mapping between the robot task space (in general a six-dimensional space of robot-tip coordinates) and the joint space (dimension equal to the number of degrees of freedom of the robot). The mapping from the joint to the task space is called the forward kinematics, while the opposite is the inverse kinematics. These mappings can be represented as recursive or as closed-form algebraic models. A closed-form model defines a functional which explicitly relates the coordinates from the joint space with the coordinates from the task space, and vice versa. A model in closed-from is easier for manipulation, and it admits a straightforward mathematical analysis [53]. As high accuracy of computation can be achieved faster with the closed-form models, they are preferable for real-time control.

Dynamic models relate motions, velocties, and accelerations with applied control inputs (forces/torques). A variety of methods are available for their derivation, and, as with kinematic models, they can be represented in various forms. Algebraic recursive models are computationally less expensive and can be represented compactly. As such, they are traditionally considered as more suitable for real-time control applications [153] than the algebraic models in closed-form. Having in mind advanced computation power of modern computers, the online implementation of closed-form dynamic models for control purposes becomes possible. Such models enable explicit analysis of each dynamic effect, such as inertia, centripetal/Coriolis, gravity, and friction, and a direct evaluation of their contributions to the overall robot dynamics. Direct insight into the model structure makes control design more straightforward, as it facilitates compensation for each particular effect. For example, a closed-form model of gravity can be used in the compensation of gravity loads. Unfortunately, the derivation of compact models in closed-form, in particular inverse kinematic and dynamic models, is usually not easy, even if software for symbolic computation is used. The derivation demands a series of operations, accompanied by permanent simplification of intermediate results. When a model is derived, it is useful to establish its correctness. Comparing it with some recursive representation of the same kinematics or dynamics is straightforward with software packages specialized for robotics, e.g., Robotica [128] and the Robotics Toolbox for MATLAB [41].

1.3.2 Estimation of model parameters

When a model is available, the next step is obtaining its parameters, so the model corresponds to the real system. Kinematic parameters, such as link lengths, twist angles and link offsets, are usually known with better accuracy than inertial ones: link masses, locations of their centers of mass, and link moments of inertia. The kinematic parameters are either directly provided by the robot manufacturer or can be achieved by direct measurements on the robot. Adjustment of these parameters can be done via kinematic calibration [7]. As for the inertial parameters, they are rarely provided by the manufacturer, but are rather reconstructed by the robot users. Friction modelling and estimation are also important issues, as friction may cause control problems, e.g., static errors or stick-slip, possibly leading to limit cycles.

The literature on estimation of inertial and friction parameters is extensive [10-12,32,35,44,57,58,72,73,89,96-98,100,101,105,112,121,133,134,145,169,170]. When estimating inertial parameters, for each robot link one should determine one mass, three Cartesian coordinates of the center of mass and six elements of the inertia tensor [7,53]. In this general case, for a robot with n links, there exist $n \times 10$ unknowns. In practice, instead of estimating each parameter, it is more common to identify algebraic combinations that constitute a so-called base parameter set (BPS). Elements of this set can be uniquely identified in experiments. There exists a representation of the dynamic model linear in the elements of the BPS [7,12,100,153]. The regression matrix depending on the robot trajectory, i.e., the motions, velocities, and accelerations, premultiplies the BPS in vector form to equate the control inputs. The total energy is also linear in the BPS [12,100,101]. The linearity property facilitates the estimation of the BPS.

1.3.2.1 Friction modelling and estimation

Friction is a complex interaction between the contact surfaces of the interacting objects, which can be represented with a number of models [10,11,35,44,72,73,89,105,134,145,169]. These models are used for theoretical analysis of friction phenomena and for control purposes. Two friction regimes are observed – presliding and sliding [10,11,134]. Presliding occurs if the interacting objects are not moving relative to each other, while sliding arises after the relative motion is established. The level of friction prior to the sliding regime is called the static friction, or the stiction. It is determined by balancing the static forces acting on the interacting objects. Sophisticated friction models, e.g. the LuGre model [13,35,105,134,145,169], characterize the presliding regime by stiffness and damping of dynamic interaction between the contact surfaces. At the beginning of the sliding regime, the friction force reduces as the velocity is increasing – this is known as the Stribeck effect. Beyond this stage, the friction force increases as the velocity is getting higher.

The basic friction model, covering just Coulomb and viscous effects, admits a linear parameterization, which enables simultaneous estimation of its parameters with the BPS elements [32,57,100,101,170]. Contrary to the basic one, more involved friction models are typically not linear in some parameters, and their parameters should be determined separately from the BPS. The three-sigmoidal-function model considered in [72] is an example of such a friction model. If a control designer adopts a model which is not linear in all parameters, then he/she has the following options: (*i*) to identify friction parameters separately from the BPS and use the fixed estimates for friction compensation [10,11,72,73,89,96,97,105,134,169] (*ii*) to estimate friction parameters on-line, simultaneously with using the model for friction compensation [35,44,145], (*iii*) to give up of a structured friction model and adopt some non-model-based friction compensation strategy, including the direct sensing of friction using force sensors [145]. The first option is often used in robot control, and it will be considered in this thesis.

1.3.2.2 Estimation of inertial parameters

Once a strategy for friction modelling, estimation, or measurement is decided, one may proceed with estimation of the BPS. For that purpose an identification experiment has to be designed. During the experiment, the robot performs specific motions, implemented using a decentralized control approach. This means use of decoupled PID (proportional, integral, derivative) controllers to generate control inputs. The signals needed for the parameter estimation are the control inputs and motion coordinates, i.e., displacements, velocities and, if necessary, accelerations. These data is obtained by direct measurements and/or are reconstructed using filters or observers. If the identification experiment is carried out just to collect data, and the elements of the BPS are determined afterwards by processing the data adequately, then we speak about batch estimation. If data is collected and processed to compute the BPS elements as the identification experiment is in progress, then the estimation is online.

In general, we can distinguish between the least squares-like [7,32,57,100,101,170], the maximum-likelihood [133], and the adaptive techniques for estimation of the BPS elements [7,157,158]. If the inertial and friction parameters are jointly estimated, then one should consider a friction model which admits a linear representation in its parameters. In this thesis, two estimation algorithms will be considered. The first one is a batch least-squares algorithm [96,97], which is facilitated by an efficient estimation of the motion coordinates via Kalman filtering. The second algorithm is aimed for online estimation [30], and it is based on a batch-adaptive technique for motion control of a robot manipulator with unknown dynamics, in repetitive motion tasks. The control technique was originally proposed in [8], and its latest version has appeared in [71]. The latest version features improved robustness against disturbances and parasitic effects, as well as faster convergence characteristics. Being computationally cheap, the batch adaptive control technique admits a relatively simple online implementation.

No matter if performed batch or online, the reliability of the parameter estimation depends on the trajectory executed by the robot during the identification experiments. This so-called excitation trajectory should be carefully designed, and this is done by optimizing some property of the information matrix. Such a matrix is formed by vertically stacking the regression matrices that correspond to different points of the excitation trajectory [12,32,57,58,100,101,170]. The condition number of the information matrix, its largest singular value, its Frobenius condition number, the determinant of the weighted product between the information matrix and its transpose, are some choices for the performance criteria to be optimized. No matter which performance criterion is adopted, the resulting optimization problem is most likely non-convex, which means that the global optimal excitation trajectory cannot be obtained easily. Different initial conditions may lead to different excitation trajectories, all corresponding to local minima of the optimization criterion. As argued in [12], “engineering judgment, patience, and good luck are required to find a good trajectory.”

Computing the excitation trajectory using calculus of variations is suggested in [12]. The optimization problem was to minimize the condition number of the Persistent Excitation matrix, created using the information matrix. The objective was to increase the immunity of estimation in the presence of measurement noise. The optimization algorithm generated a complete excitation trajectory, point by point. Such trajectory representation was referred to as “the general one,” which means that the space of trajectories to be searched is not restricted. This, in general, may contribute to an inefficient optimization process. The same problem is inherent to the method of generating excitation trajectories presented in [58]. Once again, the trajectory is computed point by point, narrowing the search space only by imposing constraints on permissible ranges of robot motions, velocities, and accelerations. Furthermore, both approaches do not explicitly specify the frequency content of the computed trajectories. This is an additional drawback, as control of the frequency content is an important precaution for avoiding frequency components that may excite the robot flexibility. The intentional parameterization of an excitation trajectory is a way to impose the desired frequency content, and to narrow the search space for the purpose of faster optimization. For example, in [32] and [170] the excitation trajectories are postulated as finite Fourier series with predefined fundamental frequency and with coefficients to be determined by optimizing some performance criterion. The resulting trajectories are periodic, which enables continuous repetition of the identification experiment and averaging of measured data to improve signal-to-noise ratio. Additional merits are that, by design, one may enforce the desired frequency resolution and the frequency content of the trajectories.

The excitation trajectories are realized on the robotic systems using PID feedback controllers. The resulting control inputs and the motion coordinates are measured and recorded for estimation of the BPS elements. Theory offers several estimation strategies, ranging from the least-squares to maximum-likelihood algorithms. The success of sophisticated algorithms strongly depends on an accurate knowledge of statistical properties of the signal noise and of other disturbances affecting the signals in use. Some algorithms were experimentally compared in [133]. The most involved estimation algorithm based on the maximum-likelihood method indeed showed the best performance, but the achieved improvements over the simpler algorithms do not seem so striking. An alternative for advanced estimation techniques is to improve the quality of the signals measured on the robotic system, by reducing the impact of disturbances [19-21,26,59]. A comparative study of several digital techniques for velocity estimation from position measurements is given in [26]. The effectiveness of these techniques for estimation of acceleration was not argued, so their usefulness in obtaining data of sufficient quality for the BPS estimation cannot be immediately anticipated. On the other hand, one may use Kalman filtering [59], as formulated in [19-21], to optimally reconstruct all motion coordinates including acceleration, in the presence of measurement noise and uncertainty in system dynamics. If the reconstructed motion coordinates are used, then even the simplest least-squares estimation of the BPS elements gives satisfactory results.

1.3.3 Experimental model validation

After the estimation is finished, experimental validation of the model has to be done. A rigorous two-step validation strategy was established in [7]. The objective of the first step is to test how accurately the model reproduces outputs of PID controllers. If this accuracy is satisfactory, one proceeds to the second validation step: verification of the model's utility for control purposes.

In a number of publications on robot modelling and identification, just the first step of the validation strategy is considered. For instance, in [170], the validation is performed along a trajectory interpolating randomly chosen points in the robot workspace. The interpolation takes care of constraints on joint motions, velocities, and accelerations. It is expected that qualitative agreement of torques rendered by the model and measured torques under PID control, along such a random trajectory, is a sufficient guarantee of model correctness. A similar qualitative validation can be found in [32] and [133]. In [32], validation is performed along a smooth trajectory defined in the joint space. Choice of a sine wave as the validation trajectory, as in [133], is probably the simplest validation one might consider, as it tests the model for just one harmonic component. Satisfactory results for a single harmonic do not necessarily imply the same for other frequency components.

In [7], experimental validation was carried-out using a fifth order polynomial trajectory, defined in joint space. The trajectory was chosen to ensure fast joint motions, requiring full authority of all actuators. First, the considered dynamic model was validated qualitatively, i.e., by comparing its output with measured torques under PD control. Then, the model was used to generate control inputs in addition to the PD feedback. The latter test verified that the model-based control realizes smaller tracking errors, when compared with the errors achieved with the simply PD control.

In this thesis, the experimental model validation will be performed similarly as in [7], but along a trajectory with a more involved shape. The validation trajectory will be a sequence of letters with a prescribed velocity profile, defined in the task space. Such a trajectory can be considered sufficiently general to cover motions that a robot normally performs in practice. For example, industrial robots typically realize movements along concatenated straight-line and curved paths in their task spaces. Writing of letters is indeed a concatenation of straight-line and curved primitives. Additionally, the handwriting may impose significant demands on the robot dynamics due to varying velocity levels and discontinuous accelerations. The handwriting in robotics was presented in [80] and [81]. Surveys of related research results are available in [143] and [144], also presenting the sequence of letters that is used for model validation in the thesis.

1.3.4 Unmodelled dynamics

Dynamic models are used in model-based control algorithms. These models contribute to the performance of the control algorithms to the extent they match the real robot dynamics. In practice, the model, especially a rigid-body one derived by physical reasoning (first principles modelling), can cover the real dynamics only within the low-frequency range. Actual dynamics include flexible effects at higher frequencies that are caused by elasticity in the joints and by distributed link flexibility [45]. If one requires high control performance, the influence of flexible modes must be taken into account during the control design. Flexible dynamics can be captured via data-driven (experimental) modelling, which utilizes input/output data observed from the considered system. System identification theory [113,140] is used to translate experimental data into models (data-driven models).

The quality of robot motion control can be improved by using both physical and data-driven dynamic models in the control design. The idea of utilizing data-driven models in the control design is not new as long as control of (linear) motion systems is concerned. Motion control systems are used throughout industry (e.g., steering of machine tools) and in various electrical appliances (e.g., copiers, printers, scanners, optical disc drives) to generate precise mechanical moves. Experience with these systems teaches us that ultimate limits of performance and robustness of motion control can be achieved only if all peculiarities (flexibilities, disturbances) of the motion system are taken into account during the control design [166]. Similar reasoning will be applied in this thesis for robot motion control.

1.3.5 Robot motion control

The accurate execution of a reference robot trajectory is the objective of the robot motion problem. This thesis is concerned with realization of the reference trajectory of the robot end-effector, while the motion control of mobile robots is not considered. The reference trajectory is executed in the robot task space which dimension is not higher than six, since unique positioning and orientation of the end-effector require at most three Cartesian and three angular coordinates. A task desired from the robot is first analyzed and divided into a sequence of simpler subtasks that are represented in terms of motion primitives (straight lines or curves). Each primitive is assigned a suitable velocity profile to establish adequate coordination of the robot operation with time. These motion primitives are translated from the task space into the joint reference trajectory via the inverse kinematics mapping.

Since the robot is a highly nonlinear system, robot motion control can be facilitated by suitable linearization of the robot dynamics. Linearization can be achieved by appropriate mechanical design or by adequate control action. Use of transmission elements of high reduction ratios in the robot joints can have linearizing effect to the robot dynamics. These elements are placed between the joint actuators and the links, and their purpose is to increase driving effect of the actuators to the corresponding links. The linearization via control action usually means implementation of the robot dynamic model in the control law to dynamically compensate for nonlinearities. Control inputs that are applied to the joint actuators are products of the control law.

The joint servo control systems realize the reference joint trajectories by applying the control inputs to the actuators, while obtaining sensory information about the actual joint displacements. These servo systems consist of appropriate hardware and software resources, and they basically close the loops around the robot to transform it into a motion system that can actively move and manipulate with the environment. The joint servo systems are responsible for realizing accurate tracking of the given references, as well as for good attenuation of disturbances and parasitic effects. Intuitively, one would expect that better quality of motion control in the joint space contributes to higher performance in the task space. This is, though, not necessarily the case, as explained in the next paragraph.

There are a number of criteria to evaluate the performance of motion control. A few will be listed here. Accuracy of tracking a motion reference is the usual performance criterion. Intuitively, it is expected that if the joint reference is realized with a higher accuracy, then the task motion should be accurate as well. However, the relation between accuracies in the joint and task spaces is not as straightforward as the previous statement implies. The kinematics mappings between the task and joint spaces are intrinsically nonlinear, and therefore the tracking accuracies in these spaces are nonlinearly related. Aside from joint configuration, flexibilities and disturbances may influence position of the robot end-effector. In the presence of these effects, even highly accurate execution of the joint reference may give unsatisfactory accuracy in the task space. Therefore, flexible and disturbance effects must be taken into account in the design of motion controllers.

The ability to accurately realize motions of versatile velocity levels is another performance criterion. The control bandwidth can be used to evaluate such ability. A higher bandwidth implies the potential of a manipulator control system to realize faster motions. Additionally, higher bandwidth means the capability to compensate for dynamic effects and disturbances of broader harmonic content. Although often addressed in manipulator motion control, there is no unique definition of the control bandwidth in robotics. In [158], the control bandwidth is formulated as a range of frequencies contained in the control input. If the manipulator dynamics are dominantly linear, and linear feedback control strategies are used, then one may resort to conventional definitions of the control bandwidth [52]. The linear dynamics can be realized by decoupling the robot joints either using high-gear transmission mechanisms between the joint actuators and the robot links, or by some nonlinear control compensation. The control bandwidth of a given joint servo control loop can be formulated as the cross-over frequency, i.e., the first zero-crossing of the open-loop gain of the servo-loop. The open-loop gain is the product of all transfer functions in the loop. Because of linearity, a higher bandwidth implies a higher open-loop gain, which may contribute to better tracking accuracy.

Imposing a high feedback gain for the sake of better tracking accuracy is unacceptable if it also excites parasitic dynamics (flexibilities) and/or amplifies disturbances (e.g. measurement noise). Therefore, a level of rejection of the parasitics and disturbances can be adopted as additional control performance criterion.

Stability and robustness issues are naturally important in robot motion control. Stabilisation of robot motions is achieved with a feedback control component. Motion control is robustly stable if despite uncertainty in the robot dynamics and disturbances, the stability is preserved. There are two kinds of uncertainties. The first ones are parametric, and they arise if physical values of robot inertial and/or friction parameters are not known exactly. The second ones are unmodelled dynamic effects, e.g., flexibilities and friction. As examples of disturbances, one may think of cogging forces and quantization noise. The former are typical for direct-drive manipulators [13], while the latter arises if incremental encoders are used as position sensors. Better stability margins enhance robustness of motion control, since they guarantee the safe operation despite uncertainties and disturbances. With linear decoupled joint dynamics and linear feedback control solutions, standard gain and phase margins [52] can be considered as control performance criteria.

1.3.6 Motion control approaches

Increasing demands on the performance and robustness of manipulator motion control has led to the development of versatile control approaches. Robust control strategies are introduced to stabilize manipulator motions under uncertainty and disturbance conditions. More advanced strategies should ensure that the desired performance of motion control is preserved despite uncertainties and disturbances. This property is refereed to as robust performance.

Decentralized PID (proportional, integral, derivative) and its variants PD and PI are conventional feedback control solutions. The majority of industrial robot manipulators are equipped with high-gear transmission mechanisms, and they can be described by linear and decoupled rigid-body dynamics. Linear conventional controllers are suitable for such dynamics. They are appealing because of their efficiency in tuning and low computational costs. However, if high-quality motion control performance is needed, conventional controllers may lead to unsatisfactory results. High-quality performance can be defined as accurate tracking the reference trajectory, robust against disturbances and parasitic dynamics. Use of conventional controllers means making serious tradeoffs between feasible static accuracy, system stability, and rejection of disturbances and damping of parasitic dynamics and high frequencies. For example, smaller proportional action gives larger gain and phase margins for system stability, sacrificing static accuracy. A higher gain improves static accuracy, but also amplifies quantization noise and other high-frequency disturbances. Inclusion of integral action improves static accuracy, but reduces stability margins. It is shown in [39] that these tradeoffs become critically conflicting with a digital implementation of the control law, as the sampling rate decreases. The simplicity of conventional controllers can be too restrictive to provide compensation of each dynamic effect encountered in the robotic system: dynamic couplings between the joints, friction, backlash, flexibility, and time-delay. Usually, conventional controllers can handle only a limited number of these effects, together with other control objectives, e.g., prescribed control bandwidth and reduction of the position error. As non-compensated effects may strongly influence motion performance, more sophisticated control strategies are needed. The situation becomes even more critical if no transmission elements are present between joint actuators and links, which is typical for direct-drive robots [13]. Then, the nonlinear dynamic couplings directly apply upon each joint actuator and their compensation requires use of more advanced control schemes.

Advanced manipulator control methods are capable of simultaneously addressing several control objectives: stability robustness, disturbance rejection, controlled transient behaviour, optimal performance, etc. Of course, it is not possible to achieve arbitrary quality with each control objective independently, since they are mutually dependent and are often conflicting. A detailed survey of advanced control methods is available in [150]. Nonlinear representatives, such as adaptive [137] and sliding-mode [55,84,158,175] methods, can improve system performance in the presence of uncertainty in the robot dynamics and external disturbances (e.g., variable load). The non-linearity of these methods, however, does not facilitate a quantitative prediction of system performance for a given robustness level. This is a limiting factor for their widespread application in practice, where it is often very important to know in advance worst-case motion accuracy for a given harmonic content of the reference trajectories.

Stability robustness, disturbance rejection, and controlled transient response can be jointly and directly imposed using feedback schemes based on H_∞ control theory [15,49,103,185]. These schemes enable quantitative prediction of motion performance, given bounds on modelling uncertainty and disturbances. Moreover, for available knowledge on the system dynamics, parasitic effects, and disturbances, motion performance can be optimized. These are the reasons that make H_∞ controllers appealing solutions for practical problems and motivate their application in robotic systems. A number of contributions to H_∞ robot control are summarized next.

Paper [36] presents a frequency domain H_∞ control design for an industrial robot, decoupled by means of gearboxes with high reduction ratios. The feedback controllers have been designed assuming just a low-order rigid dynamics in the robot joints. The effectiveness of the design has been confirmed in simulation, only. Other frequency domain H_∞ designs are available in [159,167]. In these contributions, nonlinear robot dynamics have been decoupled using a computed-torque control method [7,129,153,158] and feedback linearization [7,129,153,158], respectively. Similar with [36], only simulation results have been provided. Major control objective of all three designs was robust stability in the presence of disturbances and uncertainties in the plant dynamics, while the performance was not optimized.

Experimental effects of H_∞ robot motion controllers designed in the frequency domain have been discussed in [127,132,149]. A conservative formulation of the plant model has been employed in [127]. The plant representation was obtained by neglecting nonlinear terms of a robot rigid-body dynamic model. Bounds on the neglected terms were treated as modelling uncertainty. H_∞ feedback controllers were designed to ensure stability of the plant with uncertainty bounds. In [132], the controlled system was a robotic manipulator equipped with gearboxes of high reduction ratios. The H_∞ feedback was designed for a simple and hence conservative plant modelled assuming only linear dynamics of DC servo actuators. Coulomb friction and the remaining dynamic couplings between the robot joints were treated as disturbances, together with measurement noise. The purpose of the H_∞ controller was stability of the robot motions and disturbance rejection. Experimental results, presented in both references [127] and [132], showed that the H_∞ controllers provide better tracking accuracy than feasible with conventional feedback controllers. However, the performance achieved with these robust feedbacks was certainly not maximized, having in mind conservative plant modelling. The paper [149] was an attempt to maximize tracking performance of a robot in a specified motion task, using an H_∞ feedback control design. The nominal plant was determined by linearizing a comprehensive model of the robot rigid-body dynamics with DC actuators. A set of perturbed plants comprised variations between the comprehensive model and the nominal plant. Among all linear H_∞ controllers that stabilize the set of perturbed plants, a nonlinear optimization determines one that minimizes the squared integrated worst-case tracking error for the specified motion task. The resulting controller was successfully verified in experiments on an industrial robot with gearboxes of high reduction ratios. However, this control design was constrained to rigid-body dynamics, only.

References [36,127,132,149,159,167] are essentially linear approaches to H_∞ feedback control design. Nonlinear H_∞ control designs for rigid robot dynamics have been studied in[16,37,38,51,171]. These studies formulate conditions under which the disturbance rejection problem and internal stability of rigid robot manipulators can be analytically solved in a nonlinear setting. Among them, only [37] had the additional requirement of maximizing the tracking accuracy in a prescribed motion task. All these studies are purely theoretical and are verified only in simulations.

Some control approaches are dedicated to the problems of motion control of robot manipulators with significant flexibility. Their objective is to counteract elastic effects, which is especially important if lightweight materials are used for robot constructions. The purpose of lightweight materials is to enable faster robot movements with the same actuators applied. As the stiffness of lightweight constructions is in general not sufficient to avoid their elastic bending, quick movements may cause undesirable vibrations of the robot end-effector. Such vibrations deteriorate the accuracy of trajectory tracking in the robot task space.

There are two causes of robot flexibility: lumped elasticity in the transmission of torque [48,142,147] and distributed link flexibility [24,45,47,102,168]. The former one is characteristic to elastic transmission elements, e.g., harmonic drives, belts, or long shafts. The latter one is a result of using lightweight materials and constructions to reduce mass and inertia of the links. Lightweight materials and construction can increase the link compliance, which may lead to structural vibrations of significant amplitude.

A part of the research on robot flexibility is devoted to integrating the elastic effects into models of robot dynamics [24,45,85,102,142,147,168]. For the sake of modelling, it is common to treat robot links as slender beams and to employ some of the standardized elastic models, e.g. Bernoulli-Euler, Rayleigh, or Timoshenko beams. Dynamic models with flexible effects included can be used for simulation studies and in control designs.

Literature offers various designs of model-based controllers that counteract elastic effects in robotics [45,47,48,63,102,114,115,120,184]. The models are typically implemented as feeedforward compensators of nonlinear and flexible dynamics [45]. Stability is ensured by means of linear or nonlinear feedback controllers, that often require feedback from the whole state of the robot. This implies the presence of additional sensors for the deformation variables, besides position sensors (encoders) mounted on the joint motors. The deformation variables represent link/joint deflections, and they are measured using strain gauges, accelerometers, visual systems, etc. These additional sensors can be avoided by including a state observer in the control law, at the expense of a higher complexity of the control design and profound dependency on the model quality.

It is apparent that simulation studies on vibration compensation dominate the experimental ones in the robotics literature. In the experimental studies, the most frequently encountered are single [63,115] or two-link planar flexible robot arms [102,184]. There are only a few references treating more complex spatial robot kinematics, e.g. [114,120].

In this thesis, motion control techniques are treated in Chapters 5, 6, and 7. These techniques departure from the surveyed ones in the sense that they also account for experimental characteristics of a robot that are not covered with the robot dynamic model derived via physical reasoning. In Chapters 5 and 6, these experimental characteristics are captured via data-driven modelling, while in Chapter 7, the modelling step is even omitted and the control designs are based on mere input/output data observed from the robot. Incorporating the robot experimental characteristics into the control design aims at improving the quality of motion control, and it is the subject of the following discussion.

1.4 An approach to improve control performance

Robotics research has led to a good understanding of the basic issues in robot control. The modelling of robots, especially kinematics and rigid-body dynamics, is well understood. The majority of the robot motion control methods are based on the assumption that a robot consists of rigid bodies without additional (parasitic) dynamics. The additional dynamics are due to robot flexibilities and uncompensated friction. Apart from parasitics, disturbances, e.g., measurement noise and cogging force also affect robot operation and influence quality of motion control. When applied on a real robot which almost inevitably incorporates parasitic and disturbance effects, a control method neglecting these effects can achieve only limited position accuracy and control bandwidth. A possible way to alleviate the conservatism of control methods generally accepted in robotics is to combine them with control strategies that explicitly consider the experimental characteristics of the plant to be controlled. This section distinguishes adequate control strategies that can be ported to the robotics field.

1.4.1 Model-based data-driven control design

The problem of maintaining a desired control performance, in the presence of parasitic dynamics and disturbances, has been attracting researchers from the Systems and Control area for many years. This problem is especially important in the field of motion control, where the control performance requirements are often limited only by physical properties of the controlled plant. The next part describes a strategy originating from the motion control field, which enables one to realize optimal control performance when confronted with parasitic effects and disturbances.

This strategy is explained by an example of motion control of optical and magnetic disk drive systems (CD and DVD players, hard disks). Industry manufacturing these drives is regularly confronted with the demand from the market for higher data readings. This demand is translated into the requirement for higher bandwidths of the feedback control loops closed around a drive. Arbitrary high bandwidths are not possible, since they are limited by flexible modes of the disk drive mechanism. A controller implemented in the feedback loop is usually tuned around the maximum crossover frequency admissible by the actual dynamics of the drive in the loop. This tuning should provide sufficient stability and performance robustness against flexible effects, manufacturing tolerances, and typical disturbances encountered in the system. Information about the drive's dynamics and tolerances is captured via data-driven modelling (system identification). Measured input/output data also reveal disturbance effects. The system identification captures nominal characteristics of the drive's dynamics and represents them by a parametric model of sufficient order. Differences between the model of the nominal dynamics and the measurements, that are caused by modelling errors and plant-to-plant variations from manufacturing, are also captured by a parametric model (uncertainty modelling). The obtained nominal and uncertainty models are explicitly used in the controller design which implements methods from the robust control theory, such as H_∞ and μ–synthesis control. These methods provide formalism for controller tuning to realize the desired control performance. Such tunings are the optimal ones for the given models of the nominal plant dynamics and uncertainty.

The described strategy is clearly focused on experimental characteristics of the controlled plant, and it is thoroughly used in the form of loop-shaping design in industrial motion systems to improve quality of motion control. However, this strategy is hardly encountered in the robotics area, although it seems a solution for the conservatism of the motion control methods applied in robotics. Therefore, this thesis will propose a functional combination of the considered strategy and nonlinear robot control as a way to improve the performance of motion control. In particular, Chapter 5 and 6 will present several control designs all featuring the following steps: (*i*) nominal dynamic compensation of robot nonlinearities using a model of a robot rigid-body dynamics with friction, (*ii*) data-driven modelling of the robot dynamics that are not covered with the nominal model-based compensation of nonlinearity, and (*iii*) design of feedback controllers using the data-driven models to meet performance and robustness specifications. The obtained feedback controllers will be applied in robot motion control simultaneously with the nominal model-based compensation. The described control strategy will be effective in compensating for more versatile effects present in the robot than the nominal control based on the rigid-body dynamics with friction. Chapter 5 will present feedback control designs performed in the joint space of a robot. Chapter 6 will deal with attenuation of vibrations at the tip of the robot. These vibrations are caused by robot flexibility, and they will be attenuated using a compensator designed in the robot task space. All control designs in Chapters 5 and 6 will be experimentally evaluated on the RRR robot.

Use of data-driven modelling is one possibility to alleviate conservatism of the motion control methods generally accepted in robotics. However, its effectiveness is still restricted by the quality of the experimentally obtained models. Systems and Control theory also offers solutions to improve control performance while avoiding the plant modelling step. These solutions are summarized next.

1.4.2 Data-based control

Although emerging in many aspects from the early days of control applications, data-based (DB) control still remains a discipline which framework and scope cannot be uniquely identified within the control area. As observed in [33], the works of Ziegler and Nichols on tuning industrial regulators based on empirical data observed from the controlled plant [186], traced the path for this long present but yet immature control field.

Coarsely speaking, in DB control designs, the need for (non) parametric plant models is omitted, since the control action is only based on data observed from the controlled system. DB designs found application in various fields (e.g., mechatronics, gas and oil industries, etc.), showing satisfactory control performance. However, the stability theory covering DB control is still not well developed, which is the formidable weakness impeding wider acceptation of DB control. On the other hand, the promising trend is increasing number of publications dealing with the stability issues in DB control.

What drives someone's interest in DB control? Seemingly, there is no unique answer to this question. For example, some researchers are motivated by the observation that creating the control input directly from data requires less parameters describing the plant dynamics than generating the same input from the equivalent model-based controller. Other researchers deliberatively try to avoid the plant modelling step in the control design. Sometimes they are constrained on data, especially if their plants are of such complexity that makes the modelling too costly and time consuming. Some contributors emphasize that on the data level a designer can directly obtain controllers of desired complexity, thus avoiding an eventual controller model reduction step. Other contributors expect that the empirical data are informative enough to closely represent current plant dynamics and disturbance effects, so control actions adapted with the observed dynamics/disturbances should result in improved control performance. Obviously, the reasons for DB control are not coherent. The following survey should illuminate the particular motives and application areas of the most prominent DB control techniques found in the literature. Some of them will be found suitable for application in robot motion control, and the particular reasons for such application will be illuminated.

1.4.2.1 Tuning of industrial controllers

Original rules for tuning conventional industrial controllers, i.e., PID (proportional, integral, derivative) and the variants PD and PI, are due to Ziegler and Nichols [186]. These rules can be applied without any prior knowledge about the plant dynamics, but only by evaluating some properties of time-domain signals observed from the plant. Typically, one analyzes oscillations in the plant's response that are induced by a proportional feedback controller tuned to force the closed-loop system to a small stability margin. Upon the properties of these oscillations, the tuning rules determine the controller parameters that should be stabilizing for the plant. Such a tuning strategy can be considered as data-based, and because of simplicity in implementation, it has a broad scope of applications in industry.

Observing that the tuning rules of Ziegler and Nichols may have limited effects on control performance motivated the derivation of more sophisticated data-based tuning techniques, e.g., the internal model control method, the integral square error method, etc. These are overviewed in [14] and [122]. When implementing these techniques, a control designer usually has not much freedom in the definition of the control specifications, but is rather restricted to already “built-in” performance criteria. This restriction is overcome in more advanced data-based control strategies, overviewed next.

1.4.2.2 Knowledge-based disturbance suppression

A rather empirical strategy for suppression of deterministic disturbances encountered in control systems is sketched in [154]. In this strategy, the experience with disturbance effects is formalized by developing a database of observed disturbance properties. Typical features to be recorded are: characteristic waveforms of disturbance components in the measured signals, spectral characteristics of disturbances, statistical properties of disturbance occurrences, etc. The knowledge stored in the database can be used for detection and diagnosis of disturbances from the measured signals. After diagnosis, appropriate control action can be activated to suppress the disturbance effects.

The described strategy is applied in [183] to compensate for undesired vibrations arising in precision machines. A feedforward control signal is computed off-line from the measured position error which is caused by vibrations. Upon detection of vibrations during online control, the feedforward in control input is applied to counteract undesired effects. In [173], an online strategy for suppressing deterministic disturbances is suggested. A feedforward disturbance compensation signal is iteratively updated online, each time after a disturbance of the known waveform is detected. The feedforward update law is based on the theory of iterative learning control, which is considered next.

1.4.2.3 Iterative learning control

As already mentioned, control designs utilizing data-driven models are indeed capable of improving the performance of motion control, but at the expense of high controller order. Apart from controller complexity, the robot joints are usually very stiff because of high feedback gain within the closed-loop bandwidth. If the motion task is a repetitive one, which is common for industrial robots, the need for a complex feedback controller of high gain can be circumvented by means of iterative learning control (ILC). The combination of weakly tuned model-based feedback and an ILC component can provide motion control of high performance.

Generally speaking, ILC is a technique for improving the transient response and tracking performance of systems that execute repetitive operations. This technique is motivated by the observation that if the system controller is fixed and if the system’s working conditions are the same during each repetition of the reference operation (trial), then much of the errors in the output response are repeated in any trial. These errors can be recorded during each system operation and then used to compute modifications to the control input that will be applied to the system during the next trial. These modifications are defined in the form of an update law, which should ensure that the new input signal produces smaller errors than the previous one. Iterations are stopped if no more performance improvement can be achieved, which means that the learning process has been converged and the optimal input signal is derived.

When compared with adaptive control schemes [137,157,158], ILC is limited by the repetitive nature of the desired operation. Moreover, no explicit parameter identification is performed and the learning process should be restarted when tackling a different task. On the other hand, the learning algorithm can be made robust against parasitic dynamics, as opposite to standard adaptation schemes, while the need for iterative operation is not restrictive in many situations. Another relevant advantage of ILC is simplicity of online implementation, since ILC design is typically performed off-line and only the resulting control signal is applied in the experiment.

The development of ILC stems originally from robotics where repetitive motions show up naturally in many applications. Examples of contributions where ILC is applied in robotics are [7,9,23,46,64,67,71,82,83,106,124,126,130,131,152]. In the rapidly increasing literature, a number of ILC algorithms of different complexity have been suggested.

Depending on whether the mapping from the reference trajectory and the previous measurements (errors and control inputs) to the new control signal is linear or nonlinear, one may distinguish between linear (e.g., [9,46,82,124]) and nonlinear (e.g., [7,23,64,71]) ILC algorithms. If refinement of the control input is only based on the measurements from the previous iteration, then the ILC algorithm is of the first order (e.g., [9,46,82,124]). High order ILC algorithms use measurements from more trials [40].

The analysis of the conditions that guarantee convergence of the iterative scheme can be performed either in the time domain [9,23,67,71,125] or in the frequency domain [23,46,124,125,130,131,152]. Commonly, a time-domain ILC design requires very little knowledge about the system dynamics [125]. In robot control, such a design is usually based on an approximate robot dynamic model of which parameters are only roughly known [7,71], or even on linearized dynamics [46,124,130,152]. However, it is stated in [131] that an ILC design which uses all available information about the system dynamics can realize better performance of motion control than the designs based on approximate knowledge of these dynamics. The known dynamics can be explicitly included in the convergence analysis, and the maximum effect of ILC on the tracking performance can be anticipated. A frequency-domain ILC design is particularly suitable for including the information about the known dynamics, since it provides deeper engineering insight during the design phase, where tools like frequency response function analysis, filtering, and frequency-domain stability criteria can be used directly. The only limitation is that the frequency-domain ILC design is performed upon linearized robot dynamics, so the nonlinear effects cannot be directly integrated into the design.

This thesis considers a frequency-domain ILC technique which update law firstly appeared in [124]. This law is applicable on the linearized robot dynamics, and it has been already used in robot motion control in, e.g., [130]. In this thesis, the linearization is accomplished via nonlinear model-based compensation. Although already suggested as a possible solution for robot motion control (see, e.g., [46]), it seems that the combination of nonlinear model-based and iterative learning control has not been practically investigated yet, especially for motion control of direct-drive robots. To the best of the author's knowledge, this should be the first study of ILC design for direct-drive robots, where the frequency-domain analysis is enabled by nonlinear model-based compensation. This study is already presented in [29]. Unlike other frequency-domain ILC techniques used in robotics, the approach presented in section 7.2 provides formalism for tuning the learning filters based on linearized dynamics that are experimentally measured on the robot. It should be admitted, though, that ILC design presented in [130] also resorts to identified robot dynamics. Strongly decoupled via gearboxes of high reduction ratios, these dynamics were identified in time domain and then represented by linear parametric models. Bounds on modelling uncertainty were not found, so maximum effect of ILC on the control performance could not be anticipated. On the other hand, the approach presented in section 7.2 takes into account the nominal model of the linearized dynamics together with experimentally determined modelling uncertainty, which enables realistic prediction of maximal performance improvement via ILC. The effectiveness of the ILC design is experimentally demonstrated on the RRR robot in section 7.2.

1.4.2.4 Pulse response based control

A possible strategy to avoid the modelling step is to compute a control action using the plant's pulse response data only. One related method is used in [18] to solve a minimum time control problem for flexible structures. This method does not require an explicit model of the plant dynamics, but captures these dynamics via the measured plant's response to pulses in the control input. An optimal control profile that results in the desired plant's output is obtained by convolution of the pulse responses. When applied for control of a flexible spacecraft, this method circumvents the complex modelling step which simplifies the control design and avoids errors caused by modal truncation and other modelling approximations. However, this method is does not take into account eventual effect of noise in the considered system.

In [54], a data-based LQG (linear quadratic Gaussian) control problem is solved. Opposite to classical LQG control [59], the data-based solution does not require any state-space representation of the plant. Instead, the plant dynamics is captured via Markov parameters, which are equal to the values of the plant's impulse response at discrete-time instants. Unfortunately, this method is not practical for implementation, since it designs a batch form dynamic LQG controller using the Markov parameters computed from all past input/output data of the plant. Another limitation is that the optimal control signal is derived with the assumption that no system noise is present. Both problems have been resolved by the data-based LQG control method suggested in [155,156]: a finite number of data points is used to obtain a finite horizon recursive dynamic controller, and the system noise is introduced in the controller design. However, this method is limited to finite-time regulator problems and cannot be applied to trajectory tracking.

The data-based control methods presented in [54] and [155,156] derive the control actions assuming that the Markov parameters are already available, but do not explain how to get these parameters. An algorithm presented in [139] can be used to obtain these parameters. The algorithm provides Markov parameters from almost any set of input/output data, not requiring knowledge of the initial conditions and the stability of the plant.

By combining the Markov parameters estimation from [139] and data-based LQG control from [155,156], one may realize infinite time data-based LQG control applicable to tracking problems, as done in [6]. This reference presents simulation but also experimental results of data-based LQG control. The simulation results show that the data-based controller preserves stability in closed-loop with a plant which dynamics is time-varying. The experimental results obtained on the RRR robot are just preliminary but promising pioneer attempts to practically implement a data-based LQG control algorithm.

1.4.2.5 Simultaneous perturbation stochastic approximation algorithm

The simultaneous perturbation stochastic approximation (SPSA) is optimization algorithm [161] which can be used for online control of nonlinear stochastic systems of unknown dynamics. A typical control problem is to steer the system state coordinates along the reference state trajectories. A desired control performance is specified in terms of one-step-ahead cost functions which penalizes the states and control inputs. The control inputs are output of some function approximator (FA), i.e., the controller is represented in the form of a polynomial, neural-network, splines, wavelets, etc. Associated with the FA is a set of tuning parameters that must be determined such as to optimize the adopted cost function. The SPSA algorithm optimizes the cost function, while not requiring modelling information describing the relationship between this function and the parameters to be optimized. Data obtained under closed-loop conditions, namely, the state measurements and control inputs, are the only information supplied to the algorithm. Starting with an initial "guess" at a solution, the SPSA algorithm processes the supplied data to estimate optimal controller tuning. The estimates are updated on an iteration-by-iteration basis using the gradient of the cost function. This gradient is also estimated directly from data. The efficiency of gradient estimation is the major advantage of the SPSA over other data-based gradient estimation algorithms, such as finite-difference approximations to the gradient [17]. With a finite-difference gradient approximation, each parameter update (iteration) requires a number of data measurements proportional to the number of the tuning parameters, while the SPSA algorithm requires only two measurements. Such efficiency facilitates real-time application of the SPSA algorithm.

After being introduced as an approach to general optimization problems [160], the SPSA algorithm has become attractive for control of complex systems with highly nonlinear and time-varying dynamics [162,163]. Typical applications are noise cancellation, process control, wastewater treatment, etc. In these applications, the credit is given to satisfactory control performance of the SPSA algorithm despite lack of prior knowledge about the plant dynamics. Stability analysis is usually omitted. Robotic applications of the algorithm can be found in [117-119,135]. In [117], inverse dynamics of a robot manipulator is modelled by a neural-network. The model can be used for control purposes. References [119] and [118] present experimental results of neural-network based motion control of one degree-of-freedom flexible robot arm. Tracking control of a rigid robot manipulator with two degrees-of-freedom is considered in [135]. The tuning parameters of the underlying neural-network based controllers are estimated via the SPSA algorithm. The considered applications confirm that the algorithm can realize motion control of a manipulator which dynamics is not known beforehand. However, it appears that for reaching the optimal (highest) position accuracy, a huge number of parameter updates is needed due to the low rate at which the parameter estimates converge to the "optimal" values. Namely, the SPSA algorithm has the property of bringing the estimates to within the vicinity of the optimum in relatively few time points but then taking a long time to complete the convergence to the optimum [163]. This property is still a formidable weakness of the algorithm, which makes it less practical solution for robot motion control where precise control is needed within a short time. Consequently, the SPSA algorithm is not considered further in this thesis.

1.4.2.6 Unfalsified control concept

In the concept of unfalsified control [27,28,31,86,99,116,146,148,174,182], control objectives are specified together with a set of candidate controllers. The candidate controllers belong to the same class, i.e., they are of identical complexity but have different parameter settings. Based on input-output signals observed from the controlled plant, obtained either under open-loop or closed-loop operating conditions, it is evaluated if a given controller setting is consistent with the objectives or not. An inconsistent controller is discarded from the candidate set, i.e., it is falsified, while the control action is realized with the optimal member from the set of consistent controllers. The consistent controllers are also called unfalsified controllers. The optimal controller setting is the most consistent with the control objectives among all unfalsified parameter settings. To facilitate computation of the optimal setting, linear controller parameterization is often used, while the control objectives are represented in terms of quadratic cost functions. At the moment of switching from the falsified controller to the optimal unfalsified one, the controller states are left unchanged.

The unfalsified control is a general data-based concept for optimal selection of the input to the plant, rather than a particular control design technique. Within this concept, one can realize the prescribed control performance for the same plant, using controllers of various forms and complexities. In [148], H_∞ optimal control performance was realized for the plant with time-varying dynamics. The selection of the controller class was rather arbitrary, i.e., the adopted linear controllers of linear parameterization were not based on the knowledge of the plant dynamics. On the other hand, a nonlinear controller class, implemented in [174] for manipulator motion control, was based on the rigid-body model of the manipulator dynamics. The regressor form of this dynamic model facilitated linear parameterization of the controller, and the elements of the base parameter set were used as the controller tuning parameters. Since the similar controller parameterization is also used in conventional adaptive control [137], the performance of unfalsified concept of adaptive control was compared with the conventional adaptive control strategy. It appeared that the unfalsified adaptive controller admits discontinuous changes in the parameter updates and, consequently, it was more robust against sudden changes of the manipulator inertial properties than the conventional adaptive controller with continuous parameter update rule. However, better control performance of the unfalsified controller was at the cost of much higher computational effort. Unfalsified adaptive control is also applied to nonlinear systems that exhibit multi-periodic and chaotic behaviour [99], missile autopilot system [28], electromechanical servos [116], etc. Unfalsified control concept also provided the basis for controller identification based on the measured data [27,31,182] and for a general purpose algorithm for automatic tuning of conventional industrial controllers [86]. This concept is applied in experimental settings ranging from flexible [182] and servo [116] electromechanical laboratory set-ups to industrial process control [146]. The experimental case studies confirm theoretical expectations that controllers tuned via unfalsified control concept exhibit satisfactory performance in the face of evolving uncertainties and parameter variations.

Although not pursued further in this thesis, unfalsified control is undoubtedly a prominent data-based tool for control law selection. Still, its implementation typically requires significant computational burden and more contributions are needed to facilitate real-time capability. Stability analysis is also required.

1.4.2.7 *Iterative feedback tuning*

Iterative feedback tuning (IFT) is a direct feedback control design method [42,43,50,68-70,74-79,87,88,110,111,180]. For a plant to be controlled, which dynamics is not known, a control designer selects a feedback controller of desired structure and complexity (linear or nonlinear), but with some number of parameters free to tune. The desired control performance is specified in terms of a cost function, which penalizes the difference between the desired and actual response of the system in closed-loop, together with the control input to the plant. The controller parameters are tuned through an iterative procedure, which performs numerical optimization of the cost function. During each iteration, the controller with the latest tuning is experimentally implemented on the plant and the closed-loop input-output data are measured. The measurements are used to compute a change of the tuning parameters along the gradient direction of the given cost function, such that the actual value of the cost is reduced. As the cost is decreasing, the control performance is getting better. The optimal controller tuning corresponds to the minimum of the cost function. Under the assumptions of a linear and time-invariant plant dynamics and closed-loop stability throughout iterations, convergence of the tuning parameters to stationary values can be established. Since the cost function can be non-convex with respect to the tuning parameters, the stationary controller tuning can correspond to some local optimum of the cost. In general, it is hard to achieve global optimal controller tuning.

The IFT is a method which attracts many researchers and industrial practitioners, because of simple implementation and capability to provide significant performance improvements with no need for the plant modelling step. The fact that the plant itself is used to generate the necessary information for improved closed-loop control has the advantage that the method is also able to cope with certain nonlinearities. Thorough presentations of IFT methodology are provided in [77] and [75]. Initially suggested for tuning of the single-input-single-output controllers [78], the method has also been proven effective for tuning of multivariable controllers [74]. Stability and robustness aspects are analyzed in [43,75,180]. In [180], it is emphasized that there is still no purely data-based condition for ascertaining the closed-loop stability throughout iterations. The same reference suggests that for a more reliable stability guarantee, a model of the plant should be estimated together with the bound of the modelling uncertainty. These estimates can be used to ascertain the stability of the closed-loop, which is in accordance with the strategy suggested in [43].

So far, many interesting results on IFT have been obtained and only some of them are addressed here. A study carried out in [110] shows that industrial conventional controllers (PID and PI) tuned via IFT outperform PID and PI tuned using other standard data-based methods addressed in subsection 1.4.2.1. Various control specifications are possible with IFT, e.g., specifications of the settling time and absolute error are discussed in [111] and [87], respectively. Some initial results on H_∞ loop-shaping control design via IFT are presented in [79,180]. Furthermore, evaluations of IFT with numerical simulations and experiments have been reported, considering both linear and nonlinear plants. Just to name a few: DC-servo system with backslash [77], rolling mills [76], thermal cycling modules [50], two-mass-spring system with friction [70], magnetic suspension system [88], Smith predictor [42], etc.

Robotics applications can be found in [69] and [68]. Reference [69] presents a control design for an academic example of a rigid robot manipulator. The objective of the design was to decouple motions between the manipulator joints, while minimizing the error in realizing the reference joint trajectory. Standard PD feedback controllers were used for stabilization and trivial proportional controllers for motion decoupling. Gains of the decoupling and feedback terms were successively derived via IFT. In particular, the tuning was carried out for each decoupling controller separately, while keeping the feedback gains fixed at an initial guess. In the second stage, the feedback controllers were tuned simultaneously, while the decoupling controllers were active in the loops. Quality of the obtained tuning was verified in simulation. The value of the given study is just to verify the applicability of the IFT method to a nonlinear multivariable plant, while the design itself seems rather conservative with respect to the realized tracking accuracy, closed-loop bandwidth, as well as stability and performance robustness. Additionally, it was observed that the achieved optimal controller tuning strongly depends on the reference joint trajectory. The influences of disturbance effects and parasitic dynamics have not been analyzed. Reference [68] presents an extension of [69] only in respect of adding the Coulomb and friction effects to the rigid-body dynamic manipulator model. The given IFT design just confirmed intuitive expectation that feedback gains needed to reach optimal motion performance in the presence of friction effects are higher than the gains in the frictionless case.

In this thesis, an IFT design for a direct-drive robot is considered. Feedback motion controllers of the prescribed complexity are determined via IFT. Since the motion control of high performance cannot be realized without dynamic compensation of the robot nonlinearities, the IFT control design is applied together with a nonlinear control law based on the rigid-body robot dynamic model with friction. This should be the first work that jointly considers a nonlinear robot motion control and an IFT design. Another question to be investigated is the influence of various behaviours contained in the measured data on the outcome of the IFT design. This is done by comparing the results of the same IFT algorithm applied on two sets of data presumably representing the same robot dynamics. In particular, one data set is synthetically generated from a data-driven robot model of high quality, while the other set is directly measured on the robot. These analyses are carried out for the RRR robot in section 7.3.

1.4.2.8 The virtual reference feedback tuning

Another strategy for direct design of a feedback controller based on a set of input-output measurements is called the virtual reference feedback tuning (VRFT) [33,34,65,66,107,108]. This strategy permits one to tune a controller within a specified class with no need for an intermediate plant modelling step. The feedback tuning is performed by minimization of a cost function which is defined in terms of data collected from the plant. In contrast to the iterative feedback tuning, the minimization is accomplished in one-shot, so no iterations are needed. The designer is free to choose a controller class of desired complexity, while the control objectives are specified by a suitable selection of reference closed-loop transfer functions: sensitivity, input sensitivity, or complementary sensitivity [52]. The control design via VRFT offers a solution to the model reference control problem when no mathematical description of the plant dynamics is available. Since the reference model of the closed-loop transfer function can be directly assigned, the VRFT design method provides an important degree of freedom in the definition of the control specifications.

The underlying idea on VRFT was originally proposed in [65] and subsequently developed in [33,34,107,108] as a complete and ready-to-use method. In references [33,34], the VRFT method is presented for tuning of one degree-of-freedom (d.o.f) controllers. The VRFT design for shaping the complementary sensitivity function is presented in [33], while methods for shaping the sensitivity and input sensitivity functions are explained in [34]. The reference [108] presents the extension of the VRFT method to a two d.o.f. controller setting. Shaping the closed-loop transfer functions is performed by tuning the linearly parameterized controllers. Apart from loop-shaping, VRFT is applied for approximate feedback linearization of nonlinear plants [66]. In that setting, the VRFT design also covers nonlinearly parameterized controllers. Reference [107] suggests a test to anticipate the stability of the closed-loop consisting of the plant and of the designed feedback controller. This test assumes that the plant dynamics is unknown and resorts only to the measurements used in VRFT design. Most of the contributions treat the VRFT strictly on the simulation level. The only experimental study presented so far is available in [34].

Upon VRFT methodology, a new data-based design of feedback controllers is derived in section 7.4. The relation with VRFT is explicitly addressed in subsection 7.4.3.

1.4.2.9 Stability issue

After surveying a number of data-based (DB) control techniques, the issue of closed-loop stability is the last aspect to be addressed in this section.

When the plant is unknown, it is hard to directly asses if the feedback controller is stabilizing or not. Inline with this statement goes the observation that there is a very limited number of contributions dealing with the stability issue within the DB control field. A few rare exceptions are summarized next.

Pulse response based control is rooted in linear quadratic Gaussian optimal control theory [59]. Stability concepts used in this theory can also be applied with pulse response based control methods. Other DB methods, i.e., control using the simultaneous perturbation stochastic approximation algorithm, unfalsified control, iterative feedback tuning, and the virtual reference feedback tuning, are not provided with compatible stability tests, yet. Compatibility means being in line with the adopted framework that only a set of measurements is available, so apart from control design, these data should be utilized to ascertain the closed-loop stability. Instead upon measurements, the addressed DB methods evaluate stability using the identified (data-driven) models of the plant dynamics. These models are obtained by fitting their free parameters into the measurements using, e.g., a prediction error identification method [113].

A stability analysis presented in [22], considers an ellipsoidal region in the parameter space that is determined around the identified model. The ellipsoid contains the real system at a certain probability level. The criterion for controller validation is that the controller is stabilizing for all models within the ellipsoid. Since both the plant model and the controller are available in parametric form, it is possible to use standard stability tests.

In [107], a Bayesian probabilistic approach is used to determine the probability that the controller is stabilizing or not. Around the identified values, the plant model parameters are randomly varied according to a normal distribution. The Bayesian approach determines the required number of random parameter sets that are required to guarantee that the true plant is covered by these sets with the desired level of confidence. The criterion for controller validation is that the controller is stabilizing for all plants corresponding to the random parameter sets.

Reference [43] presents a model-based mechanism to improve control over the iterative feedback tuning procedure. During each design iteration, one identifies the sensitivity and complementary sensitivity functions of the closed-loop consisting of the plant and of the controller. These functions are needed to estimate the closed-loop stability margins. The availability of the identified closed-loop quantities (sensitivity, complementary sensitivity, margins) provides better alternatives to the computation of steepest gradient decent directions for tuning the controller parameters.

Although barely addressed in the literature, stability is undoubtedly an important aspect of DB control. It is reasonable to believe that the attractiveness of stability analysis will be in proportion with general acceptation of DB control methods within the control community. The interest for these methods can be enhanced by increasing activities showing practical advantages of DB control. This thesis tends to be inline with such activities.

1.5 Contributions of the thesis

This thesis examines various designs of robot motion controllers that supplement nominal compensation of the robot nonlinearities. The nominal compensation is based on a model of the robot rigid-body dynamics with friction. The investigated control designs share one common feature: each of them intimately takes into account experimental characteristics of the robot under control. This means that effects of the robot dynamics not covered with the physical modelling, are also taken into account in the control design.

This thesis suggests a strategy to obtain the dynamic compensator for the robot nonlinearities. The strategy includes comprehensive modelling of the robot kinematics, rigid-body dynamics, and friction, together with a sophisticated experimental estimation of the model parameters. Both off-line and on-line approaches to parameter estimation are investigated. The considered strategy does not bring essentially a new theory, but it contributes with merging of various concepts available in the literature. Such merging is not found elsewhere in the literature.

Unlike other advanced methods for parameter estimation (e.g., [133,170]), the off-line parameter estimation in this thesis does not require identification of statistical properties of signal noises. This is because of using a Kalman observer to reconstruct all required motion coordinates (joint displacements, velocities, and accelerations). This observer filters out effects from the coordinates that are not due to the rigid-body dynamics. The filtered coordinates are used in a least-squares algorithm for parameter estimation, which is less complex than the algorithms suggested elsewhere in the literature. Although less complex, it still provides relevant parameter estimates, because it is not affected with parasitic and disturbance effects.

Online parameter estimation is based on a batch adaptive control algorithm [71] which, so far, has not been used for identification purposes. This algorithm contributes to the robustness and efficiency of estimation.

A writing task is suggested for firm validation of the robot kinematic and dynamic models. This task consists of more versatile motion primitives than other tasks proposed in the literature for model validation.

The thesis promotes the idea that the quality of a robot dynamic model should be verified both qualitatively and in a control application. On one hand, the qualitative validation is standard in the literature, and it means comparison of the control inputs rendered from the model with the inputs generated by conventional servo controllers in the robot joints. On the other hand, implementation of the model in a control algorithm is not a typical way for model validation (found only in [7]) but is promoted here as highly desirable, since it provides a definite answer if the model is adequate for control applications.

As a case-study, kinematic and dynamic models of the RRR robot are derived in closed-form. These models are general enough to cover the complete class of serial robots with the rotational joints implemented as the waist, shoulder, and elbow.

This thesis illuminates that the rigid-body dynamic model with friction is still not sufficient for a perfect match to the real robot dynamics, as, in practice, robot dynamics contains additional effects than those covered by the model. A procedure to identify these additional effects is proposed. This procedure originates from the motion control field, and, so far, it is barely used in robotics.

The identified additional dynamics are used in the robustifying linear feedback control designs (H_∞ and μ-synthesis). Identification of the robot dynamics not covered with the dynamic compensation of nonlinearities, and the robust control design dedicated to these identified dynamics, is a strategy not applied before to direct-drive robots. In this thesis, the strategy is shown beneficial, since it achieves higher quality of motion control than the control methods conventionally used for these robots. Additionally, the strategy provides deeper engineering insight during the robot control design phase, where tools like frequency response, filtering, and Bode plots can be used directly.

The thesis shows that use of linear robust feedback control design methods (H_∞ and μ-synthesis) enables quantitative prediction of the position accuracy, which is an important advantage with respect to nonlinear feedback control approaches, such as adaptive and sliding-mode control.

The disturbance based control design cycles [166] for fine controller tuning in robotics are investigated, and shown to improve disturbance rejection ability of motion control.

A fair treatment of a recent discrete-time sliding mode control (DSMC) algorithm [61] is given, and its utility in robot motion control is analyzed. The contributions are theoretical analysis and experimental results showing that the theoretical merits of the algorithm, namely, finite time reaching of ideal chattering-free DSMC and stability robustness against disturbances and parasitic effects, are achieved at the expense of unreasonably high feedback gains that may cause undesirable effects, such as noise amplification and excitation of parasitic dynamics.

A method for reducing vibrations at the tip of the robot is suggested. The vibrations are caused by structural flexibility of the robot. The key contributions are: (*i*) time-efficient identification of elastic effects and their incorporation into the model of the robot rigid-body dynamics, (*ii*) application of the resulting non-linear dynamic model to simplify the problem of vibration compensation to an ordinary set-point regulation problem, which is solvable using a well-developed linear control theory, and (*iii*) robust compensation of the tip vibrations.

A frequency-domain procedure for tuning an iterative learning controller (ILC) is applied to direct-drive robots. This should be the first study of ILC for direct-drive robots, where the frequency-domain design of the learning controllers is enabled by dynamic compensation of the robot nonlinearities. In contrast with majority of the frequency-domain ILC techniques used in robotics [23,46,124,125,130,131,152], the approach applied in this thesis provides a formalism for tuning of the learning filters. Thanks to this formalism, quantitative prediction of performance improvement via ILC becomes possible.

The work demonstrates a feedback control design using an iterative feedback tuning (IFT) algorithm. The tuning algorithm suitable for robot motion control is derived using a general IFT methodology. It is proven that this algorithm provides an unbiased estimate of the gradient of the considered cost function, which is important for optimal controller tuning. It is also shown that the IFT can be performed simultaneously with the dynamic compensation of the robot nonlinearities. Such a functional combination of nonlinear robot control and IFT is not seen elsewhere in the literature, and it enables optimal controller tuning with respect to all effects contained in the input/output data observed from the robot.

Finally, the work reported here contributes with an original data-based method for controller design which enables simultaneous shaping of the closed-loop sensitivity and the complementary sensitivity transfer functions. When evaluated in the robot motion control problem, the method is capable to tune controllers of chosen structure and complexity such that the motion control of high performance is realized.

1.6 Outline of the thesis

The reminder of the thesis consists of eight chapters.

The objective of data-driven robot motion control is formally stated in Chapter 2. This chapter gives an overview of typical robot motion control techniques and discusses their effects and performance limitations. Then, the data-driven control is suggested as a strategy for performance improvements.

Chapter 3 presents the steps preceding the design of model-based robot motion controllers: (*i*) modelling of the robot kinematics and rigid-body dynamics with friction, (*ii*) estimation of the model parameters, (*iii*) model validation, and, (*iv*) data-driven modelling of the effects not covered by the rigid-body dynamics with friction. These steps are formulated for a general serial robot manipulator with an arbitrary number of joints.

Chapter 4 practically demonstrates the modelling and identification concepts formulated in the previous chapter. A robot of spatial kinematics and direct-drive actuation is used as an experimental set-up. Hardware and software components of the considered RRR robot are described. Models of the robot kinematics and dynamics are derived and presented in Appendix. The estimation of model parameters and the model validation are illustrated with experimental results.

Chapter 5 presents several feedback control designs that utilize data-driven models of the robot dynamics. These designs are applied simultaneously with the dynamic compensation of the robot nonlinearities. The following designs are considered: 1) a conventional PD (proportional, derivative) feedback design, 2) an H_∞ feedback design within the disturbance-based control design cycles, 3) a feedback design for robust performance using μ-synthesis, and 4) a design of discrete-time sliding-mode feedback controllers. Theory behind each design is formulated for the general serial robot and its practical demonstration is given for the RRR robot. Quality of the considered designs is supported by experimental results obtained on the RRR robot.

Chapter 6 presents a method for reducing vibrations arising at the tip of a robot because of robot flexibility. This method combines a model of the robot rigid-body dynamics with a data-driven model of the robot flexibility. The combined model facilitates design of a robust vibration compensator in the robot task space.

Chapter 7 examines several data-based approaches to robot motion control: 1) iterative learning control, 2) iterative feedback tuning, and 3) a data-based shaping of the closed-loop sensitivity and complementary sensitivity functions. All considered approaches are treated both theoretically and in experiments on the RRR robot.

The conclusions and recommendations for further research are presented in Chapter 8.

Chapter 2

Problem formulation

The problem of robot motion control is addressed in the previous chapter in the perspective of a literature survey. Here, this problem is formulated in mathematical terms, which is needed for putting forward the particular problem of interest in this thesis, namely, data-driven robot motion control. This chapter presents the motion control methods that are generally accepted in robotics. Conservatism of these methods in realistic applications is illuminated, especially their limiting effects on control performance. Some control techniques that are used in the general field of motion control and in the emerging data-based control field, are suggested as possible remedies for the observed limitations. Strategies for combining these techniques with the control methods from the robotics area, are announced. The chapter is organized as follows. Section 2.1 describes a general solution strategy for the robot motion control problem and in more details discusses the control part of the strategy. Robot motion control strategies are summarized in section 2.2. The research questions of interest in this thesis, and the problem statement are given in section 2.3. The general solution strategy for the given problem statement, which will be followed in the remainder of the thesis, is presented in the last section.

2.1 Robot motion control

A typical architecture for solving the robot motion control problem is shown in Fig. 2.1. This problem is divided into three stages:

- motion planning,
- trajectory generation,
- trajectory execution.

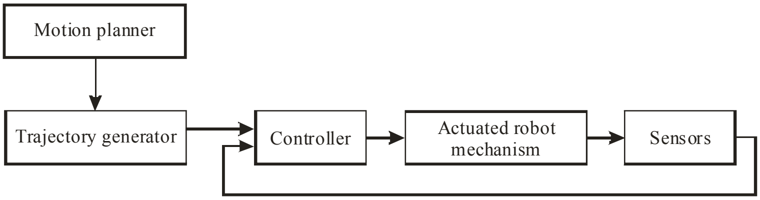


Figure 2.1: The components of the robot motion control problem.

Given a motion objective, e.g., handling some object by the robot end-effector, motion planning includes activities such as checking if the robot can move from one point to another without hitting an obstacle and determining how the work should be performed on the specified object. In industrial applications, planning is mainly done off-line by a system separate from the robot control system. This is performed by the operator or by a computer program, e.g., a CAD/CAM tool. In autonomous robotics, online motion planning is performed depending on the task objective and available sensory information about the position of the robot with respect to the neighbouring objects. On the planning level, the motion task is separated into motion primitives, e.g., movement from A to B along a straight line, from B to C along a curve, etc.

Trajectory generation is the problem of generating trajectories with position, velocity, and acceleration given as functions of time. These trajectories are restricted to the motion primitives obtained on the planning level. The problem also includes the consideration of the actual robot dynamics and kinematics, since the trajectories must be feasible, i.e., the robot must execute the trajectories that are generated. In some cases it is also a question of finding the optimal trajectory in terms of time or power consumption. The motion planning and the trajectory generation are often made in the robot task space. The trajectory is defined by the set of positions and orientations of the robot's end-effector relative to the base coordinate frame. This trajectory must be translated into the joint space to determine the reference motion compatible with the robot control system.

By appropriate control, motions of the joints are steered along the joint references, causing the end-effector to follow the trajectory calculated by the trajectory generator. In this thesis, emphasize is on the control part of the architecture shown in Fig. 2.1. In particular, control strategies to accurately realize the reference motions are considered. To facilitate presentation of these strategies, it is useful to introduce a general motion control system of interest in this thesis and to explain the meaning of its elements.

2.1.1 Motion control system

Consider the motion control system depicted in Fig. 2.2. Its basic components are the plant and the feedback and feedforward controllers. The plant is a controlled object, which in robotics represents the robot mechanism (chain of links and joints), actuators in the joints, and sensors. The actuators activate the joint displacements that result in the movement of the robot end-effector. Sensors measure mechanical variables, i.e., displacements, velocities, and accelerations. These are the controlled variables that define actual motion of the plant, i.e. the plant's output $\breve{y}$. The motion system can also be affected by disturbances η, that include external disturbances caused by environmental effects and sensor (measurement) noise. Varying load handled by the robot end-effector is an example of an external disturbance, while noise due to quantization effect is an example of measurement noise which arises if incremental encoders are used as position sensors. The measured output y of the plant is thus determined by

$$y = \breve{y} + \eta \,. \tag{2.1}$$

The reference input r is the desired motion of the plant. For the robot, r is the vector of the reference joint trajectories. The difference between the measured output and the reference represents the error signal:

$$e = y - r \tag{2.2}$$

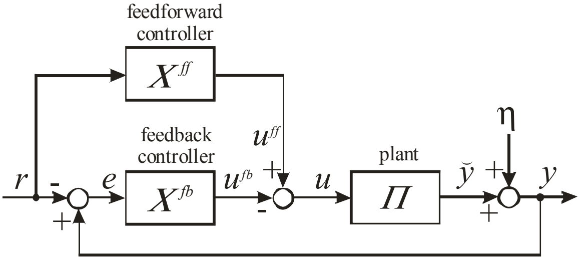


Figure 2.2: A motion control system.

By applying an appropriate control strategy, the controlled variables are steered along the reference ones

$$\breve{y}(t) \xrightarrow[t\to\infty]{} r(t) \,, \tag{2.3}$$

where t denotes time. The problem of motion control is to determine the control input to the plant $u(t)$ so as to guarantee execution of the desired motion while satisfying given transient and steady-state requirements. These requirements encompass stability, performance, and robustness. The control input to the robot is the vector of torques/forces that should be developed by the joint actuators. In general, the input u consists of feedforward u^{ff} and feedback u^{fb} components. These components are delivered by the corresponding controllers that implement suitable control strategies. The feedforward is used for linearity and performance improvements. In robotics, the linearity improvement means dynamic linearization of the robot dynamics. Performance improvement means enhancing the accuracy in realizing the reference trajectory (small e), both in transient phase and in steady-state. Additionally, the feedforward can compensate for disturbances that are repetitively appearing in the control loop (e.g. friction). However, it can hardly be used to reject disturbances that are occasionally occurring, e.g., stochastic noise and non-repeatable deterministic disturbances like mechanical shocks. These disturbances are under responsibility of the feedback control, together with closed-loop stability and stability robustness. The feedback also contributes to position accuracy, transient response characteristics, and control bandwidth. Most of these motives for feedback are conceptually explained next in mathematical terms [25,104].

To simplify presentation, it is assumed that $u^{ff} = \eta = 0$, which implies $y = \breve{y}$ and $u = -u^{fb}$. The plant and the feedback controller are represented as input-output (I/O) mapping systems with the maps Π and X^{fb}, respectively:

$$y = \Pi(u) = \Pi(-u^{fb}), \tag{2.4a}$$

$$u^{fb} = X^{fb}(e). \tag{2.4b}$$

Consequently, the feedback system shown in Fig. 2.2 is represented by the error equation (2.2) and by the I/O map of the series connection of the plant followed by the feedback controller:

$$y = \Pi(-u^{fb}) = \Pi\left(-X^{fb}(e)\right) = -\Lambda(e), \tag{2.5}$$

where Λ is the result of map composition. Equation (2.5) defines the I/O map for the feedback loop shown in Fig. 2.2, i.e., the loop I/O map. By substituting (2.5) into (2.2), one determines the feedback equation:

$$e + \Lambda(e) = -r. \tag{2.6}$$

Admissible reference signals r are limited in amplitude and should have constrained frequency content. Suppose that for a given class of admissible r, the feedback equation (2.6) has a bounded solution e. Also suppose that for this class of the references, the “gain” of the map Λ is large, i.e.,

$$\|\Lambda(e)\| >> \|e\|, \tag{2.7}$$

with $\|.\|$ denoting some signal norm [104]. If condition (2.7) holds, then the feedback equation can be rewritten as

$$\Lambda(e) \approx -r. \tag{2.8}$$

By virtue of (2.7) and (2.8), it follows that

$$\|r\| >> \|e\|. \tag{2.9}$$

The given analysis shows that if the loop gain is large, then the error is small compared with the reference input, which, by virtue of (2.2), implies

$$y \approx r. \tag{2.10}$$

If e is bounded for every bounded r (assumption for the feedback equation), then the closed-loop system by definition is BIBO (bounded input, bounded output) stable. The approximate identity

(2.10) remains valid as long as the system is BIBO stable and the loop gain is large. Since the plant dynamics is never completely known and its properties are not always favourable, the purpose of the feedback is to realize a large gain of the map Λ, which ensures that the map from r to y (I/O map of the feedback system) is almost independent of the plant map Π. The property that the approximate identity (2.10) holds in spite of uncertainty about the plant dynamics is called robustness of the feedback system with respect to plant uncertainty.

Expression (2.10) also indicates the linearity improvement by feedback control, since the I/O map of the feedback system is approximately linear no matter how nonlinear the plant I/O map Π is. Another consequence of (2.10) is the bandwidth improvement by feedback control. The I/O map of the feedback system is close to unity over those frequencies for which the loop gain is large. If the feedback controller enlarges this frequency range, it also increases the bandwidth. A further useful property of feedback is reduction of disturbances. Assume $\eta \neq 0$ and, for simplicity of presentation, let $r = 0$ in addition to $u^{ff} = 0$. Then, the feedback system shown in Fig. 2.2 is represented by the equations:

$$y = \breve{y} + \eta, \tag{2.11a}$$

$$\breve{y} = -\Lambda(e), \tag{2.11b}$$

$$e = -y. \tag{2.11c}$$

By eliminating $\breve{y}$ and e from (2.11a-c), one obtains the feedback equation

$$y + \Lambda(-y) = \eta. \tag{2.12}$$

Assume that (2.12) has bounded solution y for any bounded η, i.e., that the closed-loop system is BIBO stable. In this case, if the gain of the loop map Λ is large in the sense that

$$\|\Lambda(-y)\| >> \|y\|, \tag{2.13}$$

then we have

$$\Lambda(-y) \approx \eta \tag{2.14}$$

and in turn

$$\|y\| >> \|\eta\|. \tag{2.15}$$

The fact that the output y of the feedback system is small compared with the disturbance η implies that the effect of the disturbance is much reduced at the plant output. It is the role of feedback to ensure the sufficiently large gain of the loop map Λ.

The above analysis shows that high feedback gain is the desirable property for better motion accuracy, linearity and bandwidth improvements, as well as for disturbance reduction. However, the gain cannot be arbitrarily increased, since actuators have limited driving capabilities and high gain can also endanger closed-loop stability. If the system is unstable, the feedback equation has no bounded solution and the benefits of the feedback are lost. Feedback controller generates control action based on the measured plant output. The associated measurement errors and measurement noise may cause loss of accuracy.

2.1.2 Categorization of the motion control systems

With respect to its dedicated function, the motion control system shown in Fig. 2.2 falls into the following categories [25]:

- Regulator system: its primary role is keeping y at a predetermined constant value $r(t) = \mathrm{r}$ ($\dot{r}(t) = 0$), despite the effects of load changes and other disturbances; an example could be maintaining the desired configuration of the robot while performing some assembly task (e.g., screw driving),

- Servo or positioning system: the reference motion $r(t)$ ($\dot{r}(t) \neq 0$) is known in advance, and the motion control should realize accurate change of $y(t)$ as commanded by $r(t)$; this also covers the function of the regulator; an example is accomplishing the reference robot trajectory which corresponds to a predefined spray painting motion task,
- Tracking system: equivalent function with the servo motion control system, with the difference that the reference motion $r(t)$ is not known in advance; for instance, $r(t)$ represents a trajectory which is created online by the motion planner of an autonomous robot, or $r(t)$ corresponds to measured or observed trajectory which should be tracked by the robot (e.g., synchronization with motion of another robot).

Control techniques implemented in the servo and tracking control problems can differ with respect to admissible control design strategies and achieved performance effects.

If the motion system is restricted to repeatedly execute some predefined reference $r(t)$, i.e., the servo control system is considered, then both feedforward and feedback control components can be optimized with respect to the motion reference. In this case, u^{ff} can be derived via some iterative learning control algorithm to optimize position accuracy. The repetitive nature of the motion task facilitates performance improvement, since it is possible to apply non-causal data processing algorithms during the learning process to compensate for phase lag in the control action. The phase lag is inherent with online control, and it has negative effects on control performance. While u^{ff} is optimized for performance, u^{fb} can be optimized to reject disturbances that appear occasionally or are stochastic, and hence cannot be effectively captured during learning. The described strategy is used in industrial robot applications, where repetitive motions are often needed in manufacturing tasks.

If the reference of the servo control system is not repetitive, then u^{ff} cannot be obtained by learning but still can significantly contribute to performance improvement. For instance, if the plant features a time-delay, then u^{ff} can be computed using a known plant model and applied to the plant adequately forwarded in time to compensate for the delay. Another advantage of servo control is the availability of the reference trajectory whose position, velocity, acceleration time-histories are highly accurate. Their accuracies contribute to the motion accuracy, since both feedforwad and feedback control action depend on the reference trajectory. A feedback controller useful for servo control tasks should be designed using strategies that ensure effectiveness of u^{fb} along the reference trajectories of different waveforms. Of course, feedback can handle only those references whose harmonic contents are covered with the bandwidth of the closed-loop system. As seen in the previous subsection, feedback should ensure as high as possible loop gain within the bandwidth for better motion accuracy. Outside the bandwidth, feedback is required to provide adequate gain reduction to avoid amplification of flexible dynamics and high frequency noises.

Generating the reference $r(t)$ online impedes accurate execution of the motion task. In a tracking control system, the feedforward component can still be computed using the known plant model, but its effectiveness now depends on the accuracy of online obtained motion reference. Quality of feedback control is also influenced by the accuracy of the reference. If computation of the control actions requires the knowledge of the reference position, velocity, and acceleration time-histories, which is the case in robot motion control, then all these references should be determined online. If these are observed from sensor signals, which happen when a robot synchronizes its motions with another one, then the measurement noise can negatively influence motion performance. Filtering the measurements and use of the state observers are techniques to reduce the effect of noise, at the cost of increased computational complexity. If not all mechanical variables are measured (e.g., only position measurements are available), then causal data-processing of the measurements is needed for online reconstruction of the required variables. Online data-processing introduces phase lag and numerical errors that can also reduce the quality of motion control. Like a feedback controller useful for the servo tasks, a feedback controller implemented for trajectory tracking should be capable to realize arbitrary trajectories. Therefore, equivalent strategies can be used to design feedback controllers of servo and tracking motion control systems.

This thesis deals with servo control of robot manipulators, which means that the reference robot trajectory, i.e., the reference joint motions, velocities, and accelerations, are all known in advance. However, feedback control designs will not be restricted to some specific motion task but will aim at effectiveness for arbitrary references. Thus, the feedback designs of interest in this thesis should also be useful for tracking purposes. Emphasis is on control of direct-drive robots, which means that the plant dynamics is essentially nonlinear and there are significant couplings between the motions of the robot joints. Feedback and feedforward control components should realize accurate motion, robustly against nonlinearity, parasitic dynamics, and disturbance effects. Specific requirements for control designs will be formulated in section 2.3. The requirements will be motivated by limitations of the motion control strategies generally accepted in robotics. These strategies are addressed in the next section, and their limitations of concern in this thesis are illuminated.

2.2 Robot motion control strategies

Presentation of the robot motion control strategies requires formulation of the robot dynamics. Let us represent the dynamics of a realistic serial robot manipulator with n actuated joints in standard form [53,153,181]:

$$\begin{aligned}\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t)+\mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t))+\mathbf{g}(\mathbf{q}(t))+\boldsymbol{\tau}^{f}(\mathbf{q}(t),\dot{\mathbf{q}}(t),t)=\boldsymbol{\tau}(t)&\\ +\boldsymbol{\tau}_{\mathrm{v}}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\ddot{\mathbf{q}}(t),t),&\end{aligned} \tag{2.16}$$

where $\mathbf{q}\,,\dot{\mathbf{q}}\ \ddot{\mathbf{q}}\in\mathbb{R}^{n}$ are the vectors of joint motions, velocities, and accelerations, respectively, $\mathbf{M}\in\mathbb{R}^{n\times n}$ is the positive definite inertia matrix, $\mathbf{c}\,,\mathbf{g}\,,\boldsymbol{\tau}^{f}\in\mathbb{R}^{n}$ are the vectors of Coriolis/centripetal, gravity and friction effects, respectively, $\boldsymbol{\tau}\in\mathbb{R}^{n}$ is the vector of the control inputs (forces/torques), $\boldsymbol{\tau}_{\mathrm{v}}\in\mathbb{R}^{n}$ is the collection of all perturbations from the rigid-body dynamics with friction (e.g. flexibilities and disturbances), and t is time.

The dynamics (2.16) are highly nonlinear and coupled. Use of gearboxes as transmission elements between the joint actuators and robot links can have linearizing effect on these dynamics [153]. Let $\boldsymbol{\theta}\in\mathbb{R}^{n}$ and $\boldsymbol{\tau}_{m}\in\mathbb{R}^{n}$ denote the vectors of actuator displacements and actuator driving torques, respectively; the transmissions then establish the following relationships:

$$\begin{aligned}\boldsymbol{\theta}(t)&=\mathbf{N}\mathbf{q}(t)\,;\\ \boldsymbol{\tau}_{m}(t)&=\mathbf{N}^{-1}\boldsymbol{\tau}(t)\,,\end{aligned} \tag{2.17}$$

where $\mathbf{N}\in\mathbb{R}^{n\times n}$ is a diagonal matrix of gear reduction ratios n_i ($i=1,\ldots,n$). If the moments of inertia $J_{m,i}$ of each actuator shaft are collected in the diagonal matrix $\mathbf{J}_{m}\in\mathbb{R}^{n\times n}$, then the inertia matrix $\mathbf{M}$ can be split as follows:

$$\mathbf{M}(\mathbf{q}(t))=\mathbf{N}\mathbf{J}_{m}\mathbf{N}+\Delta\mathbf{M}(\mathbf{q}(t)) \tag{2.18}$$

where $\mathbf{N}\mathbf{J}_{m}\mathbf{N}$ is a constant matrix, and $\Delta\mathbf{M}$ is the configuration dependent matrix. Substituting (2.17) and (2.18) into (2.16), yields:

$$\boldsymbol{\tau}_{m}(t)=\mathbf{J}_{m}\ddot{\boldsymbol{\theta}}(t)+\mathbf{d}(t)\,, \tag{2.19}$$

where

$$\mathbf{d}(t)=\mathbf{N}^{-1}\Delta\mathbf{M}(\mathbf{q}(t))\mathbf{N}^{-1}\ddot{\boldsymbol{\theta}}(t)+\mathbf{N}^{-1}\left(\mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t))+\mathbf{g}(\mathbf{q}(t))+\boldsymbol{\tau}^{f}(t)-\boldsymbol{\tau}_{\mathrm{v}}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\ddot{\mathbf{q}}(t),t)\right) \tag{2.20}$$

represents the nonlinear dynamic interaction between the joint actuators. In case of high reduction ratios ($n_i >> 1$), the contribution of the interaction is small compared to the linear dynamics $\mathbf{J}_m \ddot{\boldsymbol{\theta}}$. Consequently, the manipulator dynamics (2.19) are practically decoupled and linearized, as each component of $\boldsymbol{\tau}_m$ dominantly influences the corresponding component of $\boldsymbol{\theta}$, while $\mathbf{d}$ just plays the role of a disturbance to each joint servo.

2.2.1 Decentralized motion control

Control of the decoupled dynamics (2.19) can be done using a decentralized feedback method. Denote with $\boldsymbol{\theta}_r(t)$ a reference trajectory of the actuator shaft, which is determined from the joint reference $\mathbf{q}_r(t)$ via (2.17). Define the position error as the difference between the actual and the reference motion:

$$\mathbf{e}_m(t) = \boldsymbol{\theta}(t) - \boldsymbol{\theta}_r(t). \tag{2.21}$$

A decentralized control treats each robot joint as a separate plant to be controlled and applies decoupled linear feedback control actions u_i ($i = 1, \ldots, n$) as follows:

$$\boldsymbol{\tau}_m(s) = \mathbf{J}_m \mathbf{G}_a(s) \mathbf{u}(s), \tag{2.22}$$

where $\mathbf{G}_a(s) = diag[G_{a,1}(s), \ldots, G_{a,n}(s)]$ is a diagonal matrix containing the actuator dynamics, $\mathbf{u} = [u_1, \ldots, u_n]^T$ is the vector of the feedback control actions, and s is the Laplace operator. Each u_i is decoupled from the remaining ones, as it is based just on the position error corresponding to actuator *i*.

For convenience, we may assume no actuator dynamics ($\mathbf{G}_a$ is the identity matrix) and a conventional PD feedback control law with acceleration feedforward:

$$\mathbf{u}(t) = \ddot{\boldsymbol{\theta}}_r(t) - \mathbf{K}_p \mathbf{e}_m(t) - \mathbf{K}_d \dot{\mathbf{e}}_m(t), \tag{2.23}$$

where $\mathbf{K}_p = diag[k_{p,1}, \ldots, k_{p,n}]$ and $\mathbf{K}_d = diag[k_{d,1}, \ldots, k_{d,n}]$ are matrices of positive proportional and derivative gains, respectively. The decentralized control approach (2.22),(2.23) is illustrated in Fig. 2.3.

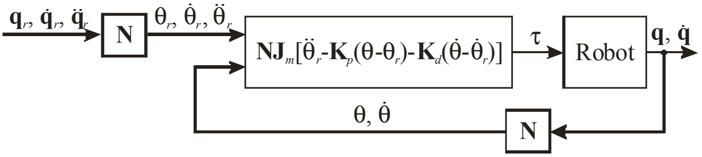

$\mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r$
N
$\theta_r, \dot{\theta}_r, \ddot{\theta}_r$
$\mathbf{NJ}_m[\ddot{\theta}_r - \mathbf{K}_p(\theta - \theta_r) - \mathbf{K}_d(\dot{\theta} - \dot{\theta}_r)]$
τ
Robot
$\mathbf{q}, \dot{\mathbf{q}}$
$\theta, \dot{\theta}$
N

Figure 2.3: Decentralized motion control with PD feedback and acceleration feedforward.

The application of the control law (2.22),(2.23) to the system (2.19) gives the error dynamics:

$$\mathbf{J}_m \left(\ddot{\mathbf{e}}_m(t) + \mathbf{K}_d \dot{\mathbf{e}}_m(t) + \mathbf{K}_p \mathbf{e}_m(t) \right) = \mathbf{d}(t). \tag{2.24}$$

It follows that the position error falls into a region of size proportional to the magnitude of $\mathbf{d}(t)$. If this region is narrow, which is the case when high reduction ratios are used, then the error can be made small by appropriate selection of the PD gains. If the gear reduction ratios are small, or if $\mathbf{N}$ is the identity matrix (direct-drive actuation), then the nonlinear coupling terms become more prominent or even dominant for fast motions in comparison with the linear part in (2.24). Consequently, the magnitude of $\mathbf{d}(t)$ is not small, and position errors become larger for the same feedback controller gains. This is the major limitation of the decentralized motion control strategy. As a remedy, one can apply a model-based compensation for $\mathbf{d}(t)$, which is explained next.

2.2.2 Nonlinear model-based compensation

Nonlinear dynamic couplings between the robot joints can be compensated using control algorithms that implement models of the robot dynamics. Typical models cover rigid-body dynamics and friction effects. These are captured with the left-hand side of equation (2.16). There are two common approaches to model-based dynamic compensation of nonlinearity [7,129,153,158]:

- inverse dynamics control

$$\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}^{f}(\mathbf{q}(t),\dot{\mathbf{q}}(t),t) + \mathbf{u}(t), \tag{2.25}$$

- computed-torque (feedback linearization)

$$\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t))\mathbf{u}(t) + \mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}^{f}(\mathbf{q}(t),\dot{\mathbf{q}}(t),t). \tag{2.26}$$

In both approaches, $\mathbf{u} = [u_1,\ldots,u_n]^T$ stands for the feedback control action based on the error between the actual and the reference joint motions:

$$\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}_r(t), \tag{2.27}$$

and possibly extended with acceleration feedforward term. The role of $\mathbf{u}$ is to stabilize the robot motion and ensure the desired control performance.

The control strategies (2.25) and (2.26) are often characterised as “centralized control”, since each component of the control inputs $\boldsymbol{\tau}$ contains dynamic interaction between the robot joints. In the control laws (2.25) and (2.26), the model-based terms are computed along the actual joint motions, velocities, and accelerations. However, the reference motions and their time derivatives can also be used for their computation. Some feedback designs applicable in the control laws (2.25) and (2.26) are presented next.

2.2.3 Inverse dynamics control with conventional feedback

Assuming perfect model-based compensation of nonlinear rigid-body dynamics including friction (2.16), i.e., in the absence of the modelling uncertainty term $\boldsymbol{\tau}_v$, a control law that guarantees global uniform asymptotic stability of the origin $(\mathbf{e},\dot{\mathbf{e}}) = (\mathbf{0},\mathbf{0})$ is given by [138]:

$$\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}_r(t) + \mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\dot{\mathbf{q}}_r(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}^{f}(\mathbf{q}(t),\dot{\mathbf{q}}(t),t) - \mathbf{K}_p\mathbf{e}(t) - \mathbf{K}_d\dot{\mathbf{e}}(t). \tag{2.28}$$

Here, $\mathbf{K}_p = diag[k_{p,1},\ldots,k_{p,n}]$ and $\mathbf{K}_d = diag[k_{d,1},\ldots,k_{d,n}]$ are matrices of positive proportional and derivative gains, respectively. Obviously, the model-based terms employ both the reference and the actual joint trajectory. The control law (2.28) is illustrated in Fig. 2.4.

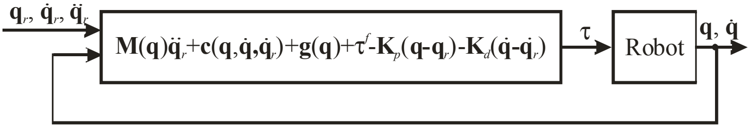

$\mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r$
$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{c}(\mathbf{q},\dot{\mathbf{q}},\dot{\mathbf{q}}_r) + \mathbf{g}(\mathbf{q}) + \tau^f - \mathbf{K}_p(\mathbf{q}-\mathbf{q}_r) - \mathbf{K}_d(\dot{\mathbf{q}}-\dot{\mathbf{q}}_r)$
τ
Robot
$\mathbf{q}, \dot{\mathbf{q}}$

Figure 2.4: Inverse dynamics motion control with PD feedback.

Although globally stabilisable and promising better control performance than a decentralized controller, the law (2.28) still features some limitations. For instance, it is not obvious how to tune the PD gains to realize the desired performance specifications, e.g., prescribed error reduction, bandwidth constraint, and attenuation of undesired high-frequency effects (quantisation noise and flexibilities). The feedback gains are hence tuned by trial and error on the real system, until a satisfactory trade-off between stability, amount of error reduction, and impact of parasitics and disturbances is reached. Such a tuning strategy is not plausible in practice, since it does not allow the designer to quantitatively anticipate feasible performance characteristics. Another disadvantage of the law (2.28) is that the PD feedback part acts independently of the configuration dependent dynamics, which may cause perturbations at neighbouring joints [7]. Namely, a corrective PD torque/force at one joint perturbs the other ones, whereas ideally the corrective torques/forces should be decoupled, and each of them should influence only the motion of the corresponding joint. This problem is circumvented in computed-torque control, addressed next.

2.2.4 Computed-torque control

The idea of computed-torque is to algebraically transform a system of equations describing nonlinear robot dynamics into a linear system by compensating all the coupling nonlinearities in (2.16). Feedback correction terms are applied to linear decoupled dynamics, that represent the plant to be controlled. A typical computed-torque control law is given by (2.26) with

$$\mathbf{u} = \ddot{\mathbf{q}}_r - \mathbf{K}_p \mathbf{e} - \mathbf{K}_d \dot{\mathbf{e}}, \qquad (2.29)$$

where $\mathbf{K}_p = diag[k_{p,1},\ldots,k_{p,n}]$ and $\mathbf{K}_d = diag[k_{d,1},\ldots,k_{d,n}]$ are once again matrices of positive proportional and derivative gains, respectively. Contrary to inverse dynamics control law (2.28), the PD feedback controller in (2.29) sends its output through the dynamic model. This is illustrated in Fig. 2.5.

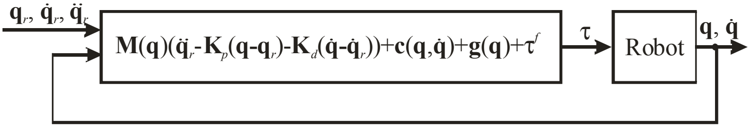

$\mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r$
$\mathbf{M}(\mathbf{q})(\ddot{\mathbf{q}}_r-\mathbf{K}_p(\mathbf{q}-\mathbf{q}_r)-\mathbf{K}_d(\dot{\mathbf{q}}-\dot{\mathbf{q}}_r))+\mathbf{c}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{g}(\mathbf{q})+\boldsymbol{\tau}^f$
$\boldsymbol{\tau}$
Robot
$\mathbf{q}, \dot{\mathbf{q}}$

Figure 2.5: Computed-torque motion control with PD feedback.

The reference acceleration is added to the PD output, because the feedback control action $\mathbf{u}$ is used to correct acceleration in the robot joints

$$\ddot{\mathbf{q}} = \mathbf{u}. \qquad (2.30)$$

The last relation results from application of (2.26) to (2.16), assuming no uncertainty term $\boldsymbol{\tau}_v$. The system (2.30) reveals the decoupled linear dynamics. Indeed, when controlling the motion q_i in joint *i*, each element u_i of $\mathbf{u}$ acts at a decoupled plant of double integrator dynamics. Application of the law (2.29) to the system (2.30) results in the error dynamics

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \mathbf{0}. \qquad (2.31)$$

By choosing the gains in $\mathbf{K}_p$ and $\mathbf{K}_d$, theoretically it is possible to place the poles of the error characteristic equation at arbitrary locations. Realistic robot dynamics (2.16) feature the nonzero term $\boldsymbol{\tau}_v$, which affects the error dynamics as follows

$$\ddot{\mathbf{e}} + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = \boldsymbol{\eta}, \qquad (2.32)$$

where

$$\boldsymbol{\eta} = \mathbf{M}^{-1}(\mathbf{q})\boldsymbol{\tau}_v(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}},t). \qquad (2.33)$$

The indicated inverse exists by the positive definiteness of the inertia matrix. The presence of $\boldsymbol{\eta}$ puts more restrictions on $\mathbf{K}_p$ and $\mathbf{K}_d$, since nonlinear couplings between the error dynamics and uncertainty may cause $\mathbf{e}$ to become unbounded if $\mathbf{K}_p$ and $\mathbf{K}_d$ and chosen too high. This limits the possible bandwidth in the closed-loop. Robust and adaptive feedback control methods are introduced to deal with uncertainties in the system dynamics.

In robotics, typical robust feedback designs are based on the theories of variable structure systems with sliding-modes [158,175] and on application of Lyapunov's direct method for stability analysis [158]. To illustrate these designs, consider the computed-torque law (2.26) with $\mathbf{u}$ defined as

$$\mathbf{u} = \ddot{\mathbf{q}}_r - \mathbf{K}_p\mathbf{e} - \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{u}_a . \tag{2.34}$$

Here, $\mathbf{u}_a \in \mathbb{R}^n$ is an extra control term to be chosen. Application of (2.26),(2.34) to the robot dynamics (2.16), results in the error dynamics:

$$\ddot{\mathbf{e}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \mathbf{u}_a + \boldsymbol{\eta} , \tag{2.35}$$

where $\boldsymbol{\eta}$ is determined by (2.33). The state-space description of (2.35) is

$$\dot{\mathbf{x}}^e = \mathbf{A}^e\mathbf{x}^e + \mathbf{B}^e(\mathbf{u}_a + \boldsymbol{\eta}) , \tag{2.36a}$$

$$\mathbf{x}^e = \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix}; \ \mathbf{A}^e = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{I}_n \\ -\mathbf{K}_p & -\mathbf{K}_d \end{bmatrix}; \ \mathbf{B}^e = \begin{bmatrix} \mathbf{0}_{n\times n} \\ \mathbf{I}_n \end{bmatrix}, \tag{2.36b}$$

where $\mathbf{0}_{n\times n} \in \mathbb{R}^{n\times n}$ contains zeros only and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. Using a time-varying bound $\rho(\mathbf{x}^e, t)$ on the uncertainty $\boldsymbol{\eta}$

$$\|\boldsymbol{\eta}\| \le \rho(\mathbf{x}^e, t) , \tag{2.37}$$

with $\|.\|$ denoting some signal norm [104], the goal is to design the additional control term $\mathbf{u}_a$, such that the state trajectory in (2.36a,b) is bounded or, if possible, converges to zero. It is not easy to determine ρ in (2.37) as a time-varying function, since in general $\boldsymbol{\eta}$ is unknown.

In sliding mode control designs, the time-varying ρ is substituted with the time-independent vector $\boldsymbol{\rho}$, whose elements ρ_i ($i = 1,\ldots,n$) are bounds on the corresponding elements of $\boldsymbol{\eta}$. The simplest sliding-mode controller results from choosing the extra input $\mathbf{u}_a$ according to

$$\mathbf{u}_a = \boldsymbol{\rho}\,\mathrm{sgn}(\mathbf{s}) , \tag{2.38}$$

$$\mathbf{s} = [s_1,\ldots,s_n]^T ; \ s_i = e_i + \lambda_i\dot{e}_i \ (i = 1,\ldots,n).$$

Here, λ_i prescribes the desired error dynamics on the sliding manifold $s_i = 0$, and $\mathrm{sgn}(.)$ is the sign function applied componentwise on the vector $\mathbf{s}$.

An alternative approach is to use the so-called theory of guaranteed stability of uncertain systems, based on Lyapunov's direct method. The matrix $\mathbf{A}^e$ in (2.36a,b) is Hurwitz, which means that for every positive symmetric $\mathbf{Q} \in \mathbb{R}^{n\times n}$, there exists a positive symmetric $\mathbf{P} \in \mathbb{R}^{n\times n}$, satisfying the Lyapunov equation

$$(\mathbf{A}^e)^T\mathbf{P} + \mathbf{P}\mathbf{A}^e = -\mathbf{Q} . \tag{2.39}$$

This follows from the Lyapunov stability theory for linear time invariant systems [158]. Using the matrix $\mathbf{P}$, the control term $\mathbf{u}_a$ can be chosen as

$$\mathbf{u}_a = \begin{cases} -\rho(\mathbf{x}^e, t)\dfrac{(\mathbf{B}^e)^T\mathbf{P}\mathbf{B}^e}{\overline{\sigma}\left((\mathbf{B}^e)^T\mathbf{P}\mathbf{B}^e\right)} & \text{if } \overline{\sigma}\left((\mathbf{B}^e)^T\mathbf{P}\mathbf{B}^e\right) \neq 0; \\ \mathbf{0}_n & \text{otherwise,} \end{cases} \tag{2.40}$$

where $\overline{\sigma}(.)$ denotes the maximum singular value [62]. With the Lyapunov function $V = (\mathbf{x}^e)^T\mathbf{P}\mathbf{x}^e$, it can be shown that $\dot{V}$ is negative definite along solution trajectories of the system given by (2.36a,b).

Practical application of both robust designs will lead to control *chattering*, especially with a sampled implementation of the control laws (2.38) and (2.40). The control chattering phenomenon, i.e., the finite amplitude oscillations of the control signal of high but finite frequency, negatively influences the performance of motion control, since it can excite flexible dynamics of the robot and cause vibrations in the joint motions. Many refinements and extensions to the two approaches for robust computed torque control have been proposed. Mainly the idea is to obtain less conservative uncertainty bounds and to smoother the chattering in the control signal [165]. Less conservative bounds are preferable, since they contribute to more adequate feedback gains and better control over the excitation of the parasitic dynamics. Disturbance observers are used to reduce the conservatism on the disturbance bounds, at the cost of increased computational complexity. Although increased computational effort is not really an issue in modern robot control, the problem which still remains is high nonlinearity of the resulting control action which impedes quantitative prediction of the motion control performance, e.g., the amount of error reduction and attenuation of the parasitics, closed-loop bandwidth, etc. Linearization of the discontinuous feedback action within the so-called boundary layer is a method for smoothing the chattering. However, this method destroys the desired properties of sliding mode control, e.g., finite time reaching the sliding manifold and invariance to disturbances [175]. Although some more conceptual strategies to counteract the chattering exist, e.g., robust chattering free control algorithm presented in [61], to date they are barely applied in robotics.

Adaptive computed-torque control [158] can also be used to cope with the uncertainty in the robot dynamics. This technique makes use of a model of the robot rigid-body dynamics and applies online adaptation of the model parameters such as to ensure accurate execution of the reference trajectory despite uncertainties. However, similarly to the above robust methods, nonlinearity of adaptive control prevents quantitative prediction of the motion control performance.

The above survey of generally accepted robot motion control strategies is certainly not an exhaustive one, but should be sufficient to point at the problems of concern addressed in this thesis. These problems are more or less characteristic to the majority of the robotics control algorithms. Possibilities how data-driven approaches can be used to circumvent the observed problems are discussed next.

2.3 Research questions and problem statement

After the problems of concern have been indicated at a general level, now is the time to distinguish particular issues that will be directly approached in the thesis. These issues will be formulated as the research questions that should be answered in the following chapters.

In this thesis, the requirements for an effective robot motion servo control are:

- accurate execution of a given robot trajectory which is specified in terms of time-histories of the reference motions, velocities, and accelerations;
- robustness against disturbances and parasitic robot dynamics;
- opportunity to quantitatively predict effects of control on position accuracy, closed-loop bandwidth, and attenuation of disturbances and parasitic dynamics;
- ease of implementation on modern control platforms.

The given requirements should be achieved by an adequate contribution of both feedback and feedforward components of the input to the robot. Complexity of nonlinear robot dynamics and strong couplings between the motions in different joints are the main obstacles to realize all the requirements. As indicated in the previous section, robotics research provides an important basis for motion control design, especially in the aspects robot modelling, identification, and model-based nonlinearity decoupling. However, control strategies from the robotics field often rely on too restrictive assumptions that impede achievement of all requirements. On the other hand, there is a wealth of systems and control theory that has not yet been exploited in robotics. Especially, developments in robust and data-based motion control seem to have potential to supplement robotics strategies. Therefore, it would be useful to distinguish those techniques from the motion and data-based control field that could contribute in realization of the given control requirements.

Model-based compensation of the robot nonlinearities is the basic mechanism of advanced robotics motion control schemes. Effectiveness of these schemes strongly depends on the quality of the underlying robot models. Theory offers a sufficient knowledge to come up with relevant models of the robot kinematics and rigid-body dynamics. Providing these models requires algorithms for derivation of the model equations and achievement of accurate values of the model parameters. Obtaining the parameters that are in high agreement with their physical values is performed by measurements and experimental estimation. Accurate estimation of parameters from the dynamic equations often involves complex data processing techniques and use of advanced estimation algorithms to reduce influences of the parasitic and disturbance effects. Unfortunately, the majority of practitioners working in robotics is lacking (deep) knowledge of advanced estimation procedures and appreciate less complex strategies to obtain the desired parameters. Therefore, the question arises if it is possible to develop procedures that can deliver parameter estimates of sufficient quality, but applying algorithms of lower complexities whose implementation would be more plausible in practice? Simplified procedures would be of benefit for both batch and online parameter estimation.

Even comprehensive representations of rigid-body dynamics supplemented with relevant friction models, have limited degrees of freedom and cannot represent the full complexity of the true dynamics. The models do not represent well deviations from the assumed model structure, which are always present to some extent. Thus, on any particular trajectory, a model of the rigid-body dynamics with friction will have some errors. Robust and adaptive control techniques used in robotics can compensate for modelling uncertainty, but their nonlinear nature raises additional issues illuminated in the previous section.

What is required is an additional level of modelling that allows representation of fine details of the dynamics. Fulfilment of this requirement would allow development of motion control strategies that consider the robot as a physical system rather than as a particular mathematical description of it. Data-driven modelling, based on system identification theory, is a tool to capture fine details of system dynamics in (linear) motion control systems. This modelling helps us discriminating the tighter margins between the nominal plant dynamics and additional phenomena present in the real system. Uncertainty in the dynamics and disturbances can be described by parametric models to facilitate application of linear robust control designs that enable quantitative prediction of control effects on the motion performance. The last property seems quite appealing to exploit the use of linear robust control designs in robotics. Therefore, new questions can be posed here. Is it useful to apply data-driven modelling in robotics to get better insight into discrepancies between the modelled rigid-body dynamics with friction and actual robot behaviour? Especially, is data-driven modelling applicable on highly nonlinear direct-drive robots? If these answers are affirmative, is it then possible to fuse linear robust control designs with nonlinear robot motion control strategies in order to realize all control requirements indicated at the beginning of this section?

The given questions certainly do not suggest linear robust control designs as the ultimate alternatives to nonlinear robust approaches. Still, linear theory offers useful techniques for controller tuning for the desired performance. One of these tools is the feedback controller tuning via loop-shaping. Use of the loop-shaping strategy in a nonlinear robust control design would certainly help in the realization of the given control requirements. One opportunity is to apply the loop-shaping in tuning a nonlinear sliding-mode controller suggested in [61]. Such a tuning should be facilitated by the particular structure of this sliding-mode controller. Additionally, this controller promises robust and chattering-free control despite its sampled time implementation, while its use in robotics is still not investigated. Hence, would it be worth to verify if it is possible to tune the given sliding-mode controller for the desired performance using the loop-shaping strategy, and to examine if this controller could be useful in robot motion control?

The primary objective of the robot servo motion control is to execute the desired robot tip trajectory in the task space. If the robot dynamics are rigid enough, accurate realization of the reference in the joint space should result in accurate tip motion, although one should also keep in mind nonlinear relation between the position accuracies in the two spaces. The influence of the flexible dynamics can be rejected in the joint space to the extent they are exhibited in the measurements obtained from the position sensors in the joints. If the flexibilities are so profound, it is common in robotics to extend the sensory information with the measurements of the actual tip position. This enhances performance robustness against the flexibilities. The major challenge here is how to adequately combine information obtained from the different sources when designing robust motion controllers. Robotics research already offers some solutions, that typically require intimate knowledge of the robot dynamic models with flexibilities. Derivation of the model equations based on first principles is well-known in theory, while obtaining accurate values of the model parameters is still a challenge because of high model complexity. An alternative could be data-driven modelling of flexibilities directly in the robot task space, and using these to supplement already available accurate models of the rigid-body dynamics with friction. Therefore, would it be useful to investigate if data-driven modelling in the robot task space can facilitate design of robust motion controllers in the presence of significant robot flexibility?

Real robots tend to exhibit various kinds of behaviours caused by flexibilities, varying friction, cogging force, disturbances, etc. The richness of such behaviours makes it hard task to capture them all with the data-driven modelling and thus take into account in the controller design. Capturing more effects increases the model complexity and in turn complexity of the model-based controllers. To come up with a controller of admissible complexity for practical implementation, one may resort to approximate modelling, which understands neglecting some system behaviours that itself might be of importance for the control performance. Another possibility is to apply model reduction on the controller of high complexity, such as to facilitate its practical implementation with minimal performance degradation. Accurate modelling, adequate model-based controller design, and eventually careful model reduction, are the steps that promise successful achievement of the given control requirements, but at the cost of an additional effort from the control designer. It is possible that one cannot afford such an effort, because of time restrictions and/or limited opportunities to organize experiment on which the control design will be based so that the maximum amount of information about the robot dynamics is acquired. The latter problem arises if the robot is already applied in some task, and its isolation for identification experiments would have unacceptable economical consequences. On the other hand, failures in the modelling, controller design, and/or controller's model reduction, may lead to significant performance degradation, even instability. Therefore, would it be useful to consider an alternative approach for model-based design of robot motion controllers?

Data-based control design strategies aim at directly finding the controllers, without an intermediate identification step. These strategies utilize data observed from the considered system. One of these, namely, iterative learning control (ILC) is utilized in robotics for improving the motion performance along the repetitive trajectories. What can be observed, though, is that the majority of robotics literature recommends ILC when a little knowledge about the actual robot dynamics is available. On the other hand, experience with linear motion systems shows that integrating more knowledge about the system dynamics into the ILC design promises better motion performance and even quantitative prediction of the performance improvement. These benefits are rarely recognized in the robotics literature on ILC, although they are of high value regarding the control requirements formulated at the beginning of this section. References [130,131] are just isolated exceptions that promote ILC design based on more intimate knowledge of the robot dynamics for the sake of better performance. Still, these references are restricted to geared robots with high reduction ratios and do not explore combination of dynamic compensation of the robot nonlinearities and ILC. Having these in mind, the question arises if it is possible to realize all the requirements via ILC, especially if direct-drive robots are the controlled plants.

An ILC design derives the control input as a time-domain signal, and its applicability is restricted to repetitive motion tasks. Data-based designs of feedback controllers are also available. In general, the designer can choose the structure and complexity of the controller, while a data-based algorithm provides optimal controller tuning with respect to given control specifications and information about the plant dynamics contained in the measured data. Among several data-based strategies surveyed in subsection 1.4.2, iterative feedback tuning and virtual reference feedback tuning (VRFT) appear as particularly interesting for application in robotics.

The iterative feedback tuning (IFT) enables optimal tuning along the prescribed motion references, even if very little information about the plant dynamics is available. Lack of information about the robot dynamics is typical for industrial robots, and, therefore, their feedback controllers are usually of lower complexity (conventional PID, PD, PI, etc.) to facilitate tuning by trial end error. Despite resorting to minimum a priori information about the plant, IFT has a strong theoretical basis which makes it a more structured approach to tune the conventional controllers, but also controllers of higher complexity that enable realization of more versatile control objectives. To date, no experimental results on IFT in robotics have been reported in the literature. Therefore, it would be instructive to experimentally examine IFT of robot feedback controllers of different complexities, in order to gain experience about practical utility of this data-based strategy for robot motion control.

It is already emphasized that IFT is mainly aimed for controller tuning along the prescribed references, and the literature on IFT suggests that optimal tunings are not necessary identical if the reference inputs are different. A feedback controller is effective for larger classes of reference motions, if it is tuned along the trajectory which sufficiently excites all the plant dynamics (persistent excitation). It is theoretically possible to perform IFT along the persistently exciting trajectories, but in practice this might require a huge effort, because of weak guarantees that the outcome of an IFT algorithm will be stabilizing for the plant. The IFT algorithm optimizes a nonconvex objective function, and without any knowledge of the plant dynamics no stability guaranties can be given. The practical experience with IFT shows that stability is more likely endangered if the measured data are richer with the behaviours, which is exactly the case with persistent excitation, unless the initial guess at the controller tuning is very close to the optimal one [77]. It is in general difficult to get an adequate initial tuning, and the persistent excitation may increase the number of iterations before the optimum is reached. Thus, less exciting trajectories are preferable when feedback is tuned via IFT and no plant model is used to ascertain the stability.

Contrary to IFT, the reference feedback tuning (VRFT) performs optimization of a convex objective functions in one-shot, and its application is preferred if data are obtained along persistently exciting trajectories. Hence, the VRFT appears more appropriate for design of feedback controllers supposed to realize versatile motion references. Additionally, the VRFT methodology gives more freedom for performance specifications than IFT, as this is performed almost like in model-based loop-shaping control designs. Seemingly, VRFT theory offers a strong foundation for development of a data-based version of loop-shaping, and this possibility is worth of investigating in this thesis. If this would be shown possible, an advantage would be efficiency of the data-based design, since no intermediate plant modelling step would be needed. This efficiency is plausible in robotics, having in mind that the robots are multivariable plants involving a number of feedback loops to be designed. However, specification of feasible performance objectives in the VRFT methodology cannot be done without good insight into the plant dynamics, which is exactly the case with model-based loop-shaping designs. For at least compatible control performance with model-based designs, the VRFT needs the knowledge of the plant dynamics as required by these designs. Nevertheless, the VRFT optimizes a controller of prescribed structure and complexity for the desired performance while utilizing mere data observed from the plant. These data always exhibit more plant behaviours than captured by parametric descriptions of the plant dynamics used in model-based designs. Therefore, given the performance specifications, data-based loop-shaping should result in controller tuning which is optimized with respect to more versatile plant behaviours than the tuning obtained in the model-based design.

Having in mind the research questions posed so far, it can be concluded that contributions to robot motion control are possible. The literature survey in the previous chapter and the questions posed here, have illuminated the problems that need to be solved to actually reach the desired innovation. These can be covered with the following problem statement:

Find and explore suitable data-driven approaches that can contribute to the quality of robot motion control. These approaches must take into account as much as possible the actual robot dynamics, and they should allow flexible performance specifications, quantitative prediction of the control performance, compatibility with general robot configurations, and straightforward real-time implementation on modern control platforms.

Provide adequate control design procedures and test them experimentally on a direct-drive robot to examine their potential impact on robotics practice.

The next section announces a strategy to respond to the given problem statement. This strategy will be followed in the rest of the thesis.

2.4 The solution strategy

A suitable solution strategy to the problem statement formulated in the previous section requires balanced utilization of three components:

- fundamental theory,
- tools that implement the theory,
- methodology that describes how to use these tools to solve a particular motion control problem.

The key points in the strategy are:

1. Exploring the available systems and control theory for useful extensions to the robotics research field. Determining possibilities with the available theory, and tailoring the theory whenever required to yield a theoretical framework suitable for the robot motion control problem at hand.
2. Development of the required tools. Using the available software and hardware that can aid in the design and implementation of the motion controllers. Where required, development of custom tools.
3. Creating a synthesis of both theory and tools, aiming at as high as possible quality of robot motion control. Development of proper control design procedures. Demonstration of these procedures on the experimental set-up and assessment of their capabilities in experiments.

To facilitate each control design, a model-based compensation of robot nonlinearities will be applied. Since this compensation will employ models of the robot rigid-body dynamics and friction, accurate modelling of the robot kinematics and dynamics will be needed, together with choosing some relevant parametric description of friction effects. Parameters of the kinematic models will be determined by direct measurement on the robot, while the parameters of the rigid-body dynamic and friction models will be obtained by experimental estimation. Implementation of the computed-torque control law may require information about the actual positions, velocities, and accelerations in the robot joints. In this thesis, only position measurements will be used, while the other states will be reconstructed via Kalman filtering. For the purpose of model-based feedback control designs, data-driven modelling will be used to capture effects not covered with the models of the rigid-body dynamics and friction. In particular, the additional dynamics will be captured by frequency response function measurements. Table 2.1 summarizes the modelling concepts to be used.

Application of H_∞ and μ-synthesis feedback control designs will be examined in the robot motion control problem. These linear robust designs will be performed in the robot joint space, and they will be based on the data-driven models of the parasitic dynamics. In particular, H_∞ design will be used for optimal controller tuning via loop-shaping in an experimental iterative procedure. The purpose of the iterative procedure will be controller tuning for counteracting the disturbance and parasitic effects exhibited in the experiments. μ-synthesis will be performed non-iteratively. Its objective will be controller tuning for robust performance. The possibility of controller tuning via loop-shaping will be tested also in a nonlinear robust discrete-time sliding-mode control design. The performance of robust feedback control designs will be confronted with the performance of conventional PD feedback. A linear H_∞ controller will be designed in the robot task space, in order to realize robust attenuation of the vibrations at the tip of the robot. These vibrations arise due to profound flexibility of the robot structure.

TABLE 2.1: SUMMARY OF THE MODELLING CONCEPTS

Modelling concept	Purpose
Kinematics	relation between the motion coordinates from the joint and task spaces of the robot
Rigid-body dynamics	model-based compensation of the robot nonlinear dynamics with the computed-torque control law
Friction	model-based fiction compensation
Kalman filtering	reconstruction of actual velocities and accelerations in the robot joints based on the position measurements
Frequency response measurements	capturing the parasitic robot dynamics

Opportunities to accomplish robot control design without direct use of parametric models of the additional dynamics will be explored in data-based control designs. Iterative learning control will be used to derive optimal feedforward components of the control inputs in the repetitive motion tasks. Given a motion reference, iterative feedback tuning will be used for optimal feedback control tuning along this reference. This tuning will be based on mere data observed from the robot, and no model of the robot additional dynamics will be utilized. A new data-based control design which enables simultaneous shaping of the closed-loop sensitivity and complementary sensitivity functions will be proposed. This design will enable specification of the desired performance as in model-based loop-shaping control designs. Specification of a feasible performance, consistent with the limits imposed by the physical realizability, will need the knowledge of the additional dynamics.

Table 2.2 summarizes the control designs to be examined in this thesis. Each of these designs will be used together with model-based compensation of robot nonlinearities.

TABLE 2.2: SUMMARY OF MOTION CONTROL DESIGNS

Design	Motivation
Conventional PD feedback	model-based design of low complexity
H_∞ feedback	model-based iterative tuning for the desired performance
μ-synthesis feedback	model-based tuning for robust performance
Discrete-time sliding-mode feedback	model-based chattering-free sliding-mode control
Task space H_∞ feedback	model-based robust attenuation of tip vibrations for robots exhibiting profound flexibility
Iterative learning control	performance improvement along repetitive motion tasks
Iterative feedback tuning	iterative model-free optimal feedback tuning along prescribed motion reference
Data-based loop-shaping feedback tuning	non-iterative optimal design of restricted complexity feedback controllers

All items addressed in this section will be worked-out in detail in the rest of the thesis.

Chapter 3

Robot modelling and identification

Accurate and fast motions of robot manipulators can be realized using control schemes that employ models of the robot kinematics and dynamics. This chapter presents the steps preceding the design of model-based motion controllers: (*i*) modelling of robot kinematics and rigid-body dynamics with friction, (*ii*) estimation of the model parameters, (*iii*) model validation, and, (*iv*) identification and modelling of *additional* robot dynamics not covered by the rigid-body dynamic model with friction. Quality achieved in each of these steps contributes to the performance of motion control. Literature offers different methods that could be used in the considered steps. Some of these methods require complex designs of identification experiments and advanced algorithms for parameter estimation, in order to reduce influences of parasitic and disturbance effects. High complexity is not appealing for wider acceptance of such methods among control engineers in robotics. Therefore, this chapter tends to offer much simpler strategy for parameter estimation, which is still capable to deliver estimates of sufficient quality. Other contributions of this chapter are the recognition of the dependence between the additional robot dynamics and closed-loop performance, as well as the suggestion of a particular data-driven modelling procedure to capture these additional dynamics. Theories underlying each method considered in the chapter are well-known and are sufficiently elaborated in the literature. However, particular selection of the methods and their effective coupling are hardly seen elsewhere in the robotics literature. The chapter is organized as follows. Sections 3.1 presents a standard method for kinematic modelling of a robot using the Denavits-Hartenberg's convention. This convention is also used in the approach to model the rigid-body robot dynamics in Section 3.2. Friction modelling is presented in section 3.3. Estimation of friction and inertial parameters of the robot dynamics, and design of the identification experiments are discussed in section 3.4. A Kalman filtering technique to achieve joint motions, velocities, and acceleration suitable for parameter estimation is presented in section 3.5. Use of a writing task for model validation is suggested in section 3.6. Identification and modelling of the robot dynamics not covered with the rigid-body model with friction is explained in section 3.7. The chapter summary is given at the end.

3.1 Kinematic modelling

Consider a serial robot manipulator with *n*-joints, shown in Fig. 3.1. A minimal kinematic parameterisation for this manipulator can be established according to the well-known Denavits-Hartenberg's (DH) convention [53,153]. This convention systematically assigns coordinate frames to the robot joints and induces the following DH parameters: twist angles α_i, link lengths a_i, joint displacements q_i, and link offsets d_i, where $i = 1,\ldots,n$. Joint *i* has assigned the coordinate frame *i*-1. The position and orientation of the i^{th} coordinate frame with respect to the previous one (*i*-1) can be specified by the homogenous transformation matrix:

$$ {}^{i-1}_{\;\;i}\mathbf{T}(q_i) = \begin{bmatrix} {}^{i-1}_{\;\;i}\mathbf{O}(q_i) & {}^{i-1}_{\;\;i}\mathbf{x}(q_i) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{3.1} $$

The skew-symmetric rotation matrix ${}^{i-1}\mathbf{O}_i \in \mathbb{R}^{3\times 3}$ defines the orientation of frame '*i*'with respect to frame ‘*i*-1’:

$$ {}^{i-1}_{i}\mathbf{O}(q_i) = \begin{bmatrix} \cos q_i & -\cos\alpha_i \sin q_i & \sin\alpha_i \sin q_i \\ \sin q_i & \cos\alpha_i \cos q_i & -\sin\alpha_i \cos q_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix}. \tag{3.2} $$

The position vector ${}^{i-1}_{i}\mathbf{x} \in \mathbb{R}^3$ points from the origin of the frame ‘*i*-1’ to frame ‘*i*’:

$$ {}^{i-1}_{i}\mathbf{x}(q_i) = \begin{bmatrix} a_i \cos q_i & a_i \sin q_i & d_i \end{bmatrix}^T . \tag{3.3} $$

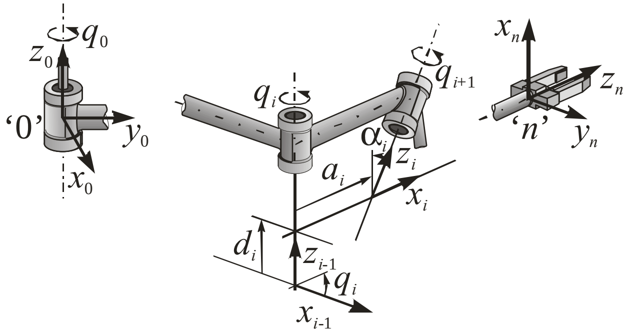


Figure 3.1: A general serial robotic manipulator with *n* joints (taken from [153]).

Let $\mathbf{q}$ denote the vector of generalized coordinates, i.e., linear and angular displacements of prismatic and rotational joints, respectively:

$$ \mathbf{q} = [q_1 \; q_2 \ldots \; q_n]^T . \tag{3.4} $$

Then, the forward kinematics can be computed as a product of homogeneous transformations between the adjacent coordinate frames:

$$ {}^{0}_{n}\mathbf{T}(\mathbf{q}) = {}^{0}_{1}\mathbf{T}(\mathbf{q}) {}^{1}_{2}\mathbf{T}(\mathbf{q}) \ldots {}^{n-1}_{n}\mathbf{T}(\mathbf{q}) = \begin{bmatrix} {}^{0}_{n}\mathbf{O}(\mathbf{q}) & {}^{0}_{n}\mathbf{x}(\mathbf{q}) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}. \tag{3.5} $$

The orientation and position of the tip frame ‘*n*’ with respect to the base (inertial) frame ‘0’ are determined by ${}^{0}_{n}\mathbf{O}$ and ${}^{0}_{n}\mathbf{x}$, respectively. In the general case, both ${}^{0}_{n}\mathbf{O}$ and ${}^{0}_{n}\mathbf{x}$ depend nonlinearly on the generalized coordinates, and thus it is not always possible to explicitly express $\mathbf{q}$ in terms of the tip position and orientation coordinates. Consequently, there is no general closed-form representation of the inverse kinematics (IK), and numerical techniques are often used to solve IK [153].

Rather than computing $\mathbf{q}$ given the mapping (3.5), the IK is often solved starting from a differential representation of the manipulator kinematics [153]:

$$ \begin{aligned} \begin{bmatrix} {}^{0}_{n}\dot{\mathbf{x}} \\ {}^{0}_{n}\boldsymbol{\omega} \end{bmatrix} &= \begin{bmatrix} {}^{0}_{n}\mathbf{J}_P(\mathbf{q}) \\ {}^{0}_{n}\mathbf{J}_O(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}} \\ &= {}^{0}_{n}\mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \end{aligned} \tag{3.6} $$

where ${}^{0}_{n}\mathbf{J}_P \in \mathbb{R}^{3\times n}$ and ${}^{0}_{n}\mathbf{J}_O \in \mathbb{R}^{3\times n}$ are the matrices relative to the contributions of the joint velocities $\dot{\mathbf{q}}$ to the end-effector linear velocity ${}^{0}_{n}\dot{\mathbf{x}}$ and angular velocity ${}^{0}_{n}\boldsymbol{\omega}$, respectively. The matrices ${}^{0}_{n}\mathbf{J}_P$ and ${}^{0}_{n}\mathbf{J}_O$ form the manipulator geometric Jacobian ${}^{0}_{n}\mathbf{J} \in \mathbb{R}^{6\times n}$ and can be computed as follows:

$$
\begin{aligned}
{}_n^0\mathbf{J}_P &= [{}_1^0\mathbf{j}_P \ldots {}_n^0\mathbf{j}_P],\; {}_n^0\mathbf{j}_P \in \mathbb{R}^3 \text{ and } {}_i^0\mathbf{j}_P = \begin{cases} {}_{i-1}^{\;\;0}\mathbf{o} & \text{for a prismatic joint} \\ {}_{i-1}^{\;\;0}\mathbf{o} \times ({}_n^0\mathbf{x} - {}_{i-1}^{\;\;0}\mathbf{x}) & \text{for a rotational joint,} \end{cases} \\
{}_n^0\mathbf{J}_O &= [{}_1^0\mathbf{j}_O \ldots {}_n^0\mathbf{j}_O],\; {}_i^0\mathbf{j}_O \in \mathbb{R}^3 \text{ and } {}_i^0\mathbf{j}_O = \begin{cases} \mathbf{0}_{3\times 1} & \text{for a prismatic joint} \\ {}_{i-1}^{\;\;0}\mathbf{o} & \text{for a rotational joint.} \end{cases}
\end{aligned}
\tag{3.7}
$$

Here, ${}_{i-1}^{\;\;0}\mathbf{o}$ is the unit vector of joint *i* axis (frame ‘*i*-1’, see Fig. 3.1) expressed in the base frame ‘0’. This vector is given by the third column of the orientation matrix ${}_{i-1}^{\;\;0}\mathbf{O}$:

$$
{}_{i-1}^{\;\;0}\mathbf{o} = {}_1^0\mathbf{O}(q_1) \ldots {}_{i-1}^{i-2}\mathbf{O}(q_{i-1})^0\mathbf{o}\,, \text{ with } {}^0\mathbf{o} = [0\;\;0\;\;1]^T\,. \tag{3.8}
$$

The mapping (3.6) is called the differential kinematics of first order. The second order differential kinematics is obtained by taking the time derivative of (3.6):

$$
\begin{bmatrix} {}_n^0\ddot{\mathbf{x}} \\ {}_n^0\dot{\boldsymbol{\omega}} \end{bmatrix} = {}_n^0\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + {}_n^0\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}, \tag{3.9}
$$

where $\ddot{\mathbf{q}}$ is the vector of joint accelerations and ${}_n^0\dot{\mathbf{J}}$ is the time derivative of the geometric Jacobian. The differential kinematic models facilitate the IK solution:

$$
{}_n^0\dot{\mathbf{q}} = {}_n^0\mathbf{J}^{\#}(\mathbf{q}) \begin{bmatrix} {}_n^0\dot{\mathbf{x}} \\ {}_n^0\boldsymbol{\omega} \end{bmatrix} \text{ or } \ddot{\mathbf{q}} = {}_n^0\mathbf{J}^{\#}(\mathbf{q}) \left(\begin{bmatrix} {}_n^0\ddot{\mathbf{x}} \\ {}_n^0\dot{\boldsymbol{\omega}} \end{bmatrix} - {}_n^0\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \right), \tag{3.10}
$$

where ‘$^{\#}$’ denotes a matrix pseudoinverse [62]. If the number of the joints coincides with the number of the end-effector coordinates, then ${}_n^0\mathbf{J}$ is a square and a normal matrix inverse is used in (3.10) instead of the pseudoinverse. The differential IK models (3.10) enable direct computation of joints velocities or accelerations, while the joint motions should be found by numerical integration. The IK algorithms implementing the solutions (3.10) are normally enhanced with additional feedback loops that counteract numerical errors caused by integration [153].

3.2 Dynamic rigid-body modelling

A general dynamics of the robot in standard form is given by equation (2.16) in section 2.2. This general presentation includes the rigid-body dynamics, friction, and a collection of additional effects, e.g., flexibilities and external disturbances. The part of the model (2.16) representing the rigid-body dynamics can be derived using the Euler-Lagrange formalism [153].[53,100,181] The dynamic rigid-body model with friction is repeated here for convenience:

$$
\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{c}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}^f(\mathbf{q}(t), \dot{\mathbf{q}}(t), t) = \boldsymbol{\tau}(t)\,. \tag{3.11}
$$

In the given model, all variables have identical meaning as in (2.16), with $\mathbf{q}$ defined by (3.4). Elements of the inertia matrix $\mathbf{M}$ and of the vectors of Coriolis/centripetal $\mathbf{c}$ and gravity $\mathbf{g}$ effects, can be determined using the homogenous transformation (3.1):

$$
\begin{aligned}
m_{i,k} &= \sum_{j=\max(i,k)}^{n} \mathrm{Tr}\left[\left(\frac{\partial}{\partial q_k} {}_j^0\mathbf{T} \right) \mathbf{J}_j \left(\frac{\partial}{\partial q_i} {}_j^0\mathbf{T} \right)^T \right],\; c_i = \sum_{k=1}^{n}\sum_{l=1}^{n} c_{i,k,l}\dot{q}_k\dot{q}_l\,, \\
c_{i,k,l} &= \sum_{j=\max(i,k,l)}^{n} \mathrm{Tr}\left[\left(\frac{\partial^2}{\partial q_k \partial q_l} {}_j^0\mathbf{T} \right) \mathbf{J}_j \left(\frac{\partial}{\partial q_i} {}_j^0\mathbf{T} \right)^T \right], \\
g_i &= \sum_{j=i}^{n} \left(-m_j [0\;\;0\;-\mathrm{g}\;\;0] \left(\frac{\partial}{\partial q_i} {}_j^0\mathbf{T} \right) \mathbf{r}_j \right),\; i,k,l = 1,\ldots,n\,.
\end{aligned}
\tag{3.12}
$$

Here, m_j is the mass of link j, g is the acceleration due to gravity, $\mathbf{r}_j$ contains the homogenous coordinates of the center of mass of link j related to the coordinate frame 'j'

$$\mathbf{r}_j = [x_j \;\; y_j \;\; z_j \;\; 1]^T , \tag{3.13}$$

while $\mathbf{J}_j$ is the inertia tensor with elements:

$$\begin{gathered} j_{j,11} = \frac{-I_{xx,j} + I_{yy,j} + I_{zz,j}}{2} ; \; j_{j,22} = \frac{I_{xx,j} - I_{yy,j} + I_{zz,j}}{2} ; \\ j_{j,33} = \frac{I_{xx,j} + I_{yy,j} - I_{zz,j}}{2} ; \; j_{j,12} = j_{j,21} = I_{xy,j} ; \; j_{j,13} = j_{j,31} = I_{xz,j} ; \\ j_{j,23} = j_{j,32} = I_{yz,j} ; \; j_{j,14} = j_{j,41} = m_j x_j ; \; j_{j,24} = j_{j,42} = m_j y_j ; \\ j_{j,34} = j_{j,43} = m_j z_j ; \; j_{j,44} = j_{j,44} = m_j . \end{gathered} \tag{3.14}$$

Here, $I_{xx,j}$, $I_{yy,j}$ and $I_{zz,j}$ are the principal moments of inertia of link j, and $I_{xy,j}$, $I_{yz,j}$ and $I_{xz,j}$ are products of inertia of the same link.

The dynamic rigid-body model (3.11), excluding friction, can be represented linearly in elements of the so-called base parameter set (BPS). These elements are combinations of the inertial parameters used in the equations above, and they constitute the minimum set of parameters whose values can determine the dynamic model uniquely [12,32,53,57,58,100,121,133,153,170]. A linear parameterisation of the model (3.11) has the form:

$$\mathbf{R}\big(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t)\big)\mathbf{p} + \boldsymbol{\tau}^f\big(\mathbf{q}(t), \dot{\mathbf{q}}(t), t\big) = \boldsymbol{\tau}(t) , \tag{3.15}$$

where $\mathbf{R} \in \mathbb{R}^{n \times p}$ is the regression matrix, and $\mathbf{p} \in \mathbb{R}^p$ is the vector of the BPS elements. The regressor form of the dynamic model is suitable for estimation of the inertial parameters. According to reference [121], a robot with rotational joints only has $7n - 4\ell$ BPS elements, where n is the number of joints and ℓ is the number of robot links connected by joints whose axes are always parallel to the first joint (the $(\ell + 1)^{\text{th}}$ joint axis is the first that is perpendicular to the predecessor). If the first joint is parallel to the gravity vector, then the number of BPS elements is $7n - 4\ell - 2$.

3.3 Friction modelling

The dynamic models (3.11) and (3.15) also feature the friction term $\boldsymbol{\tau}^f$. In a real system, friction cannot be disregarded if high-performance motion control is required. Hence, modelling and compensation of friction effects are important issues in robot motion control. A basic friction model covers just Coulomb and viscous effects:

$$\alpha_{1,i} \operatorname{sgn} \dot{q}_i + \alpha_{2,i} \dot{q}_i = \tau_i^f \;\; (i = 1, \ldots, n), \tag{3.16}$$

where τ_i^f is the i^{th} element of $\boldsymbol{\tau}^f$, while $\alpha_{1,i}$ and $\alpha_{2,i}$ are Coulomb and viscous friction parameters, respectively. The given model admits a linear representation in the parameters $\alpha_{1,i}$ and $\alpha_{2,i}$, and this enables a regressor representation of the dynamic model (3.15), where the new regression matrix is formed by vertical stacking $\mathbf{R}$ and the rows containing the friction terms, while the parameter vector consists of $\mathbf{p}$ and friction parameters. The regressor form allows simultaneous estimation of the BPS and the friction parameters [32,170].

However, friction is a very complex interaction between the contact surfaces of the interacting objects, whose description may require a more comprehensive model than (3.16) [10,11,35,44,72,73,105,134]. Presliding and sliding are two commonly recognized friction regimes. The level of friction prior to the sliding regime is called static friction, or stiction. At the beginning of the sliding regime, the decreasing friction force for increasing velocity is known as the Stribeck effect. Beyond this stage, the friction force increases as the velocity is increasing, which is caused by the viscous effect. A time lag between the velocity and friction force can be detected, which implies a dynamic, rather than just static nature of the friction. Several static and dynamic models were proposed to represent the given properties of friction. The models that are relevant in this thesis are presented next.

A frequently used static friction model has the form [11,134]:

$$\gamma(\dot{q}_i)\operatorname{sgn}\dot{q}_i + \alpha_{2,i}\dot{q}_i = \tau_i^f\,;$$
$$\gamma_i(\dot{q}_i) = \alpha_{0,i}\exp\left(-(\dot{q}_i / v_{s,i})^2\right) + \alpha_{1,i}\ \ (i = 1,\ldots,n), \tag{3.17}$$

where $\gamma(\dot{q}_i)$ is the Stribeck curve for steady-state velocities, $v_{s,i}$ is the Stribeck velocity, and $\alpha_{0,i}$, $\alpha_{1,i}$ and $\alpha_{2,i}$ are static, Coulomb, and viscous friction parameters, respectively. A typical friction curve represented by the model (3.17) is depicted in Fig. 3.2. The stiction, Stribeck, Coulomb, and viscous friction effects are clearly visible in the figure. The model describes the sliding friction regime. Because of the parameter $v_{s,i}$, the model (3.17) cannot be represented linearly in all friction parameters ($v_{s,i}$, $\alpha_{0,i}$, $\alpha_{1,i}$, and $\alpha_{2,i}$). Thus, a regressor form of a robot dynamic model is not possible with the model (3.17). This prevents the simultaneous estimation of the friction parameters and the BPS elements, using, for example, some time-efficient least-squares technique.

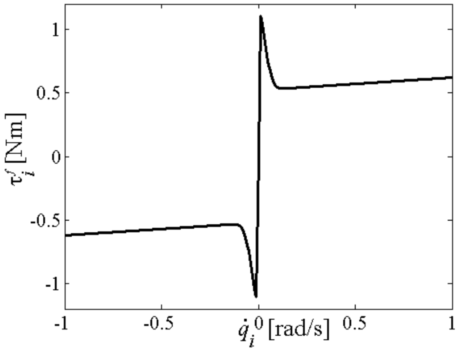


Figure 3.2: A typical friction curve corresponding to the friction model (3.17).

Dynamic friction models aim at covering more versatile friction phenomena. Among others (see surveys in [11,134]), the Dahl and the LuGre models are currently the most frequently cited, analysed, and implemented dynamic models. They cover a number of static and dynamic friction phenomena and facilitate their representation in a compact way. The LuGre model is more comprehensive than the Dahl's model and has the following form:

$$\sigma_{0,i}z_i + \sigma_{1,i}\dot{z}_i + \alpha_{2,i}\dot{q}_i = \tau_i^f\,;$$
$$\dot{z}_i = \dot{q}_i - \sigma_{0,i}\frac{|\dot{q}_i|}{\gamma(\dot{q}_i)}z_i\ \ (i = 1,\ldots,n), \tag{3.18}$$

where z_i is a pre-sliding deflection of materials in contact, $\gamma(\dot{q}_i)$ is defined by (3.17), while $\sigma_{0,i}$ and $\sigma_{1,i}$ are the stiffness of` pre-sliding deflections and viscous damping of the deflections, respectively. Notice that the static model (3.17) corresponds to the sliding behaviour ($\dot{z}_i = 0$) of the LuGre model. A linear parameterisation of the LuGre model is obviously not possible, and, consequently, the regressor form of a dynamic model cannot be created.

A three-sigmoidal-functions friction model is also used [72,96,97]:

$$\sum_{k=1}^{3} f_{i,k}[1-2/(\exp(2w_{i,k}\dot{q}_i)+1)]+\alpha_{2,i}\dot{q}_i = \tau_i^f \;\; (i=1,\dots,n), \tag{3.19}$$

where $f_{i,k}$, $w_{i,k}$, and $\alpha_{2,i}$ are unknown parameters. This static model can represent Stribeck, Coulomb, and viscous effects. However, it cannot be linearly parameterised, and, consequently, the regressor form of a dynamic model is not possible with this model.

When the robot model is completed, the next step is to obtain realistic values for the model parameters. These parameters are acquired via estimation procedures, that are explained in the next section.

3.4 Estimation of model parameters

Robot dynamics should be identified under closed-loop operating conditions, to avoid problems because of inherent instability of these dynamics. A typical drawback of closed-loop identification is the mutual correlation of disturbance components in the excitation and response signals. This correlation can cause a biased result in the identification [113]. Nowadays, there is a good understanding of bias and variance issues for linear systems identified in closed-loop. Unfortunately, these issues have still restrictive interpretations if nonlinear dynamics are identified under closed-loop conditions. In the nonlinear case, low presence of noise and disturbances in the closed-loop system is typically assumed. This assumption reduces the correlation problem, but increases the conservatism of the related bias and variance analysis. Whenever possible, open-loop identification experiments will be used in this thesis to avoid problems of closed-loop identification. Where necessary, identification will be performed in closed-loop, but with adequate precaution against the effect of disturbances in the loop.

In this section, methods for estimation of the friction parameters are explained. Friction parameters are estimated separately from the BPS elements if the regressor form of a robot dynamic model cannot be created. Then, two methods for the estimation of the BPS elements are described.

3.4.1 Friction parameters

Friction is an individual characteristic of each robot joint. Having no coupling effect with other joints, friction can be analysed and identified for every joint separately. Although less time-efficient, separate friction identification gives the opportunity to use open-loop identification experiments, which contributes to the quality of estimation. Unknown parameters of a friction model associated with joint *i* can be collected in the vector $\mathbf{p}_i^f \in \mathbb{R}^{s_i}$. In the experiment, the joint is driven with a sufficiently exciting input τ_i, and the corresponding motion q_i is measured. The remaining joints are kept locked by means of appropriate hardware or by feedback control. In the latter case, the feedback controllers must be tuned such that the motions in the remaining joints, possibly caused by the couplings effects and by gravity, are prevented. Here, an identification procedure will be described for a serial robot manipulator with rotational joints only. Extension to robots with prismatic joints is straightforward. If only joint *i* is allowed to move, then the dynamics of the moving part of the robot mechanism can be modelled as an inertial system influenced by gravity. Euler-Lagrange modelling method results in the following equation:

$$\ddot{q}_i = [\tau_i - \tau_i^f(q_i,\dot{q}_i,t,\mathbf{p}_i^f) - c_i \sin q_i]/J_i, \tag{3.20}$$

where J_i is the equivalent moment of inertia of the joint, c_i is a coefficient of the gravity term, and $\tau_i^f(q_i,\dot{q}_i,t,\mathbf{p}_i^f)$ is some adopted friction model. The term c_i vanishes if the joint axis is parallel with the direction of gravity. The parameters J_i and c_i are usually not known exactly, so they need to be estimated together with $\mathbf{p}_i^f$.

An effective time-domain identification procedure elaborated in [72] can be carried-out to estimate the parameters characteristic to the sliding regime of $\tau_i^f(q_i, \dot{q}_i, t, \mathbf{p}_i^f)$, together with J_i and c_i. This procedure resorts in a recursive extended Kalman filtering (EKF) technique for nonlinear continuous-time processes. Let $\tau_i^{f,sr}(q_i, \dot{q}_i, t, \mathbf{p}_i^f)$ represent the sliding regime of the adopted friction model. A state-space representation of (3.20), which corresponds to the sliding friction regime, is

$$\begin{aligned}\dot{\mathbf{x}}_i^{f,sr}(t) &= \boldsymbol{\mu}_i\left(\mathbf{x}_i^{f,sr}(t)\right) + \varsigma_i \tau_i(t) + \mathbf{w}_i(t);\\ \mathbf{y}_i^{f,sr}(t) &= q_i(t) + \eta_i(t)\\ &= (\mathbf{c}_i^f)^T \mathbf{x}_i^{f,sr}(t) + \eta_i(t),\end{aligned} \tag{3.21a}$$

with

$$\mathbf{x}_i^{f,sr} = \begin{bmatrix} q_i \\ \dot{q}_i \\ \mathbf{p}_i^f \\ J_i \\ c_i \end{bmatrix};\ \boldsymbol{\mu}_i = \begin{bmatrix} 0 \\ -\left(\tau_i^{f,sr}(q_i, \dot{q}_i, t, \mathbf{p}_i^f) + c_i \sin q_i\right)/J_i \\ \mathbf{0}_{s_i\times 1} \\ 0 \\ 0 \end{bmatrix};\ \varsigma_i = \begin{bmatrix} 0 \\ 1 \\ \mathbf{0}_{s_i\times 1} \\ 0 \\ 0 \end{bmatrix};$$

$$(\mathbf{c}_i^f)^T = [1\ 0\ \mathbf{0}_{1\times s_i}\ \ 0\ 0]. \tag{3.21b}$$

Here, $\mathbf{w}_i \in \mathbb{R}^{4+s_i}$ is a zero mean Gaussian process noise with covariance matrix $\mathbf{Q}_i(t) \in \mathbb{R}^{(4+s_i)\times(4+s_i)}$, and $\eta_i \in \mathbb{R}$ is a zero mean white measurement noise with covariance $r_i(t) \in \mathbb{R}$. The process and measurement errors are assumed uncorrelated. In the state-space model (3.21a,b), the unknown parameters $\mathbf{p}_i^f$, J_i, and c_i are treated as constants. These unknowns are determined recursively, by the EKF procedure [59],[72],[97] which uses the excitation τ_i and the response q_i measured at discrete-time instants $t_k \in \{t_1, \ldots, t_f\}$. The update law at t_k is

$$\hat{\mathbf{x}}_i^{f,sr}(t_k)^+ = \hat{\mathbf{x}}_i^{f,sr}(t_k)^- + \mathbf{k}_i^f(t_k)[y_i(t_k) - (\mathbf{c}_i^f)^T \hat{\mathbf{x}}_i^{f,sr}(t_k)^-] \tag{3.22}$$

where '^' denotes estimate, while '$^-$' and '$^+$' denote initial and updated estimates at t_k, respectively. The vector gain $\mathbf{k}_i^f \in \mathbb{R}^{4+s_i}$ is computed by:

$$\mathbf{k}_i^f(t_k) = \mathbf{P}_i^f(t_k)^- \mathbf{c}_i^f / [(\mathbf{c}_i^f)^T \mathbf{P}_i^f(t_k)^- \mathbf{c}_i^f + r_i(t_k)], \tag{3.23}$$

while the covariance matrix $\mathbf{P}_i^f \in \mathbb{R}^{(4+s_i)\times(4+s_i)}$ is updated with

$$\mathbf{P}_i^f(t_k)^+ = [\mathbf{I}_{4+s_i} - \mathbf{k}_i^f(t_k)(\mathbf{c}_i^f)^T]\mathbf{P}_i^f(t_k)^-, \tag{3.24}$$

where $\mathbf{I}_{4+s_i} \in \mathbb{R}^{(4+s_i)\times(4+s_i)}$ is the identity matrix. Between t_k and t_{k+1}, the state estimate and error covariance are propagated by integration of:

$$\begin{aligned}\dot{\hat{\mathbf{x}}}_i^{f,sr}(t) &= \boldsymbol{\mu}_i\left(\hat{\mathbf{x}}_i^{f,sr}(t)\right) + \varsigma_i \tau_i(t);\\ \dot{\mathbf{P}}_i^f(t) &= \mathbf{M}_i^f\left(\hat{\mathbf{x}}_i^{f,sr}, \tau_i(t)\right)\mathbf{P}_i^f(t) + \mathbf{P}_i^f(t)\left(\mathbf{M}_i^f\left(\hat{\mathbf{x}}_i^{f,sr}, \tau_i(t)\right)\right)^T + \mathbf{Q}_i(t).\end{aligned} \tag{3.25}$$

The matrix $\mathbf{M}_i^f \in \mathbb{R}^{(4+s_i)\times(4+s_i)}$ is the Jacobian of the nonlinear vector function $\boldsymbol{\mu}_i$. The EKF procedure (3.22)-(3.25) requires an initial guess of $\hat{\mathbf{x}}_i^{f,sr}(t_0)$. The confidence or error variance is expressed by the initial diagonal covariance matrix $\mathbf{P}_i^f(t_0)$. Its non-zero elements on the main diagonal represent uncertain initial state estimates, while zero diagonal elements represent infinite confidence on the initial estimates. The EKF procedure updates the estimates, until convergence of all unknown parameters is reached.

If the LuGre mode (3.18) is adopted for $\tau_i^f(q_i, \dot{q}_i, t, \mathbf{p}_i^f)$, then the parameters characteristic to the pre-sliding friction regime, i.e., $\sigma_{0,i}$ and $\sigma_{1,i}$, can be estimated using frequency response function (FRF) measurements [73],[97]. This procedure requires an experiment where the considered joint is driven with a random excitation (noise), and the FRF from the excitation τ_i to the motion q_i is measured. A low noise level is needed, so as to keep friction within the pre-sliding regime. Joint position sensors of high resolution are thus necessary, since very small displacements have to be detected. The estimation procedure employs a linearisation of (3.18) around zero $\dot{q}_i$, z_i, and $\dot{z}_i$, defined in Laplace domain:

$$H_i(s) = \frac{Q_i(s)}{\tau_i(s)} = \frac{1}{J_i s^2 + (\sigma_{1,i} + \alpha_{2,i})s + \sigma_{0,i} + c_i}. \tag{3.26}$$

The parameters J_i, $\alpha_{2,i}$, and c_i must be known beforehand. The transfer function (3.26) is fitted on the measured FRF to determine the unknown $\sigma_{0,i}$ and $\sigma_{1,i}$.

3.4.2 Base Parameter Set

Elements of the BPS are estimated using identification experiments on the robot. During the experiments, the robot is normally controlled with decoupled PD (proportional, derivative) controllers to realize a specific trajectory. This trajectory must be carefully chosen such that the applied input $\boldsymbol{\tau}(t)$ sufficiently excites the system dynamics. If the applied $\boldsymbol{\tau}(t)$ and the resulting $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$, and $\ddot{\mathbf{q}}(t)$ are collected in the experiment, and post-processed to determine elements of the BPS, then the estimation is done batch. If the estimation of the BPS is performed as the identification experiment is going on, then we speak about online estimation. The online estimation is more efficient than the batch estimation, but it requires powerful computer hardware for robot control and data processing. The online estimation is also needed for model-based motion control of robots with frequently changing loads. Two estimation algorithms, namely, a standard batch least-squares and an online one, are presented in the next part. These algorithms also indicate two possibilities for dealing with friction, i.e., identification of friction separately from the BPS, and the simultaneous estimation of the BPS and the friction parameters.

3.4.2.1 Batch least-squares estimation

Assume that already available friction models associated with the robot joints reconstruct real friction with sufficient accuracy. Then, the term $\boldsymbol{\tau}^f$ is completely determined in a robot dynamic rigid-body model (3.11) and in the corresponding regressor representation (3.15). Let ζ samples of each element of $\boldsymbol{\tau}$, $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ have been collected during the identification experiment. These samples correspond to time instants $t_1, t_2, \ldots, t_\zeta$. By virtue of (3.15), we may form the following system of equations:

$$\boldsymbol{\Phi} \cdot \mathbf{p} = \mathbf{y}, \tag{3.27}$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathbf{R}(\mathbf{q}(t_1), \dot{\mathbf{q}}(t_1), \ddot{\mathbf{q}}(t_1)) \\ \mathbf{R}(\mathbf{q}(t_2), \dot{\mathbf{q}}(t_2), \ddot{\mathbf{q}}(t_2)) \\ \vdots \\ \mathbf{R}(\mathbf{q}(t_\zeta), \dot{\mathbf{q}}(t_\zeta), \ddot{\mathbf{q}}(t_\zeta)) \end{bmatrix}, \tag{3.28}$$

$$\mathbf{y} = \left[\left(\boldsymbol{\tau}(t_1) - \boldsymbol{\tau}^f(t_1)\right)^T, \ \left(\boldsymbol{\tau}(t_2) - \boldsymbol{\tau}^f(t_2)\right)^T, \ldots, \ \left(\boldsymbol{\tau}(t_\zeta) - \boldsymbol{\tau}^f(t_\zeta)\right)^T \right]^T. \tag{3.29}$$

If the input $\boldsymbol{\tau}(t)$ sufficiently excites the manipulator dynamics, then the least-squares estimate of $\mathbf{p}$ is:

$$\hat{\mathbf{p}} = \boldsymbol{\Phi}^{\#}\mathbf{y}\,, \tag{3.30}$$

where '$^{\#}$' denotes a matrix pseudoinverse [62]. There are several criteria evaluating sufficiency of excitation. The usual one is the condition number of the information matrix $\boldsymbol{\Phi}$, defined as the ratio between maximal and minimal singular values of $\boldsymbol{\Phi}$. If this number is closer to one, the excitation is considered to be better for a reliable estimation of $\mathbf{p}$.

3.4.2.2 Online estimation using a batch adaptive control

Online estimation of the BPS elements is classically performed via adaptive techniques, as explained in [158] and [157]. Here, the batch adaptive control algorithm presented in [8,71] is used instead of adaptive techniques. Reasons are higher robustness of this algorithm against measurement noise and against the initial guess of the estimates, together with the fast convergence of the estimates to the steady-state values [71].

Assume that friction τ_i^f in joint i is represented with a model linear in unknown parameters $\mathbf{p}_i^f \in \mathbb{R}^{s_i}$:

$$\left(\mathbf{r}_i^f(q_i, \dot{q}_i, t)\right)^T \mathbf{p}_i^f = \tau_i^f\,, \tag{3.31}$$

where $\mathbf{r}_i^f \in \mathbb{R}^{s_i}$ is the corresponding regression vector. Then, the dynamic model (3.15) can be rewritten as:

$$\mathbf{R}^*(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), t)\mathbf{p}^* = \boldsymbol{\tau}(t)\,, \tag{3.32a}$$

with

$$\mathbf{R}^* = \begin{bmatrix} \mathbf{R}\left(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t)\right) & \mathbf{0}_{n\times s_1} & \cdots & \mathbf{0}_{n\times s_n} \\ \mathbf{0}_{1\times p} & \left(\mathbf{r}_1^f(q_1(t), \dot{q}_1(t), t)\right)^T & \cdots & \mathbf{0}_{1\times s_n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{1\times p} & \mathbf{0}_{1\times s_1} & \cdots & \left(\mathbf{r}_n^f(q_n(t), \dot{q}_n(t), t)\right)^T \end{bmatrix},$$

$$\mathbf{p}^* = [\mathbf{p}^T, (\mathbf{p}_1^f)^T, \ldots, (\mathbf{p}_n^f)^T]^T\,. \tag{3.32b}$$

The vector $\mathbf{p}^* \in \mathbb{R}^{p+s_1+\ldots+s_n}$ is the collection of all parameters to be estimated, i.e., of the BPS elements and the friction parameters.

During the identification experiment, the robot is supposed to realize some repetitive reference motion $\mathbf{q}_r(t)$. This can be done if the robot is controlled using a control law, consisting of a conventional PD controller and of the output $\mathbf{v} \in \mathbb{R}^n$ of a batch adaptive controller:

$$\boldsymbol{\tau}(t) = -\mathbf{K}_p\mathbf{e}(t) - \mathbf{K}_d\dot{\mathbf{e}}(t) + \mathbf{v}(t)\,, \tag{3.33}$$

where $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{n\times n}$ are diagonal matrices of positive proportional and derivative gains, respectively, and $\mathbf{e}$ is the difference between the actual and the reference joint motion:

$$\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}_r(t)\,. \tag{3.34}$$

If (3.33) is applied to (3.32), then the closed-loop system takes the form:

$$\mathbf{R}^*\left(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), t\right)\mathbf{p}^* + \mathbf{K}_p\mathbf{e}(t) + \mathbf{K}_d\dot{\mathbf{e}}(t) = \mathbf{v}(t)\,. \tag{3.35}$$

If the joint motions, velocities, and accelerations in (3.35) are equal to the references, the system (3.35) takes the form

$$\mathbf{R}^*\big(\mathbf{q}_r(t), \dot{\mathbf{q}}_r(t), \ddot{\mathbf{q}}_r(t), t\big)\mathbf{p}^* = \mathbf{v}_r(t), \tag{3.36}$$

The residual error dynamics, with input $\delta\mathbf{v} = \mathbf{v}_r - \mathbf{v}$, is given by

$$\begin{aligned} \mathbf{R}^*\big(\mathbf{q}_r(t), \dot{\mathbf{q}}_r(t), \ddot{\mathbf{q}}_r(t), t\big)\mathbf{p}^* - \mathbf{R}^*\big(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), t\big)\mathbf{p}^* \\ - \mathbf{K}_p\mathbf{e}(t) - \mathbf{K}_d\dot{\mathbf{e}}(t) = \delta\mathbf{v}(t). \end{aligned} \tag{3.37}$$

With appropriate choices of $\mathbf{K}_p$ and $\mathbf{K}_d$ [8], the dynamics (3.37) can be made passive for the output

$$\delta\mathbf{w} = \dot{\mathbf{e}} + \beta s(\mathbf{e}), \tag{3.38}$$

where $\beta \in \mathbb{R}$ is a positive scalar, and $s(\mathbf{e}) = \partial S(\mathbf{e}) / \partial\mathbf{e}$ with $S(\mathbf{e})$ defined in [8] by

$$S(\mathbf{e}) = \begin{cases} -\mathbf{e} - 0.5\pi + 1, & \mathbf{e} < -0.5\pi \\ 1 - \cos\mathbf{e}, & -0.5\pi \le \mathbf{e} < 0.5\pi \\ \mathbf{e} - 0.5\pi + 1, & \mathbf{e} > 0.5\pi. \end{cases} \tag{3.39}$$

Passivity means that the dynamics (3.37) cannot store more energy than supplied to it from outside via $\delta\mathbf{v}$, and that the condition $\int_0^{t_f} (\delta\mathbf{v})^T \delta\mathbf{w}\,\mathrm{d}t \ge \frac{1+\gamma}{2}\int_0^{t_f} (\delta\mathbf{w})^T \mathbf{T}\delta\mathbf{w}\,\mathrm{d}t$ holds with γ representing a positive scalar and $\mathbf{T}$ being a positive definite matrix of appropriate dimension.

Before presenting a batch adaptive update law, several definitions and assumptions are needed:

$$\mathbf{R}_r := \mathbf{R}(\mathbf{q}_r, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r), \; \mathbf{L}_r := \int_0^{t_f} \mathbf{R}_r^T \mathbf{R}_r \mathrm{d}t. \tag{3.40}$$

Here, t_f denotes the duration of one repetition of $\mathbf{q}_r(t)$. The matrix $\mathbf{L}_r$ is assumed to be nonsingular, which holds if $\mathbf{q}_r(t)$ persistently excites the robot dynamics. The inner product of two vector valued time functions $\mathbf{u}_i(t)$ and $\mathbf{v}_j(t)$ of equal dimension is defined by:

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle_{\mathbf{T}} = \int_0^{t_f} \mathbf{u}_i^T(t)\mathbf{T}\mathbf{v}_j(t)\mathrm{d}t. \tag{3.41}$$

The inner product of two matrices $\mathbf{U}(t)$ and $\mathbf{V}(t)$ of equal number of rows, with $\mathbf{u}_i(t)$ and $\mathbf{v}_j(t)$ representing the i^{th} and j^{th} columns $\mathbf{U}(t)$ and $\mathbf{V}(t)$, respectively, is defined by:

$$\langle \mathbf{U}, \mathbf{V} \rangle_{\mathbf{T}} = \begin{bmatrix} \langle \mathbf{u}_1, \mathbf{v}_1 \rangle_{\mathbf{T}} & \langle \mathbf{u}_1, \mathbf{v}_2 \rangle_{\mathbf{T}} & \cdots \\ \langle \mathbf{u}_2, \mathbf{v}_1 \rangle_{\mathbf{T}} & \langle \mathbf{u}_2, \mathbf{v}_2 \rangle_{\mathbf{T}} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}, \tag{3.42}$$

where $\mathbf{T}$ is a positive definite weighting matrix of appropriate dimension.

The batch adaptive update law is given by

$$\mathbf{p}_{k+1}^* = \mathbf{p}_k^* + \mathbf{K}_r \langle \mathbf{R}_r, \delta\mathbf{w}_k \rangle_{\mathbf{T}}; \; \mathbf{v}_{k+1} = \mathbf{R}_r \mathbf{p}_{k+1}^*, \tag{3.43}$$

where k is the number of repetition of $\mathbf{q}_r(t)$ $(k = 1, 2, \dots), \mathbf{T} \in \mathbb{R}^{n \times n}$ is the weighting matrix, and $\mathbf{K}_r \in \mathbb{R}^{(p+s_1+\ldots+s_n)\times(p+s_1+\ldots+s_n)}$ is the positive definite matrix called adaptation gain. If $\mathbf{K}_r$ satisfies

$$\mathbf{K}_r = \mathbf{L}_r^{-1} \langle \mathbf{R}_r, \mathbf{R}_r \rangle \mathbf{L}_r^{-1}, \tag{3.44}$$

then the convergence of the error (3.34)

$$\| \mathbf{e}_k(t) \| \xrightarrow[k \to \infty]{} 0 . \tag{3.45}$$

can be proven [71], with $\| \ \|$ denoting a signal norm. The convergence (3.45) implies convergence of the elements of $\mathbf{p}_k^*$ to steady-state values ($\mathbf{p}_k^* = \mathbf{p}_{k+1}^*$). Theoretically, there is no restriction on the initial guess $\mathbf{p}_0^*$ of the parameter vector in (3.43). Practically, when the inputs in $\boldsymbol{\tau}$ are bounded, $\mathbf{p}_0^*$ is constrained to some region corresponding to admissible control inputs.

In the presented batch adaptive control algorithm, the tuning parameters are contained in the matrices $\mathbf{K}_p$, $\mathbf{K}_d$ from (3.33), and in $\mathbf{T}$ from (3.43). The scalar β in (3.38) is tuned, as well. The PD gains should ensure passivity of the system (3.37) from the input $\delta\mathbf{v}$ to the output $\delta\mathbf{w}$ given by (3.38). The matrix $\mathbf{T}$ and the scalar β influence the rate of convergence of the parameters to the steady-state values.

The steady-state values represent estimates of the BPS elements and friction parameters. As with the batch least-squares estimation, the quality of the estimates strongly depends on the choice of a trajectory the robot realizes during the identification experiment. An appropriate design of such a trajectory is explained next.

3.4.3 Excitation trajectory

The robot trajectory, which is realized during the identification experiment, is called an excitation trajectory. There are several designs of such a trajectory available in literature, and they are briefly summarized in [32]. A design suggested in [32] and [170] postulates an excitation trajectory as finite Fourier series:

$$q_i^{exc}(t) = q_{0,i} + \sum_{j=1}^{N_i} [a_{i,j} \sin(j 2\pi \Delta f_i t) - b_{i,j} \cos(j 2\pi \Delta f_i t)] / j \ \ (i = 1,2,3). \tag{3.46}$$

Here, Δf_i [Hz] is the desired resolution of the frequency content, the number of harmonics N_i determines the frequency content of the trajectory, and the free coefficients $q_{0,i}$, $a_{i,j}$, and $b_{i,j}$ ($i = 1,\ldots,n$; $j = 1,\ldots,N_i$) should be determined such that all manipulator rigid-body dynamics are sufficiently excited during the identification experiment. The given excitation trajectory is periodic, which enables continuous repetition of the identification experiment and averaging of measured data to improve the signal-to-noise ratio. Additional advantages are direct control of the resolution and of the trajectory harmonic content. The control of the harmonic content is an important precaution for avoiding frequency components that excite the flexible modes of the robot.

An excitation trajectory can be realized on robotic systems using a PD (proportional, derivative) feedback controller:

$$\boldsymbol{\tau} = -\mathbf{K}_p(\mathbf{q} - \mathbf{q}^{exc}) - \mathbf{K}_d(\dot{\mathbf{q}} - \dot{\mathbf{q}}^{exc}), \tag{3.47}$$

where $\mathbf{K}_p, \mathbf{K}_d \in \mathbb{R}^{n \times n}$ are diagonal matrices of positive proportional and derivative gains, respectively. The applied $\boldsymbol{\tau}$ sufficiently excites the system rigid-body dynamics if the free coefficients in (3.46) are determined by optimising some property of the information matrix $\boldsymbol{\Phi}$ (e.g., the condition number of $\boldsymbol{\Phi}$). No matter which property is adopted, the resulting optimization problem is non-convex and nonlinear. The non-convexity can be illustrated with a trivial example of minimizing the condition number $\aleph_i$ of the information matrix $\boldsymbol{\Phi}_i$ corresponding to the dynamics (3.20) excluding friction:

$$\begin{bmatrix}\tau_i(t_1)\\ \vdots \\ \tau_i(t_\zeta)\end{bmatrix} = \begin{bmatrix}\ddot{q}_i(t_1) & \sin q_i(t_1)\\ \vdots & \vdots\\ \ddot{q}_i(t_\zeta) & \sin q_i(t_\zeta)\end{bmatrix}\begin{bmatrix}J_i\\ c_i\end{bmatrix} = \mathbf{\Phi}_i\begin{bmatrix}J_i\\ c_i\end{bmatrix}. \qquad (3.48)$$

To simplify presentation, q_i is postulated in the form of (3.46), with $N_i = 1$, $b_{i,j} \equiv 0$, and $\Delta f_i = 0.1$ [Hz]. Thus, only two coefficients $q_{0,i}$ and $a_{i,1}$ remain to be determined by optimising the condition number $\aleph_i$. This number is computed at a grid corresponding to $-\pi \le q_{0,i} \le \pi$ [rad] and $-20 \le a_{i,1} \le 20$ [rad]. The range for the computed $\aleph_i$ is very large, which impedes visualization. Therefore, Fig. 3.3 presents $\aleph_i^{-1}$ instead of $\aleph_i$. The existence of local maxima for $\aleph_i^{-1}$, see Fig. 3.3, shows that the optimization problem is non-convex. Because of non-convexity, different initial conditions may lead to different exciting trajectories, all corresponding to local minima of the performance criterion. Consequently, the optimisation determines a suboptimal, rather than the globally optimal exciting trajectory. Still, experience shows that the use of suboptimal trajectories gives satisfactory results.

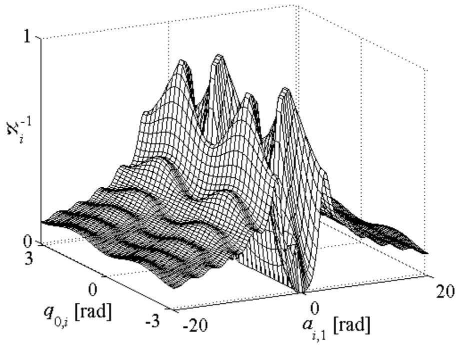


Figure 3.3: An illustration of the non-convexity problem.

The presence of disturbances and parasitic effects in the collected $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$, affect the quality of estimation of the parameters describing the rigid-body robot dynamics. The quality of estimation is better if disturbances and parasitics are reduced to a minimum, since these are not captured by the rigid-body dynamic model. High-frequency quantisation noise, cogging force, and vibrations caused by flexibilities are commonly found in robot manipulators. Carefully tuned Kalman observers [20,21,59] filter the effect of disturbances and parasitics, while reconstructing the components of actual joint motions, velocities, and accelerations that correspond to the rigid dynamic behaviour. The design of appropriate Kalman observers is discussed next.

3.5 Online reconstruction of motion coordinates

Reconstruction of motion coordinates, i.e., of joint motions, velocities, and accelerations, using a Kalman observer will be formulated in discrete-time, assuming a sampled data-acquisition during the identification experiment. It is proven in [21] that at sufficiently high sampling rates, typical in modern robotics, the discrete Kalman observer associated with the motion of a robot joint i ($i = 1, \ldots, n$) can be constructed assuming an all-integrator model for this motion. This model includes a correcting term ν_i, representing the model uncertainty. It accounts for the difference between the adopted model and the real joint dynamics. The correcting term is a realization of the zero mean white process noise ξ_i, filtered through a linear, stable, all-pole transfer function. If ν_i is used, to reconstruct velocity, a second-order model is required at least. If the acceleration needs to be estimated, the model must be at least of the third-order. Here, a third-order model is considered, assuming that ξ_i is filtered just through a single integrator. Initially, the model is formulated in continuous time.

If only the motion q_i is measured in joint *i*, the model for the Kalman observer design is

$$\ddot{q}_i(t) = \ddot{q}_{r,i}(t) + \nu_i(t); \quad \dot{\nu}_i(t) = \xi_i(t); \quad y_i(t) = q_i(t) + \eta_i(t), \tag{3.49}$$

where $\ddot{q}_{r,i}$ is the reference joint acceleration, and η_i is the measurement noise. In the design of the Kalman observer, the deviation from the desired motion $q_{r,i}$, i.e., $e_i = q_i - q_{r,i}$, its time derivative $e_i^{\omega} = \dot{e}_i$, and the model uncertainty ν_i, can be considered as the states to be reconstructed. The corresponding state-space model is obtained from (3.49):

$$\begin{aligned} \dot{\mathbf{x}}_i(t) &= \mathbf{A}_i \mathbf{x}_i(t) + \mathbf{h}_i \xi_i(t); \\ y_i(t) &= e_i(t) + q_{r,i}(t) + \eta_i(t), \end{aligned} \tag{3.50a}$$

where

$$\mathbf{x}_i(t) = \begin{bmatrix} e_i(t) \\ e_i^{\omega}(t) \\ \nu_i(t) \end{bmatrix}; \ \mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}; \ \mathbf{h}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (i = 1, \ldots, n). \tag{3.50b}$$

Let T_s denotes the sampling time used for both data-acquisition and control. A discrete-time system having identical solutions with the system (3.50a,b) at $t = kT_s$ is

$$\begin{aligned} \mathbf{x}_i(k+1) &= \mathbf{E}_i(T_s)\mathbf{x}_i(k) + \gamma_i(T_s)\xi_i(k); \\ y_i(k) &= \mathbf{c}_i^T \mathbf{x}_i(k) + q_{r,i}(k) + \eta_i(k), \end{aligned} \tag{3.51a}$$

where

$$\mathbf{x}_i(k) = [e_i(k),\ e_i^{\omega}(k),\ \nu_i(k)]^T,$$

$$\mathbf{E}_i = e^{\mathbf{A}_i T_s} = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, \ \gamma_i = \int_0^{T_s} e^{\mathbf{A}_i \varsigma} \mathbf{h}_i d\varsigma = \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix}, \ \mathbf{c}_i = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{3.51b}$$

The notation kT_s is rationalized with k, for convenience.

A Kalman observer for optimal state reconstruction in the presence of model uncertainty and measurement noise is:

$$\hat{\mathbf{x}}_i(k+1) = \mathbf{E}_i(T_s)\bar{\mathbf{x}}_i(k), \ \bar{\mathbf{x}}_i(k) = \hat{\mathbf{x}}_i(k) + \mathbf{k}_i[y_i(k) - q_{r,i}(k) - \mathbf{c}_i^T \hat{\mathbf{x}}_i(k)], \tag{3.52}$$

where $\bar{\mathbf{x}}_i$ denotes the updated estimate of $\mathbf{x}_i$, and $\mathbf{k}_i \in \mathbb{R}^n$ is a vector of constant gains that are determined by $\mathbf{E}_i$ and γ_i, and by the covariances of η_i and ξ_i. Assuming a white, zero mean quantisation noise, uniformly distributed within a resolution increment $\Delta\theta_m$ of the position sensor, a straightforward choice for the covariance of η_i is $\Delta\theta_m^2/12$ [20,21]. A simple but effective choice of the covariance of ξ_i is the square of the maximum jerk of the reference joint motion [20]. In practice, these choices serve as initial guesses, while the final tuning is most effectively done on-line. Use of Bode plot from the observed e_i to the reconstructed $\bar{e}_i$ can be also instructive for filter tuning. The Bode plot reveals frequencies where harmonics will be amplified with the filter, as well as frequencies that the filter will attenuate. By analyzing the power spectrum of the error signal, one may locate spectral components caused by disturbance and parasitic effects, that need to be filtered-out. The gain vector $\mathbf{k}_i$ can be retuned to reshape the Bode plot according to the needs.

As the observer (3.52) reconstructs deviations from the reference joint motions and velocities, together with the model uncertainty, the motion coordinates are reconstructed as follows:

$$\bar{q}_i = q_{r,i} + \bar{e}_i\,,\ \bar{\dot{q}}_i = \dot{q}_{r,i} + \bar{e}_i^{\omega}\,,\ \bar{\ddot{q}}_i = \ddot{q}_{r,i} + \bar{\mathrm{v}}_i\,. \tag{3.53}$$

3.6 Model validation

Writing tasks may require complex non-smooth motions, whose execution may require significant dynamic forces/torques. One representative writing task [143,144] is presented in Fig. 3.4. It consists of writing a sequence of letters in the vertical *xz*-plane. The letters can be described piecewise in closed-form and there is a possibility of imposing an arbitrary velocity profile:

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{z}^2(t)}\ . \tag{3.54}$$

Freedom in prescribing the velocity profile enables one to pose very demanding dynamic tasks, which is demonstrated with the analysis presented in [144]. The piecewise closed-form representation and the possibility of imposing the velocity can be exploited for a rigorous verification of robot kinematic and dynamic models.

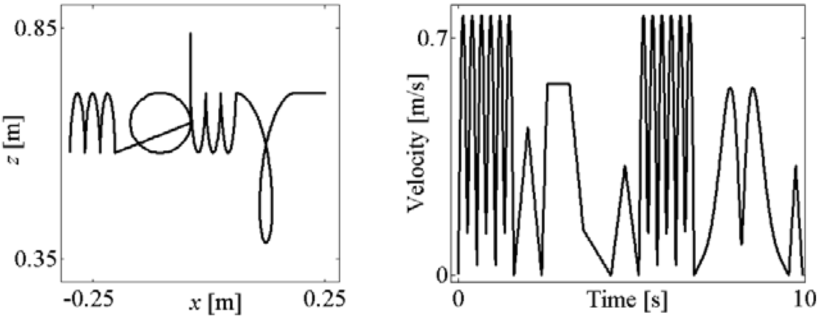


Figure 3.4: Reference trajectory in the task space:
(left) tip path, (right) tip-velocity profile.

Fig. 3.4 defines the reference trajectory of the robot-tip. A derived inverse kinematic model can be used to determine the corresponding reference joint motions $\mathbf{q}_r(t)$. Along these motions a dynamic model can be validated. The organization of the validation experiment is illustrated in Fig. 3.5. The objective is to test the agreement of the torques $\bar{\boldsymbol{\tau}}$ rendered by the model, which is fed by on-line reconstructed joint motions, velocities, and accelerations (see equations (3.51)-(3.53)), and “measured” torques under PD control:

$$\boldsymbol{\tau} = -\mathbf{K}_p(\bar{\mathbf{q}} - \mathbf{q}_r) - \mathbf{K}_d(\bar{\dot{\mathbf{q}}} - \dot{\mathbf{q}}_r)\,. \tag{3.55}$$

If $\bar{\boldsymbol{\tau}}$ is close enough to $\boldsymbol{\tau}$, the dynamic model is considered appropriate for control purposes.

3.7 Dynamics not covered with a rigid-body dynamic model with friction

In model-based motion control, a dynamic model is used to compensate for the nonlinear and coupled manipulator dynamics. Consider a motion control problem solved, for example, using the computed-torque control strategy described in subsection 2.2.2. The computer-torque control law (2.26) is repeated here for convenience [153,158]:

$$\boldsymbol{\tau}(t) = \mathbf{M}(\mathbf{q}(t))\mathbf{u}(t) + \mathbf{c}(\mathbf{q}(t), \dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}^f(\mathbf{q}(t), \dot{\mathbf{q}}(t), t). \tag{3.56}$$

Once again, $\mathbf{u} = [u_1, \ldots, u_n]^T$ represents a feedback control action which should stabilize the manipulator motion and enforce the desired accuracy in realizing the reference trajectory $\mathbf{q}_r(t)$. The remaining variables are as in the dynamic model (3.11). When applied to (3.11), the control law (3.56) reduces the motion control problem to the linear case, which can now also be defined in the Laplace domain:

$$\mathbf{q}(s) = \mathbf{P}(s)\mathbf{u}(s), \quad \mathbf{P}(s) = diag[\frac{1}{s^2}, \ldots, \frac{1}{s^2}]. \tag{3.57}$$

The i^{th} ($i = 1, \ldots, n$) diagonal element of $\mathbf{P}(s)$ describes the dynamics from the feedback control action u_i to the displacement q_i. This is the dynamics of the plant P_i to be controlled.

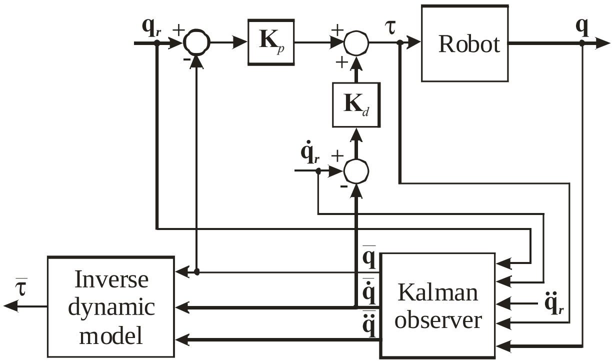


Figure 3.5: Experimental validation of a robot dynamic model.

The plants $P_1, \ldots, P_n$, defined by (3.57), have perfectly rigid dynamics. Each plant can be described by a frequency response function (FRF) with a linear magnitude plot of –2 slope and constant phase at -180$^{\text{o}}$. Such a FRF can be established only if (3.56) perfectly matches the real robot dynamics. In practice, this is hardly achievable, as no robot is perfect in its realization and, hence, the rigid-body model employed in (3.56) is hardly an accurate description of the real robot dynamics. Instead of (3.57), one should rather consider control of a more general transfer function matrix $\mathbf{P}$ that captures parasitic robot dynamics:

$$\mathbf{q}(s) = \mathbf{P}(s)\mathbf{u}(s), \quad \mathbf{P}(s) = \begin{bmatrix} P_{1,1}(s) & \cdots & P_{1,n}(s) \\ \vdots & \ddots & \vdots \\ P_{n,1}(s) & \cdots & P_{n,n}(s) \end{bmatrix}. \tag{3.58}$$

The transfer functions on the main diagonal of $\mathbf{P}$ represent dynamics from u_i to q_i ($i = 1, \ldots, n$). The cross-terms represent interactions between the joints that remain after imperfect compensation of the real robot dynamics. If any of these cross-terms is not negligible when compared with the transfer functions on the main diagonal, we can conclude that the dynamic model used in the computed-torque control law is not sufficiently accurate. If the magnitudes of the cross-terms are smaller than the magnitudes of the diagonal elements, then the transfer matrix $\mathbf{P}$ can be considered practically diagonal.

If $\mathbf{P}$ is a diagonal matrix, then feedback control design is reduced to *n* single-input, single-output systems. A feedback design for joint *i* ($i = 1, \ldots, n$) is focused on the corresponding plant with the transfer function $P_{i,i}(s)$. This plant is not a mere double integrator, as by virtue of (3.57) one would expect, but a higher order dynamical system with resonance and anti-resonance frequencies that emerge because of flexibilities. A time delay can often be experienced, as well. The flexible dynamics can change for various operating conditions, as resonance frequencies and their relative damping may vary with the manipulator configuration. The dynamics not compensated with the rigid-body dynamic model, influence feasible levels of the motion control performance and robustness [29,92,94,96]. Therefore, it is worthwhile spending additional effort on determining them. A method capable of, at least, local identification of additional dynamics not covered with the rigid-body dynamic model, is explained next. For each robot joint *i* ($i=1,\ldots,n$), this technique identifies a FRF of the corresponding additional dynamics. Hereafter, different letters will be used for the parametric plant model and for the corresponding FRF: the model of dynamics associated with joint *i* will be denoted with $P_{i,i}$ and the corresponding FRF with $G_{i,i}(j\omega)$ (ω is angular frequency).

The identification makes use of the computed-torque law (3.56) with

$$\mathbf{u}(t) = -\mathbf{K}_p(\mathbf{q}(t) - \mathbf{q}_r(t)) - \mathbf{K}_d(\dot{\mathbf{q}}(t) - \dot{\mathbf{q}}_r(t)) + \boldsymbol{\upsilon}(t), \tag{3.59}$$

where $\mathbf{K}_p = [k_{p,1},\ldots,k_{p,n}]$ and $\mathbf{K}_d = [k_{d,1},\ldots,k_{d,n}]$ are matrices of positive proportional and derivative gains, respectively, and $\boldsymbol{\upsilon} = [\upsilon_1,\ldots,\upsilon_n]^T$ stands for external excitation. The identification is performed under closed-loop conditions, since the plant dynamics are not necessarily stable. The PD controller should stabilize the robot. The external excitation could be, for example, noise. Typically, in each experiment $\boldsymbol{\upsilon}$ contains random excitation only in one element, say υ_i ($i=1,\ldots,n$), and zeros elsewhere ($\upsilon_j = 0$, $j=1,\ldots,n$, $j \neq i$). With a sampled-data implementation of (3.56) with (3.59), the random excitation cannot cover the complete frequency spectrum (white noise), but it has a finite bandwidth determined by the sampling frequency.

During the identification experiment, the robot performs some reference motions. For example, if the dynamics for joint *i* is identified, then only this joint is moving with a low constant velocity ($\dot{q}_{r,i} \neq 0$), while the remaining ones are locked in fixed positions ($q_{r,j}(t) = \mathrm{q}_{r,j}$, $\dot{q}_{r,j} \equiv 0$, $j=1,\ldots,n$, $j \neq i$). The low-velocity movements are needed to reduce the effect of friction, if it is not sufficiently well compensated with the friction model. The motions are performed within the mechanical limits in the joints. If there are no such limits, full revolutions can be made in the rotational joints. Assuming only $\mathbf{q}$ is measured, the joint velocity $\dot{\mathbf{q}}$ can be determined as backward difference between two successive position measurements, divided with the time between the measurements.

A FRF $G_{i,i}(j\omega)$ of the dynamics in joint *i* can be found from the cross power spectrum $\Phi_{uq,i}$ between $u_i(t)$ and $q_i(t)$, and the autopower spectrum $\Phi_{uu,i}$ of $u_i(t)$ [113]:

$$G_{i,i}(j\omega) = \Phi_{uq,i}(j\omega) / \Phi_{uu,i}(\omega). \tag{3.60}$$

Because closed-loop identification is used, disturbance components of $u_i(t)$ and $q_i(t)$ are mutually correlated. If q_i is measured from an incremental encoder, quantization noise is a typical disturbance. As discussed at the beginning of section 3.4, the correlation of disturbance components can cause a biased identification result. Therefore, identification is more reliable if the presence of disturbances in the measured q_i is low. The reliability of identification can be evaluated based on the coherence of the measurements. The coherence function for joint *i* is determined as [113]

$$\Gamma_i(\omega) = \Phi_{uq,i}^2(\omega) / (\Phi_{qq,i}(\omega)\Phi_{uu,i}(\omega)), \tag{3.61}$$

where $\Phi_{qq,i}$ is the autopower spectra of $q_i(t)$. Provided all autopower spectra are nonzero, Γ_i is ideally 1 at all ω, independently of $q_i(t)$ and $u_i(t)$. The frequencies ω where Γ_i is below 1 correspond to spectral components of $q_i(t)$ that are corrupted by disturbances. The influence of disturbances is particularly complex if actual joint motions and velocities are used in the nonlinear model-based terms of the computed-torque law (3.56). To alleviate disturbance effect, $\mathbf{M}$, $\mathbf{c}$, $\mathbf{g}$, and $\boldsymbol{\tau}^f$ in (3.56) can be computed along the reference joint motions and velocities.

A solution to enhance reliability of identification in closed-loop is to determine the plant's FRF after the transfer from υ_i to $\mathbf{u}$ has been identified. Since υ_i is an external excitation, it is not correlated with an eventual disturbance component of $\mathbf{u}$, which circumvents the problem of biased identification. The transfer from υ_i to $\mathbf{u}$ defines the vector

$$\mathbf{s}_i(j\omega) = [S_{1,i}(j\omega), \ldots, S_{i,i}(j\omega), \ldots, S_{n,i}(j\omega)]^T. \tag{3.62}$$

The element $S_{i,i}(j\omega) = 1/[1 + (j\omega k_{d,i} + k_{p,i})G_{i,i}(j\omega)]$ is the sensitivity function of the feedback loop in joint *i*, while $S_{j,i}(j\omega) = 1/[1 + (j\omega k_{d,j} + k_{p,j})G_{j,i}(j\omega)]$, $j = 1, \ldots, n$, $j \neq i$, represents the couplings between joint *i* and the other joints. If these couplings are small, then the matrix $\mathbf{P}$ in (3.58) is diagonal. In this case, one can determine the plant's FRF $G_{i,i}(j\omega)$ from the measured $S_{i,i}(j\omega)$. However, due to measurements in the closed-loop, only those components of $S_{i,i}(j\omega)$ that are beyond the bandwidth of the closed-loop system in joint *i* are reliable, since the coherence is poor at lower frequencies. The poor coherence is a consequence of a feedback controller that has the filtering effect on the signals in the loop. The feedback controller strongly attenuates harmonic components of u_i within the bandwidth, which reduces signal to noise ratio for these harmonics and in turn leads to poor coherence. Therefore, a low closed-loop bandwidth is preferable in joint *i*. On the other hand, sufficiently high bandwidths of the feedback loops in the remaining joints are needed to prevent any motion in these joints due to noise excitation in joint *i*. Therefore, PD settings are different for joint *i* and the remaining joints. Finally, the plant's FRF is determined by:

$$G_{i,i}(j\omega) = (1/S_{i,i}(j\omega) - 1)/(j\omega k_{d,i} + k_{p,i}). \tag{3.63}$$

Separate identification experiments are needed to identify the plant FRF's in the other joints.

The shape of the identified FRF $G_{i,i}^k(j\omega)$ depends on the actual configuration in the other joints $q_{r,j}^k(t) = \mathrm{q}_{r,j}^k$ ($j = 1, \ldots, n$, $j \neq i$). For feedback control design, it is often convenient to have some nominal representation of the plant dynamics, say $G_{i,i}^{\mathrm{o}}(j\omega)$. Assume FRF's are identified for N_i ($k = 1, \ldots, N_i$) different configurations: $\{G_{i,i}^1(j\omega), \ldots, G_{i,i}^{N_i}(j\omega)\}$. Then, there are at least two possibilities to determine the nominal dynamics in joint *i*:

$$G_{i,i}^{\mathrm{o}}(j\omega) = \arg \min_{G_{i,i}(j\omega)} \max_{k} \left| \frac{G_{i,i}^k(j\omega) - G_{i,i}(j\omega)}{G_{i,i}(j\omega)} \right|, \text{ or,} \tag{3.64a}$$

$$G_{i,i}^{\mathrm{o}}(j\omega) = \frac{1}{N_i} \sum_{k=1}^{N_i} G_{i,i}^k(j\omega). \tag{3.64b}$$

A choice between the solutions (3.64a) and (3.64b) can be case dependent, although, in principle, the latter one gives smoother magnitude and phase FRF plots. Differences between $G_{i,i}^k(j\omega)$ and $G_{i,i}^{\mathrm{o}}(j\omega)$ can be treated as perturbations from the nominal plant dynamics. These perturbations can be explicitly taken into account by the feedback control design, as it will be shown in Chapter 5.

The nominal FRF represents dynamics not compensated with the dynamic rigid-body model, as a set of data. For control design it is more convenient to represent these dynamics with some parametric model in a transfer function or state-space form. There is a number of solutions to fit a parametric model into FRF data [113,140,141,151]. The least-square fit procedure using an output error model structure [151] is used in this thesis.

3.8 Summary

This chapter deals with the problems of robot modelling and identification, with the purpose of model-based motion control.

Derivations of robot kinematic and rigid-body dynamic models were explained. The emphasis was on closed-form modelling, since it facilitates analysis and manipulation of the models, as well as control design. The price to pay, though, is complexity of derivation. Use of a writing task is suggested for firm validation of the models, since writing incorporates motion primitives that robots typically realize in practice, and whose execution may induce significant dynamic loads.

Friction modelling was also discussed. Several models were presented, each having some purpose in the coming chapters: in the next chapter, the standard Coulomb plus viscous model and the LuGre friction model will be used, while the three-sigmoidal-friction model will be used in all other chapters. Several techniques to estimate parameters of the robot dynamic model with friction were suggested. Recursive Extended Kalman filtering can be used to estimate parameters of a friction model which cannot be represented linearly in the unknown parameters. This technique will be demonstrated in the next section when presenting experimental identification of a part of the LuGre model representing the sliding friction regime, as well as the identification of the three-sigmoidal-friction model. Frequency-domain identification can be used to estimate parameters of a part of the LuGre model representing the presliding friction regime. This technique will be practically demonstrated in the next chaper, too. Two algorithms for estimation of robot inertial parameters were presented: the batch least-squares one and the online algorithm based on a batch adaptive control technique. Guidelines for design of a trajectory that sufficiently excites the robot dynamics in the identification experiment, were given. The batch algorithm is simple for implementation, but it provides accurate estimates in the absence of disturbances in the motion coordinates (joint motions, velocities, and accelerations). Disturbance-free motion coordinates were achieved via online Kalman filtering. The online algorithm ensures higher efficiency of parameter estimation, since it delivers relevant estimates as the identification experiment is in progress. However, its practical implementation requires powerful control hardware, while for batch estimation, conventional hardware for data acquisition and control is sufficient.

Finally, it was indicated that the rigid-body dynamic model is still not sufficient for a perfect match to the real robot dynamics, as in practice, robot dynamics may contain flexible effects not covered by this model. A procedure to identify these flexible dynamics was proposed. With these additional dynamics available, more advanced feedback control designs become possible, as will be presented in chapter 5.

Chapter 4

The RRR robot arm

The previous chapter theoretically formulates concepts for robot modelling and identification that are suitable for motion control design. This chapter should verify usability of the given concepts by their practical demonstration on a realistic direct-drive robot manipulator. The kinematics of this manipulator is often seen in industry, so the kinematic and dynamic models to be presented are also valuable for industrial case studies. This chapter presents experimental results obtained when identifying friction effects, the rigid-body and flexible dynamics of the considered robot. Experimental experience gained during identification can be instructive for industrial case studies, as well. The obtained kinematic, dynamics, and friction models are used in the control designs considered in Chapters 5, 6, and 7. The chapter is organized as follows. Section 4.1 introduces the experimental robot set-up. The robot kinematic and rigid-body dynamic models in closed-form are presented in section 4.2. The correctness of these models is established in section 4.3. Friction modelling and estimation are considered in section 4.4. In section 4.5, the results of estimation of the robot inertial properties are presented. The results of the experimental validation of the estimates are provided, as well. The presence of additional dynamics not covered with the rigid-body dynamic model with friction is illuminated in section 4.6. The chapter summary is given at the end. Part of the material presented in this chapter has also been published in [30,96-98].

4.1 Experimental set-up

The robot arm shown in Fig. 1.1 is the subject of all case studies considered in the thesis. This is the custom-built multipurpose experimental set-up, suitable for research of versatile control problems in robotics [176-179]. The robot mechanism has three rotational joints, implemented as the waist, shoulder, and elbow. Since the abbreviation RRR is normally used to indicate serial kinematics with three rotational joints [53], the considered robot is referred to as RRR robot. The open (serial) kinematic structure is established by alternating the actuators and the links in series.

A schematic description of the RRR robot set-up is given in Fig. 4.1. The robot mechanism is actuated by brushless DC direct-drive motors from Dynaserv DM-series [177], with nominal torques of 60, 30, and 15 [Nm], respectively. Motors are of outer-rotor type with internal encoders and bearings, and each of them is directly coupled to the corresponding robot link. Since no gearboxes are used for the coupling, the RRR robot belongs to the class of direct-drive robots [13]. Integrated optical encoders provide 655360 pulses per revolution, which is equivalent to the angular resolution of approximately 10^{-5} [rad]. To facilitate unconstrained rotation of all joints, the power and the sensor signals to and from the second and the third motor are transferred via sliprings.

The motors are driven by power amplifiers with built-in current controllers. Both encoders and amplifiers are connected to a PC-based control system. This system consists of a MultiQ-PCI board for data acquisition and control (specifications given in Table 4.1) [1], combined with a real time Windows application Wincon [4] that runs Simulink [5] generated code using Real-Time Workshop [3] on a PC. Such system facilitates the design and real-time implementation of controllers in Simulink. Typically, the controllers run at a sampling frequency of $f_s = 1000$ [Hz] ($T_s = 1$ [ms]), which appears sufficient for the control designs to be considered in this thesis. Namely, the corresponding

Nyquist frequency of $f_s / 2 = 500$ [Hz] well covers the robot dynamics of interest for the control designs, that all are below 230 [Hz]. The value of 230 [Hz] was obtained by experimental analysis in the frequency domain performed according to the procedure explained in section 3.7. Some illustrative results of that analysis will be presented in section 4.6. The presence of parasitic robot dynamics limits the feasible closed-loop bandwidths. The control designs presented in Chapters 5, 6, and 7 achieve bandwidths of maximum 20 [Hz], for which the sampling frequency of 1000 [Hz] is high enough.

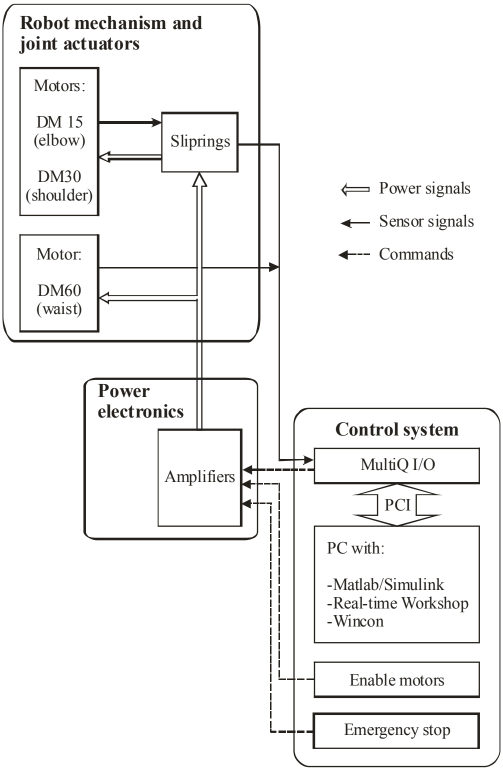


Figure 4.1: Components of the RRR robot set-up.

The power electronics ensure linearity between the commanded voltages u_i at the output of the MultiQ board (control voltage) and the torques τ_i generated by the motors:

$$\tau_i = k_i^{\tau u} u_i \ (i = 1,2,3). \tag{4.1}$$

The gains are $k_1^{\tau u} = 12$, $k_1^{\tau u} = 6$, and $k_1^{\tau u} = 3$ [90]. The vector of torques τ_i $(i = 1,2,3)$ represents the control input to the robot dynamics:

$$\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \tau_3]^T. \tag{4.2}$$

Because of direct-drive actuation, the dynamics of the RRR robot are highly nonlinear and coupled. The rigid-body dynamics dominate in the low frequency range. Friction in the robot joints can be observed, too. Aside rigid-body dynamics and friction, several other effects can be excited, e.g. vibrations at the robot base and high frequency resonances. There is also a time delay between the input generated in the control system, and the corresponding joint angular response. All these effects influence the performance of the motion control, and they must be taken into account in the control design by adequate modelling and identification. The analytical and experimental (data-driven) procedures that provide appropriate models are explained next.

TABLE 4.1: GENERAL SPECIFICATIONS OF MULTIQ-PCI BOARD

Interface	Parameter	Description
BUS	Type	PCI, 32-bit, 33 [MHz]
Encoders	Counter width	24 [bits]
	Number of channels	6
Digital input/output	Number of channels	48
Analog to digital conversion	Resolution	14 [bits]
	Input range	Programmable as ±5 or ±10 [V]
	Number of channels	16
	Conversion time	~ (17+23*n*) [μs], where *n* = # channels read
Digital to analog conversion	Resolution	13 [bits]
	Output range	±10 [V]
	Number of channels	4
	Conversion time	~ 17 [μs] / channel

4.2 Robot kinematics and dynamics in closed-form

4.2.1 Kinematics

A wire frame kinematic model of the RRR robot is shown in Fig. 4.2. The coordinate frames are assigned according to the DH (Denavits-Hartenberg's) notation [53,153], which establishes a minimal kinematic representation with the parameters presented in Table 4.2.

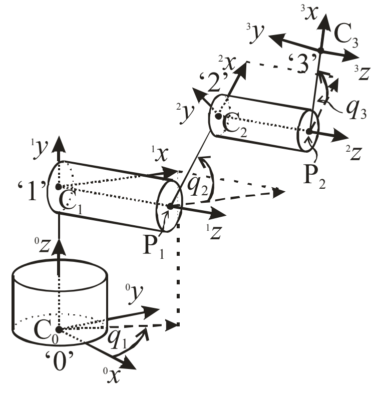


Figure 4.2: Kinematic model of the RRR robot.

TABLE 4.2: DH PARAMETERS OF THE RRR ROBOT

i	α_i [rad]	a_i [m]	q_i	d_i [m]
1	$\pi/2$	0	q_1	$C_0C_1=0.56$
2	0	$P_1C_2=0.2$	q_2	$C_1P_1=0.169$
3	0	$P_2C_3=0.415$	q_3	$C_2P_2=0.09$

The number of degree-of-freedom is $n=3$, and the vector of joint angular displacements is

$$\mathbf{q}=[q_1\ q_2\ q_3]^T . \tag{4.3}$$

The matrix (3.5) of homogenous transformation, representing position ${}^0_3\mathbf{x}$ and orientation ${}^0_3\mathbf{O}$ of the tip coordinate frame '3' relative to the coordinate frame of the base '0', is given by:

$$ {}^{0}_{3}\mathbf{T}(\mathbf{q}) = {}^{0}_{1}\mathbf{T}(\mathbf{q})\,{}^{1}_{2}\mathbf{T}(\mathbf{q})\,{}^{2}_{3}\mathbf{T}(\mathbf{q}) = \begin{bmatrix} {}^{0}_{3}\mathbf{O}(\mathbf{q}) & {}^{0}_{3}\mathbf{x}(\mathbf{q}) \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{4.4} $$

and is obtained as a product of homogenous transformations ${}^{i-1}_{i}\mathbf{T}(\mathbf{q})$ ($i = 1,2,3$) between the circumjacent coordinate frames. Particular expressions for ${}^{0}_{3}\mathbf{O}$ and ${}^{0}_{3}\mathbf{x}$, defining the forward kinematics, are given by (A.1) and (A.2), respectively, in Appendix.

A closed-form model of the inverse kinematics (IK), given by (A.3a,b), is determined by solving the nonlinear system of equations (A.2) for $\mathbf{q}$. The IK model in closed-form facilitates:

(i) direct recognition of kinematic singularities, by equalizing denominators of rational expressions with zero;

(ii) control of the robot posture, by specifying a sign in the expression for q_3 ('-' for elbow-up, '+' for elbow down);

(iii) computation of an accurate IK solution, since the closed-form equations requires less operations than computing IK by any iterative numerical method.

The geometric Jacobian ${}^{0}_{3}\mathbf{J}(\mathbf{q})$ and its time-derivative ${}^{0}_{3}\dot{\mathbf{J}}(\mathbf{q})$ are needed in the differential kinematic models (3.6) and (3.9). The Jacobian is derived using the formulas (3.7), (3.8), and (3.2), and it is given in closed-form by (A.4). The product of ${}^{0}_{3}\dot{\mathbf{J}}(\mathbf{q})$ and $\dot{\mathbf{q}}$, which is the second term of (3.9), is given by (A.5).

4.2.2 Dynamics

The rigid-body dynamic model of the RRR robot is derived in standard form (3.11), using formulas (3.12)-(3.14). For convenience, this form of the model is repeated here:

$$ \mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t)) + \mathbf{g}(\mathbf{q}(t)) + \boldsymbol{\tau}^{f}(\mathbf{q}(t),\dot{\mathbf{q}}(t),t) = \boldsymbol{\tau}(t)\,. \tag{4.5} $$

For the sake of identification of the robot inertial properties, the model is transformed into regressor form (3.15), also repeated here for convenience:

$$ \mathbf{R}(\mathbf{q}(t),\dot{\mathbf{q}}(t),\ddot{\mathbf{q}}(t))\mathbf{p} + \boldsymbol{\tau}^{f}(\mathbf{q}(t),\dot{\mathbf{q}}(t),t) = \boldsymbol{\tau}(t)\,. \tag{4.6} $$

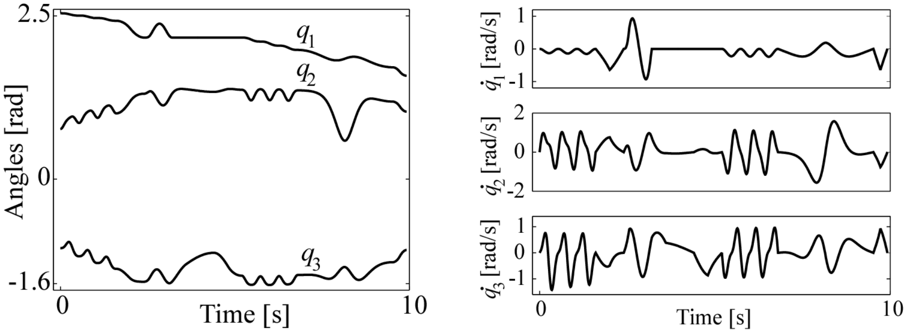
The vector $\mathbf{p}$ contains 15 elements of the BPS (Base Parameter Set), which is in agreement with the formula $7n - 4\ell - 2$ given in section 3.2 (taken from [121]). Indeed, the arguments of this formula that correspond to the RRR robot are $n = 3$ and $\ell = 1$. Closed-form expressions of the elements of $\mathbf{p}$ and $\mathbf{R}$ are given by formulas (A.6) and (A.7), respectively. These expressions reveal the complexity of the nonlinear robot dynamics, but also enable independent analysis of particular dynamical effects. With the elements of (A.7) containing the gravitational constant g, one may assemble gravity terms, with the elements containing joint accelerations one recovers inertial terms, while the remaining elements define Coriolis/centripetal terms.

4.3 Evaluating the correctness of the kinematic and dynamic models

To establish correctness of the kinematic and dynamic models (A.2),(A.3a,b) and (A.6),(A.7), respectively, the writing task shown in Fig. 3.4 is considered. With the numerical values of the DH parameters given in Table 4.2, the IK model (A.3a,b) computes the joint motions shown in Fig. 4.3a. The equivalent joint velocities are presented in Fig. 4.3b. The motions are compared with the corresponding IK solution computed using a recursive numerical routine, implemented in the Robotics Toolbox for Matlab [41]. The guaranteed accuracy of the recursive solution was better than 10^{-10} rad. The differences between the IK solutions computed with the model (A.3a,b) and with the Robotics

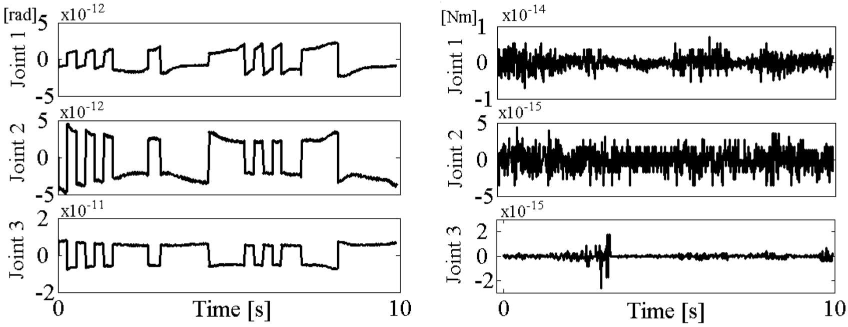
Toolbox are within the accuracy of the latter ones, see Fig. 4.4a. This verifies the correctness of (A.3a,b).

To establish the correctness of the rigid-body dynamic model (A.6),(A.7), all inertial parameters are assigned arbitrary non-zero values, positive for masses and principal moments of inertia, without sign constraint for products of inertia and coordinates of centers of masses. The model computes torques corresponding to the joint space trajectory shown in Figs. 4.3a,b. The torques are also calculated using a recursive numerical routine implemented in the Robotics Toolbox. The differences between the two solutions are caused by round-off errors only, see Fig. 4.4b. This verifies correctness of the equations (A.6),(A.7).



a) b)

Figure 4.3: The joint space trajectory for the writing task: a) motions and b) velocities.



a) b)

Figure 4.4: Differences between the closed-form and the recursive solutions of: a) inverse kinematics and b) inverse dynamics.

4.4 Friction modelling and estimation

Here, two parametric descriptions of friction presented in section 3.3 are taken into account: the dynamic LuGre model (3.18) and the static three-sigmoidal-functions model (3.19).These descriptions are considered here for two reasons: 1) to point out effective methods to capture friction properties in the presliding and sliding friction regimes, and 2) to examine if use of a more comprehensive parametric description (here the LuGre model) adds to the quality of friction reconstruction in the motion tasks where the robot is supposed to perform fast movements. Fast movements induce significant dynamic loads for the robot control system, and hence they are of concern for rigorous benchmarking of the motion control designs to be examined in the later chapters. Higher motion velocities are also preferable in industrial robot manufacturing, since they enable higher throughput.

The parameters of the LuGre model $v_{s,i}$, $\alpha_{0,i}$, $\alpha_{1,i}$, and $\alpha_{2,i}$ ($i = 1,2,3$), that correspond to the sliding friction regime, as well as the parameters of the neural-network model $f_{i,k}$, $w_{i,k}$, and $\alpha_{2,i}$ ($i = 1,2,3$; $k = 1,2,3$), were estimated using a recursive extended Kalman filtering (EKF) method explained in section 3.4.1. Aside from the friction parameters, two more unknowns J_i and c_i were also estimated (see (3.20)). The parameters $\sigma_{0,i}$ and $\sigma_{1,i}$ of the LuGre model corresponding to the pre-sliding regime were estimated using the frequency response function measurements (FRF), also addressed in section 3.4.1.

The open-loop identification experiments were performed separately for each robot joint to estimate the parameters characteristic to the sliding friction regime. The excitation torques $\tau_i(t)$ ($i = 1,2,3$) were manually generated online, by changing the gain of the constant control input with a slider control utility. Each excitation had high and low levels, to induce fast and slow angular motions $q_i(t)$. Different velocity levels were needed to acquire information about all effects covered by the adopted friction model. The measured $\tau_i(t)$ and $q_i(t)$ were processed off-line using the EKF method (3.21)-(3.25). The obtained estimates that correspond to the LuGre friction model are presented in Table A.1 in Appendix, while the estimates corresponding to the three-sigmoidal-functions model are given in Table A.2.

The FRF measurements were made to estimate $\sigma_{0,i}$ and $\sigma_{1,i}$, based on the linearization (3.26) of the LuGre model. Already estimated J_i, $\alpha_{2,i}$, and c_i were taken into account. To measure a FRF for joint *i*, this joint was excited with noise of a bandwidth up to 200 [Hz] with RMS (root mean squares) below the static friction level to prevent leaving the pre-sliding region, while the other joints were kept fixed using a PD control. The FRF was obtained by averaging 50 time series of 4096 samples at the sampling frequency of 1 [kHz], with a Hanning window and 50 % overlap. Since linearization (3.20) is valid only locally, the level of the applied noise had to be low. Consequently, several FRF measurements were performed for each joint, with different levels of noise. The FRF corresponding to the lowest noise level for which the pre-sliding behaviour could be observed, was finally selected. On this FRF, the linearization (3.20) was fitted to estimate $\sigma_{0,i}$ and $\sigma_{1,i}$. For illustration, Fig. 4.5 presents the FRF measured in the third joint and the corresponding fit. Clearly, at low frequencies a zero slope can be observed, indicating the stiffness instead of an inertial effect (-2 slope). The obtained estimates for all three joints are presented in Table A.1.

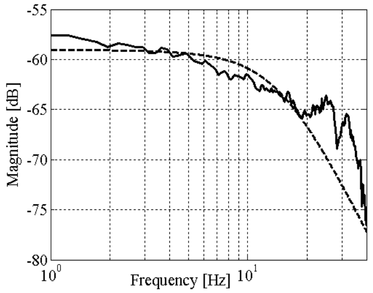


Figure 4.5: Measured (solid) and modelled (dashed) magnitude plots of the frequency response function representing presliding friction regime in the third robot joint.

To validate the friction models, two tests were performed. The first one was performed with the LuGre friction model, only. Its objective was to investigate the performance of this model when applied for online friction compensation. For illustration, the test results for the third joint are presented here. At the beginning of the test, this joint was in its equilibrium position, i.e., it was oriented vertically and downwards. Then, a sinusoidal input was applied, just to enable gravity to affect the joint motion $q_3(t)$. The friction compensation using the LuGre model was active from the very beginning of the experiment. After some time, the sinusoidal input was switched-off, but the friction compensation remained active. The joint motion was governed by the dynamics (3.20), with $\tau_3^f(t)$ representing the model-based friction compensation. The time history of the measured $q_3(t)$ is shown in Fig. 4.6. The presented motion clearly features stable oscillation around the equilibrium, which by virtue of (3.20) occurs only if friction compensation is adequate. Indeed, the resistive friction torque is so high in the third joint, and when the same experiment was performed without any friction compensation, this torque caused sudden stop of the joint motion immediately after the sinusoidal input had been switched-off. Similar result was obtained for the second joint, too. These confirm correctness of the LuGre friction models associated with joints 2 and 3.

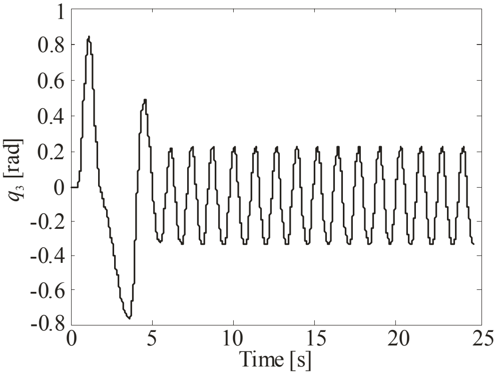


Figure 4.6: Stable swinging of the third joint with LuGre model-based friction compensation.

The second validation test should show how accurate the friction models reconstruct the experimentally measured $q_i(t)$. For illustration, the validation of the three-sigmoidal-functions friction model associated with the third robot joint is presented here. In the experiment, the torque $\tau_3(t)$ shown at the top in Fig. 4.7 was applied as the feedforward input. Gravity was compensated by closing the feedback loop via $\hat{c}_3 \sin q_3$, where $\hat{c}_3$ is the estimate given in Table A.2. During the experiment, the response $q_3(t)$ was measured. The velocity $\dot{q}_3(t)$ was determined by numerical filtering–the result of numerical differentiation of $q_3(t)$ was processed with a Butterworth filter of the fourth order. The identical input $\tau_3(t)$ was applied to system (3.20), employing the friction model (3.19) which parameters took values given in Table A2. The simulated fractions of $\tau_3(t)$ are presented in Fig. 4.7: friction torque $\tilde{\tau}_3^f(t)$ computed via (3.19), gravitational load $\hat{c}_3 \sin \tilde{q}_3(t)$, and inertial load $\hat{J}_3 \ddot{\tilde{q}}_3(t)$. Here, '~' denotes simulated variables. The plots shown in Fig. 4.7 reveal significant contribution of friction to the joint dynamics. Similar contribution of friction can be observed in the other joints, too. To verify that the estimates from Table A.2 are correct, the measured $q_3(t)$ and $\dot{q}_3(t)$ were compared with the simulated $\tilde{q}_3(t)$ and $\dot{\tilde{q}}_3(t)$. Two simulations were carried out: in the first one, the friction torque $\tilde{\tau}_3^f(t)$ was disregarded; the second simulation took $\tilde{\tau}_3^f(t)$ into account. The plots presented in Fig. 4.8a correspond to the first simulation and show no match between the measured and the simulated motion coordinates. On the other hand, $\tilde{q}_3(t)$ and $\dot{\tilde{q}}_3(t)$ obtained in the second simulation, presented in Fig. 4.8b, are in high agreement with the measurements, which validates the friction model with the estimated parameters. Similar results hold for the other joints, too, while no essential improvements were observed using the LuGre friction model. Similar performance of the three-sigmoidal-functions model and of the LuGre model is not surprising, since they were validated in the motion task in which the sliding friction regime had a dominant role. Apparently, both models perform well in the sliding friction regime, while the LuGre model is supposed to perform better during the stick-slip regime of motion, i.e., when the presliding friction behaviour is more involved.

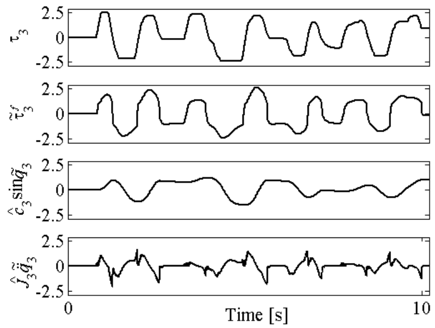
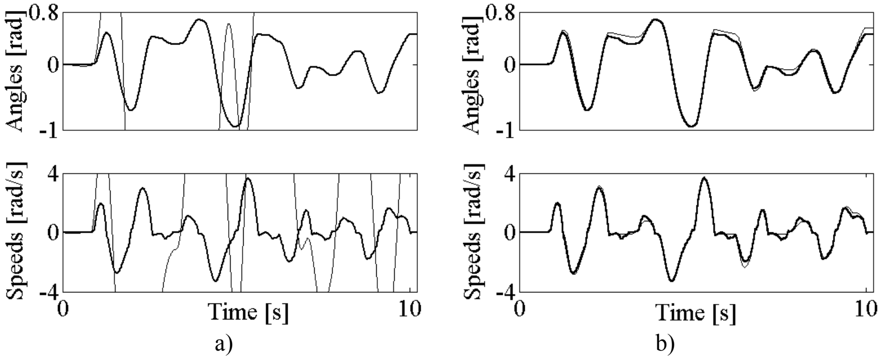


Figure 4.7: The three-sigmoidal-functions friction model validation for the third joint: the input torque and its friction, gravity, and inertial fractions.



a) b)

Figure 4.8: Measured (thick) and simulated (thin) angular displacements and velocities in the third joint: a) friction torque disregarded in simulation and b) friction torque taken into account in simulation.

Although motion tasks with very low velocities and prone to stick-slip are important in robotics (e.g., welding), this thesis gives emphasis to faster motions, since they can excite flexible dynamics and might require advanced control concepts to be realized adequately. Along such motions, the three-sigmoidal-functions model gives a satisfactory performance. Moreover, this model is static and it requires no integration of model equations, which makes its implementation much simpler than of the dynamic LuGre model. Consequently, in the next chapters, the three-sigmoidal-functions model will be primarily used for friction compensation.

4.5 Estimation of the base parameter set

As explained in subsection 3.4.3, a reliable estimation of the elements of the BPS (A.6) requires a custom designed excitation trajectory to be realized in the identification experiment. For the RRR robot, motions $q_1(t)$, $q_2(t)$, and $q_3(t)$ of the excitation trajectory were postulated as in (3.46), with the frequency resolution of $\Delta f_i = 0.1$ [Hz] ($i = 1,2,3$). Thus, each motion had a cycle period of 10 [s]. From an experimental study, it was known that the motions with the harmonic content below 10 [Hz] hardly excite the flexible dynamics. Consequently, the number of harmonics was selected as $N_i = 100$ ($i = 1,2,3$). The next step was computing the coefficients $q_{0,i}$, $a_{i,j}$, and $b_{i,j}$ ($i = 1,2,3$;

$j=1,\ldots,N_i$) in (3.46), that minimise the condition number of the information matrix $\mathbf{\Phi}$ given by (3.28). This matrix was created by vertical stacking the regression matrices $\mathbf{R}\left(\mathbf{q}^{exc}(t_k),\dot{\mathbf{q}}^{exc}(t_k),\ddot{\mathbf{q}}^{exc}(t_k)\right)$, with $\mathbf{q}^{exc}=[q_1^{exc}\ q_2^{exc}\ q_3^{exc}]^T$ and $t_k \in \{0,\ 0.1,\ 0.2,\ \ldots,\ 10\}$ [s]. The elements of $\mathbf{R}$ are given by (A.7). The $q_{0,i}$, $a_{i,j}$, and $b_{i,j}$ coefficients were found using the routine for constrained optimization *fmincon.m* implemented in *Optimization Toolbox for Matlab* [2]. The optimization takes care about constraints on permissible velocity and acceleration ranges in the robot joints: $|\dot{q}_1| \le 2\pi$ [rad/s], $|\dot{q}_2| \le 3\pi$ [rad/s], $|\dot{q}_3| \le 3\pi$ [rad/s], and $|\ddot{q}_i| \le 60$ [rad/s^2] ($i=1,2,3$). The achieved (suboptimal) condition number was 3.1. The determined coefficients were inserted into (3.46) to give the motions shown in Fig. 4.9.

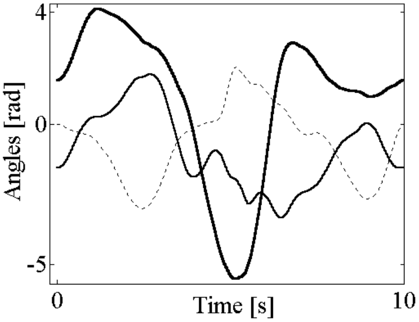


Figure 4.9: The motions of the excitation trajectories: q_1^{exc} - thick solid, q_2^{exc} - thin solid, and q_3^{exc} - dotted.

4.5.1 Batch least-squares estimation of the base parameter set

The excitation trajectory was realized with the feedback controller (3.47), which used the online reconstructed joint motions and velocities as feedback coordinates. The reconstruction was done according to the equations (3.50)-(3.53). Fig. 4.10a presents the difference between the measured and reconstructed joint motions $\tilde{y}_i$ and $\bar{q}_i(t)$ ($i=1,2,3$), respectively. Since all these differences are below 5×10^{-4} [rad], the reconstruction accuracy can be considered as good enough. Figs. 4.10b,c show $\dot{\bar{q}}_i(t)$ and $\ddot{\bar{q}}_i(t)$ ($i=1,2,3$) to indicate a low presence of noise in the reconstructed coordinates, which is important for the reliability of estimation.

The PD gains in (3.47) were tuned with "trial and error" to ensure reasonable tracking of the excitation trajectory, with a low level of noise in the control inputs τ_i ($i=1,2,3$) presented in Fig. 4.11. The friction torques $\tau_i^f(t)$ ($i=1,2,3$), shown in Fig. 4.12, were reconstructed online based on the model (3.19) with the parameters given in Table A.2.

After collecting all the data used in (3.28) and (3.29), the batch least-squares estimation (3.30) of the BPS elements could be carried out. To facilitate estimation, one may take advantage of the symmetric mass distribution in the links 2 and 3, along the positive and negative directions of the axes ${}^2y$ and ${}^3y$, respectively (see Fig. 4.2 and photo of the RRR robot in Fig. 1.1). These symmetries imply that y_2 and y_3 coordinates of the centres of masses of links 2 and 3 are identically zero. By virtue of the (A.6), it follows that the elements p_4 and p_{15} of the BPS are also identically zero. This reduces the dimensionality of the problem, as instead of all 15 parameters, only 13 elements of the BPS need to be estimated. As apparent from the plots shown in Fig. 4.13, the least-squares estimation results in the convergence of all 13 estimated elements of $\hat{\mathbf{p}}$. Their steady-state values are given in Table A.3, in the Appendix.

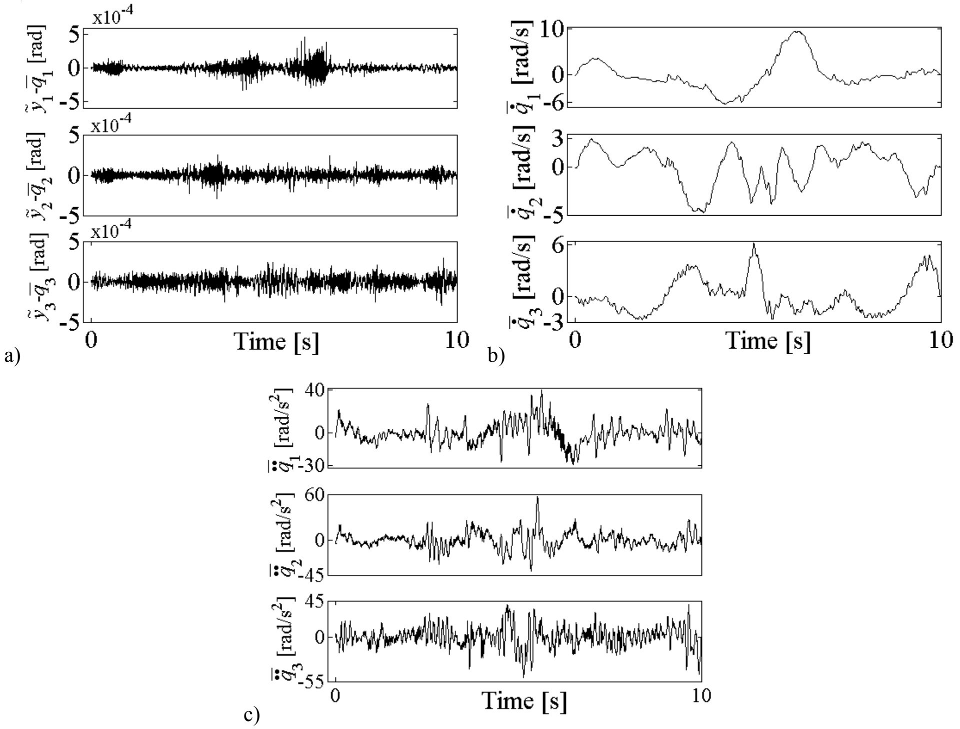


Figure 4.10: On-line reconstructed motion coordinates: a) accuracy in reconstructing the joint motions – $y_i(t) - \bar{q}_i(t)$ ($i = 1,2,3$), b) joint velocities, and c) joint accelerations.

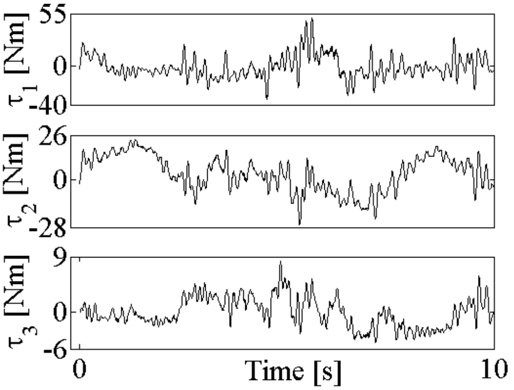


Figure 4.11: Control inputs from the identification experiment.

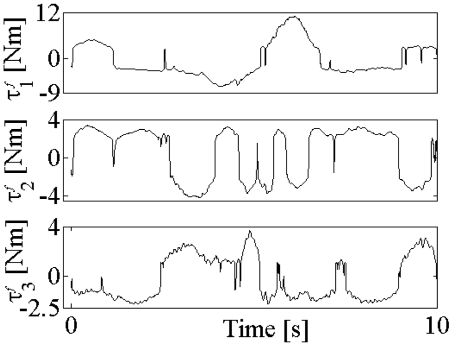


Figure 4.12: On-line reconstructed friction torques.

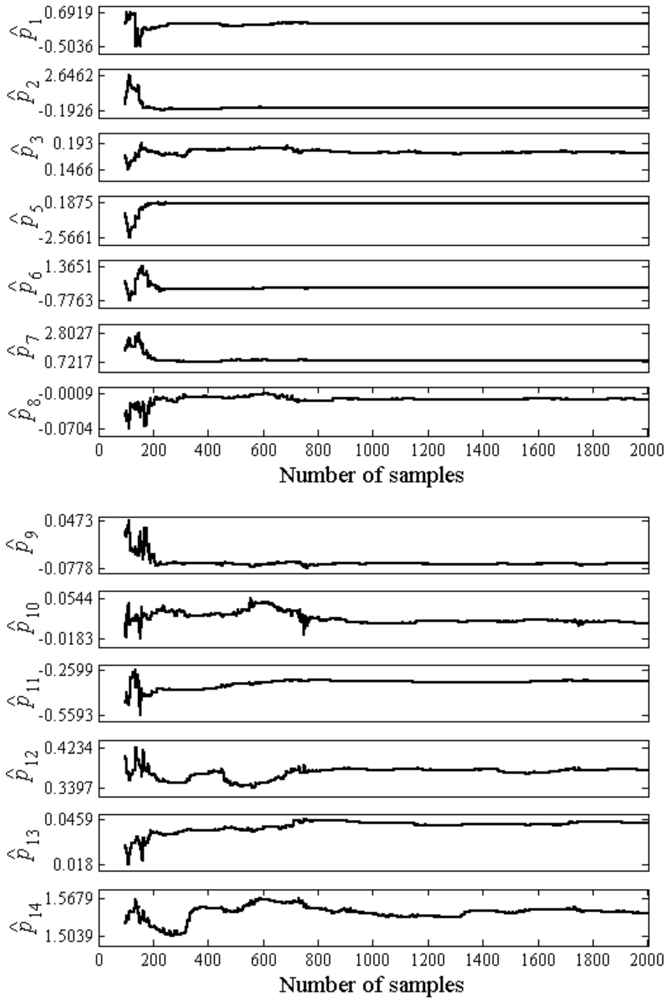


Figure 4.13: The convergence of the BPS estimates obtained with the batch least-squares method.

4.5.2 Online estimation of the base parameter set using the batch adaptive control

To facilitate a regressor representation of the RRR robot rigid-body dynamic model with friction (3.32), required by the online estimation method presented in subsection 3.4.2.2, friction in each joint i ($i = 1,2,3$) was described using the basic model (3.16) covering only Coulomb and viscous effects. This model admits a linear representation in its parameters $\alpha_{1,i}$ and $\alpha_{2,i}$:

$$\left(\mathbf{r}_i^f(q_i, \dot{q}_i, t)\right)^T \mathbf{p}_i^f = \tau_i^f , \tag{4.7a}$$

with

$$\mathbf{r}_i^f = [\operatorname{sgn} \dot{q}_i \;\; \dot{q}_i]^T \text{ and } \mathbf{p}_i^f = [\alpha_{1,i} \;\; \alpha_{2,i}]^T . \tag{4.7b}$$

Consequently, the vector $\mathbf{p}^*$ of parameters to be estimated had 19 elements (13 of the BPS and 3×2 friction parameters).

The excitation trajectory used in the batch least-squares estimation, shown in Fig. 4.9, was also used for online parameter estimation. To demonstrate the full potential of online estimation, three different identification experiments were performed. In the first experiment, the last link was removed from joint 3. In the second experiment, this link of 1.22 [kg] was attached again to the joint. In the third experiment, an extra load of 0.45 [kg] was attached to the tip of the link. For each experiment, $\mathbf{p}_0^* = \mathbf{0}_{19}$ was used as the initial guess of the parameter vector. Consequently, the extra input $\mathbf{v}$ in (3.33) was also zero during the first repetition of the excitation trajectory. The feedback gains $\mathbf{K}_p$ and $\mathbf{K}_d$ in (3.33) were weakly tuned for each joint separately, such as to stabilize the system and to avoid the excitation of parasitic dynamics (flexibilities). The scalar β in (3.38) and the weighting matrix $\mathbf{T}$ in (3.43) were tuned such as to achieve fast convergence of the estimates to steady-state values.

Fig. 4.14 shows the results of the identification experiments with the different loads. By inspection of the given plots, it can be seen that the estimates are updated after every 10 [s], i.e., at the end of each repetition of the excitation trajectory. The estimates reach their steady-state values after as few as nine trials. It can also be noticed that the steady-state values of $\hat{p}_2$, $\hat{p}_3$, $\hat{p}_4$, $\hat{p}_8$, $\hat{p}_{11}$, $\hat{p}_{12}$, and $\hat{p}_{13}$, are not the same when estimated with different loads. This can be expected, as these BPS elements depend on the inertial properties of the third robot link, as seen from (A.6). The variances in the load attached to the last joint clearly induce differences in the estimates. The steady-state values of the estimates obtained in three experiments are given in Tables A.4, A.5, and A.6.

One may observe that the estimates achieved with the batch least-squares algorithm and with the online method, presented in Tables A.3 and A.5, respectively, are not identical, but are in good agreement. As discussed in [170], it is hard, if even impossible, to estimate physical values of all elements of the BPS. The elements of small magnitude may have insignificant contribution to the total robot dynamics and thus it is difficult to obtain accurate values for them. In the case of the RRR robot, p_6 is an element of small magnitude. Still, if the estimates are determined for the purpose of model-based control, the parameter uncertainty corresponding to BPS elements of small magnitude should not be, in principle, important for the control performance [170]. To verify whether this is the case for the RRR robot, the obtained estimated were experimentally validated in a writing task, and the achieved results are presented next.

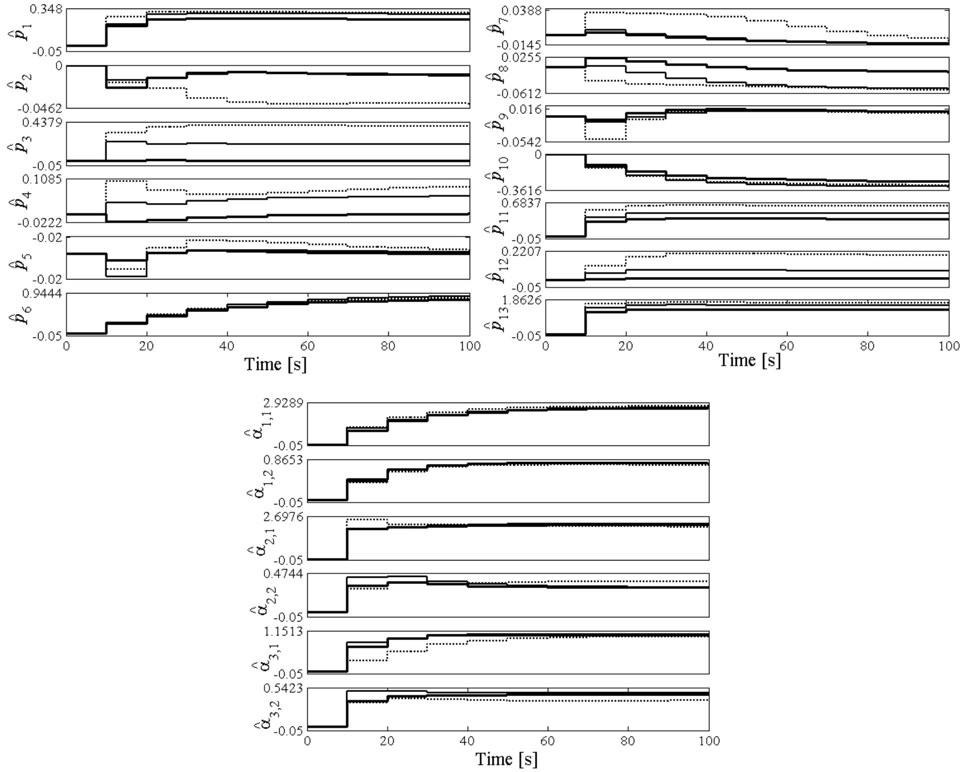


Figure 4.14: The convergence of the estimates of the BPS elements and of friction parameters obtained with online estimation: no last link (black thick), with the last link (black thin), and with an extra mass at the tip of the last link (dotted).

4.5.3 Experimental model validation

To evaluate the quality of the estimates, the validation experiment was performed as explained in section 3.6. Under PD control (3.55), the RRR robot was realizing the writing task shown in Fig. 3.4, which joint motions and velocities are presented in Figs. 4.2a,b. The objective is to check how well the rigid-body dynamic model (A.6),(A.7) with friction models (3.19), respectively, (3.16), reconstructs PD generated torques.

The torques generated with the PD controllers and the torques rendered by the dynamic model whose parameters were estimated as explained in subsection 4.5.1, are presented in Figs. 4.15a-c. Apparently, there is a close match between the experimental and the reconstructed torques in all three joints. A similar match can be observed in Figs. 4.16a-c, showing experimental and model-based torques of the second joint obtained with different loads at the third joint. These loads were as in the identification experiments explained in subsection 4.5.2. In each validation experiment, the model parameters were assigned the estimates obtained in subsection 4.5.2, corresponding to the given load. In Fig. 4.16c, it can be seen that the output of the PD controller is clipped at its maximum permissible value (30 [Nm]). The saturation occurs as the extra load attached at the robot-tip requires more actuation than permissible. Under normal operation conditions, i.e., without extra load at the tip (see Figs. 4.16a,b), no saturation occurs. High agreement between the measured and the reconstructed control torques can also be observed for the other two joints, in the validation experiments with all three loads.

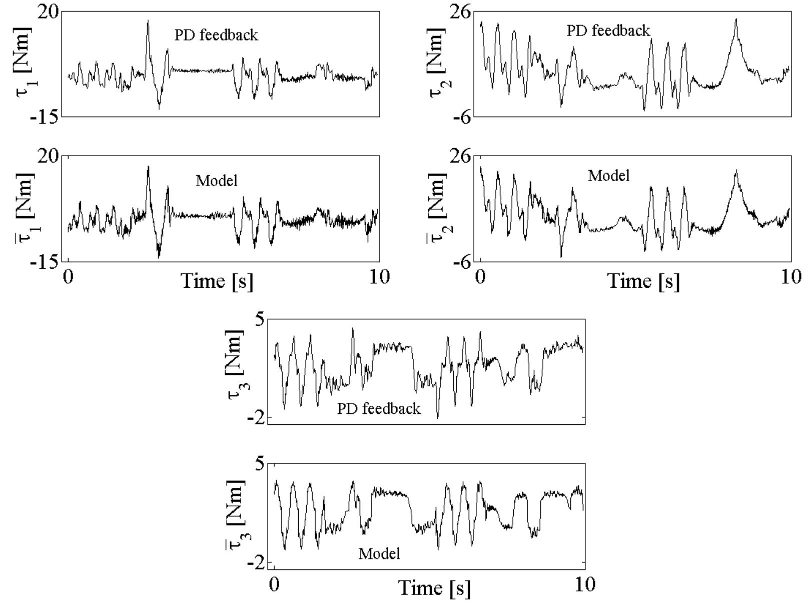


Figure 4.15: Results of the validation experiments with the writing task: the control torques were produced with the decentralized PD control, and with the rigid-body dynamic model in closed-form augmented with the three-sigmoidal-functions friction model; the batch least-squares estimates of the base parameter set were used.

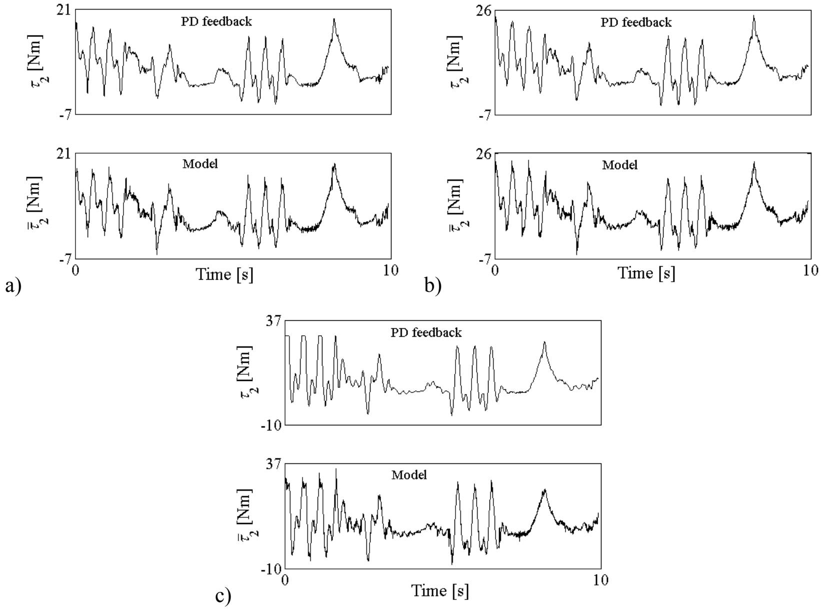


Figure 4.16: Results of the validation experiments with the writing task; the control torques were produced with the decentralized PD control, and with the rigid-body dynamic model in closed-form augmented with the basic friction model; the online obtained estimates of the base parameter set and of friction parameters were used: a) without the last link attached, b) with the last link attached, c) with the last link and an extra mass attached.

The presented validation results imply a satisfactory estimation of the rigid-body dynamics with friction. Moreover, since the rigid-body dynamic model with friction reconstructed the control torques as the experiment was in progress, it follows that the model itself admits on-line implementation. Sufficient quality in reconstructing the realistic rigid-body dynamics with friction and the real-time capability, justify implementation of the model for control purposes. To investigate the performance of model-based motion control, another validation experiment was carried-out. The rigid-body dynamic model (A.6),(A.7) with friction (3.19) was implemented as a feedforward controller to augment the basic PD feedback input:

$$\boldsymbol{\tau}(t) = \mathbf{R}\left(\mathbf{q}_r(t), \dot{\mathbf{q}}_r(t), \ddot{\mathbf{q}}_r(t)\right)\mathbf{p} + \boldsymbol{\tau}^f(t) - \mathbf{K}_p\left(\overline{\mathbf{q}}(t) - \mathbf{q}_r(t)\right) - \mathbf{K}_d\left(\overline{\dot{\mathbf{q}}}_r(t) - \dot{\mathbf{q}}_r(t)\right). \qquad (4.8)$$

The experimental servo set-up corresponding to the model-based controller (4.8) is presented in Fig. 4.17. The controller was tested in the same writing task considered before. Six control cases were studied; the writing task was realized via: 1) PD feedback only, 2) PD feedback augmented with model-based friction compensation, 3) PD feedback augmented with model-based compensation of friction and inertial effects, 4) PD feedback augmented with model-based compensation of friction and centripetal/Coriolis effects, 5) PD feedback augmented with model-based compensation of friction and gravity effects, and 6) PD feedback augmented with the complete dynamic model with friction. Individual dynamic effects were extracted from model (A6),(A7), as explained in subsection 4.2.2. Identical PD gains were used for all case studied. These gains were weakly tuned, allowing a considerable discrepancy between the reference and executed motions, if the handwriting was executed using the PD feedback only. The reconstructed joint motions in all six case studies were processed via the forward kinematics model (A.2), enabling visualization of the executed letters presented in Figs. 4.18a,b. By inspection of the given plots, one notices that the first four case studies, shown in Fig. 4.18, achieve similar accuracy in handwriting. Thus, augmenting the PD feedback with the model-based friction compensation (case 2), as well as separate model-based compensation of inertial (case 3) and centripetal/Coriolis effects (case 4), do not improve motion performance. However, the PD feedback augmented with the model-based compensation of friction and gravity effects (case 5 in Fig. 4.18b) significantly improves the accuracy of handwriting. This indicates that use of the model-based controller in addition to the basic PD feedback contributes to better motion performance. As apparent in Fig. 4.18b, the best accuracy was achieved with the PD feedback augmented with the complete model (case 6). In this case study, there is no visible discrepancy between the reference and executed sequences of letters. This demonstrates that the motion controller implementing the complete dynamic model with friction achieves performance superior to controllers implementing just fractions of the complete model. Numerical verification of the given statements is presented in Table 4.3. This table shows standard deviations of the Cartesian errors between the reference and the reconstructed tip motions. The standard deviations corresponding to the last two cases are clearly smaller than the deviations achieved in the cases 1-4. Finally, it can be claimed that the estimated dynamic model successfully passed the experimental verification with the writing task.

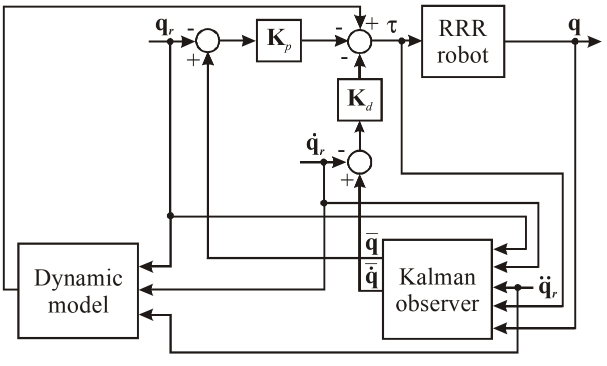


Figure 4.17: Servo set-up for experimental model-based control.

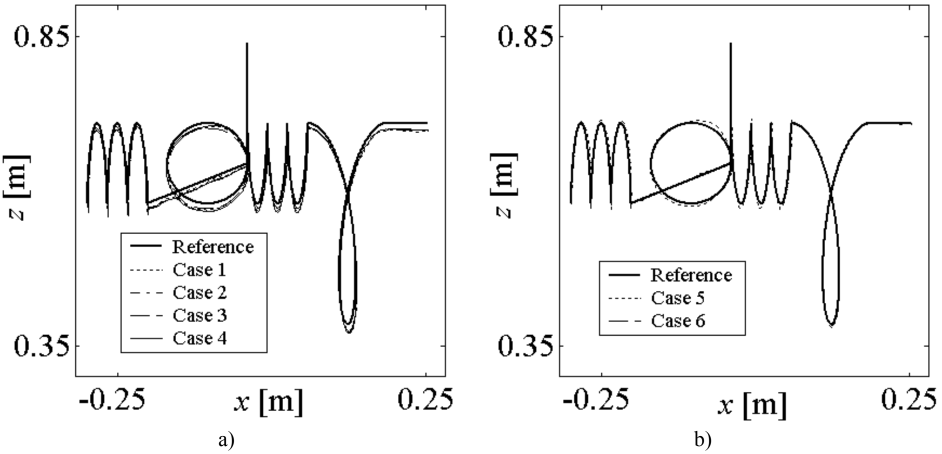


Figure 4.18: Experimental results of model-based control.

TABLE 4.3: STANDARD DEVIATIONS OF THE TRACKING ERRORS IN THE TASK SPACE X10^{-3} [M]

Axis	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
x	1.0586183	0.8937624	0.8514829	0.9144705	0.7174938	0.3269994
y	1.7499008	1.6946019	1.4393078	1.7196776	0.4644522	0.2522228
z	4.5700627	3.7535460	3.3575332	3.7666039	1.7217765	1.0738162

4.6 Parasitic linear dynamics

The motivation and procedure to identify and model parasitic robot dynamics are explained in section 3.7. The identification procedure is applied to the RRR robot. For illustration, Fig. 4.19 presents frequency response functions (FRF) measured in joint 1, for two different configurations in the remaining joints. The left-hand side FRF is obtained for the configuration $[q_2, q_3] = [0, \pi/2]$ [rad], which, according to the kinematic parameterisation shown in Fig. 4.2, corresponds to a horizontal orientation of the second link and vertical upwards position of the last link. The right-hand side FRF corresponds to a fully horizontally stretched arm, i.e. $[q_2, q_3] = [0, 0]$ [rad].

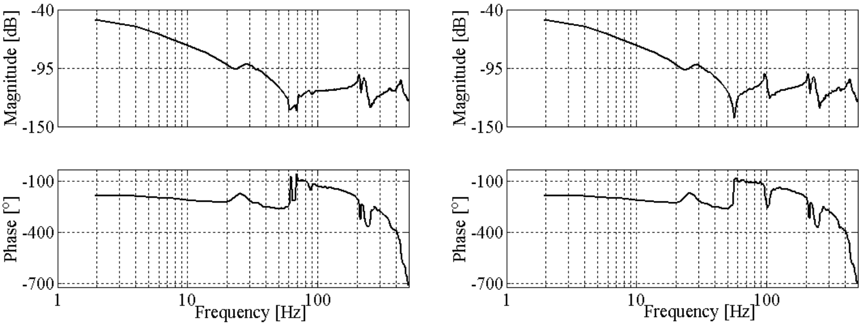


Figure 4.19: Flexible dynamics for first joint corresponding to two fixed configurations of the remaining joints: (left) $[q_2, q_3] = [0, \pi/2]$, (right) $[q_2, q_3] = [0, 0]$.

Both FRF's reveal rigid dynamics at low frequencies, i.e., the dynamics of a double integrator. Flexible effects can be observed at higher frequencies. A modest resonance is apparent at 28 [Hz], caused by insufficient stiffness in mounting the robot base to the floor. Being highly damped and with a temporary phase lead (see the phase plots), it is not a problem for stability. However, it may cause vibrations of the robot mechanism and consequently degrade the robot performance. At higher frequencies (above 80 Hz), more profound resonances can be noticed. Their frequencies and damping factors depend on the robot posture. They can contribute to amplification of vibrations and noise levels in the covered frequency ranges, which would degrade the performance of motion control. If amplified too much by feedback, these resonances may even destabilise the closed-loop system.

Apart from flexibilities, an additional peculiarity of the RRR dynamics can be observed by inspection of the phase plots shown in Fig. 4.19. Namely, there is a frequency dependent phase lag, superimposed to the phase changes due to flexibilities. The phase lag can be related to the time-delay between the control action computed in the control system described in section 4.1 and the corresponding joint angular response. The time-delay can be easily identified from the FRF's measured in the third joint, as these contain less flexible effects. The phase plot shown in Fig. 4.20 illustrates how the delay has been identified. The phase is plotted linearly against the frequency to reveal a linear phase lag. The linear lag can be described as $-360^{\circ}\delta f$, where δ is the time-delay in [s] and f is the frequency in [Hz]. As the phase drops for 360° until 500 [Hz], it follows that $\delta = 1/500 = 2$ [ms], i.e. $\delta = 2T_s$ (since $T_s = 1$ [ms]). The same time delay is also present in the other two joints.

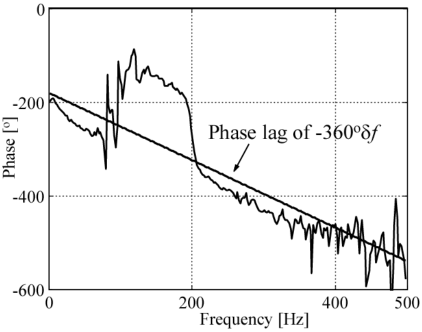


Figure 4.20: Phase plot of dynamics measured in joint 3 reveals time delay of $2T_s$.

The resonance effects and the time-delay are characteristics of the RRR robot. Similar problems are also often seen in industrial practice. To achieve motion control of high performance, these problems have to be captured by the modelling part of the control design. This part of the control design is explained in section 3.7, and here it is demonstrated for joint 1 of the RRR robot.

To determine a nominal representation of the dynamics in joint 1, we first identified FRF's $G_{1,1}^{k}(j\omega)$ ($k = 1,\ldots,N_1$) for $N_1 = 16$ static positions in joints 2 and 3: [0 0], [0 $\pi/2$],..., [$3\pi/2$ $3\pi/2$] [rad]. These positions span a complete revolution in both joints, making a grid of different robot postures. This grid can be considered dense enough to cover variations in the parasitic dynamics associated with joint 1. The identification of these dynamics was done under closed-loop conditions, as explained in section 3.7. Identification experiments were repeated for all robot postures from the grid. In each experiment, joint 1 was realizing the low speed trajectory shown in Fig. 4.21.

The identified FRF's are depicted in Fig. 4.22 with the grey lines. By inspection of their magnitude plots, one may notice that they are less steep than –2 in the frequency range up to 4 Hz. This range is within the bandwidth of the closed-loop system established in identification experiments. As pointed out in section 3.7, the spectral components of a FRF that are below the closed-loop bandwidth are not reliable. However, these components can be easily determined from the slope of the FRF components that belong to the frequency range of the rigid dynamics. Once the distinct FRF's

$G_{1,1}^{k}(j\omega)$ ($k = 1,\ldots,16$) are available, the nominal response $G_{1,1}^{\mathrm{o}}(j\omega)$ can be computed using (3.64b). The obtained $G_{1,1}^{\mathrm{o}}(j\omega)$ is shown in Fig. 4.22 with the black line, and it represents the nominal plant as a set of data points. A parametric model $P_{1,1}^{\mathrm{o}}$ (transfer function) can be fitted using an output error model structure with the least-square criterion [151]. The FRF of $G_{1,1}^{\mathrm{o}}(j\omega)$ and the Bode plot of the corresponding $P_{1,1}^{\mathrm{o}}(j\omega)$ of 7th order are shown in Fig. 4.23. The discrepancy between the unreliable data below 4 [Hz] and the fit is obvious. Above 4 [Hz], the fit is good up to 160 [Hz]. Taking into account that $G_{1,1}^{\mathrm{o}}(j\omega)$ was determined as an average of $G_{1,1}^{k}(j\omega)$ ($k = 1,\ldots,16$) in order to minimize the difference from the measured FRF's, it follows that uncertainty of the parametric model $P_{1,1}^{\mathrm{o}}$ is also minimal below 160 [Hz]. At high frequencies one may notice that the slope of the model magnitude plot is 0, while the phase is at -180°. This reveals that the transfer function $P_{1,1}^{\mathrm{o}}$ is non-minimum phase, which is the consequence of the time-delay. Another peculiarity is the lack of the high frequency roll-off. The FRF's of the plant were calculated from the measured sensitivity functions, as explained in section 3.7. In the frequency range where the sensitivity is equal to 1, i.e., at high frequencies, the magnitude of the plant's FRF's are very small. Hence, even minor errors in the sensitivity measurements at high frequencies can influence the shape of the corresponding plant's FRF's and cause lack of the roll-off effect. Fortunately, absence of this effect does not cause difficulties for model-based control designs, as it will be verified in the chapter.

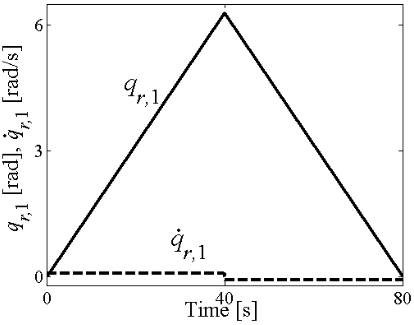


Figure 4.21: The joint trajectory used in identification experiments.

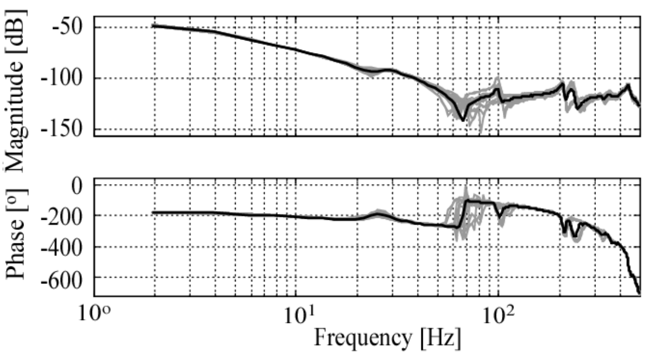


Figure 4.22: Identified FRF's $G_{1,1}^{1},\ldots,G_{1,1}^{16}$ (grey) and the nominal FRF $G_{1,1}^{\mathrm{o}}$ (black).

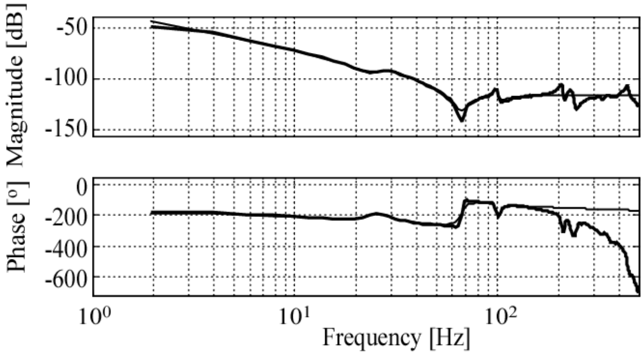


Figure 4.23: Nominal dynamics for joint 1: FRF $G_{1,1}^{\mathrm{o}}$ based on measured data (thick) and Bode plot of a parametric model $P_{1,1}^{\mathrm{o}}$ (thin).

In addition to determining $P_{1,1}^{\mathrm{o}}(j\omega)$, FRF's of the cross couplings between joint 1 and the remaining joints were measured. To illustrate this, Fig. 4.24 presents FRF's of the dynamics in joint 1 and of the interaction between joints 1 and 2, identified for $[q_2\ q_3]=[3\pi/2\ 0]$ [rad]. At each frequency within the range of the rigid dynamics (below 20 [Hz]), the amplitude of $G_{1,1}^{13}(j\omega)$ is at least two orders of magnitude higher than the amplitude of the interaction term. Above 20 [Hz] this ratio reduces, since the parasitic dynamics become more profound at higher frequencies. At the higher frequency range, interaction between joints 1 and 2 is compatible $G_{1,1}^{13}(j\omega)$, but the magnitude of the interaction is much smaller as compared with the magnitudes of the rigid-body dynamics below 20 [Hz]. Thus, the coupling effect can be disregarded. The result shown in Fig. 4.24 is also illustrative for the plant model $P_{1,1}^{\mathrm{o}}$: below 20 [Hz] the magnitude of $P_{1,1}^{\mathrm{o}}(j\omega)$ is much higher than the magnitude of any of the interaction terms in (3.58), while at very high frequencies magnitudes of the interactions are low and they hardly influence the robot motions.

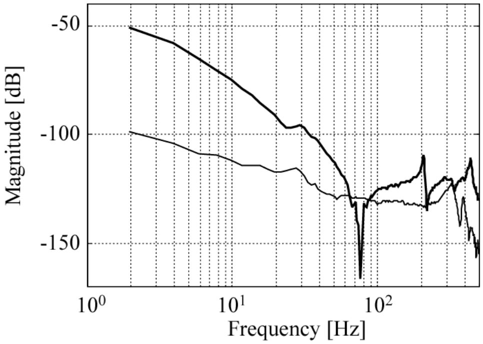


Figure 4.24: Magnitude plots of FRF's corresponding to dynamics in joint 1 (thick) and to the cross-coupling between the joints 1 and 2 (thin), identified for $[q_2\ q_3]=[3\pi/2\ 0]$ [rad].

Models of dynamics in joints 2 and 3 were obtained in a similar way. They also feature flexible effects, with negligible coupling effects at high frequencies. Consequently, the transfer function matrix given by (3.58) can practically be considered as diagonal for the RRR robot. Chapter 5 presents a number of feedback control designs applied on this robot, all resorting to the identified models.

4.7 Summary

This chapter described characteristics of the RRR robot arm, which is the experimental set-up considered hereafter in the thesis.

Closed-form kinematic and rigid-body dynamic models of the robot were presented in full detail, and their validity was established with a writing task, by comparison with program routines implemented in the Robotics Toolbox for Matlab.

Parameters of the static three-sigmoidal-functions friction model and of the dynamic LuGre friction model were experimentally estimated. Both models were validated in experiments, showing sufficient but also similar quality in describing realistic friction in the sliding friction regime. The LuGre model is expected to outperform the three-sigmoidal-functions model at very low velocities, i.e., in the stick-slip. Since the sliding regime dominates during the fast robot motions, that are of particular interest in this thesis, it appears that both friction models are suitable for online friction compensation. The static three-sigmoidal-functions friction model is computationally less expensive and consequently more preferable for online applications.

The rigid-body dynamic model reveals a high complexity of the nonlinear and coupled robot dynamics. This model was represented in the regressor form to facilitate identification of the robot inertial properties. Two identification procedures were experimentally investigated: batch least-squares estimation and online estimation based on a batch adaptive control algorithm. These procedures deliver two sets of parameter estimates. The obtained estimates are similar but not identical. The observed parameter uncertainties can be tolerated, since they do not show clear significance on the performance of model-based motion control. This was experimentally verified in the writing task. The control torques rendered by the dynamic model with parameters assigned two sets of estimates were in high agreement with the measured torques under PD control. The experimental testing of the performance of model-based motion control confirms the importance of all effects contained in the rigid-body dynamic model with friction. The best tracking performance in the writing task was achieved if the complete dynamic model with friction was implemented in the motion controller.

Finally, it was indicated that the rigid-body dynamic model with friction is still not sufficient for a perfect match to the real RRR robot dynamics, as this dynamics contains flexible effects not covered by the rigid-body dynamic model. An effect of time-delay can be observed, as well. The flexible and time-delay effects are captured by the modelling part of the control design. The obtained models are used later on in the thesis for the design of feedback motion controllers.

Chapter 5

Motion control in the robot joint space

Once the robot modelling is complete, one may proceed with the design of motion controllers that should realize the desired control specifications. Limitations of motion control designs that are generally accepted in robotics were discussed in section 2.2. The observed limitations triggered several research questions posed in section 2.3. These questions motivate investigation of the possible combination of dynamic compensation for robot nonlinearities and advanced robust model-based feedback control designs. This chapter presents several such designs, all performed in the joint space of a robot. Their common characteristic is direct accounting for the parasitic robot dynamics captured via data-driven modelling. Sections 2.3 and 2.4 explained the motives for the particular selection of these designs. Each design aims at achieving high control performance, i.e., accurate realization of the reference joint trajectory with adequate rejection of disturbances and parasitic dynamics. The theory behind each design is briefly formulated, while its practical implementation on the RRR robot is demonstrated in detail. All control designs are experimentally assessed, and the discussions of their practical effects are provided. These discussions also give answers on the research questions posed in section 2.3. The chapter is organized as follows. Section 5.1 describes the problem of motion control in the joint space and announces the solution strategy that is followed later on in this chapter. Specific requirements from feedback control designs are formulated in section 5.2. A conventional feedback design is presented in section 5.3. A feedback design based on H_∞, control theory is described in section 5.4. A μ-synthesis feedback design is considered in section 5.5. A discrete-time sliding mode feedback control design is presented in section 5.6. The chapter summary is given at the end. Parts of the material presented in this chapter have also been published in [92,94].

5.1 Problem description

The problem of maintaining a desired motion performance, in the presence of parasitic robot dynamics and disturbances, has been attracting many researchers in recent years. The parasitic dynamics encompass the uncertainties from the physical values of the robot inertial and/or friction parameters, as well as unmodelled dynamic and static effects (flexibilities and friction). As examples of disturbances, one may think of cogging forces and quantization noise. The former is common to direct-drive robots, while the latter arises if incremental encoders are used as position sensors.

Decentralized PID (proportional, integral, derivative) and its variants (PD and PI) are conventional feedback solutions in robot motion control. They are appealing because of their efficiency in tuning and low computational costs. Unfortunately, their simple form can be too restrictive to provide compensation of each effect encountered in a robotic system: friction, backlash, flexibility, time-delay, and nonlinear couplings between the joints. Usually, conventional controllers can take care of only a limited number of these effects, together with other control objectives, e.g., prescribed bandwidth and reduction of the position error. As non-compensated effects may influence the performance of robot motion control, the ultimate quality of motion performance can hardly be reached using conventional feedback only.

Industrial practice teaches us that ultimate limits of performance and robustness in motion control can be achieved only if all relevant observable characteristics of a controlled plant are captured by the control design. This reasoning has arisen during a long industrial practice in control of linear motion systems, e.g. optical storage systems [166]. The advanced control designs that take care about more versatile properties of the plant should also be capable to simultaneously realize a greater variety of control objectives than the conventional designs. However, the controllers resulting from advanced designs are typically computationally expensive, which impedes their practical implementation. Modern technology already offers solutions that facilitate online implementation of higher complexity algorithms. This makes it possible to experimentally evaluate effects of advanced designs for robot motion control.

In this chapter, several conventional and advanced feedback control designs are investigated on the RRR robot. Each design is performed in the robot joint space. The objective is to show that accounting for the robot peculiarities, e.g., flexibilities, time-delay, and the measurement noise, is beneficial for the quality of motion control. As for advanced approaches, H_∞, μ-synthesis, and discrete-time sliding mode feedback control designs are considered. The theory behind each design is already known in the literature, but their utilities in experimental motion control of direct-drive robots have hardly been explored yet. Dynamics of direct-drive robots are highly nonlinear, which particularly impedes their robust control of high performance. Inherent parasitic and disturbance effects additionally limit the feasible control performance. To deal with the nonlinearities, disturbances, and parasitics, the considered advanced control designs resort to model-based decoupling of the RRR robot dynamics, and in each of them feedback tuning is performed with a strong experimental support. These designs incorporate the following steps: (*i*) computed-torque compensation of nonlinear couplings between the robot joints using a comprehensive model of the RRR robot rigid-body dynamics with friction, (*ii*) frequency-domain identification of the remaining (flexible) dynamics, (*iii*) design of robustifying feedback controllers to meet performance and robustness specifications. These three design steps are not particularly novel as far as robust control of linear motion systems is concerned. However, they are not common in the motion control of direct-drive robots. Especially, identification of the dynamics that remain after model-based decoupling of the robot joints and the feedback design dedicated to these identified dynamics, are hardly encountered in the literature on robust robot control.

In the following, specific requirements will be given that each feedback control design should meet. Then, the particular designs will be presented. The theory behind each design will be formulated for the general case of a robotic manipulator with n degrees of freedom. The practical demonstrations of the designs will be given for the RRR robot. These demonstrations should emphasize the feedback controller tuning parts of the considered designs.

5.2 Requirements from feedback control designs

The use of computed-torque control (2.26) to compensate for nonlinear robot dynamics was addressed in subsections 2.2.2 and 2.2.4. Section 3.7 illuminated the importance of the parasitic robot dynamics not covered with the model-based compensation of nonlinearity. The parasitic dynamics of the RRR robot were presented in section 4.6 as an illustrative example. The following discussion is a natural continuation of section 3.7.

What remains after computed-torque model based compensation are mostly decoupled and linear dynamics $P_{i,i}(s)$ $(i=1,\ldots,n)$ in the robot joints. Strictly speaking, some amount of couplings $P_{i,j}(s)$ $(i=1,\ldots,n,\ j=1,\ldots,n)$ between the joints, see relations (3.58), is hardly avoided in practice. The inclusion of the coupling terms does not principally limit the feedback control designs considered in this chapter, although it makes them more involved. However, with accurate compensation of the nonlinear dynamics, each $P_{i,i}(s)$ should quantitatively dominate any $P_{i,j}(s)$, which enables n single-input, single-output (SISO) feedback designs. When dealing with practical problems, the condition of total dynamic decoupling ($|P_{i,i}| >> |P_{i,j}|$) must be verified before proceeding to the feedback design. For the RRR robot, this condition was verified in section 4.6.

Each SISO feedback control design is focused on a particular transfer function $P_{i,i}(s)$, representing the plant to be controlled. To abbreviate notation, the indices in the subscript can be replaced with a single one–hereafter, the plant for joint *i* is denoted with $P_i(s)$. As argued in section 3.7, $P_i(s)$ is a higher order dynamical system, whose resonance frequencies and their relative damping vary with robot configuration. Two procedures to identify these dynamics are suggested in section 3.7. This chapter presents several designs of the feedback controller C_i implemented in the servo-system for joint *i*, as depicted in Fig. 5.1. The objective of C_i is to realize accurate execution of a reference motion $q_{r,i}(t)$, with a minimum sensitivity to disturbances (e.g. noise) and unmodelled dynamic effects (parasitic dynamics). To enhance the performance of motion control, the known acceleration $\ddot{q}_{r,i}(t)$ is used as a feedforward control in the joint servo-loop. Therefore, the input u_i (i^{th} element of $\mathbf{u}$ in (2.26)) to the plant P_i is

$$u_i(t) = \ddot{q}_{r,i}(t) - u_i^*(t), \tag{5.1}$$

where u_i^* is the output of the feedback controller C_i based on the position error e_i

$$e_i = q_i - q_{r,i}. \tag{5.2}$$

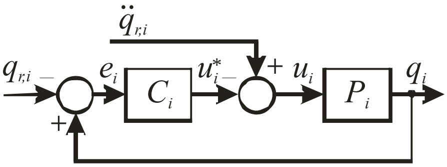


Figure 5.1: The servo control system for joint *i*.

Conventional PD, H∞, μ-synthesis, and discrete-time sliding mode designs of C_i are considered in this chapter. Each of them resorts to a loop-shaping feedback tuning strategy. This strategy enables specification of the performance objectives and feedback tuning in a very intuitive way. Its objective is to design C_i which enforces the desired shape of the transfer functions for the servo-system shown in Fig. 5.1. The considered transfer functions are

- sensitivity function $S_i = 1/(1+L_i)$;
- input sensitivity function $U_i = C_i S_i$;
- complementary sensitivity function $T_i = 1 - S_i$,

where $L_i = P_i C_i$ is the open-loop gain of the servo-loop. Key is to use the open-loop as the primary loop to shape, i.e., closed-loop specifications are transformed in the desired shape of L_i. From this, the adequate C_i follows directly given the knowledge of the plant P_i. The loop-shaping is performed according to standard design rules [166]:

r.1 stabilize the plant P_i,

r.2 ensure high open-loop gain L_i within the largest possible cross-over frequency,

r.3 allow maximum peaking in the sensitivity function S_i of 6 [dB] (stability robustness),

r.4 minimize the influence of unmodelled dynamics and disturbances on the motion control performance.

These rules are used in the control designs to be presented next.

5.3 Conventional feedback control design

The simplest use of loop-shaping is the tuning of the gains $k_{p,i}$ and $k_{d,i}$ of the conventional PD feedback controller:

$$u_i^*(t) = k_{p,i} e_i(t) + k_{d,i} \dot{e}_i(t). \tag{5.3}$$

A more advanced loop-shaping is to design a feedback controller by proper selection of the shaping filters, e.g., lead/lag, notch, low-pass, etc. An example of such a design can be found in [91]. Here, only the conventional feedback controller (5.3) is considered. As already discussed in section 3.5, the position measurement is usually corrupted with quantization noise. Furthermore, a time delay of δ [s] can be present in the robot control system, as it is the case with the RRR robot (see section 4.6). If this delay is associated with the position measurements, the following output equation can be formulated:

$$y_i(t) = q_i(t-\delta) + \eta_i(t) \quad (i = 1, \ldots, n), \tag{5.4}$$

where y_i is the position measurement corrupted with the measurement noise η_i. Forming the error e_i in (5.3) as the difference between y_i and $q_{r,i}$ does not promise a high motion performance, because of the time lag and the noise.

If a sampled time (digital) controller implementation is used, with the sampling period T_s, and if the time delay δ is an integer multiple of T_s ($\delta = pT_s$, $p \in \{0,1,2,\ldots\}$), then a Kalman observer technique similar to that explained in section 3.5 can be engaged to compensate for the time delay, as well as to optimally filter out the noise. The given assumptions are met with the RRR robot. A Kalman observer design is based on the robot dynamics that remain after application of the computed-torque control law (2.26) to the robot dynamics (2.16):

$$\ddot{\mathbf{q}}(t) = \mathbf{u}(t) + \mathbf{v}(t), \tag{5.5}$$

where

$$\mathbf{v}(t) = -\mathbf{M}^{-1}(\mathbf{q}(t))\boldsymbol{\tau}_{\mathrm{v}}(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t), t). \tag{5.6}$$

The i^{th} ($i = 1, \ldots, n$) equation of (5.5) is

$$\ddot{q}_i(t) = u_i(t) + \nu_i(t). \tag{5.7}$$

Assuming u_i of the form (5.1), and if ν_i is the integral of the zero mean white process noise ξ_i [21]

$$\dot{\nu}_i(t) = \xi_i(t), \tag{5.8}$$

then a state-space representation of (5.7),(5.8) is given by:

$$\dot{\mathbf{x}}_i(t) = \mathbf{A}_i \mathbf{x}_i(t) - \mathbf{b}_i u_i^*(t) + \mathbf{h}_i \xi_i(t);$$
$$\mathbf{x}_i(t) = \begin{bmatrix} e_i(t) \\ e_i^{\omega}(t) \\ \nu_i(t) \end{bmatrix}; \ \mathbf{A}_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}; \ \mathbf{b}_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}; \ \mathbf{h}_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{5.9}$$

where e_i is the position error defined by (5.2) and $e_i^{\omega} = \dot{e}_i$. The output equation (5.4) can be associated to (5.9):

$$\begin{aligned} y_i(t) &= q_i(t-\delta) + \eta_i(t) \\ &= q_{r,i}(t-\delta) + e_i(t-\delta) + \eta_i(t). \end{aligned} \tag{5.10}$$

By taking the assumption $\delta = pT_s$ ($p \in \{0,1,2,\ldots\}$) into account, it is possible to determine the discrete-time system

$$\begin{aligned} \mathbf{x}_i^{\exp}(k+1) &= \mathbf{E}_i^{\exp}(T_s)\mathbf{x}_i^{\exp}(k) - \mathbf{f}_i^{\exp}(T_s)u_i^*(k) + \boldsymbol{\gamma}_i^{\exp}(T_s)\xi_i(k)\,; \\ y_i(k) &= q_{r,i}(k-p) + (\mathbf{c}_i^{\exp})^T\mathbf{x}_i(k) + \eta_i(k), \end{aligned} \tag{5.11a}$$

with the state vector $\mathbf{x}_i^{\exp}$ given by

$$\mathbf{x}_i^{\exp}(k) = [e_i(k-p), e_i(k-p+1), \ldots, e_i(k), e_i^{\omega}(k), \nu_i(k)]^T, \tag{5.11b}$$

and

$$\begin{gathered} \mathbf{E}_i^{\exp} = \begin{bmatrix} \mathbf{E}_i^a & \mathbf{E}_i^b \\ \mathbf{E}_i^c & \mathbf{E}_i^d \end{bmatrix};\ \mathbf{E}_i^a = \mathbf{0}_{p\times 1};\ \mathbf{E}_i^b = [\mathbf{I}_p, \mathbf{0}_{p\times 2}];\ \mathbf{E}_i^c = \mathbf{0}_{3\times p}; \\ \mathbf{E}_i^d = e^{\mathbf{A}_i T_s} = \begin{bmatrix} 1 & T_s & T_s^2/2 \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix};\ \mathbf{f}_i^{\exp} = \begin{bmatrix} \mathbf{0}_{p\times 1} \\ \mathbf{f}_i \end{bmatrix};\ \mathbf{f}_i = \int_0^{T_s} e^{\mathbf{A}_i\varsigma}\mathbf{b}_i d\varsigma = \begin{bmatrix} T_s^2/2 \\ T_s \\ 0 \end{bmatrix}; \\ \boldsymbol{\gamma}_i^{\exp} = \begin{bmatrix} \mathbf{0}_{p\times 1} \\ \boldsymbol{\gamma}_i \end{bmatrix},\ \boldsymbol{\gamma}_i = \int_0^{T_s} e^{\mathbf{A}_i\varsigma}\mathbf{h}_i d\varsigma = \begin{bmatrix} T_s^3/6 \\ T_s^2/2 \\ T_s \end{bmatrix},\ \mathbf{c}_i^{\exp} = \begin{bmatrix} 1 \\ \mathbf{0}_{(p+2)\times 1} \end{bmatrix}. \end{gathered} \tag{5.11c}$$

Here, $\mathbf{I}_p \in \mathbb{R}^{p\times p}$ is the identity matrix, and $\mathbf{0}_{m\times n} \in \mathbb{R}^{m\times n}$ only contains zeros. With the state-space representation (5.11a-c), a Kalman observer can be designed to reconstruct the states $\mathbf{x}_i^{\exp}$, according to:

$$\begin{aligned} \hat{\mathbf{x}}_i^{\exp}(k+1) &= \mathbf{E}_i^{\exp}(T_s)\bar{\mathbf{x}}_i^{\exp}(k) - \mathbf{f}_i^{\exp}(T_s)u_i^*(k)\,; \\ \bar{\mathbf{x}}_i^{\exp}(k) &= \hat{\mathbf{x}}_i^{\exp}(k) + \mathbf{k}_i^{\exp}[y_i(k) - q_{r,i}(k-p) - (\mathbf{c}_i^{\exp})^T\hat{\mathbf{x}}_i^{\exp}(k)], \end{aligned} \tag{5.12}$$

where $\bar{\mathbf{x}}_i^{\exp}$ is the updated estimate of all states, and $\mathbf{k}_i^{\exp} \in \mathbb{R}^{p+3}$ is a vector of constant gains that are dependent on $\mathbf{E}_i^{\exp}$, $\mathbf{f}_i^{\exp}$, $\boldsymbol{\gamma}_i^{\exp}$, and on the properties of ξ_i and η_i [20,21,59].

Now, the reconstructed coordinates $\bar{e}_i$ and $\bar{e}_i^{\omega}$ can be used in the PD feedback control law

$$u_i^*(t) = k_{p,i}\bar{e}_i(t) + k_{d,i}\bar{e}_i^{\omega}(t), \tag{5.13}$$

and the loop-shaping should take into account the observer's dynamics. The block-diagram shown in Fig. 5.2 visualizes the considered control approach, featuring the vectors: $\mathbf{u}^* = [u_1^*, \ldots, u_n^*]^T$, $\bar{\mathbf{e}} = [\bar{e}_1, \ldots, \bar{e}_n]^T$, $\bar{\mathbf{e}}^{\omega} = [\bar{e}_1^{\omega}, \ldots, \bar{e}_n^{\omega}]^T$, $\mathbf{y} = [y_1, \ldots, y_n]^T$, and $\mathbf{h} = \mathbf{c} + \mathbf{g} + \boldsymbol{\tau}^f$. The matrices of position and velocity gains are formed as $\mathbf{K}_p = diag[k_{p,1}, \ldots, k_{p,n}]$ and $\mathbf{K}_d = diag[k_{d,1}, \ldots, k_{d,n}]$, respectively. Notice that a model-based compensation of the nonlinear rigid-body robot dynamics is computed along the reference joint motions and velocities rather than along the actual ones. This is common choice in practice, since it avoids the impact of disturbances inherent to the actual motion coordinates, on the input $\boldsymbol{\tau}$.

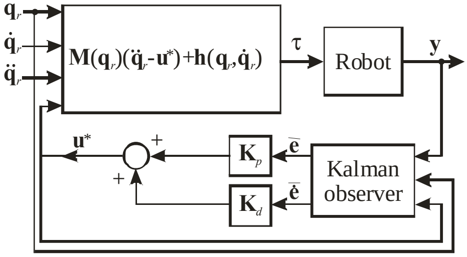


Figure 5.2: Robot motion control with PD controllers and Kalman observer in the feedback loop.

As a case-study, consider the conventional feedback control design for the first joint of the RRR robot. Similar designs are accomplished for joints 2 and 3. The design for joint 1 is based on the frequency-response function (FRF) shown in Fig. 4.23, representing its nominal dynamics. The Kalman observer is designed by taking into account the time delay of two sampling periods, which implies $p = 2$ in (5.11a-c),(5.12).

Bode plots of the PD controllers designed for joint 1 are shown in Fig. 5.3. The solid lines represent the transfer function from the error $e_1(t)$ to the control input $u_1^*(t)$ defined by (5.3). The dotted lines represent the transfer function from $y_1(t) - q_{r,1}(t - 2T_s)$ (the input to the Kalman observer, see (5.12)) to the control input u_1^* defined by (5.13). The PD gains were chosen to meet the design rules r.1-r.4 from section 5.2 as close as possible. Fine-tuning of the gains was manually done on the real system, to minimize the influences of the quantization noise, vibrations at the base, and the flexibilities at higher frequencies.

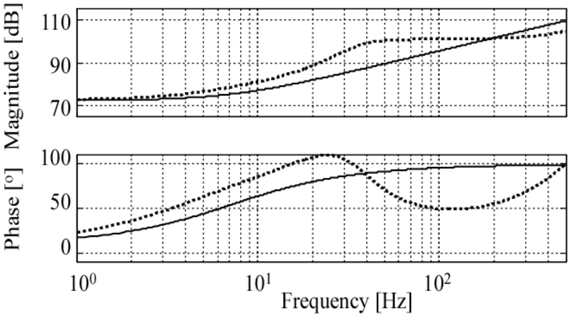


Figure 5.3: Bode plots of conventional feedback controllers defined by equations (5.3) (solid) and (5.13) (dotted).

By inspection of the Bode plots, one may notice that at low frequencies the PD controller (5.13) introduces a higher gain than the PD controller (5.3). At low frequencies, the controller (5.13) provides more phase advance than the controller (5.3). A higher gain implies better motion performance, while the phase lead implies a possibility for a higher cross-over frequency in the closed-loop. These properties can be verified from the Bode plots of the open-loop gains depicted in Fig. 5.4. By inspection of the magnitude plots, it is apparent that the cross-over frequency of the controller (5.3) is lower, and its gain is smaller below the cross-over. The higher cross-over frequency of the PD defined by (5.13) indicates a potential of this controller to realize faster robot motion. The magnitudes of the sensitivity functions of both controllers are presented in Fig. 5.5. The design rule r.3 on the maximum peaking of 6 [dB] is obviously realized in both cases. At the lower frequency range, the sensitivity achieved with (5.3) is above the sensitivity achieved with (5.13), implying a better reduction of the position error with the latter controller. However, the achieved performance is still below the one that is feasible with the advanced feedback control designs, which will be verified next.

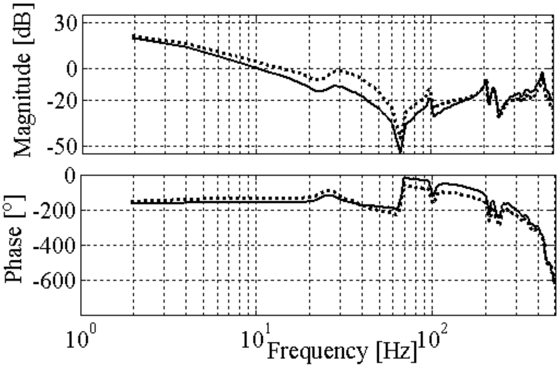


Figure 5.4: Open-loop gains of the servo control system for joint 1 achieved with the conventional feedback controllers defined by equations (5.3) (solid) and (5.13) (dotted).

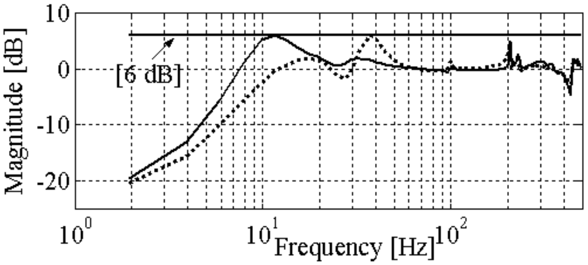


Figure 5.5: Sensitivity functions of the servo control system for joint 1 achieved with the conventional feedback controllers defined by equations (5.3) (solid) and (5.13) (dotted).

5.4 H∞ feedback control design

An optimal loop-shaping for joint i calculates a feedback controller minimising the H_2 or H_∞ norms [49,103,185] of the transfer matrix containing the weighted closed-loop transfer functions. These functions are the sensitivity, input sensitivity, and complementary sensitivity of the joint servo-system. The weightings are frequency-dependent filters, suitably selected to enforce the desired performance specifications. The availability of commercial software packages that enable H_2 or H_∞ optimization, e.g., the *μ-Analysis and Synthesis Toolbox for Matlab* [15], facilitates an optimal loop-shaping.

This section presents an optimal feedback control design that aims at shaping the sensitivity function S_i and input sensitivity function U_i of joint i ($i=1,\dots,n$) servo system. It employs two frequency dependent weighting functions W_i^S and W_i^U for performance specification. Constraining the sensitivity function enforces the desired low-frequency dynamics in the closed loop:

$$|S_i(j\omega)| \le 1/\left|W_i^S(j\omega)\right|, \ \forall\omega\in \mathbb{R}^+ \ (i=1,\dots,n), \tag{5.14}$$

where ω denotes the angular frequency defined on $\mathbb{R}^+=\{x\in\mathbb{R}|x\geq 0\}$. By shaping the sensitivity, one may impose a minimum bandwidth requirement and integral control. Attenuation of resonances at low frequencies can be enforced, as well. Bounding the input sensitivity enforces a high-frequency roll-off, which is important for robustness against high-frequency resonances and measurement noise:

$$|U_i(j\omega)| \le 1/|W_i^U(j\omega)|, \ \forall\omega\in \mathbb{R}^+ \ (i=1,\ldots,n). \tag{5.15}$$

The simultaneous specification of the closed-loop objectives (5.14) and (5.15) can be done using the block-diagram shown in Fig. 5.6. The channel from w_i to z_i^S and z_i^U is called the performance channel. By considering w_i as the input variables, and z_i^S, z_i^U as the output variables, then from Fig. 5.6 we may determine the interconnecting transfer function matrix $\mathbf{H}_i$:

$$\begin{bmatrix} z_i^S \\ z_i^U \end{bmatrix} = \mathbf{H}_i w_i, \quad \mathbf{H}_i = \begin{bmatrix} W_i^S S_i \\ -W_i^U U_i \end{bmatrix}. \tag{5.15}$$

The conditions (5.14) and (5.15) jointly hold if the H$_\infty$ norm of $\mathbf{H}_i$ is less than one:

$$\|\mathbf{H}_i(j\omega)\|_\infty < 1, \ \forall\omega\in \mathbb{R}^+ \ (i=1,\ldots,n). \tag{5.16}$$

Thus, the optimal loop-shaping procedure seeks for a controller C_i which stabilizes the servo-loop shown in Fig. 5.6 and minimizes the given H$_\infty$ cost function, ensuring the minimum is below 1. This optimization problem is known as the mixed-sensitivity problem [103].

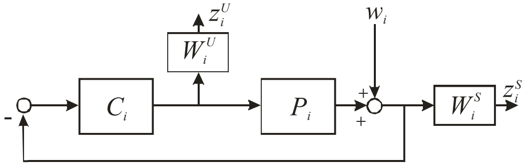


Figure 5.6: A set-up for H$_\infty$ servo design.

The loop-shaping is performed by tuning the weightings W_i^S and W_i^U in disturbance-based control design cycles [166]. This data-driven tuning takes care about parasitic and disturbance effects observed in the experiments on the robot. The tuning has to enforce adequate compensation for parasitics and disturbances. The disturbance-based control design cycles are performed according to the following algorithm:

1. a conventional PD feedback controller is used to stabilise the robot in the initial design cycle; the robot is moving along trajectories spanning the whole joint space; the position error is observed and collected, and *cumulative power spectra* (CPS's – cumulative sums of all components of the auto-power spectrum) of all error components are calculated; frequencies above the bandwidth of the joint references at which the CPS's have steeper slopes reveal undesirable effects (noise, cogging force, vibrations), which should be tackled with the feedback,
2. the weightings W_i^S and W_i^U are shaped to account for the observed undesirable effects,
3. optimal controllers C_i minimising $\|\mathbf{H}_i(j\omega)\|_\infty$, with $\mathbf{H}_i$ given by (5.15), are computed for $i=1,\ldots,n$ and implemented on the manipulator to realize the reference trajectory in the new design cycle,
4. CPS's of the new observed position errors are calculated and evaluated; if the design specifications have not been met yet, the weightings W_i^S and W_i^U are adjusted,
5. the steps 3 and 4 are repeated until further improvement of the motion performance can not be achieved; the latest determined C_i ($i=1,\ldots,n$) are adopted as the feedback controllers.

The block-diagram shown in Fig. 5.7 illustrates the robot motion control with robust controllers $\mathbf{C}=diag[C_1,\ldots,C_n]$ in the feedback loop. As in Fig. 5.2, $\mathbf{h}=\mathbf{c}+\mathbf{g}+\boldsymbol{\tau}^f$ and a model-based compensation of nonlinear rigid-body robot dynamics is computed along the reference trajectory.

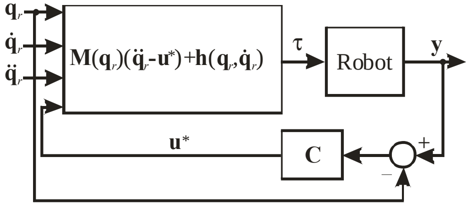

$\mathbf{q}_r$
$\dot{\mathbf{q}}_r$
$\ddot{\mathbf{q}}_r$
$\mathbf{M}(\mathbf{q}_r)(\ddot{\mathbf{q}}_r-\mathbf{u}^*)+\mathbf{h}(\mathbf{q}_r,\dot{\mathbf{q}}_r)$
τ
Robot
y
u*
C
+
–

Figure 5.7: Robot motion control with robust feedback controllers.

As a case-study, the given iterative procedure is demonstrated for joint 1 of the RRR robot. The feedback designs for other two joints are accomplished in a similar fashion. A motion task shown in Fig. 5.8 is suitable for experiments performed during the disturbance-based control design cycles. In this task, the joints perform fast displacements of π [rad] (duration 1 [s]), each time in opposite direction, with cosine velocity profiles. Thus, most of the joint space is spanned. The initial/terminal velocities and accelerations are at zero. Such a vicious movement requires full authority of the actuators as the velocities and accelerations are at maximum. Accurate realization of the fast reference trajectory calls for high bandwidth of the feedback control systems. Also, during the fast movements, flexible dynamics of the robot can easily be excited.

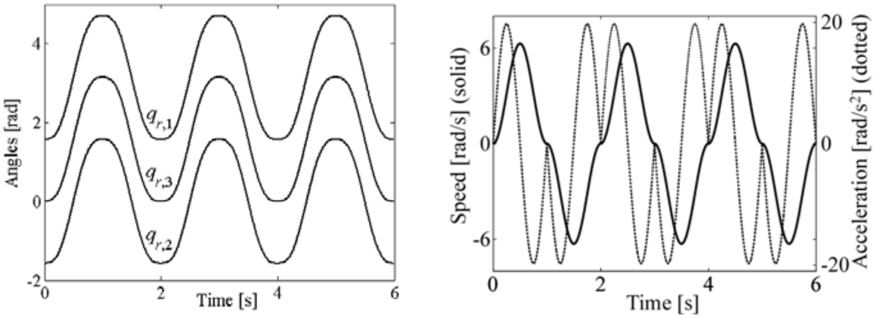


Figure 5.8: The motion task with cosine velocity profiles.

The disturbance-based control design cycles have been initialized using the PD controller shown in Fig. 5.3 with the solid lines. The parametric plant model of the 7th order, whose Bode plot is shown in Fig. 4.23, was used to solve the mixed-sensitivity problem in each design cycle. The weightings W_1^S and W_1^U were adjusted in each cycle, such as to meet the design rules r.1-r.4. as close as possible. Their final shapes are shown in Fig. 5.9. The 5th order weighting W_1^S should enforce integral action in the feedback controller, together with as high as possible reduction of the position error at low frequencies. Furthermore, it should enforce attenuation of the vibrations at 28 [Hz]. The cause of these vibrations is explained in section 4.6. The weighting W_1^U of the 3rd order should enforce suppression of the resonances at high-frequencies. The controller C_1 was calculated using the *μ-Analysis and Synthesis Toolbox for Matlab*. Its original order was 15 (plant 7 + W_1^S 5 + W_1^U 3). To facilitate experimental implementation of C_1, the original order was reduced to 11, using the model reduction routines implemented in the *μ-Analysis and Synthesis Toolbox for Matlab*. The order reduction took into account an acceptable degradation of the control performance. The experimental C_1 is presented in Fig. 5.10. Its shape is clearly more complex than of the conventional controllers depicted in Fig. 5.3, indicating a potential of this controller to simultaneously handle more versatile control objectives than the conventional ones. It is also worth noticing that the design of the

H_∞ controller does not require an explicit treatment of the time-delay problem, which was the case with the conventional controller. It considers the delay as an inherent property of the controlled plant and creates an optimal controller that handles this effect together with other control objectives.

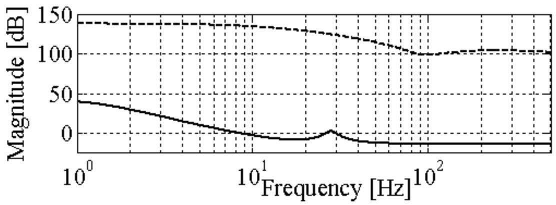


Figure 5.9: The weighting filters W_1^S (solid) and $(W_1^U)^{-1}$ (dashed).

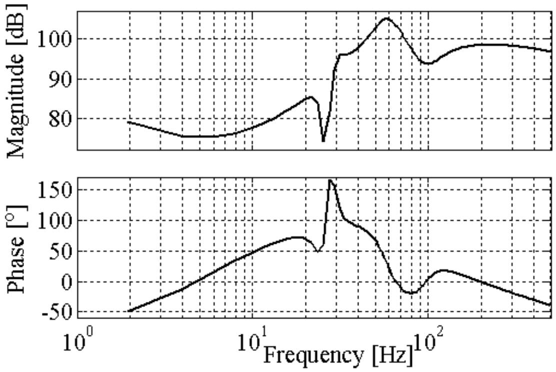


Figure 5.10: Bode plot of C_1 feedback controller calculated by H_∞ optimization during disturbance-based control design cycles.

Fulfilment of several control goals using the H_∞ controller C_1 is demonstrated by the Bode plot of the open-loop gain of joint 1 servo-system and by the magnitude plot of the sensitivity function, shown in Fig. 5.11 and Fig. 5.12, respectively. These are computed using the frequency-response function of the nominal dynamics for joint 1, shown in Fig. 4.23. For comparison, Figs 5.11 and 5.12 also show the plots that correspond to the conventional PD controllers presented in the previous section. The cross-over frequencies achieved with the H_∞ controller are higher than with the conventional feedbacks, see Table 5.1, which means that the H_∞ controller can realize faster motions more accurately. Only in the last joint, PD defined by (5.13) and the H_∞ controller yield identical cross-over. As seen from Fig. 5.11, the H_∞ controller has higher open-loop gain than the PD controllers below the cross-over. The higher gain is due to the integral action. Consequently, Fig. 5.12 indicates smaller sensitivity with the H_∞ controller than with the conventional ones below the cross-over. The higher gain and smaller sensitivity imply that H_∞ feedback better reduces the position error. Beyond the cross-over, the H_∞ controller does not excite parasitic dynamics (flexibilities). The H_∞ controllers in the other two joints also achieve higher open-loop gains and smaller sensitivities below the cross-over, while they do not excite high-frequency resonances. These are verified by experimental results, presented next.

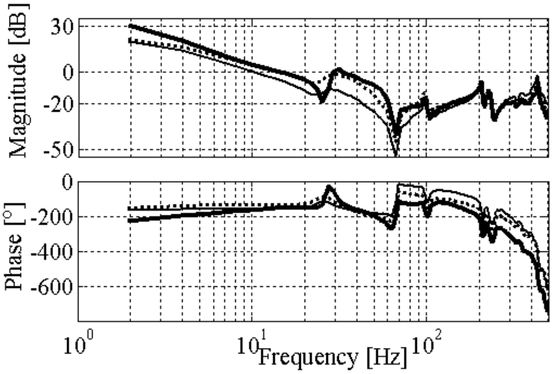


Figure 5.11: Open-loop gains of the servo control system for joint 1 achieved with the H_∞ feedback controller (thick solid) and with the conventional feedback controllers defined by equations (5.3) (thin solid) and (5.13) (dotted).

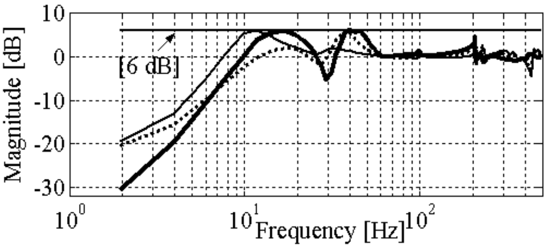


Figure 5.12: Sensitivity functions of the servo control system for joint 1 achieved with the H_∞ feedback controller (thick solid) and with the conventional feedback controllers defined by equations (5.3) (thin solid) and (5.13) (dotted).

TABLE 5.1: THE CROSS-OVER FREQUENCIES IN [HZ]

i	PD eq. (5.3)	PD eq. (5.13)	H_∞
1	10	14	15
2	10	14	19
3	10	20	20

Fig. 5.13 presents the joint position errors obtained in the motion task depicted in Fig. 5.8. The errors achieved with the H_∞ controller are compared with the errors obtained with PD controller defined by (5.13). Because of smaller gain below the cross-over, PD defined by (5.3) produces even higher errors than the PD controller defined by (5.13), so there is no need to plot them here. By inspection of the plots shown in Fig. 5.13, it becomes clear that the H_∞ controllers realize the reference motion more accurately. Integral action of these controllers especially contributes to the higher gains. The error ranges, given in Table 5.2, verify the superior motion performance with the H_∞ controllers. In joints 1 and 2, these controllers achieve errors within the range $[-1.1,1.1]\times10^{-3}$ [rad]. The range of error in joint 3 is less than two times that of the first two joints. Such accuracy can be considered as a very good for a direct-drive robot.

The contribution of parasitic effects to the motion performance can be evaluated from the CPS's of the position errors. Fig. 5.14 shows the CPS's of the errors in joint 1 achieved with both PD controllers and with the H_∞ controller. It is apparent that most of the errors' frequency contents lie within the bandwidth of the joint references (up to 5 [Hz]). Since above the bandwidth none of the

CPS's abruptly changes its slope, it appears that the influence of the parasitic effects on the trajectory execution is eliminated, i.e., the design rule r.4 is met with all feedback controllers. Inspection of the given plots reveals the lowest energy content of the CPS corresponding to the H_∞ controller. This can be quantitatively verified by comparing the final CPS values in all joints, presented in Table 5.3. The final CPS's represent the variances of the position errors. As is obvious from the table, the variances achieved with the H_∞ controllers are dramatically lower than the variances obtained with the conventional controllers.

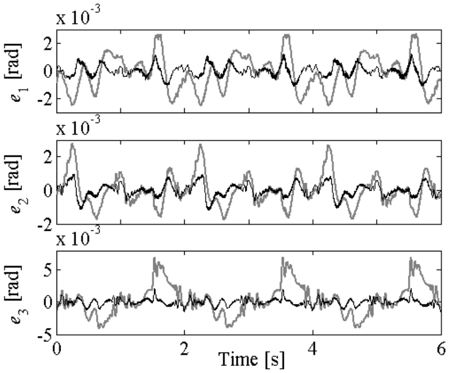


Figure 5.13: Position errors in the motion task with cosine velocity profiles achieved with the PD defined by (5.13) (grey) and with the H_∞ controller (black).

TABLE 5.2: RANGES OF THE JOINT ERRORS IN $\times 10^{-3}$ [RAD]

i	PD eq. (5.3)	PD eq. (5.13)	H_∞
1	[-5.3, 4.3]	[-2.5, 2.7]	[-1.1, 1.1]
2	[-3.9, 2.9]	[-1.7, 2.8]	[-1.1, 1.1]
3	[-10.2, 17.7]	[-4.1, 7.0]	[-1.5, 2.1]

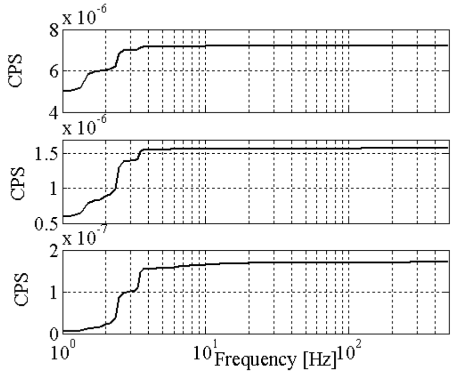


Figure 5.14: The cumulative power spectra of the position errors achieved with PD defined by (5.3) (top), PD defined by (5.13) (middle), and the H_∞ controller (bottom), respectively.

In the second experiment, the RRR robot realized the writing task shown in Fig. 3.4. The corresponding joint trajectory is depicted in Fig. 4.3. The Cartesian position errors achieved with the PD controller (5.13) and with the H_∞ controller are shown in Fig. 5.15 with grey and black lines, respectively. The latter controller performs better than the former in each direction. The H_∞ controller achieves errors (black lines) less than 0.5 [mm] along x– and y–axes, while along z–axis, the error is less than 1 [mm]. This is very accurate for a direct-drive robot.

TABLE 5.3: VARIANCES OF THE JOINT ERRORS IN $\times 10^{-6}$ [RAD^2/S^2]

i	PD eq. (5.3)	PD eq. (5.13)	H_∞
1	7.21	1.58	0.17
2	3.07	0.87	0.24
3	46.73	5.59	0.76

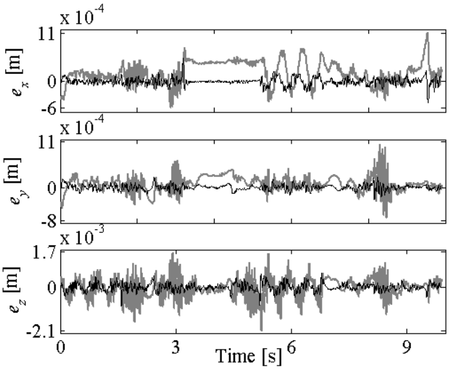


Figure 5.15: Cartesian errors in the writing task achieved with the PD defined by (5.13) (grey) and with the H_∞ controller (black).

In the feedback control design presented in this section, the optimal shaping of the closed-loop transfer functions is based on the nominal model of the plant dynamics. As observed in section 4.6, the actual plant dynamics can differ (within some bounds) from the nominal one. The considered H_∞-optimization makes the resulting controller optimal only for the nominal plant model. However, the weighting filters used in optimization were refined in the disturbance-based control design cycles, such as to make the resulting controller effective for the complete joint space. The next section presents an optimal loop-shaping that accounts for the differences from the nominal dynamics from the very beginning of the feedback control design. These differences are directly incorporated in the optimization process such as to determine feedback controllers that ensure robust control performance.

5.5 μ-synthesis feedback control design

This section deals with a feedback control design which should ensure robust stability and robust performance of the servo-loop shown in Fig. 5.1 [15]. Robust performance means a guaranteed accuracy of robot motions despite parasitic effects in the robot dynamics and disturbances affecting the robot operation. As seen in section 4.6, the parasitic dynamics change with the robot configuration, which must be taken into account during the design for robust performance. One option is to use a gain scheduling control strategy [109] to adapt the control action against the configuration dependent dynamics. This option usually demands a huge design effort, since it requires spanning the configuration space with a sufficiently dense grid of operating points and local control design for each of these points. Here, a typical issue is ensuring closed-loop stability during the transitions between the local controllers that correspond to the adjacent operating points. Advanced gain scheduling methods employ LPV (linear parametric-varying) plant representations. These commonly referred to as LPV gain scheduling methods involve the direct synthesis of configuration dependent controllers rather than their construction from a family of local controllers. Unfortunately, an LPV control design is yet cumbersome, since it requires derivation of the configuration dependent plant model upon which the controller will be synthesised.

Here, another option to ensure motion control of robust performance is considered. This option resorts to the μ-synthesis control design procedure [185], which provides unique controller tuning at all operating points. Although it is in general considered that unique tuning is more conservative than the adaptive one of the LPV gain scheduling controller, the μ-synthesis control design is considerably more time-efficient than the LPV control design, which is the essentially important advantage having in mind that the robot has many degrees of freedom for which the control designs should be accomplished. The μ-synthesis design employs the models of the nominal plant dynamics and of the modelling uncertainty. Assume that P_i^{o} represents a parametric model of nominal dynamics associated with joint i ($i = 1,\ldots,n$). The nominal model and actual plant dynamics P_i can be related using a multiplicative plant perturbation model:

$$P_i^{\mathrm{o}}(s) \rightarrow P_i(s) = P_i^{\mathrm{o}}(s)(1 + \Delta_i(s))\,. \tag{5.17}$$

Here, $\Delta_i(s)$ is a scalar multiplicative uncertainty, representing the relative difference between P_i^{o} and the real dynamics.

The uncertainty can be expanded as follows:

$$\Delta_i(s) = W_i^{\delta}(s)\delta_i(s)\,, \tag{5.18}$$

where W_i^{δ} is a stable parametric weighting function which satisfies

$$\left|W_i^{\delta}(j\omega)\right| \geq \left|\Delta_i(j\omega)\right|,\ \forall\omega \in \mathbb{R}^+, \tag{5.19}$$

and δ_i is the normalized scalar uncertainty $\left|\delta_i(j\omega)\right| \leq 1$, $\forall\omega \in \mathbb{R}^+$. Here, ω denotes angular frequency defined on $\mathbb{R}^+ = \{x \in \mathbb{R} | x \geq 0\}$.

A bound for the uncertainty can be determined from the measured frequency response function (FRF) G_i^k:

$$\left|\Delta_i(j\omega)\right| = \max_k \left|\frac{G_i^k(j\omega) - P_i^{\mathrm{o}}(j\omega)}{P_i^{\mathrm{o}}(j\omega)}\right|,\ \forall\omega \in \mathbb{R}^+\ (k = 1,\ldots,N_i). \tag{5.20}$$

Any stable weighting function, preferable of low order, can be adopted for W_i^{δ} if its magnitude closely bounds the perturbations $\left|\Delta_i\right|$ from above. According to the small gain theorem [185], a sufficient condition for robust stability of the perturbed plant (5.17) is

$$\left|T_i(j\omega)\right| < 1/\left|W_i^{\delta}(j\omega)\right|,\ \forall\omega \in \mathbb{R}^+. \tag{5.21}$$

Like with the H$_\infty$ optimal loop-shaping discussed in the previous section, the desired control performance can be specified by bounding the sensitivity and inputs sensitivity functions according to the relations (5.14) and (5.15).

A servo set-up shown in Fig. 5.16 can be used for simultaneous specification of the closed-loop objectives (5.14), (5.15), and (5.21), i.e., for joint shaping of S_i, U_i, and T_i. In this block-diagram, the channel from v_i to p_i is the uncertainty channel with the scalar scaled complex uncertainty δ_i. The channel from w_i to z_i^S and z_i^U is the performance channel. If v_i and w_i are adopted as the input variables, and p_i, z_i^S, and z_i^U as the output variables, then from Fig. 5.16 we may determine the interconnecting transfer function matrix $\mathbf{H}_i$:

$$\begin{bmatrix} p_i \\ z_i^S \\ z_i^U \end{bmatrix} = \mathbf{H}_i \begin{bmatrix} v_i \\ w_i \end{bmatrix}, \quad \mathbf{H}_i = \begin{bmatrix} -W_i^{\delta} T_i & -W_i^{\delta} T_i \\ W_i^S S_i & W_i^S S_i \\ -W_i^U U_i & -W_i^U U_i \end{bmatrix}. \tag{5.22}$$

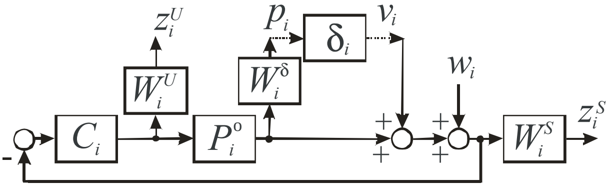


Figure 5.16: A set-up for servo design via μ-synthesis.

According to μ-analysis theory [185], to have the objectives (5.14), (5.15), and (5.21) robustly satisfied, it is sufficient that the structured singular value of the transfer function matrix $\mathbf{H}_i$ satisfies

$$\sup_{\omega} \mu_{\tilde{\mathbf{\Delta}}_i}(\mathbf{H}_i) < 1 \tag{5.23}$$

for the extended plant perturbation structure

$$\tilde{\mathbf{\Delta}}_i = \begin{bmatrix} \delta_i & \mathbf{0}_{1\times 2} \\ 0 & \boldsymbol{\delta}_i^{SU} \end{bmatrix}, \tag{5.24}$$

where $\boldsymbol{\delta}_i^{SU}$ denotes a complex uncertainty of dimension 1×2.

The objective of μ-synthesis is to construct a controller C_i which stabilizes the feedback loop shown in Fig. 5.16 and yields (5.23). If such a controller exists, robust stability and robust performance are realized. The controller can be calculated using the routines provided, e.g., in [15]. A design of such a controller is demonstrated in the case study, presented next.

The loop-shaping control design via μ-synthesis is demonstrated for joint 1 of the RRR robot. The feedback designs for other two joints are accomplished in the similar way. Section 4.6 presents joint 1 dynamics not covered with the model of the robot rigid-body dynamics with friction. The plots of FRF's $G_1^k(j\omega)$ ($k = 1,\dots,N_1$), measured for $N_1 = 16$ different robot postures, are shown in Fig. 4.22. The FRF of the nominal dynamics G_1^{o} and Bode plot of the corresponding parametric model of the 7th order are depicted in Fig. 4.23. The grey lines in Fig. 5.17 present relative differences (perturbations) in magnitudes between $G_1^1,\dots,G_1^{16}$ and P_1^{o}. By virtue of (5.20), at each ω the maximum perturbation is represented by Δ_1. The maximum perturbation is shown in Fig. 5.17 with the bold line. The dotted line in the same figure represents the magnitude of the parametric weighting function W_1^{δ} of the 2nd order, which is chosen in accordance with (5.21). As emphasized in section 4.6, the identified FRF's are not reliable below 4 [Hz], and hence W_1^{δ} does not bound the uncertainty in the lower frequency range. Fig. 5.18 shows the performance weightings W_1^S and W_1^U, that are chosen to adequately meet the control design rules r.1-r.4 from section 5.2. The weighting W_1^S of the 5th order should enforce integral action in the feedback controller, together with an as high as possible reduction of the position error at low frequencies. Furthermore, it should enforce attenuation of the vibrations at 28 [Hz]. The weighting W_1^U of the 2nd order should enforce a sufficient attenuation of resonances at high-frequencies. The weighting W_1^S is identical with the one used in H_∞ optimization presented in the previous section. The weighting W_1^U is different, as μ-synthesis with W_1^U presented in the previous section has not fulfilled the condition (5.23), while (5.23) is successfully achieved with the weighting shown in Fig. 5.18.

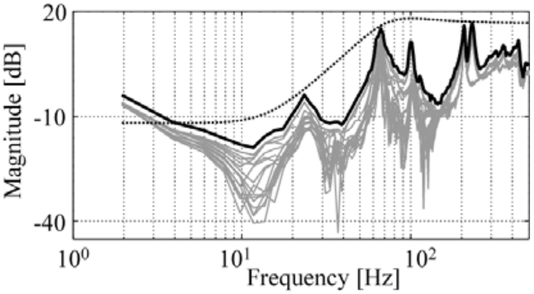


Figure 5.17: Magnitude plots of the weighting function W_1^δ (dotted), of relative differences between P_1^o and $G_1^1,\ldots,G_1^{16}$ (grey), and envelope of all differences Δ_1 (bold).

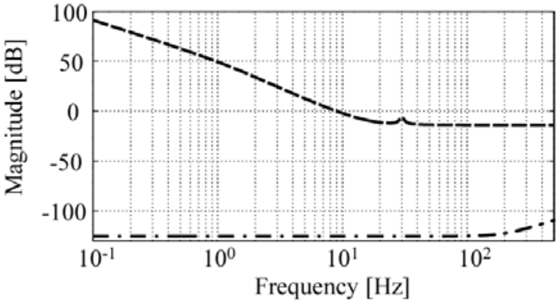


Figure5.18: Magnitude plots of the performance weightings W_1^S (dashed) and W_1^U (dash-dotted).

The μ-synthesis of the feedback controller C_1 employed iterative scaling of the 2nd order (D-scaling [15,185]) and H∞ optimization. Five "DK" iterations were necessary to determine C_1 which ensures robust performance specified by W_1^S and W_1^U. A plot of the upper bound of the achieved structured singular value $\mu_{\mathbf{H}_1}$ is given in Fig. 5.19. It is below 1, which implies a satisfactory design. For the sake of experimental implementation, the high order of the originally calculated controller C_1 was reduced to 16. The order reduction did not compromise the condition (5.39) for robust performance. The experimental controller C_1 is shown in Fig. 5.20. The controller introduces integral action at low frequencies and deals with various resonance frequencies over a broad frequency range. Its magnitude and phase plots have more involved shapes than the plots of the conventional controllers depicted in Fig. 5.3. They also differ from the plots of the H∞ controller shown in Fig. 5.10. Discrepancies are the result of the two different optimization problems solved here and in the previous section. The feedback controllers $\mathbf{C} = diag[C_1\ C_2\ C_3]$ obtained via μ-synthesis are experimentally implemented as illustrated by the block-diagram shown in Fig. 5.7.

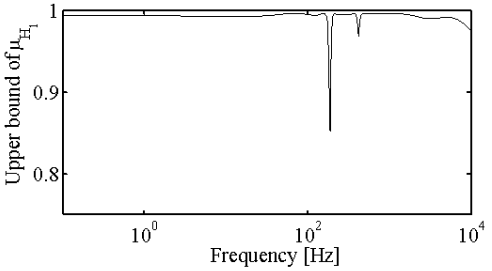


Figure 5.19: Upper bound of achieved structured singular value for joint 1.

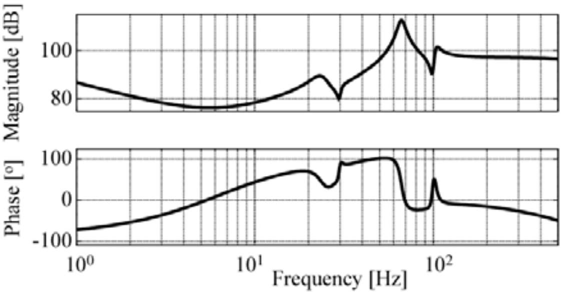


Figure 5.20: Bode plot of feedback controller C_1 calculated via μ-synthesis.

When determined based on the FRF of the nominal plant dynamics, shown in Fig. 4.23, the Bode plots of the open-loop gain and of the sensitivity function corresponding to C_1 reveal the same advantageous properties, as the plots related with the H_∞ controller from the previous section: high cross-over frequencies, high gains and small sensitivity below the cross-over and sufficient attenuation of resonance effects beyond it. In particular, the achieved cross-over frequencies are: 16 [Hz] for joint 1 (higher than with PD controllers or with H_∞ controller), 18 [Hz] for joint 2 (higher than with PD controllers and less than with H_∞ controller), and 20 [Hz] for joint 3 (higher than with PD controller defined by (5.3) and identical with PD defined by (5.13) and with H_∞ controller).

The idea of μ-synthesis is a feedback control robust against uncertainty in the plant dynamics. To verify that the achieved C_1 is robust for application within the complete joint space of the RRR robot, Fig. 5.21 presents the complementary sensitivity, sensitivity, and input sensitivity functions of the closed-loop system with the compensator C_1, calculated for P_1^{o} and for all $G_1^1,\ldots,G_1^{16}$. The top plots show that all the complementary sensitivities are bounded from above by the magnitude of W_1^δ, implying that the system remains stable for all postures–the condition (5.21) is satisfied. The middle and the bottom plots in Fig. 5.21 show the sensitivity functions and the input sensitivity functions, respectively. Obviously, all the plots are bounded with the magnitudes of $1/W_1^S$, respectively, $1/W_1^U$, except in the range of unreliable measurements (below 4 Hz). These imply that the performance specifications are robustly satisfied, i.e., conditions (5.14) and (5.15) are satisfied for all postures. Results similar to those presented in Fig. 5.21 also hold for the other two joints of the RRR robot.

The feedback controllers designed via μ-synthesis were experimentally tested in the same motion tasks, as considered in the previous section:

Task 1: fast joint displacements, presented in Fig. 5.8,
Task 2: the writing task shown in Fig. 3.4 and defined in the joint space according to Fig. 4.3.

The achieved position accuracies are good enough for a direct-drive robot:

Task 1: in joints 1 and 2, the errors remain within the range [-1,1]×10^{-3} [rad], while the range of the error in joint 3 is twice that of the first two joints;

Task 2: the Cartesian errors stay below 0.5 [mm] along each direction.

Given accuracies are even better than those achieved using the H_∞ controllers presented in the previous section.

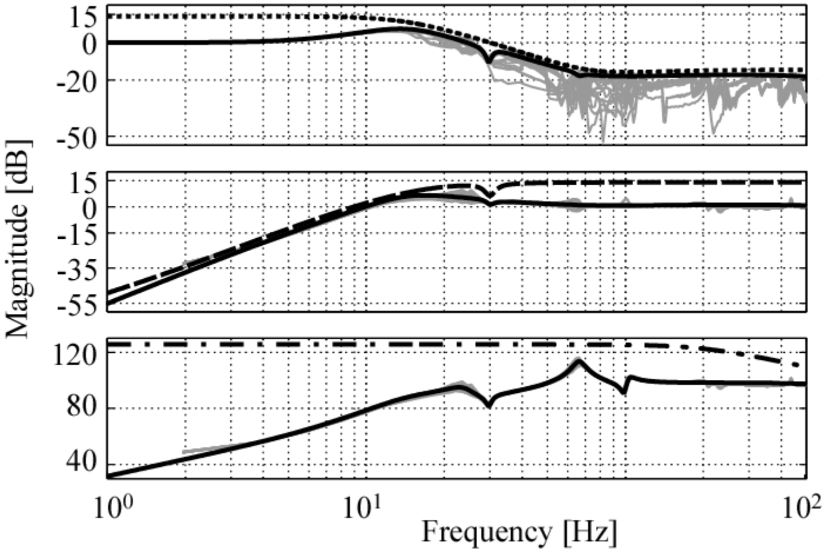


Figure 5.21: Magnitude plots of the nominal (bold) and of the perturbed (grey) complementary sensitivity functions (top), of the nominal (bold) and of the perturbed (grey) sensitivity functions (middle), and of the nominal (bold) and of the perturbed (grey) input sensitivity functions (bottom) are all below the pre-specified bounds (dotted, dashed, and dash-dotted lines, respectively).

To evaluate if the reductions of the joint errors are below the prescribed ones, the ratios between the error spectra (determined by Fast Fourier transform) and the spectra of the corresponding joint references, were found. For illustration, Fig. 5.22 presents the ratio between the spectra of $e_1(t)$ and of $q_{r,1}(t)$, corresponding to the viscous movement in Task 1. This ratio is plotted together with the inverse of the weighting function W_1^S (dashed line). Apparently, the curve corresponding to the ratio is always below the weighting. Having in mind the relation $S_1 = e_1 / q_{r,1}$, it follows that condition (5.14) is satisfied, i.e., the specified error reduction is realized. It appears that robust performance control of the RRR robot was realized in the experiment.

The cumulative power spectra of the achieved errors do not reveal excitation of high-frequency resonances, implying successful fulfilment of control design rule r.4. In Task 1, the error variances were 0.11, 0.14, and 0.72 [×10^{-6} rad^2/s^2] for joint 1, 2, and 3, respectively. They are smaller than the variances presented in Table 5.3, that were obtained with the conventional and H_∞ feedback controllers.

The advanced loop-shaping control designs presented so far, namely, the designs based on H_∞ optimization and μ-synthesis, are used to calculate linear feedback motion controllers. The last section of this chapter that comes next, explains how a data-driven loop-shaping can be used for feedback tuning of a nonlinear robustifying motion controller.

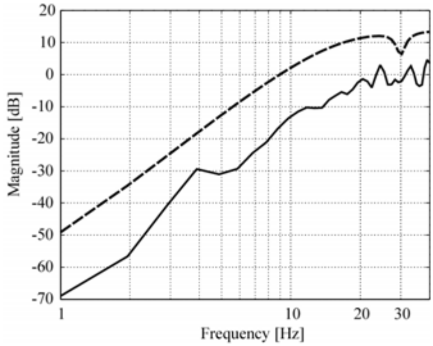


Figure 5.22: The ratio between spectra of the position error in joint 1 achieved with the μ-synthesis controller and of the position reference (solid) is below the prescribed weighting $1/W_1^S$ (dashed).

5.6 Discrete-time sliding-mode feedback control design

The previous two sections present advanced designs of linear feedback motion controllers. These designs enable robust motion control of high quality at the expense of accepting high orders of the resulting feedback controllers. This section investigates if a recently introduced nonlinear feedback control strategy, namely, a discrete-time sliding-mode control (DSMC) algorithm proposed in [60,61], can provide control performance feasible with the previous two linear designs, but implementing feedback controllers of considerably lower order.

The theory of variable structure systems with sliding mode control (SMC) has been developing for the last four decades. Some robotic applications of the theory can be found in [55,56,84,158,175]. Theoretically, when designed in continuous time, SMC features invariance to disturbances and to modelling errors, closed-loop system order reduction, and predictable transient behaviour. In practice, especially with discrete-time (sampled) implementations of SMC algorithms, control chattering occurs due to the finite duration of controller switching. Chattering is an undesirable high-frequency oscillation of the control input that may excite unmodelled dynamics present in electromechanical systems, resulting in decreased control performance.

Notable advantages of the DSMC algorithm presented in [60,61] with respect to other discrete-time sliding mode control strategies, are: finite-time reaching of ideal sliding mode in the absence of modelling errors and disturbance effects, chattering-free control, stability robustness against disturbances and modelling errors, and nice control over the transient behaviour of the controlled system. These merits seem quite appealing to investigate the use of the DSMC algorithm in robot motion control, since, so far, the algorithm was experimentally tested only for an oscillator design [60] and in the control of a simple linear unloaded DC motor [61]. In the next part, the original algorithm is accommodated for application in robotics. Later on, experimental effects of the algorithms are demonstrated on the RRR robot.

The considered DSMC algorithm deals with the dynamics (5.5) that remain after application of the computed-torque control law (2.26) to the actual robot dynamics (2.16). The dynamics (5.5) for joint i is given by equation (5.7). Using the control of the form (5.1) and with the position error defined by (5.2), the dynamics (5.7) can be rewritten as

$$\ddot{e}_i(t) = -u_i^*(t) + \nu_i(t)\,. \tag{5.25}$$

A state-space representation of (5.25) is

$$\dot{\mathbf{x}}_i^e(t) = \mathbf{A}_i^e \mathbf{x}_i^e(t) + \mathbf{b}_i^e(-u_i^*(t) + \nu_i(t))\,;$$

$$\mathbf{x}_i^e(t)=\begin{bmatrix} e_i(t)\\ e_i^{\omega}(t)\end{bmatrix};\ \mathbf{A}_i^e=\begin{bmatrix}0&1\\0&0\end{bmatrix};\ \mathbf{b}_i^e=\begin{bmatrix}0\\1\end{bmatrix}, \tag{5.26}$$

where $e_i^{\omega}=\dot{e}_i$. The DSMC design [61] requires a discrete-time equivalent of the system (5.26). A discrete-time system having identical states with (5.26) at $t=kT_s$, with T_s the sampling time and $k\in N_0=\{0,1,2,\ldots\}$, has the form:

$$\mathbf{x}_i^e(k+1)=\mathbf{E}_i^e(T_s)\mathbf{x}_i^e(k)+\mathbf{f}_i^e(T_s)(-u_i^*(k)+\nu_i(k)), \tag{5.27a}$$

where k and $k+1$ abbreviate kT_s and $(k+1)T_s$, respectively, and

$$\mathbf{E}_i^e=e^{\mathbf{A}_i^e T_s}=\begin{bmatrix}1&T_s\\0&1\end{bmatrix};\ \mathbf{f}_i^e=\int_0^{T_s} e^{\mathbf{A}_i^e\varsigma}\mathbf{b}_i^e d\varsigma=\begin{bmatrix}T_s^2/2\\ T_s\end{bmatrix}. \tag{5.27b}$$

If only position measurements (5.10) are available, featuring a time-delay of $\delta=pT_s$ (for some $p\in N_0$), then a Kalman observer technique (5.9)-(5.12) can be used to reconstruct the vector of states $\mathbf{x}_i^e$.

The DSMC design of the feedback control law u_i^* is based on the following representation of (5.27a,b):

$$\delta\mathbf{x}_i^e(k)=\mathbf{A}_i^{\delta}\mathbf{x}_i^e(k)+\mathbf{b}_i^{\delta}(T_s)(-u_i^*(k)+\nu_i(k)), \tag{5.28a}$$

where

$$\begin{gathered}\delta\mathbf{x}_i^e(k)=(\mathbf{x}_i^e(k+1)-\mathbf{x}_i^e(k))/T_s;\\ \mathbf{A}_i^{\delta}=(\mathbf{E}_i^e-\mathbf{I}_2)/T_s=\begin{bmatrix}0&1\\0&0\end{bmatrix};\ \mathbf{b}_i^{\delta}=\mathbf{f}_i^e/T_s=\begin{bmatrix}T_s/2\\1\end{bmatrix}.\end{gathered} \tag{5.28b}$$

Here, $\mathbf{I}_2\in\mathbb{R}^{2\times 2}$ is the identity matrix. The pair $(\mathbf{A}_i^{\delta},\mathbf{b}_i^{\delta})$ is controllable for any $T_s>0$. The feedback control law should be formulated in terms of $\overline{\mathbf{x}}_i^e(k)=[\overline{e}_i(k),\overline{e}_i^{\omega}(k)]^T$, where $\overline{e}_i(k)$ and $\overline{e}_i^{\omega}(k)$ are reconstructed according to (5.12).

The switching function is also based on the reconstructed state vector $\overline{\mathbf{x}}_i^e$:

$$s_i(k)=\boldsymbol{\lambda}_i^T\overline{\mathbf{x}}_i^e(k);\ \boldsymbol{\lambda}_i=[\lambda_i^p\ \ \lambda_i^d]^T, \tag{5.29}$$

where $s_i,\lambda_i^p,\lambda_i^d\in\mathbb{R}$ and $(\lambda_i^p,\lambda_i^d)\neq(0,0)$. The reaching law approach [55,56,61,84] is used for the controller design:

$$\begin{aligned}\delta s_i(k)&=(s_i(k+1)-s_i(k))/T_s=\boldsymbol{\lambda}_i^T\delta\overline{\mathbf{x}}_i^e(k)=\\ &=-\phi(s_i(k)),\end{aligned} \tag{5.30}$$

where ϕ should be chosen such that the system state trajectories reach in finite time

1. the switching manifold $s_i=0$, in the absence of disturbances and modelling errors; as $s_i=0$ is reached, the states should remain on the manifold, establishing chattering-free control; contrary to the discrete-time quasi-sliding mode [56,123], no zigzagging about $s_i=0$ should occur–the system motions must remain stable and stick to $s_i=0$,
2. a small neighbourhood of $s_i=0$, in the presence of disturbances and modelling errors; within the neighbourhood, chattering-free control should occur.

In the following, two spaces are considered:

– the *state space*, with elements of $\bar{\mathbf{x}}_i^e$ ($i=1,\dots,n$) as coordinates,

– the *reaching space* [55,56], with s_i ($i=1,\dots,n$) as coordinates.

If (5.28a,b) is substituted into (5.30), $\nu_i(k)$ is omitted since it is considered unknown, and of (5.29) is taken into account, one finds:

$$\begin{aligned} u_i^*(k) &= \frac{\boldsymbol{\lambda}_i^T \mathbf{A}_i^\delta}{\boldsymbol{\lambda}_i^T \mathbf{b}_i^\delta(T_s)} \bar{\mathbf{x}}_i^e(k) + \frac{\phi(s_i(k))}{\boldsymbol{\lambda}_i^T \mathbf{b}_i^\delta(T_s)} \\ &= \frac{\lambda_i^p \bar{e}_i(k) + \phi(s_i(k))}{0.5 T_s \lambda_i^p + \lambda_i^d}. \end{aligned} \tag{5.31}$$

An appropriate ϕ should ensure that (5.31) is stabilizing for the system (5.28a,b) in the presence of ν_i. Sufficient conditions that an appropriate ϕ must fulfil are formulated next.

Consider the system (5.28a,b) with control (5.31) and assume the following: – the collection of model uncertainties and disturbances is bounded according to

(*i*) $\left|\nu_i(k)\right| \le \mu_i$, $\forall k \in N_0$;

– the neighbourhood of the switching manifold $s_i = 0$ is given by

(*ii*) $S_i^{ss} = \{\bar{\mathbf{x}}_i^e \in \mathbb{R}^2 : |\boldsymbol{\lambda}_i^T \bar{\mathbf{x}}_i^e(k)| < \varepsilon_i T_s\}$ in the state space, or

$S_i^{rs} = \{s_i \in \mathbb{R} : |s_i(k)| < \varepsilon_i T_s\}$ in the reaching space;

– the state reconstruction (5.12) ensures

(*iii*) $\bar{\mathbf{x}}_i^e(k) \equiv \mathbf{x}_i^e(k)$.

The next theorem is obtained by tailoring Theorems 1 and 2 from [61] to make then compatible with the considered robot control problem.

Theorem 1. In order that for any initial $\bar{\mathbf{x}}_i^e(0) \equiv \mathbf{x}_i^e(0)$ there exists a number $K = K(\bar{\mathbf{x}}_i^e(0))$ such that $\bar{\mathbf{x}}_i^e(k) \in S_i^{ss}$ for $k \ge K$, it is sufficient that the following conditions hold:

(*iv*) $\phi = s_i(k)/T_s$ for $s_i(k) \in S_i^{rs}$,

(*v*) $o_i T_s \mu_i / |s_i(k)| < T_s \phi(s_i(k))/s_i(k) < 1$, for $s_i(k) \notin S_i^{rs}$, $o_i > \boldsymbol{\lambda}_i^T \mathbf{b}_i^\delta(T_s)$,

(*vi*) $\varepsilon_i > \mu_i \left|\boldsymbol{\lambda}_i^T \mathbf{b}_i^\delta(T_s)\right|$.

If $\nu_i \equiv 0$, then to have $s_i(k) = 0$ for $k > K(\bar{\mathbf{x}}_i^e(0))$, it is sufficient that (*iv*) holds and $0 < T_s \phi / s_i < 1$ is satisfied for $s_i \notin S_i^{rs}$.

Proof. Similar to the proofs of Theorems 1 and 2 from [61]. ■

Dead-beat property: if $\nu_i \equiv 0$, then (*iv*) is a necessary and sufficient condition for reaching $s_i = 0$ in one step, after $\bar{\mathbf{x}}_i^e$ enters S_i^{ss} [61].

One choice for ϕ, suggested in [61], is

$$\begin{gathered} \phi(s_i) = \begin{cases} s_i / T_s \ \text{ if } \ s_i \in S_i^{rs}; \\ \rho_i s_i + \sigma_i \operatorname{sgn}(s_i) \ \text{ if } \ s_i \notin S_i^{rs}, \end{cases} \\ 0 \le \rho_i T_s < 1; \ \ \sigma_i > o_i \mu_i \underset{(v)}{>} \boldsymbol{\lambda}_i^T \mathbf{b}_i^\delta(T_s) \mu_i, \end{gathered} \tag{5.32}$$

where

$$S_i^{rs} = \{s_i \in \mathbb{R} : |s_i(k)| < \sigma_i T_s / (1 - \rho_i T_s)\}, \tag{5.33a}$$

$$S_i^{ss} = \{\bar{\mathbf{x}}_i^e \in \mathbb{R}^2 : |\boldsymbol{\lambda}_i^T \bar{\mathbf{x}}_i^e(k)| < \sigma_i T_s / (1 - \rho_i T_s)\}. \tag{5.33b}$$

By virtue of (5.32) and (5.29), u_i^* defined by (5.31) is a continuous function within S_i^{rs}, providing a chattering-free DSMC.

The tuning of the coefficients present in the feedback control law (5.31) with ϕ defined by (5.32), is discussed next. Since by virtue of (5.28b) and (5.29), $\boldsymbol{\lambda}_i^T \mathbf{b}_i^{\delta}(T_s) = 0.5 T_s \lambda_i^p + \lambda_i^d$, then (5.32) implies $\sigma_i > (0.5 T_s \lambda_i^p + \lambda_i^d)\mu_i$, where μ_i must overbound ν_i according to (*i*). The variable ν_i is reconstructed with $\bar{\nu}_i$, according to (5.12). Since the coefficient σ_i is determined according to the actual value of $\bar{\nu}_i$, the a conservative tuning is obtained if σ_i is specifically determined for a given reference motion task. The coefficient ρ_i is subject to $0 \le \rho_i T_s < 1$, and increasing ρ_i speeds up reaching S_i^{rs} [55,56].

The coefficients λ_i^p and λ_i^d of the switching function s_i, in (5.29), should be chosen such that the dynamics of the system (5.28a,b) with control (5.31), is stable within S_i. So far, it is considered that $\bar{\mathbf{x}}_i^e \equiv \mathbf{x}_i^e$ (see (*iii*)). Assume now:

$$e_i = \bar{e}_i + \Delta e_i, \; e_i^{\omega} = \bar{e}_i^{\omega} + \Delta e_i^{\omega} \text{ with } |\Delta e_i| < \infty, \; |\Delta e_i^{\omega}| < \infty. \tag{5.34}$$

By virtue of (5.32), if $s_i \in S_i^{rs}$, then (5.31) defines a conventional PD control action:

$$u_i^*(k) = \kappa_i^p \bar{e}_i(k) + \kappa_i^d \bar{e}_i^{\omega}(k) \tag{5.35a}$$

$$\kappa_i^p = \frac{\beta_i}{T_s(0.5 T_s \beta_i + 1)}, \; \kappa_i^d = \frac{T_s \beta_i + 1}{T_s(0.5 T_s \beta_i + 1)}, \; \beta_i = \frac{\lambda_i^p}{\lambda_i^d} \tag{5.35b}$$

The scalar proportional and derivative gains κ_i^p and κ_i^d, respectively, are uniquely defined by the β_i–the slope of the switching line $s_i = 0$ (see (5.29)). If the control law (5.35a,b) is applied to (5.27a,b), and (5.34) is taken into account, one obtains:

$$\begin{bmatrix} e_i(k+1) \\ e_i^{\omega}(k+1) \end{bmatrix} = \frac{1}{0.5 T_s \beta_i + 1} \begin{bmatrix} 1 & 0.5 T_s \\ -\beta_i & -0.5 T_s \beta_i \end{bmatrix} \begin{bmatrix} e_i(k) \\ e_i^{\omega}(k) \end{bmatrix} + \begin{bmatrix} 0.5 T_s^2 \\ T_s \end{bmatrix} \cdot \left(\nu_i(k) + \frac{\beta_i \Delta e_i(k) + (T_s \beta_i + 1)\Delta e_i^{\omega}(k)}{T_s(0.5 T_s \beta_i + 1)} \right). \tag{5.36}$$

Since ν_i, Δe_i, and Δe_i^{ω} are assumed to be bounded, and assuming $T_s > 0$, a necessary and sufficient condition to have the magnitudes of the states of (5.36) monotonically decreasing is

$$-1 < (1 - 0.5 T_s \beta_i)/(1 + 0.5 T_s \beta_i) < 1 \Leftrightarrow \beta_i = \lambda_i^p / \lambda_i^d > 0. \tag{5.37}$$

As already mentioned, the PD gains in (5.35b) are uniquely defined by β_i. This seems a main shortcoming of the considered DSMC algorithm when used in robot motion control. In Fig. 5.23, κ_i^p and κ_i^d are plotted against β_i ($T_s = 1$ [ms]). The same figure shows $2\sqrt{\kappa_i^p}$ as a function of β_i, as $2\sqrt{\kappa_i^p}$ is a common choice for the derivative gain for continuous-time second order systems (relative damping equal to one) [153]. Of course, the equivalence with continuous systems is sensible only for small sampling times, realistic in modern robotics. The middle plot was obtained by zooming the top of the figure for $0 \le \beta_i \le 1.5$. Apparently, $\kappa_i^p < \kappa_i^d$ for $\beta_i < 1$. From the bottom plot, one notices the rising tendency of $2\sqrt{\kappa_i^p}$. Up to $\beta_i = 414$, κ_i^d is higher than $2\sqrt{\kappa_i^p}$, implying relative damping above one. As κ_i^p is limited by the maximal feasible closed-loop bandwidth, a realistic β_i cannot be high. For lower β_i, κ_i^d is much larger than $2\sqrt{\kappa_i^p}$, implying a high relative damping. High damping, induced by the dead-beat property implied by (*iv*), may amplify noise considerably and excite parasitic dynamics at higher frequencies. These unwanted effects are inherent to control chattering arising with nonideal sliding mode control [55,84,175]. The considered DSMC algorithm was designed to establish chatter-free control, but, unfortunately, it fails in eliminating the unwanted effects. It appears that these effects can easily arise in practical implementations of the DSMC algorithm, and their negative influence on the control performance is experimentally demonstrated next.

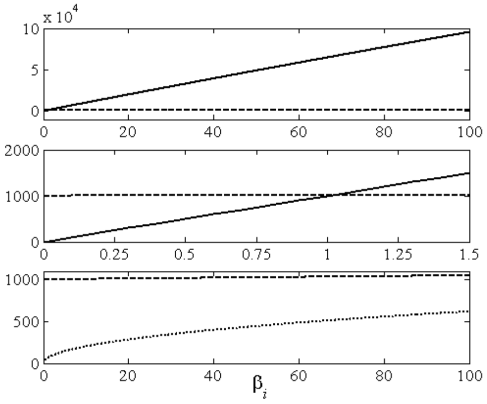


Figure 5.23: Feedback gains: κ_i^p (solid), κ_i^d (dashed), and $2\sqrt{\kappa_i^p}$ (dotted).

The experimental DSMC design is presented for the RRR robot. As mentioned above, the less conservative tuning of the coefficient σ_i ($i = 1,2,3$), used in the DSMC feedback control law (5.31),(5.32), is obtained if σ_i is specifically determined for a given robot motion task. Here, the motion task depicted in Fig. 5.24 is considered. In this task, each robot joint has to be displaced for 2π [rad] in 3 [s], with zero initial/terminal velocity and acceleration. Obviously, this task requires lower velocity and acceleration levels that the task presented in Fig. 5.8. Consequently, the task depicted in Fig. 5.24 should be less challenging for the accurate realization than the one shown in Fig. 5.8.

Initially, the reference motions shown in Fig. 5.24 were realized using PD feedback controllers:

$$\mathbf{u}^*(k) = 1000\mathbf{I}_3\bar{\mathbf{e}}(k) + 2\sqrt{1000}\mathbf{I}_3\bar{\mathbf{e}}^{\omega}(k), \qquad (5.38)$$

where $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix. The state reconstruction according to (5.12) was of sufficient quality, since the differences between the position measurements y_i, see (5.10), and the corresponding $\bar{q}_i(k-2) = q_{r,i}(k-2) + \bar{e}_i(k-2)$, $i = 1,2,3$, were all within [-2, 2]×10^{-4} [rad]. The reconstructed $\bar{\nu}_i$ ($i = 1,2,3$) were used to determine μ_i ($i = 1,2,3$), according to condition (*i*). With μ_i available, σ_i was chosen according to $\sigma_i > (0.5T_s\lambda_i^p + \lambda_i^d)\mu_i$. As indicated above, the stability of the closed-loop system for $s_i \in S_i^{rs}$ depends only on the ratio $\beta_i = \lambda_i^p / \lambda_i^d$. Thus, the number of tuning coefficients can be reduced by choosing $\lambda_i^d = 1$ ($i = 1,2,3$). By virtue of (5.37), any positive λ_i^p ($i = 1,2,3$)

ensures stability in closed-loop, which suggests that any closed-loop bandwidth should be achievable within S_i^{rs}. However, parasitic dynamics limit the maximal feasible closed-loop bandwidth within S_1^{rs}, implying that arbitrary high $\lambda_1^p = \beta_1$ is not admissible with respect to the stability in closed-loop. For instance, by taking into account the frequency-response function (FRF) of the nominal dynamics for joint 1, presented in Fig. 4.23, it appears that only $\beta_1 \leq 12$ ensures stability within S_1^{rs}. From Fig. 5.23, it is obvious that different β_i correspond to different tuning of the equivalent PD controller (5.35a,b). For $\beta_1 \leq 12$, the relative damping is unreasonably high, as it may excite flexible dynamics and amplify quantization noise. An appropriate set of PD gains, away from the "stability" boundary $\beta_1 = 12$, can be found using the loop-shaping technique. With this technique, a designer tunes the PD gains in (5.35a,b) by varying λ_1^p, to make a compromise between the conflicting demands r.1-r.4 from section 5.2 for high closed-loop bandwidths, acceptable control performance, and stability. For example, the loop-shaping for joint 1 achieved the compromise for the value of $\lambda_1^p = 5$. Unfortunately, this λ_1^p does not fulfil rule r.3, since the sensitivity function computed for the corresponding PD gains (see (5.35b)) and for the FRF shown in Fig. 4.23, features a severe peaking above 6 [dB], around 20 [Hz]. This peaking is not an issue for stability robustness of the considered DSMC algorithm, since by Theorem 1, the control law (5.31),(5.32) is guaranteed to be stabilizing under the given circumstances. However, the peaking may cause undesirable excitation of parasitic dynamics, as is confirmed by the experimental results presented below. For the selected λ_1^p and $\lambda_1^d = 1$, σ_1 was found as $\sigma_1 > (2.5T_s + 1)\mu_1$. The remaining tuning coefficient in (5.32) is ρ_i. Tentatively, $\rho_i = 0$ ($i = 1,2,3$) was chosen. Experiments with the motion task depicted in Fig. 5.24, showed no need for higher ρ_i.

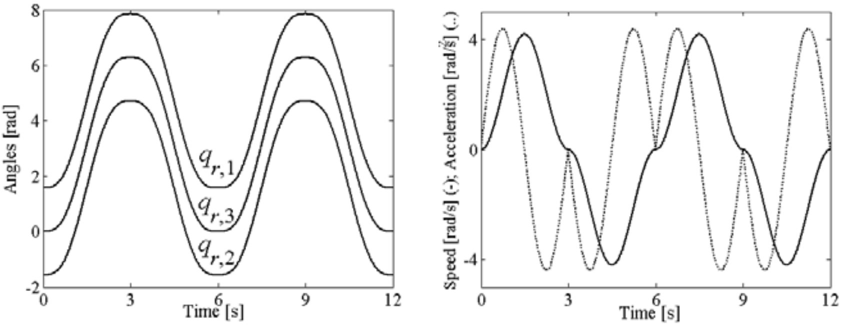


Figure 5.24: The motion task used in experiments with discrete-time sliding mode control.

In Fig. 5.25, the online reconstructed errors achieved with DSMC controllers are plotted for all three joints. The errors in joints 1 and 2 stay within the range [-1,1]×10^{-3} [rad], while in joint 3 the error peaks are almost three times higher. Closer inspection of the plots reveals the presence of oscillatory components superimposed to the error patterns. The existence of these fine oscillations is confirmed by the cumulative power spectra (CPSs) of the errors as shown in Fig. 5.26. By inspection of the CPS plots, one notices that the dominant error spectral components are in the lower frequency range, which is the range of the reference trajectory depicted in Fig. 5.24. Outside this range, jumps in the slopes of the error CPSs are present around 20 [Hz]. These jumps are due to oscillations observed in the error signals. The oscillations correspond to severe peaking around 20 [Hz] in the sensitivity function for joint 1. It appears that the oscillation arises in all three joints due to elastic couplings not compensated using the rigid-body dynamic model with friction implemented in the computed-torque control law (2.26).

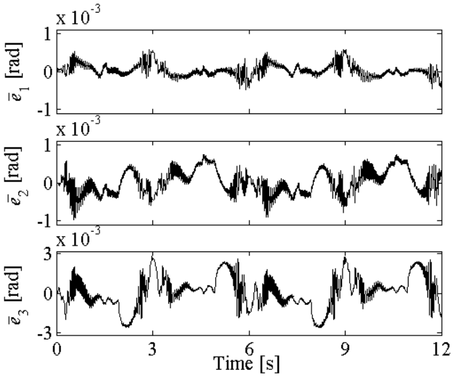


Figure 5.25: Experimental position errors reconstructed by Kalman observer.

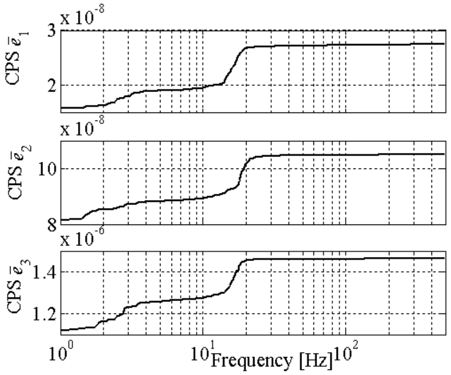


Figure 5.26: Cumulative power spectra of the position errors.

The control torques computed using (2.26), with the feedback control action realized by (5.1) and (5.31),(5.32), are shown in Fig. 5.27. The chattering effect from the SMC theory is clearly eliminated, but the presence of oscillations, related with the oscillation discussed in the previous paragraph, is obvious. Fig. 5.28 presents the time histories of the switching functions $s_i(t)$ $(i=1,2,3)$. The bounds of each plot of $s_i(t)$ match the bounds of S_i^{rs}. Apparently, each $s_i(t)$ remains within S_i^{rs}.

It appears that the DSMC of the RRR robot does not give the performance achievable with advanced feedback controllers discussed in the previous two sections. This result can be explained as follows: the feedback controllers presented in the previous two sections are based on models with flexibilities included, while the DSMC considers only the rigid-body dynamics that remains after feedback linearization (computed-torque control). Furthermore, the feedback controllers considered before also include integral action, while the DSMC does not. Finally, there is a restriction in tuning the DSMC that impedes satisfactory loop-shaping within the vicinity of the switching manifold. This restriction can easily cause excitation of flexibilities in a robotic system. Therefore, DSMC design for the dynamics that include flexibilities is an option for further research. Such design will certainly increase the order of the resulting controller and of the accompanying Kalman observer. The expected merit, though, would be robust feedback control with nice and predictable transient characteristics.

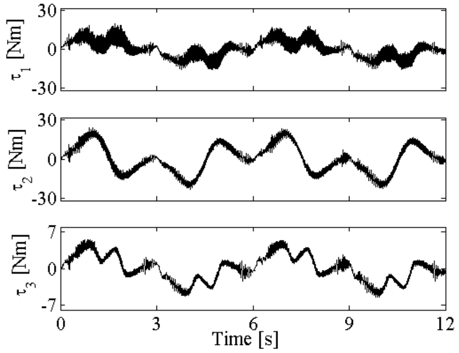


Figure 5.27: Control torques.

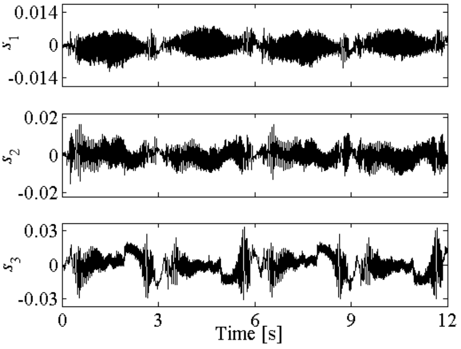


Figure 5.28: Time histories of the switching functions.

5.7 Summary

This chapter presents and examines various data-driven model-based designs of feedback motion controllers. The feedback designs and their tunings are facilitated with the computed-torque control, which is used to compensate for the robot nonlinearities. The computed-torque control decouples motions between the robot joints and enables the use of the well-known loop-shaping feedback design strategy. The loop-shaping is performed on the decoupled robot dynamics that are not covered with the model-based compensation of the nonlinearity. Several feedback designs are considered, and each of them has been experimentally tested on the RRR robot. Since the RRR robot has direct-drive actuation and spatial kinematics (gravity effect is included), the conclusions drawn from experimental control studies with this robot can be considered relevant for a broad class of direct-drive robots.

A conventional PD feedback control was investigated in the cases when the motion coordinates (joint displacements and velocities) have been corrupted by disturbance effects (measurement noise, time-delay) and when these coordinates were online reconstructed via Kalman observer that compensates for the time-delay and sufficiently filters-out the noise. Use of a Kalman observer enables higher closed-loop bandwidth and better reduction of position errors within the bandwidth, as well as sufficient reduction of parasitic effects at higher frequencies. However, the observer increases the total order of the motion controller and the computational costs.

Feedback controllers designed via H_∞ optimization and via μ-synthesis additionally increase the controller order, but for the benefit of high-quality motion control. These approaches closely account for the experimental characteristics of the robot under control. The H_∞-optimal controllers were iteratively refined via disturbance-based control designs cycles. It is true that such an approach increases off-line design and tuning effort, but it also leads to the successful achievement of the control specifications: high accuracy of executing the reference joint trajectories, and the robustness against the modelling errors and disturbances. The μ-synthesis is a conceptual approach to achieve robust performance of robot motion control. By supplying data-driven models of robot dynamics not covered with nonlinear model-based compensation, μ-synthesis calculates feedback controllers that ensure robust performance, i.e., guaranteed accuracy of robot motions in the presence of parasitic dynamics and disturbances conditions.

The utility of a discrete-time sliding mode control (DSMC) algorithm, for robot motion control, has been analysed in the last part of the chapter. With a theoretical analysis, as well as by experimental results, it was shown that the theoretical merits of the algorithm, namely, finite time reaching of ideal chattering-free DSMC and stability robustness against disturbances and modelling uncertainty, are achieved at the expense of unreasonably high feedback gains that may cause undesirable noise amplification and excitation of parasitic dynamics. Feedback tuning of the DSMC algorithm was based on data-driven models of the robot dynamics. As a possible remedy for the undesirable effects, a data-driven model which includes flexible dynamics should be incorporated into a DSMC design.

Chapter 6

Vibration isolation in the robot task space

The previous chapter discussed robot motion control in the presence of parasitic dynamics. The parasitic dynamics caused by robot flexibility were tackled at the servo control level in the robot joints. An alternative to deal with flexibilities is to counteract their effects in the robot task space. This is a common practice if the robot structure features a profound elastic behaviour. Because of elasticity, undesirable vibrations may arise at the robot tip during execution of the reference motion task. As mentioned in section 2.3, higher robustness against these vibrations can be achieved if the tip motion is directly measured in the task space, and these measurements are employed in the motion control law together with the joint position measurements. In section 2.3 it was also remarked that the derivation of the control law which ensures accurate execution of the reference tip trajectory with robust attenuation of the tip vibrations is not straightforward. This remark motivated the particular research question to be addressed in this chapter, namely, development of a simple but effective strategy for robust attenuation of the robot tip vibrations. In this strategy, a nominal motion controller designed in the robot joint space will be extended with an additional regulation system which will serve as the vibration compensator. The vibration compensator will be based on a data-driven model of the robot flexibility identified in the robot task space. An effective way of combining the nominal motion controller with the vibration compensator will be suggested. The benefits of the resulting control strategy will be experimentally verified on the RRR robot. The chapter is organized as follows. Section 6.1 describes the problem of robot-tip vibrations. Section 6.2 explains the solution strategy for these vibrations. The problem of vibrations at the tip of the RRR robot is described in section 6.3. A design of the vibration compensator for this robot is presented in section 6.4. The quality of this design is experimentally assessed in section 6.5. The chapter summary is given at the end. Most of the material presented in this chapter has also been published in [93].

6.1 Problem description

Modelling of the rigid-body robot dynamics with friction was treated in sections 3.2 and 3.3. As demonstrated in the case-study with the RRR robot, presented in section 4.6, the rigid-body dynamics dominate only in the lower frequency range, beyond which flexible effects appear and become critical for control performance. When excited during robot movements, lumped elasticity in the robot joints and/or distributed link flexibility cause undesirable bending of the robot structure, which displaces position of the robot tip from the point corresponding to the rigid robot posture. Permanent excitation of flexibility induces vibrations at the tip that degrade control performance in the robot task space.

The tip vibrations are a limiting factor for realizing faster robot motions, especially for robots of lightweight and less stiff constructions. On one hand, the purpose of a lightweight construction is to enable faster motions with the same actuators applied. On the other hand, low stiffness means that flexibilities appear at lower frequencies, and they can be excited at robot velocities that are admissible for the applied actuators. Typical manipulators subject to tip vibrations are robot arms used in space robotics. Indeed, these manipulators feature long and narrow links to reduce link inertia and demands on the joint actuators. However, the long and slender links are exactly the preconditions for elastic bending. To reduce the bending, commanded movements of these manipulators are slow.

The conflicting relation between the advantages and disadvantages of using the lightweight robot constructions is a subject of research in motion control. The literature offers a number of solutions [47,48,63,102,114,115,120,147,184]. This chapter presents a simple but original control method that attenuates vibrations at the robot tip, caused by the high flexibility in the robot mechanism. The theory behind the control method is briefly explained, while its practical application on the RRR robot is demonstrated in detail.

6.2 Mathematical formulation

Consider the dynamic rigid-body model with friction (3.11). When structural flexibility is negligible, the model determines the joint torques corresponding to certain location (position and orientation) of the robot end-effector. When flexible effects are present, the location of the end-effector deviates from its rigid-body location, as illustrated in Fig. 6.1.

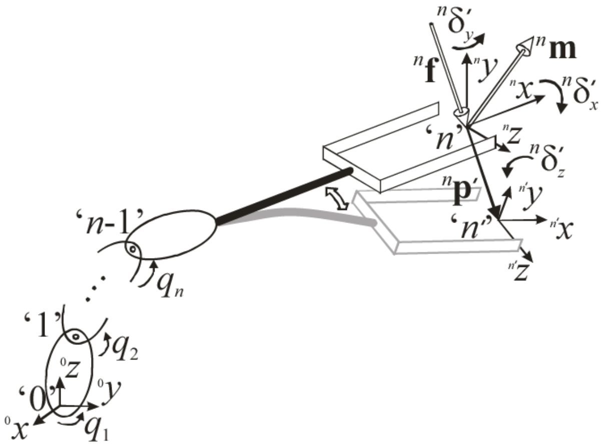


Figure 6.1: Elastic displacement of the robot end-effector from its rigid-body location.

Reference [120] suggests that elastic displacement of the end-effector can be considered as the result of an equivalent elastic force $^n\mathbf{f} \in \mathbb{R}^3$ and an equivalent elastic moment $^n\mathbf{m} \in \mathbb{R}^3$. Both factors $^n\mathbf{f}$ and $^n\mathbf{m}$ are indicated in Fig. 6.1, where they are referred to the tip coordinate frame '*n*'. These factors can be integrated into the model (3.11) using the well-known description of dynamic interaction between the end-effector and environment [153]:

$$\begin{aligned}&\mathbf{M}(\mathbf{q}(t))\ddot{\mathbf{q}}(t)+\mathbf{c}(\mathbf{q}(t),\dot{\mathbf{q}}(t))+\mathbf{g}(\mathbf{q}(t))+\boldsymbol{\tau}^f(\mathbf{q}(t),\dot{\mathbf{q}}(t),t)+\\&+\left({}_n^0\mathbf{J}(\mathbf{q}(t))\right)^T\begin{bmatrix}{}_n^0\mathbf{O}(\mathbf{q}(t)) & \mathbf{0}_{3\times3}\\ \mathbf{S}\left({}_n^0\mathbf{x}(t)\right){}_n^0\mathbf{O}(\mathbf{q}(t)) & {}_n^0\mathbf{O}(\mathbf{q}(t))\end{bmatrix}\begin{bmatrix}{}^n\mathbf{f}\left({}^n\mathbf{p}'(t)\right)\\ {}^n\mathbf{m}\left({}^n\boldsymbol{\delta}'(t)\right)\end{bmatrix}=\boldsymbol{\tau}(t),\end{aligned} \tag{6.1}$$

where ${}_n^0\mathbf{J} \in \mathbb{R}^{6\times n}$ is the robot geometric Jacobian defined by (3.6),(3.7) and which is related to the base (inertial) frame '0', the matrix ${}_n^0\mathbf{O}(\mathbf{q})$ and the vector ${}_n^0\mathbf{x} = [{}_n^0x \ {}_n^0y \ {}_n^0z]^T$ map the orientation and position of the tip frame '*n*' to '0' according to (3.5), and the skew-symmetric matrix operator $\mathbf{S}$ is defined by:

$$\mathbf{S}({}_n^0\mathbf{x})=\begin{bmatrix}0 & -{}_n^0z & {}_n^0y\\ {}_n^0z & 0 & -{}_n^0x\\ -{}_n^0y & {}_n^0x & 0\end{bmatrix}. \tag{6.2}$$

The variables ${}^{n}\mathbf{p}' \in \mathbb{R}^3$ and ${}^{n}\boldsymbol{\delta}' \in \mathbb{R}^3$ denote the vectors of translational and angular vibration coordinates, respectively, referred to '*n*'. They represent the displacement of the tip frame from its rigid-body location, as shown in Fig. 6.1. The remaining variables used in (6.1) are as defined by (3.11).

The last term in (6.1) points out how to compensate for the robot flexibility: the nominal control of the reference joint motions $\mathbf{q}_r(t)$ should be corrected with a term counteracting the effects of ${}^{n}\mathbf{f}$ and ${}^{n}\mathbf{m}$. Let the nominal motion control be realized using the computed torque strategy (2.26) with a PD (proportional, derivative) feedback:

$$\boldsymbol{\tau}^{q}(t) = \mathbf{M}(\mathbf{q}_r(t))\mathbf{u}(t) + \mathbf{c}(\mathbf{q}_r(t), \dot{\mathbf{q}}_r(t)) + \mathbf{g}(\mathbf{q}_r(t)) + \boldsymbol{\tau}^{f}(\mathbf{q}_r(t), \dot{\mathbf{q}}_r(t), t); \tag{6.3a}$$

$$\mathbf{u}(t) = \ddot{\mathbf{q}}_r(t) - \mathbf{K}_p \mathbf{e}(t) - \mathbf{K}_d \dot{\mathbf{e}}(t), \tag{6.3b}$$

where $\mathbf{e}(t) = \mathbf{q}(t) - \mathbf{q}_r(t)$ is the difference between the actual and the reference joint motions, while $\mathbf{K}_p = diag[k_{p,1},\ldots,k_{p,n}]$ and $\mathbf{K}_d = diag[k_{d,1},\ldots,k_{d,n}]$ are matrices of positive proportional and derivative gains, respectively. The vibration compensation can be achieved by adding a term

$$\boldsymbol{\tau}^{v}(t) = \left({}^{0}_{n}\mathbf{J}(\mathbf{q}(t))\right)^{T} \begin{bmatrix} {}^{0}_{n}\mathbf{O}(\mathbf{q}(t)) & \mathbf{0}_{3\times 3} \\ \mathbf{S}\left({}^{0}_{n}\mathbf{x}(t)\right){}^{0}_{n}\mathbf{O}(\mathbf{q}(t)) & {}^{0}_{n}\mathbf{O}(\mathbf{q}(t)) \end{bmatrix} \begin{bmatrix} {}^{n}\mathbf{f}\left({}^{n}\mathbf{p}'(t)\right) \\ {}^{n}\mathbf{m}\left({}^{n}\boldsymbol{\delta}'(t)\right) \end{bmatrix} \tag{6.4}$$

to the motion control law $\boldsymbol{\tau}^{q}$ defined by (6.3a,b). Here, ${}^{n}\mathbf{f}$ and ${}^{n}\mathbf{m}$ represent feedback control laws that should ensure attenuation of the linear and angular elastic tip displacements, respectively. The vibration compensator (6.4) needs to be active in the frequency range of vibrations and must not interfere with the computed torque controller. In practice, reference joint trajectories are chosen to lie below the lowest eigenfrequency of the structure. The motion controller (6.3a,b) acts within the bandwidth of the joint references, while the compensator (6.4) acts in the frequency range of elastic vibrations. As $\boldsymbol{\tau}^{q}$ and $\boldsymbol{\tau}^{v}$ approximately act complementary to each other, we can determine them separately. The complete control law is:

$$\boldsymbol{\tau} = \boldsymbol{\tau}^{q} + \boldsymbol{\tau}^{v}. \tag{6.5}$$

Design of the feedback laws $\mathbf{u}^{l}$ and $\mathbf{u}^{a}$ requires knowledge of the functions ${}^{n}\mathbf{f}({}^{n}\mathbf{p}')$ and ${}^{n}\mathbf{m}({}^{n}\boldsymbol{\delta}')$. A data-driven approach to acquire such knowledge will be presented in the case-study with the RRR robot.

6.3 Vibrations at the tip of the RRR robot

The RRR robot is used to demonstrate the utility of the control strategy formulated in the previous section. For the sake of firm experimental validation, an elastic plastic tube is used as the last robot link instead of the standard metal rod shown in Fig. 1.1. A photo of the RRR robot with the plastic tube attached is shown in Fig. 6.2. Low rigidity of the tube facilitates vibrations at the robot tip, which impedes accurate realization of the reference trajectories in the robot task space. With reference to Fig. 4.2, presenting the kinematic wire frame of the robot, during the robot movements the flexible link shows translational elastic deflections along ${}^{3}y$ and ${}^{3}z$ axes. Deflections along ${}^{3}y$ are more profound, as they arise inline with the motion of the last joint. The elastic deformations result in translational vibrations at the tip, as high torsional stiffness of the link prevents rotational vibrations. A triaxial accelerometer mounted at the robot tip is used for detecting the tip vibrations. This sensor has poor low frequency characteristics, as it cannot measure constant accelerations and does not provide reliable measurements up to a lower cut-off frequency of 3 [Hz]. However, its band of reliable measurements covers the frequency range of the tip vibrations. The sensor is mounted in the origin of the coordinate frame '3' indicated in Fig. 4.1. Its mutually orthogonal principal axes are aligned with the frame axes, which enables direct measurement of accelerations of the translational tip vibrations.

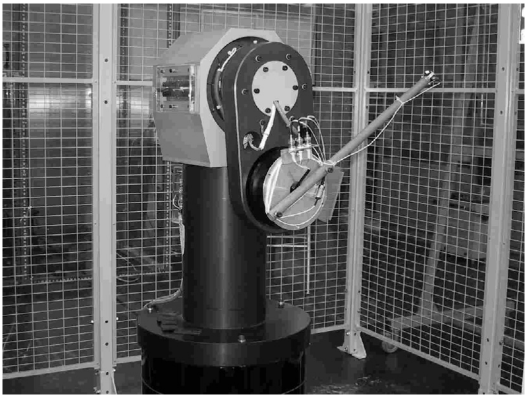


Figure 6.2: The RRR robot with the elastic tube as the last link.

The effect of the link elasticity shows up in the motion task depicted in Fig. 6.3. In this task, the joints are displaced for π [rad] in 2.5 [s], each time in opposite direction and with zero initial/terminal velocities and accelerations. Such displacements were experimentally realized using the controller (6.3a,b). The RRR robot rigid-body dynamic model (A.6),(A.7) was implemented to compensate for the robot nonlinearities. The model parameters that correspond to the plastic tube as the last robot link were obtained using the batch least-squares estimation explained in subsections 3.4.2.1 and 4.5.1. The feedback tuning, i.e., the choice of the PD gains, was primarily focused on achieving a satisfactory position accuracy in the robot joint space. For joints 1 and 2, the desired accuracy was limited to the range of $[-2,2]\times10^{-3}$ [rad], while for joint 3 the range was $[-3,3]\times10^{-3}$ [rad]. The experimentally achieved joint velocities and accelerations are shown in Fig. 6.4. They are not measured, but reconstructed using the Kalman observer (5.9)-(5.12), designed in section 5.3. The observer filters-out quantization noise caused by the incremental encoders. The given time-histories for joint 3 reveal the presence of a high frequency component superimposed to the nominal velocity and acceleration patterns. The power spectrum densities (PSD) of $\dot{\bar{q}}_3$ and $\ddot{\bar{q}}_3$, presented in Fig. 6.5, indicate a dominant harmonic component at 28 [Hz]. It is shown in the next section that this harmonic corresponds to the first eigenfrequency of the last link. It appears that the joint motions shown in Fig. 6.3 induce elastic deflections of the last link that influence measurements from the position sensor in joint 3.

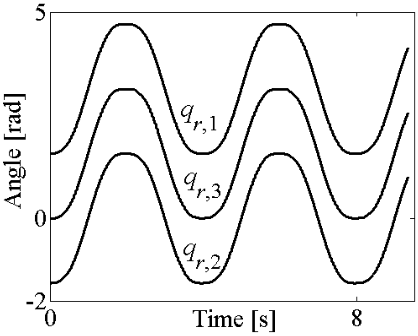


Figure 6.3: Reference joint motions.

Signals obtained from the triaxial accelerometer confirm the presence of tip vibrations induced by elastic deflections. For instance, one may investigate signals obtained from the accelerometers aligned with the ${}^{3}y$ and ${}^{3}z$ axes of the tip coordinate frame. These signals are shown in Fig. 6.6 together with the reference tip accelerations:

$$\begin{bmatrix} {}^{3}_{3}\ddot{x}_r(t) \\ {}^{3}_{3}\ddot{y}_r(t) \\ {}^{3}_{3}\ddot{z}_r(t) \end{bmatrix} = \left({}^{0}_{3}\mathbf{O}(\mathbf{q}_r(t))\right)^T [{}^{0}_{n}\mathbf{J}_P\left(\mathbf{q}_r(t)\right)\ddot{\mathbf{q}}_r(t) + {}^{0}_{n}\dot{\mathbf{J}}_P\left(\mathbf{q}_r(t)\right)\dot{\mathbf{q}}_r(t)], \tag{6.6}$$

where ${}^{0}_{n}\mathbf{J}_P \in \mathbb{R}^{3\times 3}$ is the upper half of the manipulator geometric Jacobian ${}^{0}_{n}\mathbf{J}$, as defined by (3.6). The elements of the RRR robot geometric Jacobian in closed-form are given by (A.4), in the Appendix. The elements of the vector ${}^{0}_{3}\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}}$ are also available in closed-form, and they are given by (A.5). The relation (6.6) is obtained by extracting the upper half of the second order differential kinematic model (3.9), which relates the joint space variables with the linear accelerations of the robot tip. The right-hand side of (6.6) is premultiplied by the transpose of ${}^{0}_{3}\mathbf{O}(\mathbf{q}_r(t))$ to represent the tip accelerations in the coordinate frame '3' of the tip. Such representation facilitates analysis, since the outputs from the triaxial accelerometer are referred to frame '3'.

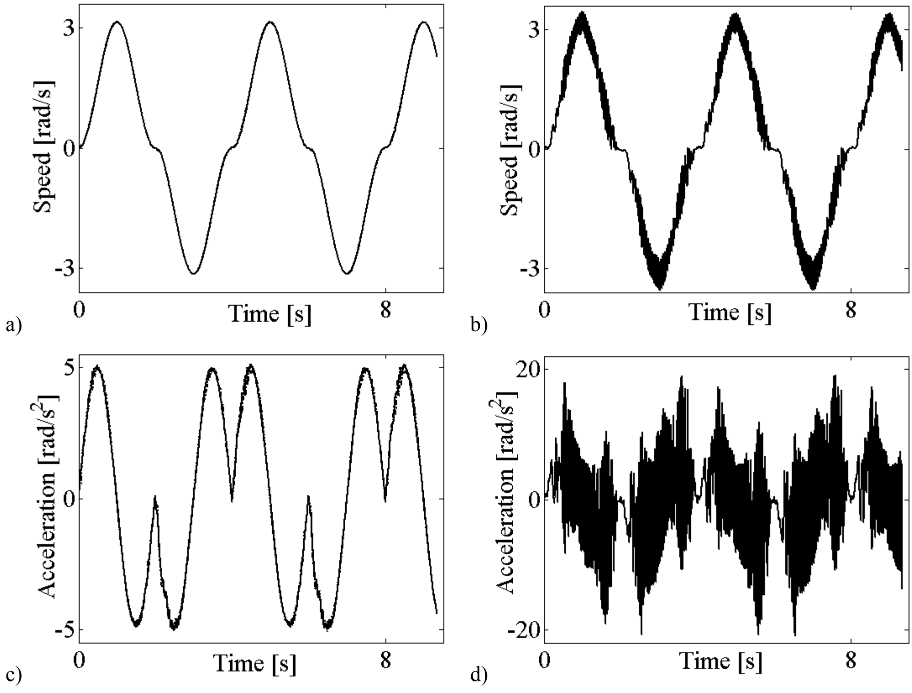


Figure 6.4: Experimental joint velocities and accelerations:
a) $\bar{\dot{q}}_1(t)$ and $\bar{\dot{q}}_2(t)$, b) $\bar{\dot{q}}_3(t)$, c) $\bar{\ddot{q}}_1(t)$ and $\bar{\ddot{q}}_2(t)$, d) $\bar{\ddot{q}}_3(t)$.

By inspection of the expected and measured accelerations along the ${}^{3}y$ and ${}^{3}z$ axes, shown in Fig. 6.6, one notices the presence of an oscillatory component in the output of the accelerometer aligned with ${}^{3}y$ axis. The PSD of this signal, depicted in Fig. 6.7, shows that the accelerometer detects an oscillation at 28 [Hz]. This harmonic has already been observed in the power spectra of $\bar{\dot{q}}_3$ and $\bar{\ddot{q}}_3$. A nice correspondence between the desired and measured time histories shown on the bottom in Fig. 6.6, confirms appropriate calibration of the installed accelerometers. Slight discrepancies are due to the poor low-frequency characteristics of the sensor. The oscillations in the output of the accelerometer aligned with ${}^{3}z$ axis are slightly influenced by tip vibrations, since they are less profound along this axis, but they are dominantly caused by sensor noise and interference between the axial accelerometers ("cross-talk" effect).

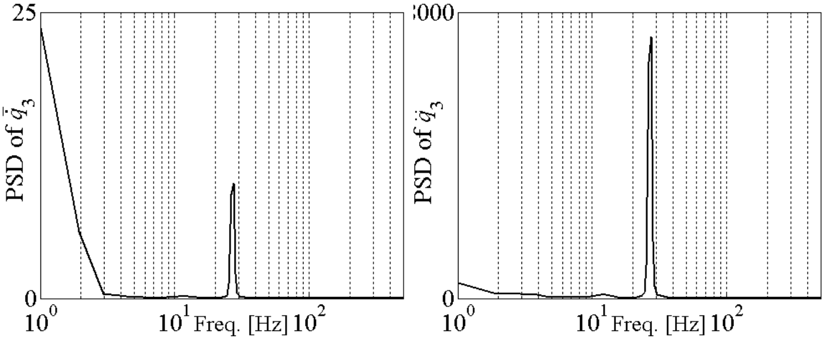


Figure 6.5: Power spectrum densities of the velocity (left) and acceleration (right) of joint 3.

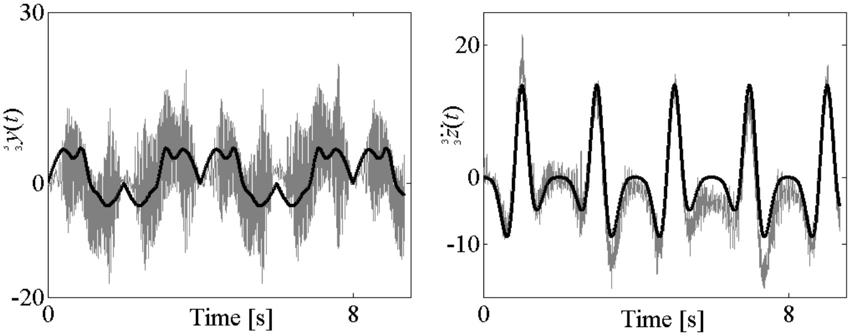


Figure 6.6: Tip accelerations: reference (solid) and accelerometer outputs (grey).

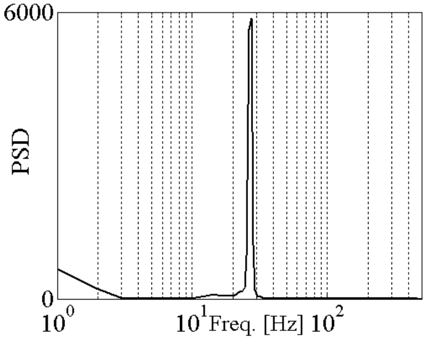


Figure 6.7: Power spectrum density of the output of the accelerometer aligned with ${}^{3}y$ axis.

The analysis performed both in the robot joint and task spaces, confirms the appearance of translational tip vibrations in the presence of the elastic robot link. The nominal computed-torque motion controller with a PD feedback has been primarily designed to enforce the desired position accuracy for a given motion reference. A simple PD feedback controller is not capable to jointly realize the desired accuracy and prevent the rise of the tip vibrations that imminently degrade the motion performance in the robot task space. Therefore, the nominal motion controller should be supported with a vibration compensator, whose design is presented next.

6.4 Design of a vibration compensator

Section 6.2 formulates a strategy to tackle the problem of tip vibrations. This strategy requires the knowledge of a relation between the equivalent force/moment acting at the tip and the corresponding linear and angular tip displacements. On the RRR robot, only translational tip vibrations can be observed. A relation between the elastic force and the translational displacements can be established either using the first principles modelling [24,102,168] or via experimental identification. The modelling approach is theoretically well-established but also more time-consuming. On the other hand, "black-box" identification does not provide the analytical dynamics underlying the tip vibrations, but it more efficiently delivers models suitable for control design. Here, the identification approach is adopted.

Accurate information about the actual velocity and especially the amplitude of the elastic tip displacement cannot be acquired from the accelerometer output signals, because of the poor accelerometers' low-frequency characteristics. Instead of considering the amplitude of the vibrations, one should rather stick to the signals directly obtained from the accelerometers, since the frequencies of these signals are equal to the frequencies of the vibrations, and their magnitudes are proportional to the amplitudes of the vibrations. The acceleration of vibrations of smaller amplitude is also lower, and vice versa. Therefore, instead of looking for the relation ${}^n\mathbf{f}({}^n\mathbf{p}')$ one can determine a relation between ${}^n\mathbf{f}$ and ${}^n\ddot{\mathbf{p}}'$, since the tip vibrations can be attenuated by counteracting the force that induces these accelerations. The vibrations occur at specific eigenfrequencies that should be covered in the band of reliable measurements of the applied accelerometers. As mentioned in the previous section, the vibrations at the tip of the RRR robot are within the band of the triaxial accelerometer. To enhance the reliability of the measurements, a Yulewalk bandpass filter of eight order, with Bode plot shown in Fig. 6.8, is used to extract useful information about the vibrations from the accelerometer signals. For each accelerometer signal, one filter is used. The filter‘s pass band covers the harmonic components of the accelerometer signal that correspond to tip vibrations, while the other signal harmonics are attenuated at the filter's output. Hence, the output of the Yulewalk filter should be used in the design of the vibration compensator.

The relationship between the equivalent elastic force and filtered acceleration is identified by three sets of measurements with random excitation. Each set identifies one Cartesian projection ${}^3f_i({}^3\ddot{p}_i')$ ($i=x,y,z$) of the relationship. By applying

$$\boldsymbol{\tau}(t)=\boldsymbol{\tau}^q(\mathbf{q}_r^{\mathrm{o}})+\left({}_3^0\mathbf{J}_P(\mathbf{q}_r^{\mathrm{o}})\right)^T{}_3^0\mathbf{O}(\mathbf{q}_r^{\mathrm{o}})^3\mathbf{f}(\nu(t)),\tag{6.7}$$

with $\boldsymbol{\tau}^q$ defined by (6.3a,b), the robot is kept in some prescribed static configuration $\mathbf{q}_r^{\mathrm{o}}\in\mathbb{R}^3$ ($\dot{\mathbf{q}}_r^{\mathrm{o}}=\mathbf{0}$). A noise excitation $\nu(t)$ is injected into the system via the element ${}^3f_i=v(t)$ ($i=x,y,z$) of the vector ${}^3\mathbf{f}\in\mathbb{R}^3$. The other elements ${}^3f_j$ ($j=x,y,z$; $j\neq i$) of ${}^3\mathbf{f}$ are identically zero. The output ${}^3\ddot{p}_i'(s)$ ($i=x,y,z$) of the corresponding accelerometer is measured as response to the excitation. From the excitation and response signals one determines the frequency response function (FRF) ${}^3\ddot{p}_i'(j\omega)/{}^3f_i(j\omega)$ ($i=x,y,z$), where ω denotes angular frequency. For illustration, Fig. 6.9 presents FRF's ${}^3\ddot{p}_y'(j\omega)/{}^3f_y(j\omega)$, obtained for 16 different postures of joints 2 and 3: $[0\ 0]$, $[0\ \pi/2]$,..., $[3\pi/2\ 3\pi/2]$ [rad]. The given FRF's show consistent relation between ${}^3f_y$ and ${}^3\ddot{p}_y'$, except in the low frequency region where the coherence of measurements was poor. A high consistency has been also observed from the FRF's that correspond to ${}^3\ddot{p}_z'(j\omega)/{}^3f_z(j\omega)$. The transfer from ${}^3f_x$ to ${}^3\ddot{p}_x'$

was not considered, as longitudinal vibrations of the last link (along ${}^3x$) are negligible.

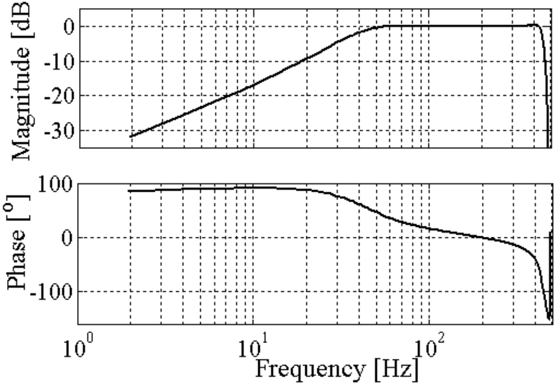


Figure 6.8: Yulewalk filter of the 8th order used for filtering the accelerometer outputs.

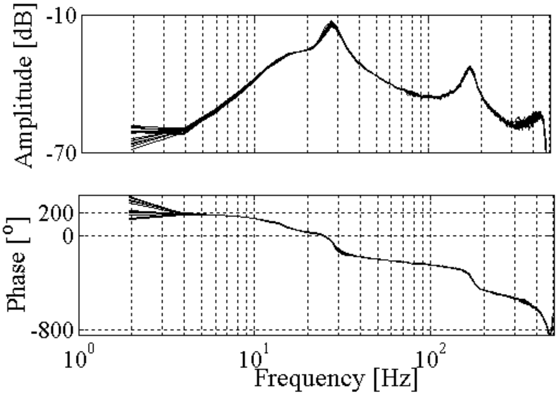


Figure 6.9: Measured frequency responses of ${}^3\ddot{p}'_z(s)/{}^3f_z(s)$.

The FRF's presented in Fig. 6.9 clearly reveal two eigenfrequencies – one at 28 [Hz] and another at 180 [Hz]. The first one has been observed in the previous section as the frequency of the tip vibrations. The second eigenfrequency was not excited in the experiments, since it was not found in the sensor signals. Therefore, it was not further investigated. The strategy which is used to identify a relationship between the equivalent elastic forces and translational vibration displacements could also be applied if a relation between the equivalent elastic torque and angles of rotational vibrations should be identified. Such a relation, however, is not needed for the RRR robot.

A suitable vibration compensator should reduce the tip vibrations at 28 [Hz], but it must not interfere with the motion controller (6.3a,b). This implies that a contribution from the compensator within the frequency range of the reference tip trajectory is not wanted. The vibration compensator should be active only beyond this low-frequency band, since it should act complementary to the nominal motion controller.

Since dominant vibrations arise along the ${}^3y$-axis of the tip coordinate frame, it is sufficient to design a compensator for this axis only. Hence, only the second component ${}^3f_y$ of the feedback compensation law ${}^3\mathbf{f}$ in (6.4) should be designed. The feedback law treats the tip vibrations in the same coordinate frame '3' where the vibrations are measured. The control objective is to keep ${}^3\ddot{p}'_y(t)$ at zero by appropriate change of ${}^3f_y(t)$. In other words, the goal is to design a feedback compensator C which stabilizes the plant $P(j\omega)={}^3\ddot{p}'_y(j\omega)/{}^3f_y(j\omega)$ and performs output regulation of P with zero as the set-point. The vibration attenuation problem is thus formulated as a linear regulation problem, which is the key contribution of the strategy presented in this chapter. The compensator was determined by shaping the open-loop gain PC. The plant P was represented by an average of the FRFs shown in Fig. 6.9, which is in accordance with formula (3.64b). The average dynamics was used for the product PC. This product was shaped to achieve: as high as possible attenuation of the eigenfrequency at 28 [Hz], closed-loop stability, and a low open-loop gain below and beyond 28 [Hz]. A low gain below 28 [Hz] is needed to avoid interference between the nominal motion controller and the vibration compensator. A low gain beyond 28 [Hz] ensures robustness against parasitic effects at high frequencies. The Bode plots of the resulting compensator and of the corresponding open-loop gain are shown in Fig. 6.10 and Fig. 6.11, respectively. The open-loop gain indicates that the regulated system is stable, the phase margin of 30° seems sufficient, and a high-frequency roll-off is realized. The magnitude plot of the sensitivity function $S=1/(1+PC)$, given in Fig. 6.12, shows attenuation of the first eigenfrequency of about 14 [dB], but a larger S on both sides of 28 [Hz]. The peaks of S are below 7 [dB], which can be tolerated.

This completes the design of the vibration compensator. Experimental effectiveness of the design is demonstrated in the next section.

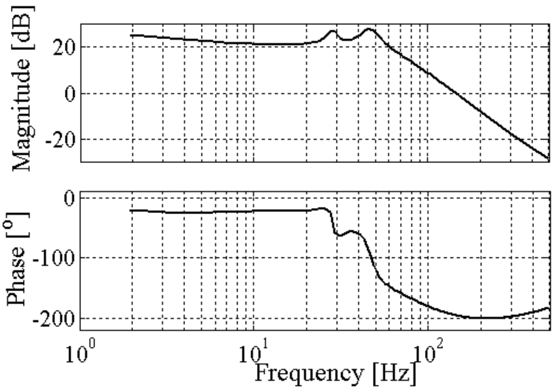


Figure 6.10: Vibration compensator.

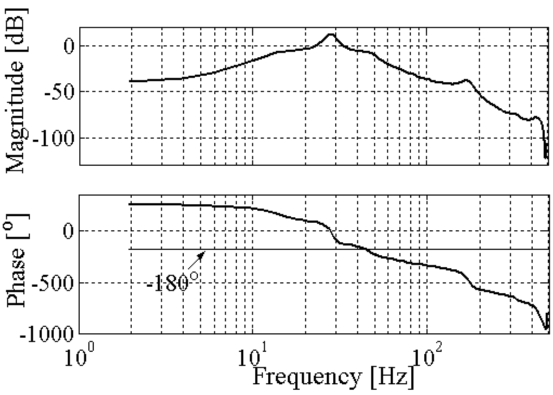


Figure 6.11: Open-loop gain with the vibration compensator.

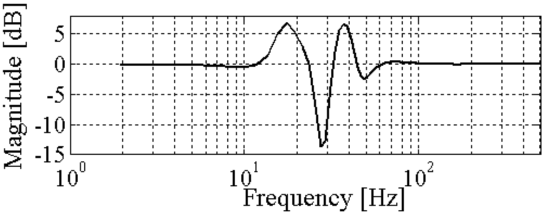


Figure 6.12: Sensitivity with the vibration compensator.

6.5 Experimental evaluation

This section presents experimental results in reducing the tip vibrations. The motion task depicted in Fig. 6.3 was considered first. When the vibration compensator was disabled, the output of the accelerometer aligned with the ${}^3y$ -axis is as shown on the left-hand side in Fig. 6.6. The accelerometer output shown on the left-hand side in Fig. 6.13 was obtained when the vibration compensator was enabled. Clearly, there is a significant attenuation of the vibrations in the latter case. This is also confirmed by the plot on the right-hand side in Fig. 6.13, showing negligible power of the accelerometer output at 28 [Hz] when the vibration compensator is active.

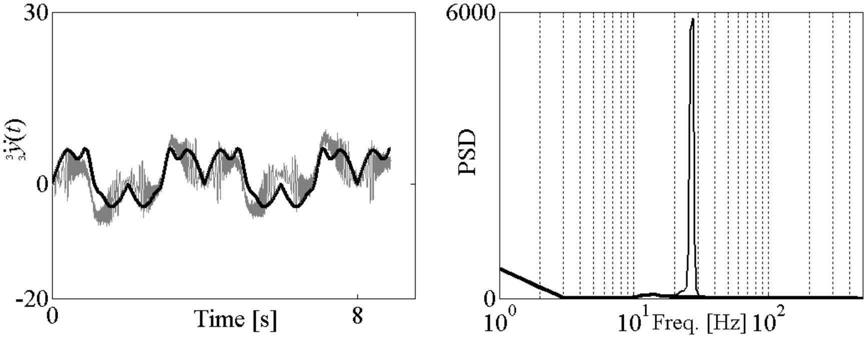


Figure 6.13: Left: reference and measured acceleration with vibration compensation enabled; right: power spectrum densities of the measured acceleration ${}^3_3\ddot{y}(t)$ without (thin) and with (thick) vibration compensation

In another experiment, the writing task depicted in Fig. 3.4 was used. The joint trajectory corresponding to this task is shown in Fig. 4.3. The objective of the experiment was to test robustness properties of the vibration compensator. For this reason, the resonance at 28 [Hz] was amplified by increasing the $k_{d,3}$ gain in the motion controller (6.3a,b). The FRF ${}^3\ddot{p}'_y(j\omega)/{}^3f_y(j\omega)$ corresponding to an increased $k_{d,3}$ differs from those in Fig. 6.9, that are used in the design of the vibration compensator. The differences occur since an increase of $k_{d,3}$ amplifies high frequency eigenmodes and tilts the FRF of the plant P. Apart from the amplified resonance, the second experiment introduced an additional difference in the operating conditions for the RRR robot: during the design of the vibration compensator no contact between the robot tip and environment was made, while for the sake of visualisation of the executed motion, in the writing task the robot tip was in contact with a whiteboard. The contact dynamics is a factor not taken into account during the compensator design. This factor can be considered as a disturbance to the regulation process. At the beginning of the experiment the vibration compensator was disabled. Realization of the reference joint motions induced significant chattering in the acceleration of joint 3 and in the output of the triaxial accelerometer. When the vibration compensator was enabled, an immediate attenuation of the vibrations occurred. This is illustrated in Fig. 6.14, representing the reference and actual accelerations of joint 3 (left), and the reference and measured acceleration along ${}^{3}y$ axis (right).

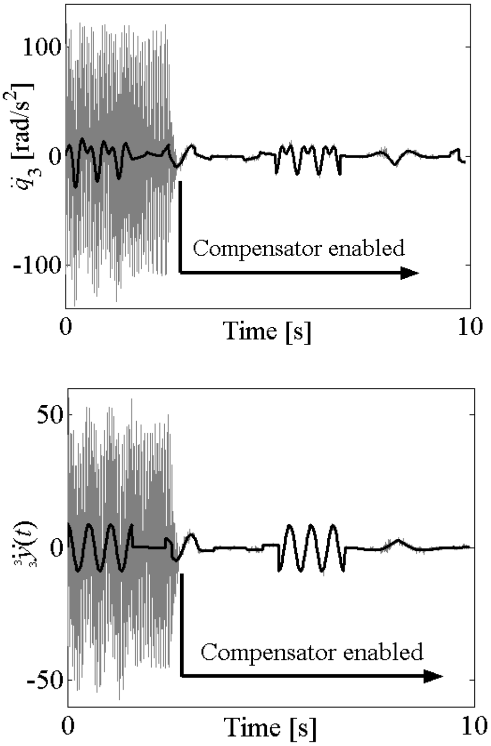


Figure 6.14: Top: reference (black) and reconstructed (grey) accelerations of joint 3; bottom: reference (black) and measured (grey) accelerations along ${}^{3}y$ -axis.

Fig. 6.15 shows a photo of the executed sequence of letters. One may notice that the compensator was enabled when writing the letter ‘o’. Before the compensator was enabled, the letters were written with considerable oscillations around the reference path. When enabled, the compensator attenuated the oscillations and the writing proceeded smoothly and accurately. Cartesian errors along ${}^{0}x$ and ${}^{0}z$ directions, presented in Fig. 6.16, reduce after the vibration compensation is activated. This example shows robustness of the compensator against variations in the plant dynamics and disturbance effects.

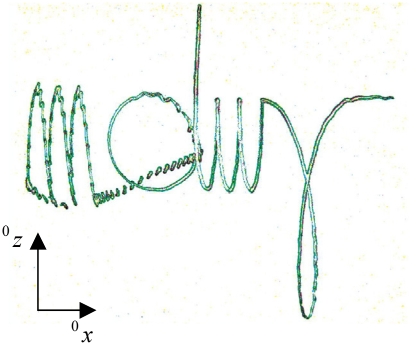


Figure 6.15: Photo of the tip path.

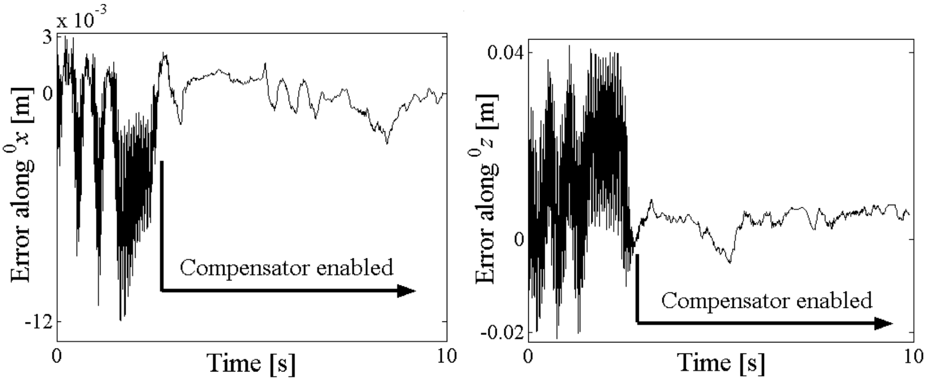


Figure 6.16: Cartesian errors in the writing task.

6.6 Summary

This chapter presents a method for reducing vibrations arising at the tip of a robot due to flexibility in the robot construction.

The method requires knowledge of the flexible dynamics and sensors for measuring vibrations. Vibrations can be measured using accelerometers. Data-driven modelling of flexible dynamics contributes to the efficiency of the modelling step. The identified model can be combined with a rigid-body model of the robot dynamics. The combined model enables suppression of tip vibrations directly in the coordinate frame where the vibrations are measured. The problem of vibration compensation is reduced to an ordinary regulation problem, solvable using well-developed linear theory. This facilitates design of a vibration compensator. A loop-shaping design was used in this chapter.

Practical effectiveness and robust performance of the compensator were experimentally verified on the RRR robot.

Chapter 7

Data-based approaches to motion control

The results of the previous two chapters confirm that knowledge of robot experimental characteristics can be beneficial for the quality of motion control. An advanced data-driven feedback control design can compensate for more versatile system dynamics and realize motion control of high performance. Since the advanced designs presented in the previous two chapters are all model-based, their performance strongly depends on the quality of the underlying models. This chapter investigates if motion control of high quality can be realized while avoiding the need for system models. The control techniques to be examined here employ only time-domain data observed from the robot. So, these techniques are data-based. The reasons for investigating the data-based control designs were explained in section 2.3. Shortly, these designs should be capable to deliver controller tuning which is optimized against actual plant behaviour, since they are based on experimental data that are rich with effects describing the relevant plant dynamics. Section 2.3 indicated the motives for particular selection of the data-based designs to be examined here. Namely, under given circumstances, e.g., repetitive nature of the motion reference, availability of a prior knowledge of the parasitic dynamics, time-efficiency, etc., one of the designs is more appropriate solution than others. The research questions related with the data-based control designs were posed in section 2.3. The answers to these questions will be given while presenting the underlying theories of these designs and during demonstrations of their practical effects on the RRR robot. The chapter is organized as follows. Section 7.1 formulates the problem of data-based control. Section 7.2 describes iterative learning control. Section 7.3 treats iterative feedback tuning. Section 7.4 develops a new data-based loop-shaping control design method. The chapter summary is given at the end. Parts of the material presented in this chapter have been also published in [29,95,136].

7.1 Problem description

A model-based design can deliver a motion controller of high performance if the implemented model is an accurate description of the plant dynamics. Modelling approximations and model inaccuracies limit the control performance and may even lead to instability. In practice, it is sometimes time consuming and too costly to model the plant accurately. Moreover, if the design resorts to a model of high complexity, the resulting controller can also be very complex, which is not practical for real time implementation. This chapter discusses another approach to control design which circumvents the need for plant modelling.

In data-based (DB) control, input-output data observed from the plant is of major importance. By functional use of this data, one may synthesize the controller or directly derive the control action. The DB control field encompasses versatile research interests, strategies, and case studies. A literature survey of this field is given in section 1.4. In this chapter, it is investigated if DB control methods can deliver high performance solutions to the robot motion control problem. The following DB techniques are considered: iterative learning control, iterative feedback tuning, and data-based loop shaping. The first two techniques are already known from the literature, but here they are formulated as ready-to-use solutions for motion control of direct-drive robots. Because of high dynamic couplings and nonlinearities, these robots are considered as very challenging for accurate motion control. The third technique is an original contribution of this thesis.

The iterative learning control (ILC) is treated to show that the quality of nonlinear model-based motion control can be significantly improved by functional use of experimentally obtained data. An ILC design enables offline derivation of sequences of additional inputs that should supplement outputs of the nominal feedback controllers. The derived control inputs have corrective effect on the feedback control signals, since use of these additional inputs results in improved motion accuracy than obtained with feedback control only. Offline derivation of the corrective signals is performed for a known reference motion task which is repetitive in nature. Thus, ILC is applicable in servo control problems only.

Contrary to ILC, iterative feedback tuning (IFT) is in general not limited to a particular motion task, and it can be used in both servo and tracking control problems. Another difference from ILC is that IFT does not deliver sequences of additional control inputs, but provides optimal feedback tuning for the desired control performance. Here, IFT is considered to show that plant models of even very high complexity still might be incapable to cover all effects present in the real system. Because of discrepancies between the model and the reality, the outcome from the control design utilizing model-driven data might not be identical with the outcome from the design utilizing experimental data.

The last technique to be presented is data-based shaping of feedback controllers for a desired closed-loop performance. This technique enables simplified off-line design of high-performance feedback motion controllers. Contrary to ILC but similarly with IFT, the data-based loop shaping can be used in an arbitrary motion task, and it is applicable in both servo and tracking control problems.

All three techniques are based on the measured input/output time-domain data that are sampled and of finite length. Typical signals of interest are control input to the plant $\{u(k)\}_{k=1,2,\ldots,N}$, the plant output $\{y(k)\}_{k=1,2,\ldots,N}$, and the error $\{e(k)\}_{k=1,2,\ldots,N}$ between y and the reference input. Here, k indicates a data point at time $t = kT_s$ (T_s is the sampling time), and N is the total number of data points. While mathematical analysis of ILC design will be facilitated assuming deterministic properties of the considered signals, derivation of IFT and data-based loop shaping control design algorithms will be simplified assuming stochastic properties of the measured data and by performing stochastic mathematical analysis.

All considered techniques will be theoretically formulated and practically demonstrated on the RRR robot. Results of their experimental implementations will be provided.

7.2 Iterative learning control

In many applications industrial robots are programmed to repeatedly execute the same task. Under similar operating conditions, the error between the reference and executed robot motions is highly repetitive in different iterations of the same task. Iterative learning control (ILC) allows iteratively compensating for and eventually removing this repetitive error. The typical formulation of the ILC problem is: given the reference robot trajectory $\mathbf{q}_r(t)$, find the components of the control inputs that steer the robot motions $\mathbf{q}(t)$ as close as possible to $\mathbf{q}_r(t)$. The appropriate input is found using an iterative search procedure. The convergence of this procedure should be guaranteed by an appropriate update law. The law defines a mechanism for refining the input based on input and tracking errors from previous iterations. The refinement process convergences to the input which minimizes the tracking error. The next section formalizes the application of one ILC algorithm for the problem of direct-drive robot motion control.

7.2.1 Learning algorithm

The considered ILC algorithm appeared first in [124], and it is applicable to linear plant dynamics. Linearization of a direct-drive robot can be achieved using the computed-torque control method (2.26) presented in subsection 2.2.2. The i^{th} element of the control input $\mathbf{u}$ is defined by:

$$u_i(t) = \ddot{q}_{r,i}(t) - u_i^*(t) + u_i^{ilc}(t), \tag{7.1}$$

where $\ddot{q}_{r,i}$ is the reference acceleration, u_i^* is the output of the known feedback controller C_i based on the position error $e_i = q_i - q_{r,i}$, and u_i^{ilc} is the output of the iterative learning controller. Section 3.7 explains procedures to identify and model linearized dynamics. The single-input single-output linear plant associated to joint *i* can be described by a transfer function P_i. Under the standard assumptions [125] that the reference trajectory $q_{r,i}(t)$ is of finite length and that the initial conditions are identical at the beginning of each execution of $q_{r,i}(t)$ (trial), the derivation of u_i^{ilc} can be explained using the block diagram shown in Fig. 7.1.

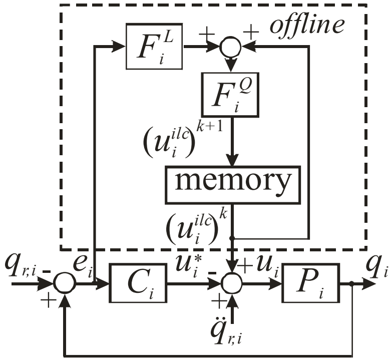


Figure 7.1: The principle scheme of iterative learning control.

The output of the learning controller is updated after each trial *k*. The update law is given in the Laplace domain, which is allowed under standard assumptions given above:

$$\left(U_i^{ilc}(s)\right)^{k+1} = F_i^Q(s)\left(\left(U_i^{ilc}(s)\right)^k + F_i^L(s)\left(E_i(s)\right)^k\right), \tag{7.2}$$

where F_i^L is the learning filter, F_i^Q is the robustness filter, while U_i^{ilc} and E_i are the Laplace transforms of u_i^{ilc} and e_i, respectively. According to Fig. 7.1, the errors in two successive trials can be calculated using the closed-loop transfer functions from U_i^{ilc} to E_i, from $Q_{r,i}$ to E_i, and from $A_{r,i}$ to E_i ($Q_{r,i}$ and $A_{r,i}$ are the Laplace transforms of $q_{r,i}$ and $\ddot{q}_{r,i}$, respectively):

$$\left(E_i(s)\right)^k = -P_i(s)S_i(s)\left(U_i^{ilc}(s)\right)^k + S_i(s)Q_{r,i}(s) - P_i(s)S_i(s)A_{r,i}(s) \tag{7.3}$$

$$\left(E_i(s)\right)^{k+1} = -P_i(s)S_i(s)\left(U_i^{ilc}(s)\right)^{k+1} + S_i(s)Q_{r,i}(s) - P_i(s)S_i(s)A_{r,i}(s) \tag{7.4}$$

where $S_i = 1/(1+P_iC_i)$ is the sensitivity function. Substitution of (7.2) and (7.3) into (7.4), gives the equation which defines the change of the tracking errors in successive trials:

$$\begin{aligned}\left(E_i(s)\right)^{k+1} &= \left(E_i(s)\right)^k F_i^Q(s)\left(1 - F_i^L(s)P_i(s)S_i(s)\right) \\ &\quad + S_i(s)\left(1 - F_i^Q(s)\right)\left(Q_{r,i}(s) - P_i(s)A_{r,i}(s)\right).\end{aligned} \tag{7.5}$$

A sufficient condition for the first term on the right hand side of (7.5) to converge to zero throughout the trials is given by:

$$\left\| F_i^Q(s)\left(1 - F_i^L(s)P_i(s)S_i(s)\right)\right\|_\infty < 1, \tag{7.6}$$

with $\|.\|_\infty$ denoting the infinity norm.

To maximize the speed of convergence, the learning filter should be chosen as:

$$F_i^L(s) = \frac{1}{P_i(s)S_i(s)}. \tag{7.7}$$

The product P_iS_i is known as the process sensitivity function. If P_i is non-minimum phase, the learning filter will be unstable if the process sensitivity is inverted directly. A possible remedy for this problem is to invert P_iS_i using the Zero phase error tracking algorithm for digital control [172]. This algorithm uses a discrete-time model of the process sensitivity and provides an approximate inversion which is stable. Even if P_iS_i is minimum phase, the parametric representation of F_i^L is not an accurate representation of the real system dynamics, but more likely it covers only the low-frequency part of these dynamics. The role of the robustness filter F_i^Q is to ensure error convergence in the presence of the modelling errors. This filter should guarantee validity of (7.6) for the frequency components at which the model is inaccurate. Since the modelling errors usually arise in the high-frequency range, F_i^Q is typically chosen to be a low-pass filter. In the simplest case, below its cut-off frequency f_Q, the robustness filter has a pass-band equal to 1, while above f_Q its amplitude is decreasing. From (7.5), it can be seen that the robustness filter modifies the sensitivity function as $S_i(s)\left(1 - F_i^Q(s)\right)$. Consequently, within the pass-band of F_i^Q, the transfer functions from $Q_{r,i}$ to E_i and from $A_{r,i}$ to E_i are both equal to zero.

The F_i^Q-filter should influence the frequency content of $(u_i^{ilc})^{k+1}$ only by its magnitude characteristics and should not introduce any phase distortion. To explain this further, the error equation (7.5) should be discussed. The objective of learning is to achieve the error in trial k+1 to be smaller than the error in trial k, i.e., $\left|(e_i)^{k+1}\right| < \left|(e_i)^k\right|$. The convergence criterion (7.6) ensures decrease of the part of $(e_i)^{k+1}$ which directly depends on $(e_i)^k$. The contribution of the other term on the right-hand side of (7.5), which depends on $q_{r,i}, \ddot{q}_{r,i}$ and also influences $(e_i)^{k+1}$, diminishes if F_i^Q is a real scalar equal to 1. Therefore, the F_i^Q-filter must not introduce any phase shift, especially within its pass-band, since this would violate the condition $F_i^Q \equiv 1$. Consequently, the F_i^Q-filtering is a non-causal operation that is performed off-line, as indicated in Fig. 7.1.

In the next part, the considered ILC design is experimentally demonstrated on the RRR robot.

7.2.2 Control design for the RRR robot

It is sufficient to demonstrate the ILC design for the first joint of the RRR robot only, since the designs for the other two joints were accomplished in a similar way. The ILC design is based on the identified frequency response functions (FRF's) G_1^k ($k = 1,\ldots,16$) shown in Fig. 4.22 and on the nominal plant model P_1^{o} depicted in Fig. 4.23. The conventional PD control law (5.3) with $k_{p,1} = 1000$ and $k_{d,1} = 2\sqrt{k_{p,1}}$ is adopted for stabilization. These gains correspond to weak tuning of the feedback controller C_1, since they provide a cross-over frequency of 8 [Hz]. This frequency is almost two times lower than the cross-overs achievable using the more complex feedback control laws presented in sections 5.4 and 5.5. The parametric model of the process sensitivity is determined by the product $P_1^{\mathrm{o}}S_1^{\mathrm{o}}$, with $S_1^{\mathrm{o}} = 1/(1 + P_1^{\mathrm{o}}C_1)$. The learning filter F_1^L is determined as explained in the previous section. The Bode plot of this filter is shown in Fig. 7.2. The same figure presents the FRF's of the process sensitivities $G_1^kS_1^k$ with $S_1^k = 1/(1 + G_1^kC_1)$ ($k = 1,\ldots,16$) and the FRF's of the products $F_1^L G_1^k S_1^k$ ($k = 1,\ldots,16$). Ideally, all these products should feature 0 [dB] magnitude and 0° phase shift. From Fig. 7.2, it can be seen that the obtained F_1^L-filter matches the inverse of the process sensitivities until approximately 20 [Hz] and in the range from approximately 32 [Hz] until 50 [Hz]. Between 20 and 32 [Hz], the varying location and damping of the first resonance mode cause the deviations from ideal magnitude and phase characteristics.

As explained in the previous section, the robustness filter F_1^Q is introduced to preserve validity of the convergence criterion (7.6) at the frequencies where the products $F_1^L G_1^k S_1^k$ ($k = 1,\ldots,16$) deviate from the ideal characteristics. The filter F_1^Q is chosen to be a fourth order low-pass Butterworth filter with a pass-band amplitude equal to one. The cut-off frequency of F_1^Q is chosen to be 18 [Hz], which is sufficient for robustness against modelling uncertainty observed in Fig. 7.2 above 20 [Hz]. If the cut-off frequency of F_1^Q-filter is compared with the cross-over frequency of 8 [Hz], it appears that the learning process should provide error reduction well above the bandwidth of the joint servo-system.

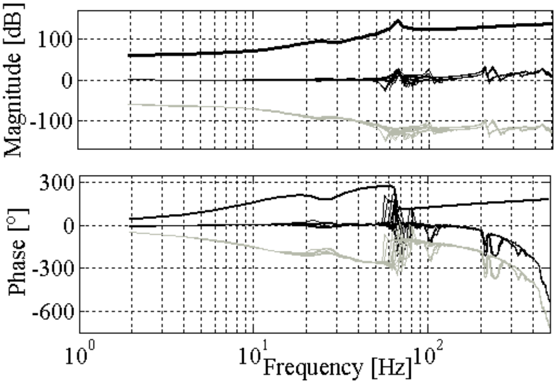


Figure 7.2: Bode plot of the learning filter F_1^L (black, thick), and the frequency response functions of $G_1^k S_1^k$ (grey) and of $F_1^L G_1^k S_1^k$ (black, thin) ($k = 1,\ldots,16$).

To demonstrate the effectiveness of the ILC design, experiments on the RRR robot have been conducted. In the experiments, learning controllers were implemented in all robot joints. Experimental results are presented in the next section.

7.2.3 Experimental results

The iterative learning control was experimentally assessed in the writing task. The original writing task, shown in Fig. 3.4, was extended with additional movements that ensure a cyclic shape of the robot tip path. The robot tip had to track the path depicted in Fig. 7.3a, starting from point 'A' and continuously moving in the indicated direction until 'A' has been reached again. The corresponding joint motions were calculated using the inverse kinematic model (A.3a,b), and they are presented in Fig. 7.3b.

During the experiment, the robot was controlled using the control law (2.26) in combination with the control input (7.1). Stabilization was ensured by means of PD feedback controllers that have been used for the design of the learning filters. The initial trial was carried out without the learning component u_i^{ilc} of the control input defined by (7.1). After each trial, the law (7.2) was used to update the learning component: the measured tracking errors were filtered with the corresponding learning filters, the filtered errors were added to the previous learning control components, and the obtained signals were filtered with the corresponding robustness filters. The updated learning component was used in the next trial.

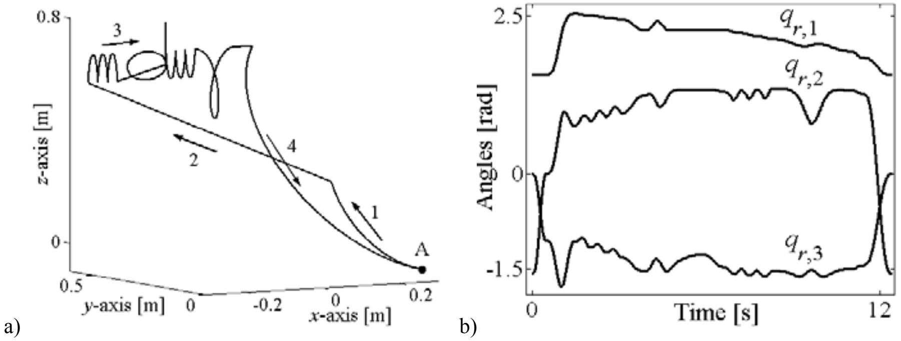


Figure 7.3: Experimental motion task: a) the robot-tip path, b) the corresponding joint motions.

The experiment with the writing task consisted of ten trials. The tracking errors obtained during the first, second, and tenth trial are depicted in Fig. 7.4. From this figure it is clear that significant error reduction was already achieved after the first trial, which means, during the initial application of the learning control component. The peak values of the errors obtained in the tenth trial are approximately ten times smaller then the errors in the first trial. The root-mean-square (RMS) values of the errors are depicted in Fig. 7.5. These plots show that the RMS of the errors in all joints keep decreasing until they reach lower bounds. Reaching these bounds indicates that the learning process has been converged. The RMS values of the final errors are over ten times smaller than before learning.

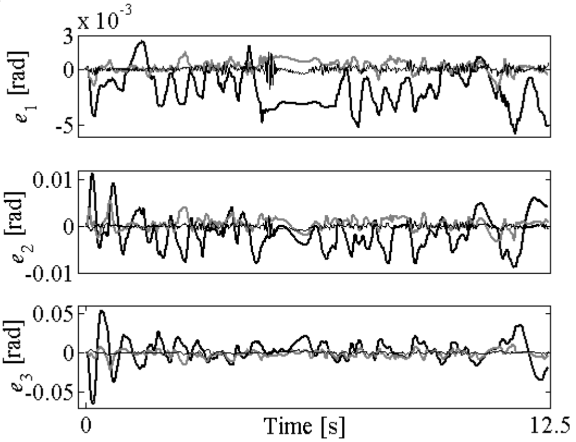


Figure 7.4: Joint tracking errors obtained during the first (black, thick), second (grey), and tenth (black, thin) trial.

The obtained errors are also analyzed in the frequency domain. Fig. 7.6 shows the power spectrum densities (PSD) of the joint errors achieved in the first, second, and tenth trial. These plots show a significant reduction of power until approximately 10 [Hz] in the first two joints and until 12 [Hz] in the third joint. Since these frequency ranges completely cover the harmonic content of the reference joint motions shown in Fig. 7.3b, it is confirmed that the learning process was effective in improving the performance of motion control. Since no learning took place in the high frequency range, there is no change in the harmonic content of the errors in this range, as apparent from Figs. 7.6a-c. From Figs. 7.6a,b, it can be seen that at the end of the learning process, the errors in the first and in the second joint have slightly amplified power between 13 [Hz] and 20 [Hz]. The amount of amplification is not an issue for the motion performance. However, it is peculiar why amplification occurs in the frequency range within the bandwidth of the corresponding robustness filters (18 [Hz] for joint 1 and 25 [Hz] for joint 2). This phenomenon might be caused by non-reproducible effects, such as variable friction in the robot joints and variable sampling times of the PC-based control system. Both effects are evident in the experiments on the RRR robot. Another reason for the amplification phenomenon could be the influence of residual nonlinear behaviours contained in the measured data. The model-based compensation strongly reduces robot nonlinearities but hardly eliminates these. The residual nonlinearities influence the outcome of iterative learning control design, which does not directly take into account nonlinear effects. The design itself assumes linear plant dynamics only. Therefore, an interesting topic for future research could be the analysis of the influence of nonlinear behaviours on the considered iterative learning control algorithm. The amplification of the harmonic components might have been avoided by reducing the pass-band amplitude of the robustness filter. However, this would slow down the convergence rate of learning, i.e., more trials would be needed for the errors to reach their lower bounds. There is no amplification of the error harmonic content between 13 [Hz] and 20 [Hz] in the third joint, as seen in Fig. 7.6c.

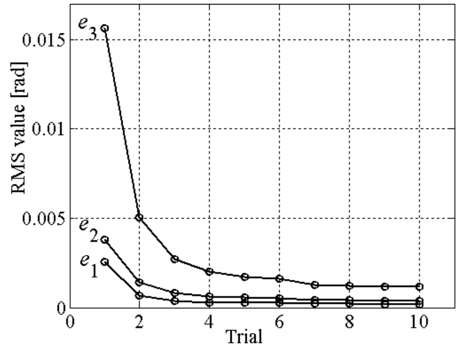


Figure 7.5: Root-mean-square values of the joint tracking errors.

Iterative learning control is a data-based technique used to derive the control component in repetitive motion. Although very effective in improving the motion control performance, ILC remains limited by the repetitive nature of the desired motion. The next two sections examine two different data-based techniques that primarily focus on feedback control design. These feedback designs are applicable in arbitrary motion tasks.

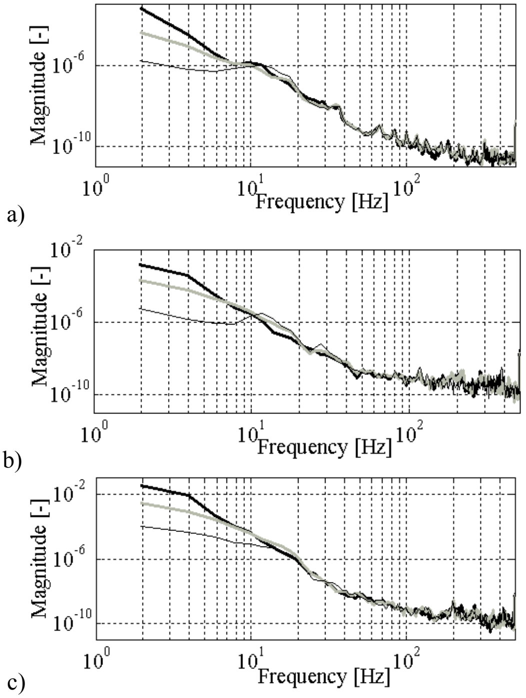


Figure 7.6: The power spectral densities of the tracking errors obtained during the first (black, thick), second (grey), and tenth (black, thin) trial: a) joint 1, b) joint 2, c) joint 3.

7.3 Iterative feedback tuning

Iterative feedback tuning (IFT) is a flexible technique for tuning controllers of a desired structure [75,77,78]. It is a gradient based approach where the parameters of the controller are updated through an iterative procedure, such as to minimize the cost function which prescribes the desired control performance objectives. The key feature is that the closed-loop experimental data is used to compute the gradient of the cost function in the direction of the controller tuning parameters. The controller complexity is arbitrary, ranging from conventional PID (proportional, integral, derivative) controllers to nonlinear controller structures.

So far tested in versatile control problems, some of them are summarized in subsection 1.4.2.7, IFT has been shown capable to significantly improve control performance without any prior knowledge of the plant dynamics, even if these dynamics feature nonlinear effects. Simulation studies of robotics applications of IFT, presented in [68,69], confirm this. However, it appears that these simulation studies end up with conservative feedback designs, since they are based on a variant of the decentralized control approach in which decoupling between the robot axes is realized via static (proportional) control. With a static decoupling, one cannot afford as high control bandwidths of the robot joint servo systems as in the case of dynamic compensation of robot nonlinearities. Static decoupling only reduces the nonlinearities in a limited motion range, while dynamic compensation cancels nonlinearities in the complete joint space of the robot. Therefore, it seems interesting to investigate IFT in combination with model-based compensation for robot nonlinearities. Furthermore, since IFT performs optimization of the control performance based on the observed data, it is instructive to demonstrate the influence of behaviours contained in these data on the outcome of the control design.

The theory of IFT is presented in the next part to the extent which is sufficient to practically realize feedback control design for a robot of intrinsic nonlinear dynamics, such as a direct-drive robot. Practical demonstration of the design is presented on the RRR robot.

7.3.1 An iterative feedback tuning algorithm

Several IFT algorithms can be found in the literature. These are summarized in [75]. Variations between algorithms are induced by the choice of the cost function, as well as by the adopted structure of the controller (one or two degrees-of-freedom). Here, the standard algorithm considered in [77] for a two degrees-of-freedom controller structure, is made compatible with the robot joint servo set-up shown in Fig. 5.1 (section 5.2). This set-up consists of a one degree-of-freedom feedback controller and of the plant representing the decoupled joint dynamics obtained after application of the computed-torque control law (2.26).

The control design via IFT is performed in the *Z*-domain, assuming sampling time T_s. For convenience, the notation used in the IFT algorithm to be presented is not identical with that in Fig. 5.1, but it corresponds to the servo set-up shown in Fig. 7.7. Here, the argument z is the complex variable from the *Z*-domain (the shift operator), while the argument $k \in \{0,1,2,\ldots\}$ abbreviates sampled time kT_s.

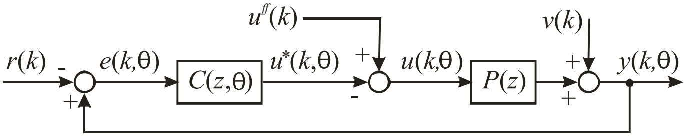


Figure 7.7: The servo set-up used for iterative feedback tuning.

In the given figure, $r(k)$ is the reference input, $u^{ff}(k)$ is the feedforward component of the control input, $v(k)$ is an unmeasurable disturbance which is assumed to be stochastic and uncorrelated with $r(k)$ and $u^{ff}(k)$, $P(z)$ represents a single-input/single-output linear-time invariant plant of unknown dynamics, and $C(z,\boldsymbol{\theta})$ is the feedback controller suitably parameterized by some parameter vector

$$\boldsymbol{\theta} = [\theta_0 \ \theta_1 \ \ldots \ \theta_n]^T . \tag{7.8}$$

The designer selects the controller's structure and complexity, which means that $C(z,\boldsymbol{\theta})$ is available in analytical form. The parameterization should be such that $C(z,\boldsymbol{\theta})$ is differentiable with respect to $\boldsymbol{\theta}$. Signals in the closed-loop, namely, the output $u^*(k,\boldsymbol{\theta})$ of the feedback controller, the input to the plant $u(k,\boldsymbol{\theta})$, the system output $y(k,\boldsymbol{\theta})$, and the error $e(k,\boldsymbol{\theta}) = y(k,\boldsymbol{\theta}) - r(k)$, are all dependent on the actual controller tuning, which is explicitly indicated by using the $\boldsymbol{\theta}$-argument.

Assume that the reference and the feedforward inputs are of finite length, i.e., $\{r(k)\}_{k=1,\ldots,N}$ and $\{u^{ff}(k)\}_{k=1,\ldots,N}$. Let $S(z,\boldsymbol{\theta})$ and $T(z,\boldsymbol{\theta})$ denote the achieved closed-loop sensitivity and complementary sensitivity functions, respectively:

$$S(z,\boldsymbol{\theta}) = \frac{1}{1+P(z)C(z,\boldsymbol{\theta})} ; \ T(z,\boldsymbol{\theta}) = 1 - S(z,\boldsymbol{\theta}) = \frac{P(z)C(z,\boldsymbol{\theta})}{1+P(z)C(z,\boldsymbol{\theta})} . \tag{7.9}$$

and let $y_r(k)$ be the desired output response to $r(k)$ for the closed-loop system, generated by a reference complementary sensitivity $T_0(z)$ selected by the designer:

$$y_r(k) = T_0(z) r(k) . \tag{7.10}$$

The difference between the achieved and the desired response is

$$\delta y(k,\boldsymbol{\theta}) = y(k,\boldsymbol{\theta}) - y_r(k)\,. \tag{7.11}$$

By virtue of Fig. 7.7, one may establish the following relationships:

$$y(k,\boldsymbol{\theta}) = T(z,\boldsymbol{\theta})r(k) + S(z,\boldsymbol{\theta})\big(v(k) + P(z)u^{ff}(k)\big), \tag{7.12a}$$

$$u(k,\boldsymbol{\theta}) = C(z,\boldsymbol{\theta})S(z,\boldsymbol{\theta})\big(r(k) - v(k)\big) + S(z,\boldsymbol{\theta})u^{ff}(k)\,. \tag{7.12b}$$

Consider the quadratic cost function of the form

$$J(\boldsymbol{\theta}) = \frac{1}{2N}\sum_{k=1}^{N}\Big\{\big[L_y(z)\delta y(k,\boldsymbol{\theta})\big]^2 + \big[L_u(z)u(k,\boldsymbol{\theta})\big]^2\Big\}, \tag{7.13}$$

where $L_y(z)$ and $L_u(z)$ are frequency dependent weightings that penalize the output difference δy and the control input u according to the designer needs. The objective of IFT is to determine the optimal controller tuning $\boldsymbol{\theta}^*$ which minimizes the cost (7.13):

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})\,. \tag{7.14}$$

A necessary condition for optimality is

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} &= \frac{1}{N}\sum_{k=1}^{N}\left[\left[\big(L_y(z)\big)^2 \delta y(k)\frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta}) + \big(L_u(z)\big)^2 u(k)\frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} \\ &= 0. \end{aligned} \tag{7.15}$$

To solve equation (7.15) for $\boldsymbol{\theta}^*$, one has the know how the signals δy and u are related with $\boldsymbol{\theta}$. These signals are generated from the closed-loop system containing the plant of unknown dynamics, which prevents establishing the required relations in analytical form.

The main contribution of the IFT methodology [75,77,78] is an unbiased estimation of the gradient $\partial J/\partial\boldsymbol{\theta}$ directly from data observed from the closed-loop system. According to (7.15), the required data are the signals δy and u, and their derivatives with respect to the parameter vector. The signals can be obtained by direct measurements, while the derivatives can be generated from the closed-loop system using the strategy which is explained next.

Analytical expressions for the derivatives are obtained using (7.11), (7.12a,b), and (7.9):

$$\begin{aligned} \frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta}) &\underset{(7.11)}{=} \frac{\partial y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta}) \\ &\underset{(7.12a)}{=} \frac{\partial T}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})r(k) + \frac{\partial S}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})\big(v(k) + P(z)u^{ff}(k)\big) \\ &\underset{(7.9)}{=} \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})\frac{T(z,\boldsymbol{\theta})}{C(z,\boldsymbol{\theta})}\big(r(k) - y(k,\boldsymbol{\theta})\big), \end{aligned} \tag{7.16a}$$

$$\begin{aligned} \frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta}) &\underset{(7.12b)}{=} \left(\frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})S(z,\boldsymbol{\theta}) + C(z,\boldsymbol{\theta})\frac{\partial S}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})\right)\big(r(k) - v(k)\big) \\ &\quad + \frac{\partial S}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})u^{ff}(k) \\ &\underset{(7.9)}{=} \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})S(z,\boldsymbol{\theta})\big(r(k) - y(k,\boldsymbol{\theta})\big). \end{aligned} \tag{7.16b}$$

Here, $\partial C/\partial\boldsymbol{\theta} \in \mathbb{R}^{(n+1)}$ represents the gradient of the controller which can be determined in analytical form using the known expression for $C(z,\boldsymbol{\theta})$:

$$\frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta}) = \left[\frac{\partial C}{\partial \theta_0}(z,\boldsymbol{\theta}),\ \frac{\partial C}{\partial \theta_1}(z,\boldsymbol{\theta}),\ldots,\ \frac{\partial C}{\partial \theta_n}(z,\boldsymbol{\theta})\right]^T. \tag{7.17}$$

By assumption, all derivatives in (7.17) exist. The expressions (7.16a) and (7.16b) indicate how to estimate the derivatives $\partial \delta y / \partial \boldsymbol{\theta}$ and $\partial u / \partial \boldsymbol{\theta}$ directly from data. Namely, given a stabilizing controller $C(z,\boldsymbol{\theta})$, one should perform two experiments on the closed-loop system shown in Fig. 7.7 according to the procedure presented in Table 7.1. In the first experiment, also known as the normal experiment, the desired r and u^{ff} are applied as the reference and feedforward control inputs, respectively. The control input to the plant u_{I} and the system output y_{I} are measured. In the second experiment, called gradient experiment, the difference $r - y_{\mathrm{I}}$ is applied as the reference input, while the feedforward component of the control input is kept on zero. The zero feedforward ensures unbiased estimation of the gradient of the cost function, which is explained below. In the gradient experiment, the error signal e_{II} and the system output y_{II} are measured.

The obtained measurements can be used in the analytical expressions (7.16a,b) to estimate the derivatives $\partial \delta y / \partial \boldsymbol{\theta}$ and $\partial u / \partial \boldsymbol{\theta}$:

$$est\left[\frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right] \underset{(7.16a)}{=} \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta}) \frac{1}{C(z,\boldsymbol{\theta})} y_{\mathrm{II}}(k,\boldsymbol{\theta}), \tag{7.18a}$$

$$est\left[\frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right] \underset{(7.16b)}{=} -\frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta}) e_{\mathrm{II}}(k,\boldsymbol{\theta}) \tag{7.18b}$$

Now, by virtue of (7.15), one may estimate the gradient of the cost function:

$$\begin{aligned} est\left[\frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta})\right] = \frac{1}{N}\sum_{k=1}^{N}\Bigg\{&\left[\left(L_y(z)\right)^2 \left(y_{\mathrm{I}}(k) - y_r(k)\right) est\left[\frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right] \\ &+ \left[\left(L_u(z)\right)^2 u_{\mathrm{I}}(k) est\left[\frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right]\Bigg\}. \end{aligned} \tag{7.19}$$

TABLE 7.1: EXPERIMENTAL PROCEDURE

Experiment	I (normal)
Reference input	$r_{\mathrm{I}}(k) = r(k)$
Feedforward input	$u_{\mathrm{I}}^{ff}(k) = u^{ff}(k)$
Measurements	$u_{\mathrm{I}}(k,\boldsymbol{\theta}) = C(z,\boldsymbol{\theta})S(z,\boldsymbol{\theta})\left(r(k) - v_{\mathrm{I}}(k)\right) + S(z,\boldsymbol{\theta})u^{ff}(k)$ $y_{\mathrm{I}}(k,\boldsymbol{\theta}) = T(z,\boldsymbol{\theta})r(k) + S(z,\boldsymbol{\theta})\left(v_{\mathrm{I}}(k) + P(z)u^{ff}(k)\right)$
Experiment	II (gradient)
Reference input	$r_{\mathrm{II}}(k) = r(k) - y_{\mathrm{I}}(k,\boldsymbol{\theta})$
Feedforward input	$u_{\mathrm{II}}^{ff}(k) = 0$
Measurements	$e_{\mathrm{II}}(k,\boldsymbol{\theta}) = S(z,\boldsymbol{\theta})\left(-r(k) + y_{\mathrm{I}}(k,\boldsymbol{\theta}) + v_{\mathrm{II}}(k)\right)$ $y_{\mathrm{II}}(k,\boldsymbol{\theta}) = T(z,\boldsymbol{\theta})\left(r(k) - y_{\mathrm{I}}(k,\boldsymbol{\theta})\right) + S(z,\boldsymbol{\theta})v_{\mathrm{II}}(k)$

The given gradient estimate is unbiased under the mild assumptions that the disturbance v_{I} in the normal experiment and the disturbance v_{II} in the gradient experiment are both zero mean and mutually uncorrelated. This can be seen from the expectation $E[.]$ of the estimate (7.19) with respect to the disturbances v_{I} and v_{II}, which is computed after the estimates of the derivatives (7.18a,b) are rewritten as:

$$\begin{aligned}
est\left[\frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right] &= \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})\frac{1}{C(z,\boldsymbol{\theta})}y_{\mathrm{II}}(k,\boldsymbol{\theta}) \\
&\underset{\text{Table 7.1}}{=} \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})\frac{1}{C(z,\boldsymbol{\theta})}\left[T(z,\boldsymbol{\theta})\left(r(k)-y_{\mathrm{I}}(k,\boldsymbol{\theta})\right)+S(z,\boldsymbol{\theta})v_{\mathrm{II}}(k)\right] \\
&\underset{(7.16\text{a})}{=} \frac{\partial \delta y_{\mathrm{I}}}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})+\omega_y(k,\boldsymbol{\theta}),
\end{aligned}\tag{7.20a}$$

$$\begin{aligned}
est\left[\frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right] &= -\frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})e_{\mathrm{II}}(k,\boldsymbol{\theta}) \\
&\underset{\text{Table 7.1}}{=} \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})S(z,\boldsymbol{\theta})\left(r(k)-y_{\mathrm{I}}(k,\boldsymbol{\theta})-v_{\mathrm{II}}(k)\right) \\
&\underset{(7.16\text{b})}{=} \frac{\partial u_{\mathrm{I}}}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})+\omega_u(k,\boldsymbol{\theta}),
\end{aligned}\tag{7.20b}$$

where

$$\omega_y(k,\boldsymbol{\theta}) = \frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})\frac{S(z,\boldsymbol{\theta})}{C(z,\boldsymbol{\theta})}v_{\mathrm{II}}(k);\tag{7.21a}$$

$$\omega_u(k,\boldsymbol{\theta}) = -\frac{\partial C}{\partial \boldsymbol{\theta}}(z,\boldsymbol{\theta})S(z,\boldsymbol{\theta})v_{\mathrm{II}}(k).\tag{7.21b}$$

The relations (7.20a,b) can be substituted into (7.19) to compute the expectation of the gradient:

$$\begin{aligned}
E\left[est\left[\frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta})\right]\right] &= \frac{1}{N}\sum_{k=1}^{N}\left\{E\left[\left(L_y(z)\right)^2\left(y_{\mathrm{I}}(k)-y_r(k)\right)\left(\frac{\partial \delta y_{\mathrm{I}}}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})+\omega_y(k,\boldsymbol{\theta})\right)\right]\right. \\
&\qquad \left.+E\left[\left(L_u(z)\right)^2u_{\mathrm{I}}(k)\left(\frac{\partial u_{\mathrm{I}}}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})+\omega_u(k,\boldsymbol{\theta})\right)\right]\right\} \\
&= \frac{1}{N}\sum_{k=1}^{N}\left\{E\left[\left(L_y(z)\right)^2\left(y_{\mathrm{I}}(k)-y_r(k)\right)\frac{\partial \delta y_{\mathrm{I}}}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right. \\
&\qquad \left.+E\left[\left(L_u(z)\right)^2u_{\mathrm{I}}(k)\frac{\partial u_{\mathrm{I}}}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right\} \\
&\qquad +\frac{1}{N}\sum_{k=1}^{N}\left\{E\left[\left(L_y(z)\right)^2\left(y_{\mathrm{I}}(k)-y_r(k)\right)\omega_y(k,\boldsymbol{\theta})\right]\right. \\
&\qquad\qquad \left.+E\left[\left(L_u(z)\right)^2u_{\mathrm{I}}(k)\omega_u(k,\boldsymbol{\theta})\right]\right\} \\
&= E\left[\frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta})\right] \\
&\qquad +\frac{1}{N}\sum_{k=1}^{N}\left\{E\left[\left(L_y(z)\right)^2\left(y_{\mathrm{I}}(k)-y_r(k)\right)\right]E\left[\omega_y(k,\boldsymbol{\theta})\right]\right. \\
&\qquad\qquad \left.+E\left[\left(L_u(z)\right)^2u_{\mathrm{I}}(k)\right]E\left[\omega_u(k,\boldsymbol{\theta})\right]\right\} \\
&= E\left[\frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta})\right]+0.
\end{aligned}\tag{7.22}$$

Notice that in the intermediate step which appears between the third and the fourth equality signs, $y_{\mathrm{I}}-y_r$ and u_{I} have been decoupled from ω_y and ω_u, respectively. This is allowed since $y_{\mathrm{I}}-y_r$ and u_{I} are affected only by v_{I}, while ω_y and ω_u are affected only by v_{II}. Since these disturbances are uncorrelated by assumption, decoupling is possible. In the last step of (7.22), the zero mean assumption of the disturbances has been used. The derived expectation of the gradient estimate is clearly unbiased, which is possible only if the feedforward component of the control input is at zero during the gradient experiment. If $u_{\mathrm{II}}^{ff} = u_{\mathrm{I}}^{ff} = u^{ff}$, then apart from y_{I} and u_{I}, the terms ω_y and ω_u would also contain u^{ff}, which would make impossible the decoupling applied above.

Given the unbiased estimate of $\partial J / \partial \boldsymbol{\theta}$, one can numerically solve (7.15) for the optimal controller tuning $\boldsymbol{\theta}^*$ by using an iterative Gauss-Newton scheme:

$$\boldsymbol{\theta}^{i+1} = \boldsymbol{\theta}^{i} - \gamma_i (\mathbf{R}^i)^{-1} est\left[\frac{\partial J}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}^i)\right], \tag{7.23}$$

where $i \in \{0,1,2,\ldots\}$ is the iteration number, γ_i is a positive real number which determines the step size, and $\mathbf{R}^i \in \mathbb{R}^{(n+1)\times(n+1)}$ is a positive definite matrix typically chosen equal to an estimate of the Hessian of J [77]:

$$\mathbf{R}^i = \frac{1}{N}\sum_{k=1}^{N}\left\{ est\left[\frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\left(est\left[\frac{\partial \delta y}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right)^T + est\left[\frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\left(est\left[\frac{\partial u}{\partial \boldsymbol{\theta}}(k,\boldsymbol{\theta})\right]\right)^T \right\}\Bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}^i}. \tag{7.24}$$

The parameter update in (7.23) is performed in a descendent direction of J. The parameter vector $\boldsymbol{\theta}^i$ is used during the i^{th} repetition of the experimental procedure described in Table 7.1. This procedure is repeated until the estimate of the gradient (7.19) approaches zero to a sufficient extent. The latest parameter update obtained from (7.23) is considered as the optimal controller tuning $\boldsymbol{\theta}^*$.

In reference [77], a formal proof can be found about the convergence of the standard IFT algorithm, which served as the basis for the algorithm developed above. The most important requirement for convergence is that the closed-loop signals should stay bounded throughout iterations, in other words, the updated controllers have to remain stabilizing for the plant. Unfortunately, there is no guarantee that the update law always provides a stable controller. Intuitively, one would expect that the actual value of the cost function should be increasing when approaching the regions of unstable controller tunings in the design space of the controller parameters. This way, the IFT algorithm would automatically avoid unstable tunings, since this algorithm searches for the minimum of the cost function. This is, though, not always the case in practice, and the controller might become unstable during the iterations before any warning is noticed. As a remedy, reference [43] suggests ascertaining the stability using an identified plant model. If this strategy is not used, the maximum one can afford is to examine Bode plot of the controller after each parameter update. If there is a significant difference with respect to the controller used in the previous iteration, then the step size in (7.23) should be reduced for precaution. By virtue of (7.18a), an estimate of $\partial \delta y / \partial \boldsymbol{\theta}$ is computed after filtering the signal through the inverse of the controller. To have this filtering operation possible, the controller must be of minimum phase. If the parameter update (7.23) delivers the controller which is of non-minimum phase, then, like in the iterative learning control strategy discussed in subsection 7.2.1, the Zero phase error tracking algorithm for digital control [172] can be used to ensure stable filtering. Literature still does not offer a rigorous analysis how persistency of excitation influences the convergence of an IFT algorithm. This remains an open issue which requires further research.

The derived IFT algorithm is practically examined on the RRR robot. The results are presented next.

7.3.2 Control design for the RRR robot via iterative feedback tuning

The RRR robot feedback control design via IFT was the subject of a comprehensive simulation and experimental study presented in [136]. The most striking problem experienced is the non-convexity of the overall optimization problem. Because of non-convexity, the quality of the outcome of IFT strongly depends on the initial controller tuning (initial guess at controller parameters), on the choice of the reference robot trajectory, as well as on the selection of the reference model in (7.10) and of the weighting filters applied in the cost function. If no knowledge about the plant dynamics is taken into account, and hence a weak initial tuning is used just to ensure stability in closed-loop, then it may turn out that the result of IFT will end up at some local optimum of the cost function, which might be even very close to the initial controller tuning. A truly "blind" control design via IFT has fewer chances to reach motion control performance feasible with the model-based loop-shaping designs discussed in Chapter 5. To achieve at least comparable performance, IFT must be carried out by adopting the controller structure and its initial tuning, together with the robot trajectory, the reference model, and the weighting functions, that are already proven effective, e.g., from some model-based design. An other option is to experiment with different controller structures, tunings, trajectories, models, and weighting. The latter option can be very inefficient, as it may require a lot of trials.

The objective of this section is not to realize control performance of high quality, but to confirm that the IFT algorithm derived in the previous section can be used to design feedback controllers of a direct-drive robot. Especially, the goal is to combine dynamic compensation for the robot nonlinearities with IFT, which is a strategy hardly seen elsewhere in the literature. Another goal is to illuminate influences of different behaviours contained in the data on the outcome of IFT. To facilitate the presentation, a low-complexity controller structure, a common robot trajectory, and simple reference model and weightings are used.

The IFT feedback control design is carried-out for the servo-system of the first robot joint. The IFT is applied on the linear dynamics P_1 for this joint which is not covered with the robot rigid-body dynamic model with friction implemented in the control law (2.26). The joint servo-system is depicted in Fig. 5.1 (section 5.2). The control input u_1 to the plant P_1 consists of the feedback control action u_1^* and of the feedforward component representing the reference acceleration $\ddot{q}_{r,1}$:

$$u_1 = \ddot{q}_{r,1} - u_1^* . \tag{7.25}$$

The control action u_1^* is the output of the feedback controller C_1. When compared with the servo setup shown in Fig. 7.7, the following correspondence can be established: $r = q_{r,1}$ (reference joint motion), $u^{ff} = \ddot{q}_{r,1}$, $u^* = u_1^*$, $u = u_1$, and $y = q_1$. The objective of IFT is to design the feedback controller C_1. This design is performed in the Z-domain.

In general, the controller can be represented by any function which is differentiable with respect to all tuning parameters. Here, a simple linear discrete-time transfer function is considered:

$$C_1(z) = \frac{\theta_0 + \theta_1 z^{-1} + \ldots + \theta_n z^{-n}}{1 - z^{-1}} . \tag{7.26}$$

This function is linear in the parameters $\theta_0, \ldots, \theta_n$ and it includes an integral action. For instance, if the order $n = 2$ is selected, then a conventional PID controller is obtained. By increasing the order, more versatile effects can be achieved (e.g., notch, roll-off, etc).

Initially, the control design for the first robot joint was carried out for $n = 4$, which enables a high-frequency roll-off effect in addition to a PID control action. This design was performed for the reference joint trajectory $q_{r,1}(t)$ shown in Fig. 7.8. The analytical equations for $q_{r,1}(t)$ enable direct specification of the velocity, acceleration, and jerk levels, which facilitate direct control of the trajectory shape and implicit control of the trajectory frequency content. The specifications corresponding to $q_{r,1}(t)$ shown in Fig. 7.8 turn out to be adequate to excite the robot dynamics persistently during the limited acceleration/deceleration periods. The initial controller tuning $\boldsymbol{\theta}^0$ was experimentally determined by trial and error, just to ensure stable motion in joint 1 along $q_{r,1}(t)$. The Bode plot of

$C_1(z,\boldsymbol{\theta}^0)$ is shown in Fig. 7.9. While joint 1 was moving, the other two were kept locked by PD feedback controllers of sufficient bandwidths. For the sake of simplicity, no special reference model was prescribed, i.e., $T_o = 1$ was chosen in (7.10). An identical choice was made for the weighting which penalizes the output error in the cost function (7.13), i.e. $L_y = 1$ was used. Finally, a constant control input weighting L_u was used to limit the controller effort and, equivalently, to constrain the gain of C_1 and the bandwidth of the joint 1 servo system.

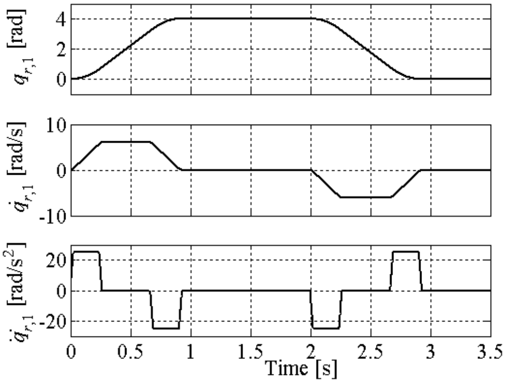


Figure 7.8: The reference joint trajectory.

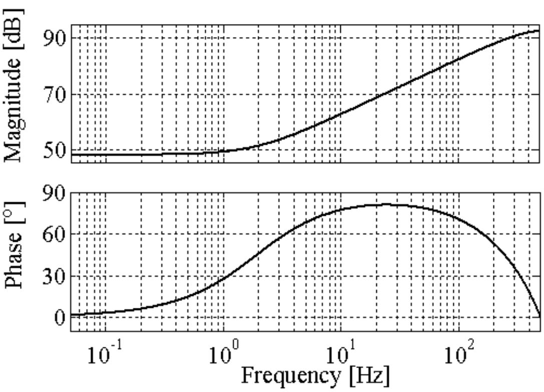


Figure 7.9: The feedback controller for joint 1 corresponding to the initial parameter tuning.

To get some experience with IFT, the feedback design was first carried out using the plant parametric model $P_1(z)$. In section 4.6, Fig. 4.23 presents the nominal data-driven FRF of joint 1 together with the Bode plot of the corresponding parametric model of the 7th order. Here, Fig. 7.10 presents the nominal FRF together with the Bode plot of the 15th order model. By comparing the plots shown in Fig. 4.23 and Fig. 7.10, it is clear that the higher order model better fits FRF data at higher frequency range. Thus, it seems justified to use the 15th order model during the IFT feedback control design. The higher order model was avoided in the control designs considered in Chapter 5, to constrain the complexity of the resulting model-based controllers from the very beginning of the designs. On the other hand, in the data-based design considered here, the controller complexity is prescribed by the designer, while the optimal controller tuning should be determined using the observed data. If these data are richer with information about the system dynamics, which should be the case if a more involved plant model is taken into account, then a more effective control design should be possible. Thus, one may have opposite preferences about the plant model complexity during model-based and data-based control designs.

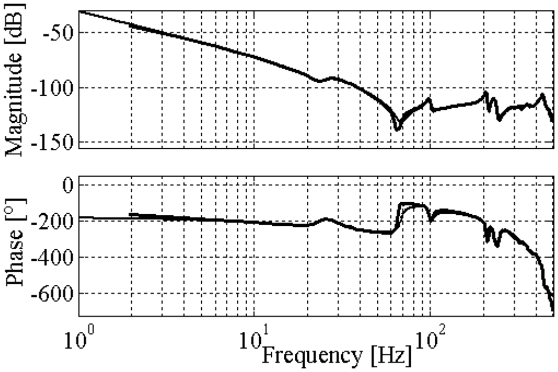


Figure 7.10: Frequency response function of the nominal plant dynamics (thick) and the Bode plot of the parametric model of the 15th order (thin).

When simulating the feedback control system depicted in Fig. 7.7 for the references ($r = q_{r,1}$, $u^{ff} = \ddot{q}_{r,1}$) shown in Fig. 7.8, the controller shown in Fig. 7.9, and the plant model shown in Fig. 7.10, the obtained amplitude of the control input u_1 was approximately 10^3 times higher than the output error $\delta y_1 = y_1 - y_{r,1} = q_1 - q_{r,1}$. This error is penalized in the cost function (7.13). Since $L_y = 1$ was adopted, a fair treatment of both terms present in the cost function was achieved by appropriate scaling of the control input via L_u. Considering the given factor of difference between u_1 and δy_1 (~10^3), constant values less than 10^{-6} seemed appropriate for L_u. An "optimal" L_u was determined by examining the sensitivity functions that correspond to the plant model and to the results of the IFT designs performed for several values of L_u, ranging from 10^{-8} to 10^{-6}. All designs were carried out in accordance with the procedure explained in the previous subsection. For the optimal controller tuning $\boldsymbol{\theta}^*(L_u)$ obtained after each design, the peak value of the magnitude of the sensitivity function was evaluated. These peak values are plotted in Fig. 7.11 against L_u. Bode plots corresponding to the optimal controllers ($C(z,\boldsymbol{\theta}^*(L_u))$), open-loop gains ($P(z)C(z,\boldsymbol{\theta}^*(L_u))$), and sensitivities, obtained for $L_u \in \{10^{-8}, 4.3\cdot 10^{-7}, 10^{-6}\}$, are plotted in Figs. 7.12, 7.13, and 7.14, respectively. The related output errors are presented in Fig. 7.15, and they were obtained for the trajectory shown in Fig. 7.8. By inspection of these figures, it can be concluded that L_u should not be chosen too small, since, e.g., $L_u = 10^{-8}$ causes unacceptable peaks in the sensitivity function. The maximum peak that can be tolerated is 6 [dB] (design rule r.3 from section 5.2). If L_u is small, the gains of the controller and of the open-loop become very high, which reduces stability margins. On the other hand, if L_u is chosen too high, the cross-over frequency decreases, which also degrades the control performance as confirmed by higher output errors shown in Fig. 7.15. From Figs. 7.11-15, it can be concluded that $L_u = 4.3\cdot 10^{-7}$ gives an "optimal" trade-off between the stability robustness and the performance.

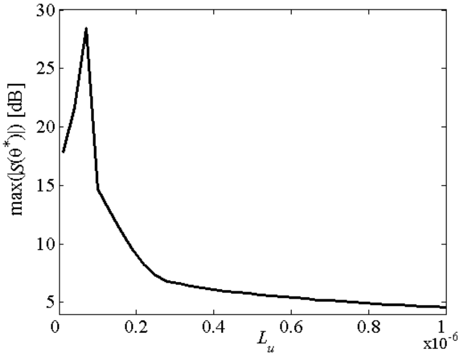


Figure 7.11: Maximal magnitudes of the sensitivity functions that correspond to different penalties on the input signal.

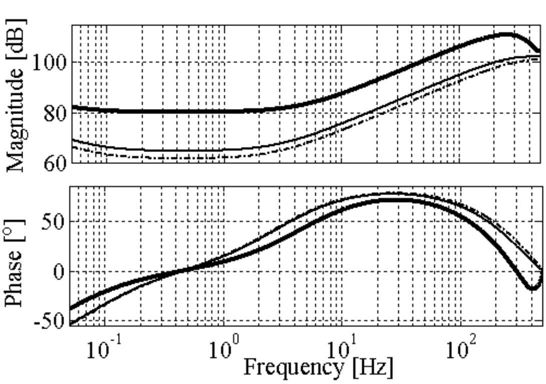


Figure 7.12: Controllers obtained from IFT designs performed for $L_u = 10^{-8}$ (solid, thick), $L_u = 4.3 \cdot 10^{-7}$ (solid, thin), and $L_u = 10^{-6}$ (dash-dotted).

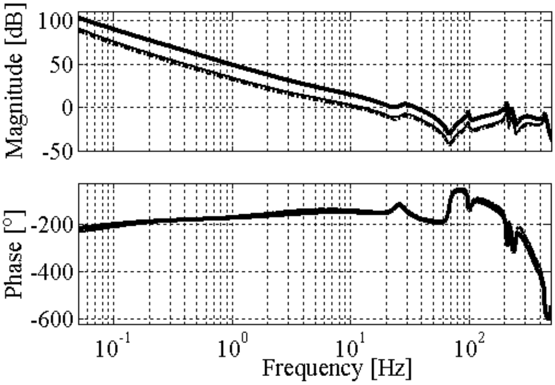


Figure 7.13: Open-loop gains computed for controllers obtained from IFT designs performed for $L_u = 10^{-8}$ (solid, thick), $L_u = 4.3 \cdot 10^{-7}$ (solid, thin), and $L_u = 10^{-6}$ (dash-dotted).

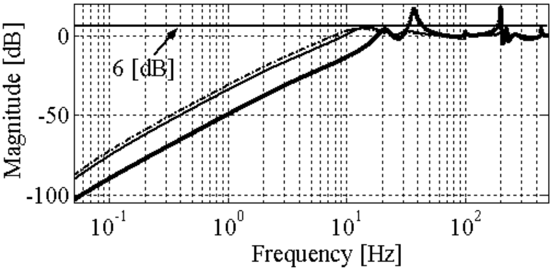


Figure 7.14: Sensitivity functions computed for controllers obtained from IFT designs performed for $L_u = 10^{-8}$ (solid, thick), $L_u = 4.3 \cdot 10^{-7}$ (solid, thin), and $L_u = 10^{-6}$ (dash-dotted).

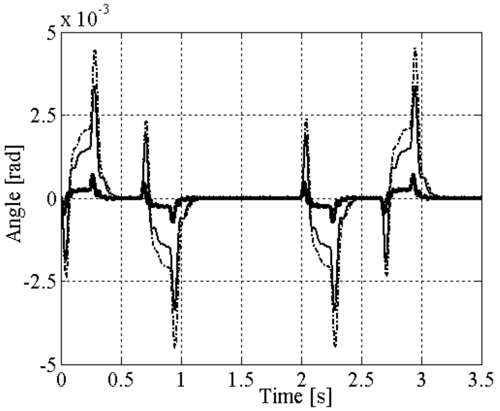


Figure 7.15: Output errors simulated with controllers obtained from IFT designs performed for $L_u = 10^{-8}$ (solid, thick), $L_u = 4.3 \cdot 10^{-7}$ (solid, thin), and $L_u = 10^{-6}$ (dash-dotted).

It should be admitted, though, that the analysis performed above is not possible without some knowledge of the plant dynamics. Without this knowledge, a well performing and a robust controller can not be easily designed via IFT. However, a possible advantage of IFT against a model-based control design is that IFT optimizes characteristics of the resulting controller while utilizing mere system data. These data can contain effects of importance for control performance that are not incorporated into the plant model. Consequently, an outcome of IFT control design can be more effective in dealing with unmodelled effects than a model-based controller.

The property of IFT to optimize its outcome against versatile behaviours encountered in the observed data is confirmed by the differences between the optimal controllers shown in Fig. 7.16. This figure presents the optimal controller designed via IFT for $L_u = 4.3 \cdot 10^{-7}$ (also shown in Fig. 7.12), together with the result of the same IFT algorithm with identical setting which utilized data experimentally measured on the RRR robot. For comparison, the controller with the initial tuning, already depicted in Fig. 7.9, is also shown. By inspection of the given plots, it is obvious that both optimal controllers provide better performance than the initial one (higher gains, integral action for larger frequency ranges), but it is also clear that these optimal controllers are not identical. The controller designed based on the experimental data features integral effect along a wider range at low-frequencies, while at higher frequencies, this controller imposes lower gain and, in turn, more profound roll-off. The better roll-off with the controller based on the experimental data is apparent from the Bode plots of the open-loop gains shown in Fig. 7.17, while the smaller sensitivity plotted in

Fig. 7.18 is the result of the stronger integral action achieved with this controller.

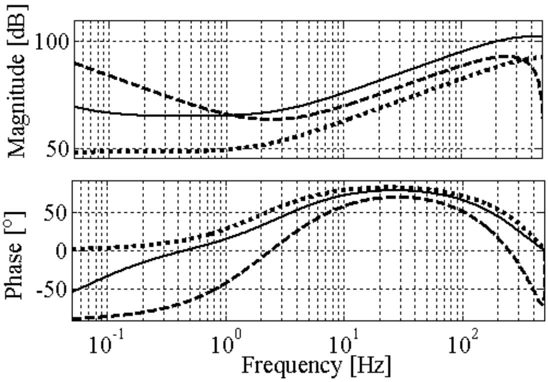


Figure 7.16: The controller with initial tuning (dotted), and the optimal controllers based on synthetic (solid) and experimental (dashed) data.

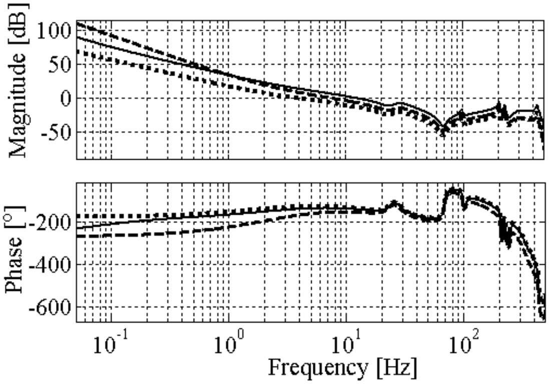


Figure 7.17: Open loop-gains corresponding to the initial controller tuning (dotted) and to the optimal controllers based on synthetic (solid) and experimental (dashed) data.

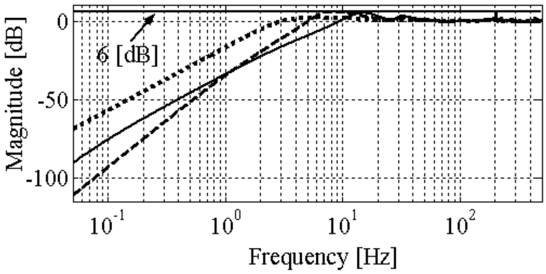


Figure 7.18: Sensitivity functions corresponding to the initial controller tuning (dotted) and to the optimal controllers based on synthetic (solid) and experimental (dashed) data.

The differences between two optimal controllers can be indeed explained by the presence of effects not covered by the plant model, e.g., residual friction and dynamic couplings, measurement noise. Namely, the experiments on the RRR robot were performed by applying the model-based compensation of robot nonlinearities, implemented in the control law (2.26). The nonlinearity was strongly reduced but was not completely eliminated. Consequently, some residual nonlinear effects were still present in the measured data, while they were not contained in the plant model. The presence of uncompensated Coulomb friction is supported by Fig. 7.19, showing the output errors obtained in simulations and in experiments. The plots given on the left-hand side present the errors achieved with the initial controller tuning. The error obtained in the experiment features a large offset due to residual friction. There was no friction in the simulation, so no offset is present in the corresponding error. The strong integral effect achieved with the optimal controller tuning eliminates the offset from the experimental error, as apparent from the plots shown on the right-hand side in Fig. 7.19. Emphasized roll-off property of the controller based on the experimental data was probably induced by the measurement noise inherent in the experiments, which mostly has high-frequency content.

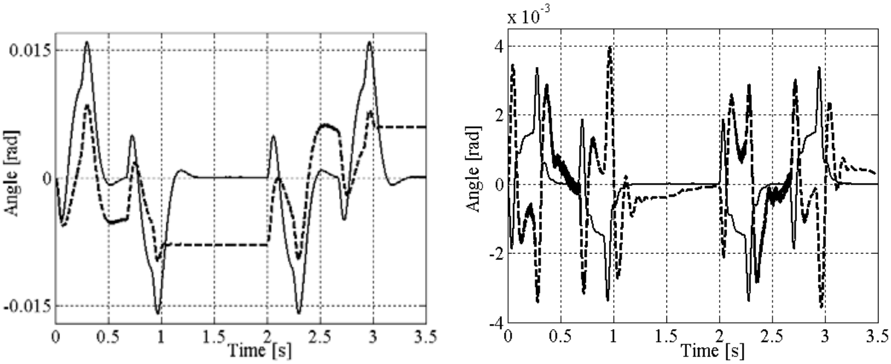


Figure 7.19: Output errors obtained in simulations (solid) and in experiments (dashed): the initial controller tuning (left), the optimal controller tuning (right).

Differences between the contents of the synthetic and experimental data result in discrepancies between the achieved optima of the cost function. The values of the cost function obtained throughout the iterations are plotted in Fig. 7.20, confirming that the cost is optimized with both synthetic and experimental data. However, the obtained minima are clearly different. As discussed at the beginning of this subsection, there is no guarantee that the global minima have been achieved. Moreover, while a constant minimum was achieved with the synthetic data, use of the experimental data brings the optimal values to a narrow band, within which slight fluctuations are still present. These fluctuations are caused by the numerical nature of the optimization process performed via IFT, which is sensitive to any difference in the information contained in the data. Data obtained in different experiments were never identical, at least because of the presence of unreproducible effects addressed in subsection 7.2.3 (varying friction and varying sampling time). Still, fluctuations of the cost functions are negligible considering the performance improvement with respect to the initial controller tuning.

The final evaluation of the IFT algorithm should confirm its capability to handle controllers of higher complexity. Thus, the order of the controller (7.26) was increased from $n=3$ to $n=7$. The optimal tuning achieved with $n=3$ while utilizing the experimentally obtained data was used to initialize the corresponding parameters of the controller of the 7th order. The initial values of the remaining parameters of the 7th order controller were all set to zero. The IFT for this controller utilized the experimentally obtained data, and the plot of the cost function against iteration number is shown in Fig. 7.21. As obvious from this plot, the application of the IFT algorithm on the higher complexity controller provides a better result. Notice that the initial value of the cost function corresponds with the optimal one obtained with the controller of the 3rd order. Clearly, the expanded controller complexity adds more freedom for performance improvement. The Bode plots corresponding to the optimal controllers of the 3rd and 7th orders are shown in Fig. 7.22, and they reveal that the 7th order controller outperforms the lower order one. Namely, the higher order controller has larger gain up to 100 [Hz], and stronger roll-off at higher frequencies. The stronger roll-off effect is confirmed from the Bode plots of the open-loop gains shown in Fig. 7.23. The higher gain of the 7th order controller ensures better error reduction at lower frequencies, which is confirmed by the lower magnitude of the related sensitivity plot shown in Fig. 7.24, as well as by lower peaks of the corresponding output error depicted in Fig. 7.25 (experimentally achieved).

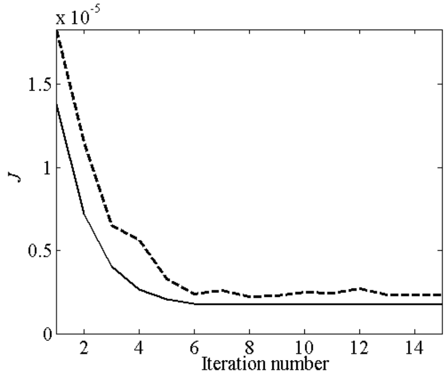


Figure 7.20: Achieved values of the cost functions thorough iterations for the controllers that are based on synthetic (solid) and experimental (dashed) data.

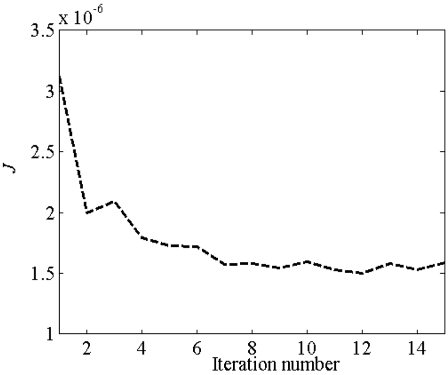


Figure 7.21: Achieved values of the cost functions thorough iterations for the higher complexity controller which is based on experimental data.

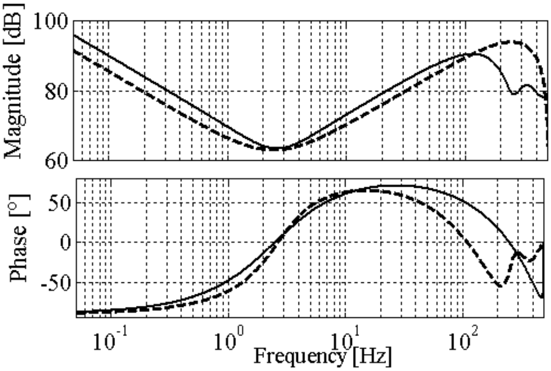


Figure 7.22: Optimally tuned controllers of the 3rd order (dashed) and of the 7th order (solid).

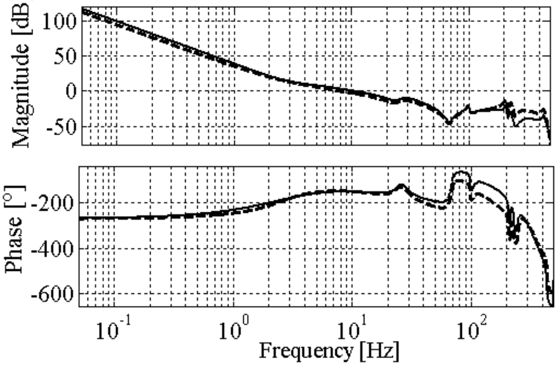


Figure 7.23: Open-loop gains computed with the optimally tuned controllers of the 3rd order (dashed) and of the 7th order (solid).

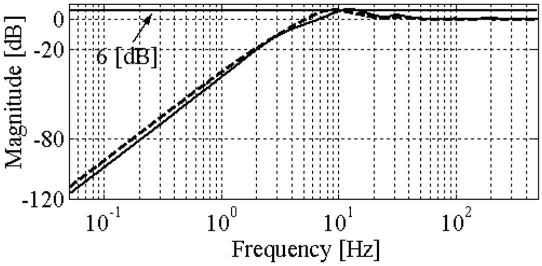


Figure 7.24: Sensitivity functions computed with the optimally tuned controllers of the 3rd order (dashed) and of the 7th order (solid).

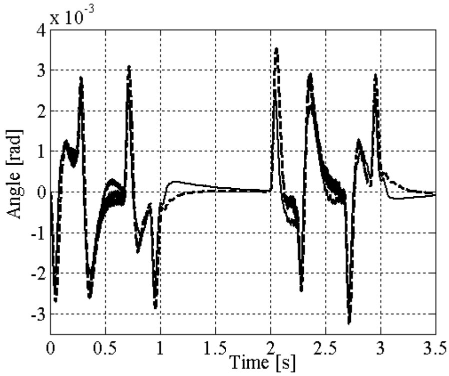


Figure 7.25: Experimentally achieved output errors with the optimally tuned controllers of the 3rd order (dashed) and of the 7th order (solid).

The original intention of the IFT methodology was to provide control performance improvement in situations where no modelling representation of the plant dynamics is available. This especially applies to process industries (e.g. oil refineries) where high complexity of underlying processes impedes reliable modelling. The IFT feedback control design considered in this section is shown capable to optimize control performance with respect to the information about the plant dynamics contained in the measurements. This indicates that IFT is not only a remedy for plants of totally unknown dynamics, but it is also an appealing tool for fine tuning of a controller which can even be obtained by some model-based design. The fine tuning means experimental optimization of the controller performance against behaviours that are not covered by the underlying model but are encountered in the observed data.

Because of the possibility to manipulate the reference model and the weighting functions, the IFT can also be considered as a special, data-based loop-shaping control design method. Another data-based method with the utility of loop-shaping is presented next.

7.4 Data-based loop-shaping control design

Conventional and advanced model-based loop-shaping feedback control designs are thoroughly treated in sections 5.3, 5.4, and 5.5. In these designs, it is instrumental to prescribe the control performance objectives in terms of the desired shapes of the closed-loop transfer functions: sensitivity, input sensitivity, and/or complementary sensitivity. The objective is to design a feedback controller (of the robot joint servo system) which realizes the prescribed transfer functions. Experiences with advanced model-based feedback designs, reported in sections 5.4 and 5.5, confirm that closely accounting for experimental properties of the robot to be controlled can significantly improve the control performance and the robustness against disturbances and parasitic dynamics. However, the common problem is the complexity of controllers that result from advanced model-based designs. To make these controllers admissible for online implementation, their complexity must be lowered via model reduction.

This section investigates if the control performance feasible with model-based motion controllers can also be realized with controllers designed using some data-based (DB) method. The requirement is that the method must allow prescribing the controller structure and the complexity at the very beginning of the feedback design. A starting point for such a design is found in the virtual reference feedback tuning (VRFT) approach [33,34,65,66,107,108]. Relying on VRFT methodology, a new DB controller design is suggested, which enables simultaneous shaping of the closed-loop sensitivity and the complementary sensitivity transfer functions. The derivation of the DB method will be explained in full detail. Its practical merits will be illustrated with experimental results obtained on the RRR robot.

7.4.1 Problem formulation

Consider the servo-system shown in Fig. 7.26, which consists of a LTI (linear time-invariant) plant P and controller C. Here, a SISO (single-input, single-output) one degree-of-freedom structure is assumed, although this is not essential for the feedback design method that will be developed. The reference input is r, the control input to the plant is u, the feedforward component of the control input is u^{ff}, the plant output is y, and the error e is defined by:

$$e = y - r \,. \tag{7.27}$$

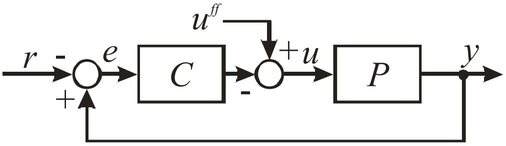


Figure 7.26: The servo set-up.

The desired control performance is specified via the desired closed-loop sensitivity and complementary sensitivity functions S_o and T_o, respectively. As discussed in section 5.4, by means of S_o, a designer specifies the desired dynamics of the closed-loop system at low-frequencies, e.g., a minimum bandwidth requirement, integral control, and a level of error reduction. T_o facilitates specification of the maximum closed-loop bandwidth. It is also important for the robustness against disturbances and plant uncertainties at the higher frequencies, e.g., resonances and noise.

The objective is to design a stabilizing controller C which closely realizes S_o and T_o. The ideal controller C_o satisfies:

$$S_o = \frac{1}{1+PC_o};\ T_o = \frac{PC_o}{1+PC_o}. \tag{7.28}$$

Strictly speaking, there is no C_o satisfying both conditions of (7.28), unless the prescribed S_o and T_o are complementary. If the complementary constraint $S_o + T_o = 1$ is fulfilled, then by specifying either S_o or T_o, the other transfer function is immediately found. The specified function must capture the desired closed-loop dynamics for both low and high frequency ranges. Although possible, specification of a single desired transfer function for the complete frequency scale usually is not a convenient task. Instead, a designer rather specifies both S_o and T_o, that are not necessarily complementary, but are rather idealistic requirements for the closed-loop dynamics at low and high frequencies, respectively. A controller C appropriately fulfilling the given control objective is the one which shapes the actual closed-loop sensitivity S $(=1/(1+PC))$ and the complementary sensitivity T $(=PCS)$ transfer functions such that $S \approx S_o$ and $T \approx T_o$, where the measure of distance from the ideal shapes is up to the designer.

The problem posed in the previous paragraph is well known in control theory. The existence of a stable controller that fulfils the control objective is addressed in [108] and is discussed in more details in [14]. To find such a controller, one can use a number of standard techniques. Among them, those based on optimization in an H_2 and an H_∞ sense are typically applied [49,103,185]. These techniques are model-based, as they determine the appropriate C using some parametric representation of the plant. Although very effective for computing a C close to realizing the desired S_o and T_o, the model-based techniques usually feature a common problem: the generalized plant model uniquely defines the structure and complexity of the resulting controller. If this model is of high order, then the order of the resulting C is also high. As the controller complexity is an important factor for implementation, a designer often has to take special care to come up with a C which fulfils the given control objective and is also suitable for application. This is realized either by using generalized plants of restricted complexity, or by applying some model reduction technique on the originally designed C.

The next section presents a data-based method that realizes the given control objective, but, contrary to the standard model-based techniques, also gives a designer the freedom to prescribe the controller structure and complexity at the very beginning of the control design. This is a direct consequence of using time-domain data to represent the plant dynamics, instead of resorting to some parametric model. Typical signals required for the data based control design are the input to the plant u and the corresponding plant output y. However, they are not the only choices. For instance, if the plant nominally operates in closed-loop (because of stability or other reasons), then any set of input and output signals that can recover information about the plant dynamics can be used. For the sake of clarity, u and y will be used in the following presentation. Assume that these signals are observed at N discrete time instants T_s, $2T_s$, …, NT_s, where T_s is the sampling time. Consequently, two sets of data points are available $\{u(k)\}_{k=1,2,\ldots,N}$ and $\{y(k)\}_{k=1,2,\ldots,N}$, where k abbreviates kT_s. If observed from the real plant, it is irrelevant if $\{u(k)\}_{k=1,2,\ldots,N}$ and $\{y(k)\}_{k=1,2,\ldots,N}$ are obtained under open-loop or closed-loop operating conditions. The necessary requirement is that $\{u(k)\}_{k=1,2,\ldots,N}$ sufficiently excites the plant dynamics, so $\{y(k)\}_{k=1,2,\ldots,N}$ is rich around frequencies of interest for control, such as the crossover frequency [34]. When experimentally obtained, these signals can be corrupted with disturbances. This is addressed in subsection 7.4.4.

7.4.2 Data-based controller design

The data-based (DB) controller design is formulated in the *Z*-domain. The two constraints on C_o given in (7.9) are replaced with a single one:

$$T_o(z) = C_o(z)S_o(z)P(z)\,. \tag{7.29}$$

where z is the complex variable from the *Z*-domain (the shift operator). The quest for C which is close enough to C_o –once again, a metric is up to the designer– is highly facilitated if the search space is restricted to the class of linearly parameterized controllers

$$C(z,\boldsymbol{\theta}) = C_p(z)\boldsymbol{\beta}^T(z)\boldsymbol{\theta}\,, \tag{7.30}$$

where $C_p(z)$ is the part explicitly assigned by the designer, and $\boldsymbol{\beta}^T(z)\boldsymbol{\theta}$ is the part to be tuned based on the observed data and on the performance specifications. The adopted controller class is used in a number of DB control designs, e.g. [33,34,65,107,108]. It is attractive since it facilitates explicit prescription of the controller structure and complexity, and, still allows tuning the free parameters contained in $\boldsymbol{\theta}$.

The part of the controller assigned by the designer can reflect some a priory experience of the plant dynamics, e.g., observation of resonance frequencies and deterministic disturbances in the measured signals. If such effects are found, then the appropriate remedy can be directly integrated in the controller. For instance, notch filters can be applied to accomodate the resonances. Furthermore, $C_p(z)$ enables the designer to explicitly impose the desired controller effects, e.g. integral action, low-pass filtering characteristic, etc.

The tuning part of the controller consists of a vector of appropriately chosen basis functions $\boldsymbol{\beta}$, and of a vector of tuning parameters $\boldsymbol{\theta}$:

$$\begin{aligned} \boldsymbol{\beta}(z) &= [\beta_0(z)\ \beta_1(z)\ \ldots\ \beta_n(z)]^T, \\ \boldsymbol{\theta} &= [\theta_0\ \theta_1\ \ldots\ \theta_n]^T. \end{aligned} \tag{7.31}$$

Unfortunately, there is no formalism yet developed for selecting the basis functions that will compose a controller whose tuning can be guaranteed to achieve the control objectives. FIR (finite impulse response) filters are typically selected for the basis functions. This selection is strongly motivated by system identification theory [113,140,141].

Within the selected class (7.30),(7.31), one is interested in the tuning (set of $\boldsymbol{\theta}$) that suitably realizes the given control objective. By virtue of (7.29), an adequate controller tuning is obtained by minimizing the model-based (MB) cost function:

$$J_{\mathrm{MB}}(\boldsymbol{\theta}) = \left\| \left(T_o(z) - C(z,\boldsymbol{\theta})S_o(z)P(z)\right)W(z) \right\|_2^2, \tag{7.32}$$

where $\|.\|_2$ denotes the standard H$_2$-norm, and W is a stable filter. As discussed in [34], this filter is chosen by the designer so as to emphasize the frequency regions where S_o and T_o are low in magnitude: low frequencies for S_o and high frequencies for T_o. In these regions, large relative errors between $C(z,\boldsymbol{\theta})$ and $C_o(z)$ may have little impact on the integral H$_2$-norm cost (7.32) because of small S_o and T_o. Therefore, W is used as weighting to compensate for low magnitudes of S_o and T_o, which balances importance of all frequency regions in the cost (7.32).

The cost (7.32) is not a unique measure of distance from the ideal controller. For instance, one might like to tune $C(z,\boldsymbol{\theta})$ such to minimize the maximal difference from C_o. In this case, an adequate measure of distance should be based on the H$_\infty$-norm instead of H$_2$-norm. On the other hand, one could form the MB cost by composing two separate MB functions, as it is the case if $C(z,\boldsymbol{\theta})$ is tuned via the virtual reference feedback tuning (VRFT) technique [108]:

$$\begin{aligned} J_{\mathrm{MB\_VRFT}}(\boldsymbol{\theta}) &= \left\| \left(S_{\mathrm{o}}(z) - \frac{1}{1+P(z)C(z,\boldsymbol{\theta})} \right) W_T(z) \right\|_2^2 \\ &+ \left\| \left(T_{\mathrm{o}}(z) - \frac{P(z)C(z,\boldsymbol{\theta})}{1+P(z)C(z,\boldsymbol{\theta})} \right) W_S(z) \right\|_2^2 . \end{aligned} \tag{7.33}$$

Although a viable alternative, the use of the cost function (7.33) leads to an increased amount of filtering operations when processing the observed data, as will be pointed out in subsection 7.4.3.

The MB cost function (7.32) is minimized for all $\boldsymbol{\theta}$ when

$$T_{\mathrm{o}}(z) = C(z,\boldsymbol{\theta}) S_{\mathrm{o}}(z) P(z) . \tag{7.34}$$

This condition can be considered as a definition of the filter, which can be used for processing the observed control input $\{u(k)\}_{k=1,2,\ldots,N}$:

$$T_{\mathrm{o}}(z) u(t) = C(z,\boldsymbol{\theta}) S_{\mathrm{o}}(z) P(z) u(t) . \tag{7.35}$$

Using the relation $y = Pu$ (see Fig. 7.7), equation (7.35) can be rewritten as

$$T_{\mathrm{o}}(z) u(t) = C(z,\boldsymbol{\theta}) S_{\mathrm{o}}(z) y(t) . \tag{7.36}$$

The last expression induces a DB cost function, which facilitates tuning of the free parameters $\boldsymbol{\theta}$:

$$J_{\mathrm{DB}}^{N}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \left[L(z) \left(T_{\mathrm{o}}(z) u(k) - C(z,\boldsymbol{\theta}) S_{\mathrm{o}}(z) y(k) \right) \right]^2 . \tag{7.37}$$

Here, L is a stable filter whose purpose is to ensure the equivalence between the MB and DB costs (7.32) and (7.37), respectively. The condition of equivalence can be derived using the same strategy as in the VRFT method [33,34,107,108], and it will be done at the end of this subsection. Notice that only observed signals are used in the cost (7.37) and no plant model is required.

Since the considered controller class (7.30),(7.31) is linear in the free parameters $\boldsymbol{\theta}$, the DB cost is quadratic in them. Thus, under assumption that $\boldsymbol{\theta}$ is not constrained, the global $\hat{\boldsymbol{\theta}}^{\mathrm{DB}}$ minimizer of the DB cost is found using the least-squares method [113,140]:

$$\hat{\boldsymbol{\theta}}_N^{\mathrm{DB}} = (\mathbf{A}_N)^{-1} \mathbf{f}_N ; \; \mathbf{A}_N = \sum_{k=1}^{N} \varphi(k) \varphi^T(k) ; \; \mathbf{f}_N = \sum_{k=1}^{N} \varphi(k) u_T(k) , \tag{7.38a}$$

where

$$\varphi(k) = \boldsymbol{\beta}(z) C_{\mathrm{p}}(z) L(z) S_{\mathrm{o}}(z) y(k) , \; u_T(k) = L(z) T_{\mathrm{o}}(z) u(k) . \tag{7.38b}$$

The subscript 'N' indicates that the tuning is based on N data points.

By recalling that

$$\| H(z) \|_2^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H(e^{j\Omega}) \right|^2 d\Omega , \tag{7.39}$$

where Ω is the normalized angular frequency related with the actual ω via $\Omega = \omega T_s$, one may find a frequency-domain interpretation of the MB cost (7.32):

$$
\begin{aligned}
J_{\mathrm{MB}}(\boldsymbol{\theta}) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |T_{\mathrm{o}} - C(\boldsymbol{\theta})S_{\mathrm{o}}P|^2 |W|^2 \, d\Omega \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|P|^2}{|1+PC_{\mathrm{o}}|^2} |C_{\mathrm{o}} - C(\boldsymbol{\theta})|^2 |W|^2 \, d\Omega.
\end{aligned} \tag{7.40}
$$

The argument $e^{j\Omega}$ is dropped for simplicity. A frequency-domain interpretation of the DB cost (7.37) is derived under standard assumptions that the observed signals are stationary and ergodic [113,140], and that the number of observed data points is sufficiently high ($N \to \infty$) for asymptotic convergence $J_{\mathrm{DB}}^{N}(\boldsymbol{\theta}) \underset{N\to\infty}{\longrightarrow} J_{\mathrm{DB}}(\boldsymbol{\theta})$, where

$$
\begin{aligned}
J_{\mathrm{DB}}(\boldsymbol{\theta}) &= E[\{L(z)[T_{\mathrm{o}}(z)u(k) - C(z,\boldsymbol{\theta})S_{\mathrm{o}}(z)y(k)]\}^2] \\
&= E[\{L(z)[T_{\mathrm{o}}(z) - C(z,\boldsymbol{\theta})S_{\mathrm{o}}(z)P(z)]u(k)\}^2] \\
&= E\left[\left\{\frac{L(z)P(z)(C_{\mathrm{o}}(z) - C(z,\boldsymbol{\theta}))}{1+P(z)C_{\mathrm{o}}(z)}u(k)\right\}^2\right].
\end{aligned} \tag{7.41}
$$

Here, $E[.]$ is the expectation operator. By using Parseval's identity [113,140], one may find the frequency-domain counterpart of $J_{\mathrm{DB}}(\boldsymbol{\theta})$:

$$
J_{\mathrm{DB}}(\boldsymbol{\theta}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{|L|^2 |P|^2 |C_{\mathrm{o}} - C(\boldsymbol{\theta})|^2}{|1+PC_{\mathrm{o}}|^2} \phi_u(\Omega) d\Omega, \tag{7.42}
$$

where the argument $e^{j\Omega}$ has been dropped again, and ϕ_u is the power spectral density (PSD) of the input signal $\{u(k)\}_{k=1,2,\ldots,N}$.

The equivalence between (7.40) and (7.42) is established following the same reasoning as in the VRFT method:

1) If $C_{\mathrm{o}}(z) \in \{C(z,\boldsymbol{\theta})\}$, then the minimizer $\hat{\boldsymbol{\theta}}^{\mathrm{MB}}$ of J_{MB} coincides with $\hat{\boldsymbol{\theta}}^{\mathrm{DB}}$ of J_{DB}, as they both correspond to C_{o}, no matter what the plant, the filters, and the desired transfer functions are. Thus, if the number N of data points is high enough, then the controller optimally tuned with respect to J_{DB} (7.37) coincides with the optimal one with respect to J_{MB} (7.32).

2) If $C_{\mathrm{o}}(z) \notin \{C(z,\boldsymbol{\theta})\}$, then only a suboptimal solution of the MB problem (7.13) is found; however, this suboptimal solution can still be determined using the DB algorithm (7.19a,b); a sufficient condition for this is:

   $$
   |L|^2 = |W|^2 / \phi_u. \tag{7.43}
   $$

   If the filter L is chosen according to (7.43), then, by virtue of (7.40) and (7.42), $J_{\mathrm{DB}}(\boldsymbol{\theta}) \equiv J_{\mathrm{MB}}(\boldsymbol{\theta})$, implying that $\hat{\boldsymbol{\theta}}^{\mathrm{DB}}$ coincides with $\hat{\boldsymbol{\theta}}^{\mathrm{MB}}$. A viable choice of the filter $L(z)$ is

   $$
   L(z) = W(z)Q(z), \tag{7.44a}
   $$

   where

   $$
   Q(z) = 1/U(z),\ |U|^2 = \phi_u. \tag{7.44b}
   $$

   The filter $Q(z)$ must be stable. As suggested in the VRFT method, $Q(z)$ can be obtained by fitting a parametric transfer function into the PSD of the observed input signal $\{u(k)\}_{k=1,2,\ldots,N}$.

Standard methods to obtain such a parametric fit are described in [113,140]. If $\{u(k)\}_{k=1,2,\ldots,N}$ is white Gaussian zero mean noise with variance σ_n^2, then Q is equivalent to $1/\sigma_n$.

7.4.3 Relation with the Virtual reference feedback tuning

The presented DB control design is akin to the VRFT technique. The most important linkage is the methodology for establishing the equivalence between the considered DB and MB cost functions. This methodology, as emphasized at several points above, follows from the VRFT [33,34,107,108]. However, a few departure points from the VRFT can also be observed:

1) In this thesis, the DB counterpart of a given MB cost is obtained in a different way than in the VRFT method: here, the DB cost is derived from the given MB function (see (7.32),(7.34)-(7.37)), while in the references [33,34,107,108] related with the VRFT, the DB cost is simply postulated, and then its equivalence with the MB function is established.
2) In the DB method suggested here, there is no inversion of the desired transfer functions S_{o} and T_{o}, while in the variants of the VRFT method presented in [33,107,108], S_{o} and T_{o} should be inverted. If any of these transfer functions is not strictly proper, after inversion it induces a non-causal filtering operation. Non-causal filtering is not needed in the method presented here.
3) The suggested DB method resorts to the single MB cost function (7.32), while in the VRFT method, the MB cost (7.33) is composed of two individual MB criteria. This difference has a consequence in the number of filtering operation that must be applied on the observed signals when tuning the controller parameters. Here, the filtering process is defined by (7.38b), and it depends on the L-filter responsible for the equivalence between MB and DB costs. The filtering process of VRFT features two L-filters, one for each function composing the cost (7.33). As seen from (7.43) and (7.44a,b), the L-filter used here does not depend on S_{o} and T_{o}, while each of two L-filters from the VRFT method [108] contains the L-filter (7.44a,b) multiplied with the corresponding desired transfer function. Consequently, more filtering operations via S_{o} and T_{o} are required in VRFT than in the DB design suggested in this thesis.

7.4.4 Effect of Disturbances

It is shown in [33,107] that disturbance effects in the observed signals can cause biased controller tuning. If a disturbance (noise) is a realization of a stochastic process, then its influence can be compensated as suggested in [33,107,108]–using the instrumental variable technique [113]. In this technique, two experiments are needed. In both experiments, identical input sequence $\{u(k)\}_{k=1,2,\ldots,N}$ is applied to the plant. If the plant operates in closed-loop, then two experiments are performed with the same reference input r. The plant's outputs $\{y(k)\}_{k=1,2,\ldots,N}$ and $\{\tilde{y}(k)\}_{k=1,2,\ldots,N}$ for each trial are measured. These outputs are not identical, since they are affected by different realizations of the noise in the two experiments. Then, the modified version of the least-squares algorithm (7.38a,b) is applied:

$$\hat{\boldsymbol{\theta}}_N^{\text{DB}} = (\mathbf{A}_N)^{-1}\mathbf{f}_N;\ \mathbf{A}_N = \sum_{k=1}^{N}\zeta(k)\varphi^T(k);\ \mathbf{f}_N = \sum_{k=1}^{N}\zeta(k)u_T(k), \tag{7.45a}$$

where

$$\varphi(k) = \boldsymbol{\beta}(z)C_{\text{p}}(z)L(z)S_{\text{o}}(z)y(k),\ \zeta(k) = \boldsymbol{\beta}(z)C_{\text{p}}(z)L(z)S_{\text{o}}(z)\tilde{y}(k)$$

$$u_T(k) = L(z)T_{\text{o}}(z)u(k). \tag{7.45b}$$

Here, ζ represents the instrumental variable. It is shown in [113], that if the noise signals in two experiments are uncorrelated, then, asymptotically, (7.45a,b) gives the same controller tuning as in the noiseless case.

Unfortunately, the instrumental variable technique cannot help with deterministic disturbances. These disturbances are not problematic if their deterministic behaviour can be understood. Knowledge of this behaviour helps one to discriminate between disturbances and information relevant to plant dynamics, when processing empirical signals observed from the plant. If, however, the deterministic behaviour is so complex that its mathematical description is too costly and time consuming, perhaps even impossible, then the designer should find another way to extract the relevant information about the plant dynamics. A simple but effective solution is to use signals synthetically generated from some plant model, instead of the signals obtained by direct measurements. This, of course, assumes that a model of sufficient quality is already available, which is not always true. If it is possible to use the synthetic data, then the influence of deterministic disturbances is safely avoided. This solution is used in the case study presented next.

7.4.5 Data-based design for the RRR robot

Once again, the RRR robot is the subject of a case study. The DB method presented in subsection 7.4.2 is used to design the robot feedback controllers. Here, the design for joint 1 is demonstrated only. The DB designs for the other joints were carried-out in a similar way.

The considered DB design applies to linear plants. As discussed in section 4.6, when the robot nonlinearities are compensated using (2.26) with the conventional PD stabilizing feedback control, a poor coherence between the excitation and response signals is achieved within the bandwidth (below 4 [Hz]) of the servo-system for experimental observation. The consequence is that the observed signals cannot reliably represent decoupled robot dynamics at low frequencies. On the other hand, the robot dynamics is deterministic, but highly nonlinear and coupled, with significant friction effects. The model-based control reduces nonlinearity but does not eliminate it. The dominant nonlinearity is also in the lower frequency range, where the problem of poor coherence has already been experienced. Therefore, the online observed signals are always corrupted with residual nonlinear effects and do not contain reliable information about the system dynamics at low frequencies. Although deterministic, the nonlinear effects are highly complex, and they can cause biased tuning of DB designed controllers.

Problems with empirical data at low frequencies can be resolved using the models of robot dynamics that remain after nonlinear model-based compensation. As explained in section 4.6, the identified frequency-response functions (FRFs), presented in Fig. 4.22, have incorrect shapes at low frequencies, because of the coherence problem and nonlinearities. However, at these frequencies the decoupled robot dynamics are rigid and can be represented with decoupled single inertias, i.e., decoupled double integrators. Therefore, the incorrect shapes can be corrected to match the right ones (of double integrators), which is done by appropriate fitting of the parametric model to the FRF data. In section 4.6, Fig. 4.23 presents the nominal data-driven FRF of joint 1 together with the Bode plot of the 7th order parametric model, while in subsection 7.3.2, Fig. 7.10 presents the FRF and the Bode plot of the model of the 15th order. The FRF has an incorrect shape below 4 [Hz], while the low-frequency characteristics of the Bode plots are correct. As already emphasized in subsection 7.3.2, the higher order model better fits FRF data at higher frequencies, which makes it a better choice for generation of signals u and y required by the DB method. The synthetic signal generation circumvents the problems experienced with empirical signals at low frequencies.

The desired sensitivity $S_{o,1}$ and the complementary sensitivity $T_{o,1}$ transfer functions for joint 1 servo-system are shown in Fig. 7.27. Index '1' refers all functions to the joint 1 servo-system. To indicate that $S_{o,1}$ and $T_{o,1}$ are not in complement, $1-S_{o,1}$ is presented in the same figure. The reference functions were chosen in accordance with the design rules r.1-r.4 given in section 5.2. In particular, $S_{o,1}$ was chosen such as to achieve high error reduction in the lower frequency range, while $T_{o,1}$ ensures robustness against high-frequency resonances. At the beginning, the slope of $S_{o,1}$ is +3, indicating the need to achieve a single integrator in the resulting controller; the plant itself introduces a double integral effect. The performance weighting function $1/W_1^S$ and the stability robustness weighting $1/W_1^{\delta}$, used in the μ-synthesis control design presented in section 5.5, were starting points for $S_{o,1}$ and $T_{o,1}$, respectively. The weighting filter W_1 from the MB cost (7.32) was chosen to adequately penalize the lower and higher frequency ranges, and it is also shown in Fig. 7.27.

The DB controller design was carried out as described in subsection 7.4.2. The necessary data was generated using the parametric model $P_1(z)$. The model was excited in open-loop with white Gaussian noise of zero mean and variance $\sigma_n = 150$. The excitation and the corresponding model output were observed with a sampling time of $T_s = 1$ [ms]. The number of collected data points was $N = 2^{15}$. The *L*-filter from the DB cost (7.37) was chosen as in (7.44a,b), with $Q = 1/\sigma_n$. The prescribed structure of the controller $C(z, \boldsymbol{\theta})$: $C_{\mathrm{p}}(z)$ was the product of one integrator (thus directly enforced in the controller structure) and two notch filters; the notches were based on our experience of resonances at 28 [Hz] and 98 [Hz] in the position measurements from the first robot joint; 12 FIR filters were used as the basis functions: $\boldsymbol{\beta}^T(z) = [1,\ z,^{-1}\ z,^{-2}\ \ldots,\ z^{-12}]$; hence, 13 tuning parameters $\boldsymbol{\theta} = [\theta_0,\ \theta_1,\ \theta_2, \ldots,\ \theta_{12}]^T$ were induced. The total order of 16 was intentional; this is the order of the feedback controller of satisfactory performance obtained after model reduction of the controller obtained via μ-synthesis (see section 5.5). To investigate if selection of the controller class of higher complexity can improve the motion control performance, the adopted controller orders for joints 2 and 3 were higher than obtained using the μ-synthesis designs. In particular, both orders were increased by a factor 3.

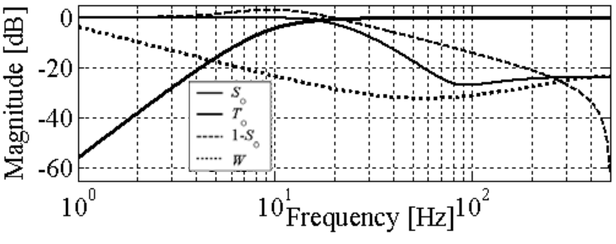


Figure 7.27: The reference models $S_{o,1}$ and $T_{o,1}$ and the weighting filter W_1.

As one of the criteria for the quality of the controller $C_1(z, \hat{\boldsymbol{\theta}}_N^{\mathrm{DB}})$ resulting from the DB design (the measure of distance from $C_{o,1}$), it was evaluated how close the achieved sensitivity function $(1/(1+P_1(z)C_1(z, \hat{\boldsymbol{\theta}}_N^{\mathrm{DB}}))$ and its complement (complementary sensitivity) are from $S_{o,1}$ and $T_{o,1}$, respectively. Another criterion was the maximum acceptable peaking in the sensitivity of 6 [dB] (design rule r.3 from section 5.2). Finally, $C_1(z, \hat{\boldsymbol{\theta}}_N^{\mathrm{DB}})$ must be stabilizing for the plant in closed-loop. This was evaluated using the Nyquist criterion [52] applied on the open-loop gain computed as the product between the plant's FRF and $C_1(z, \hat{\boldsymbol{\theta}}_N^{\mathrm{DB}})$.

The parameters $\hat{\boldsymbol{\theta}}_N^{\mathrm{DB}}$ were computed using (7.38a,b). The Bode plot of the resulting controller is shown in Fig. 7.28. By inspection of the plot, one notices that the integral action was achieved, and that effects of the enforced notches are present in the controller. Apart from the enforced ones, several other notch effects are also obvious. Induced by the resonances in the plant dynamics (see Fig. 7.10), these effects were created by the tuning part of the controller $\boldsymbol{\beta}^T(z)\boldsymbol{\theta}$. Bode magnitude plots of the achieved sensitivity and complementary sensitivity transfer functions are shown in Fig. 7.29, together with the corresponding desired transfer functions. The achieved transfer functions were computed based on the plant FRF data and $C_1(z, \hat{\boldsymbol{\theta}}_N^{\mathrm{DB}})$. Similarities between the plots in the frequency ranges of interest are acceptable, the peaking in the sensitivity is below 6 [dB], and the controller passed the stability test. Therefore, the adopted requirements for the quality of the design have been met.

As the final step, the DB controllers designed for all three joints were implemented in the robot control system and their quality was evaluated in experiments. The obtained experimental results are presented next.

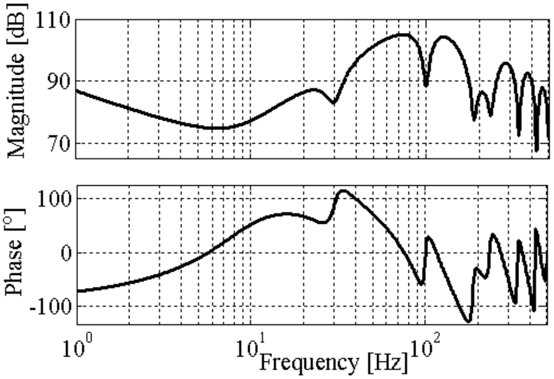


Figure 7.28: The feedback controller for joint 1 servo-system designed using the DB method.

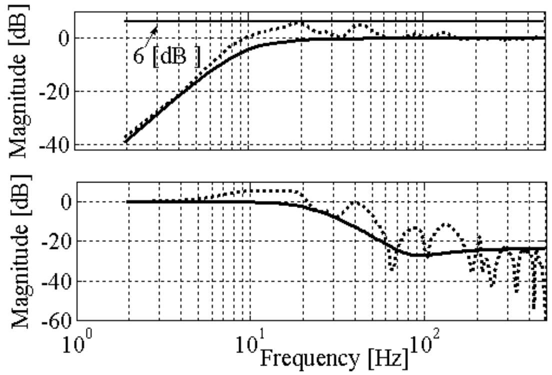


Figure 7.29: The reference (solid) and realized (dotted) closed-loop transfer functions: sensitivity (top) and complementary sensitivity (bottom).

7.4.6 Experimental results

The principle of experimental robot control was as depicted in Fig. 5.7 (section 5.4): the control law (2.26) was applied with the DB feedback controllers $\mathbf{C} = diag[C_1, C_2, C_3]$.

In the first experiment, the RRR robot was realizing the motion task presented in Fig. 5.8 (section 5.4). The achieved position errors are shown in Fig. 7.30: the errors in joints 1 and 2 stay within the range $[-1.1, 1.1] \times 10^{-3}$ [rad], and in joint 3 the error is twice that of the first two. The obtained accuracy is comparable with the accuracies achieved with advanced model-based designs considered in sections 5.4 and 5.5. The achieved cross-over frequencies of the joint servo-systems are: 15.5 [Hz] for joint 1, 19 [Hz] for joint 2, and 21 [Hz] for joint 3. They are also comparable with the cross-over frequencies presented in Table 5.1, obtained with the model-based H_∞ design explained in section 5.4.

To evaluate if the notch effects were useful, the cumulative power spectra (CPSs) of the errors are shown in Fig. 7.31. By inspection of these plots, one notices that the dominant energy of the tracking error is in the lower frequency range, which is the range of the reference trajectory shown in Fig. 5.8. Outside this range, abrupt changes in the slopes of the error CPSs are not visible, which means that no resonance was excited during the robot motion. The error variances are 0.11 [$\times 10^{-6}$ rad^2/s^2] for joint 1 and 2, and 0.45 [$\times 10^{-6}$ rad^2/s^2] for joint 3. The variance for joint 1 is lower than achieved with the H_∞ design, and it is identical with the variance achieved using the μ-synthesis control design presented in section 5.5. The variances for joints 2 and 3 are both lower than the variances achieved using the H_∞ and μ-synthesis designs. Achievement of lower variances confirms the expectation indicated in the previous subsection, that increasing the controller complexity can improve motion performance.

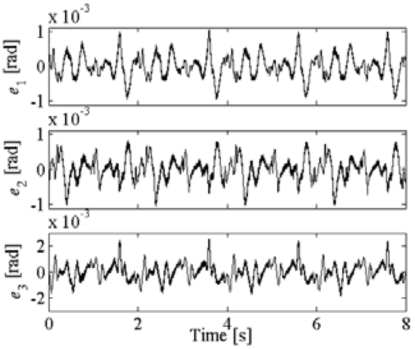


Figure 7.30: Joint position error in the first experiment.

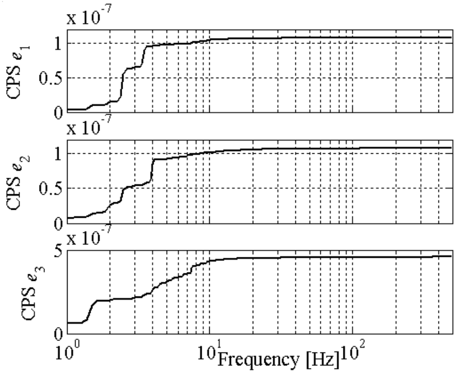


Figure 7.31: Cumulative power spectra of the joint position errors.

In the second experiment, the robot was realizing the writing task defined in the robot task and joint spaces according to Figs. 3.4 and 4.3, respectively. The achieved Cartesian errors are shown in Fig. 7.32. The peak errors along x-axis are within [-0.5,0.5] [mm], along y-axis they are within [-3,3] [mm], and along z-axis within [-1,0.8] [mm]. These high accuracies are compatible with the ones achieved with the advanced model-based strategies. CPSs of the errors do not reveal excitation of parasitic dynamics.

Having the obtained experimental results in mind, it can be claimed that the controllers resulting from the suggested DB design are capable to realize motion control of high performance. Undoubtedly prominent, the considered DB method gives opportunities for additional contributions. Just to mention three possibilities. The first objective should be overcoming the need for synthetically generated data in the DB design of the robot motion controllers. The target is a design based on empirically observed signals, only. Two possibilities can be investigated: a) organization of a dedicated experiment to improve coherence between the robot input and output signals in the complete frequency range of interest, and, b) incorporating the influence of nonlinear disturbances into the DB controller design. The second objective should be derivation of a stability test compatible with the DB method. The third objective should be adaptation of the method to extend its application from off-line controller design to an online DB control strategy. The recursive least-squares algorithm presented in [113] seems an appealing tool for achieving online capability. The expected benefit is an adaptive feature of the DB controller, which realizes the desired control performance despite time-varying plant dynamics and disturbance effects.

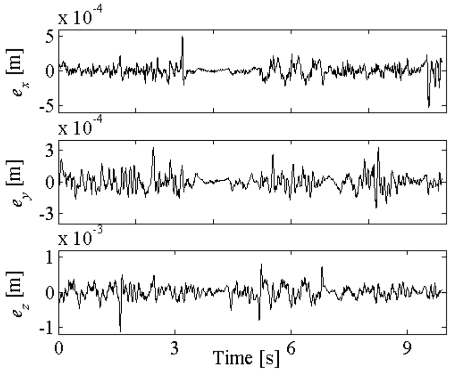


Figure 7.32: Experimental Cartesian errors in the writing task.

7.5 Summary and conclusions

Three data-based control methods were considered in this chapter: iterative learning control (ILC), iterative feedback tuning (IFT), and a new data-based method for simultaneous shaping of the sensitivity and complementary sensitivity functions. Theory behind each method was formulated as a ready-to-use solution for robot motion control. These methods were practically demonstrated on the RRR robot, and the results of their experimental evaluations are of particular value in this thesis. The pioneer experiments in data-based robot motion control presented in this chapter, provide new and instructive information about the practical merits and difficulties of the considered control methods. The most important conclusion is that all investigated methods are shown useful in robot motion control. The remaining of this section summarizes particular findings obtained during experimental studies and discusses under which circumstances each of the considered methods is more adequate choice than other ones. Also, open issues related with these methods are addressed.

If the reference motion is known in advance and has repetitive nature, then use of ILC should be encouraged, since it enables significant improvement of robot motion accuracy. It is shown that ILC design can be performed for each robot joint separately, while the robot nonlinearities are dynamically compensated using model-based control. The dynamic compensation for nonlinearities facilitates computation of the learning controllers in frequency-domain, by applying the design rules that are standard in motion control of linear systems but are rarely used in robotics. These design rules enable one to anticipate the frequency range where the learning controllers will contribute to the quality of motion control, together with the order of error reduction within the given range. The convenience of quantitative performance prediction is not typical in robot ILC, so the considered ILC design is an example how the field of motion control can contribute to robotics. The quantitative prediction is possible only if the designer has more knowledge about the plant dynamics. This finding is in contradiction of the original idea that ILC requires only little information about the robot dynamics [71]. However, the finding[82,125,126] supports the similar conclusion found in [131]. A possible weakness of the considered ILC design is utilization of only linear plant dynamics, although the design is experimentally confirmed applicable even on a highly nonlinear direct-drive RRR robot. A useful subject for further research would be explicit accounting for plant nonlinearities in the design, since this would allow more accurate performance prediction during iterative learning robot control. Another research topic could be development of a strategy that would allow derivation of adequate control inputs for all motion references that are similar with the reference for which the learning has been performed. For instance, one could accomplish ILC design for a single motion reference to obtain sequences of additional control inputs. By appropriate modification of these sequences, the developed strategy would produce additional control inputs that are adequate for any other trajectory of the same path shape but with different velocity profile. In turn, no learning would be needed for the similar trajectories, which would improve time efficiency of the control design. Finally, robustness of the convergence of the ILC algorithm against non-repeatable disturbances contained in the measurements is the additional issue which should be rigorously studied in the future.

If the reference motion task is not repetitive, but the reference robot trajectory is rather arbitrary, then instead of ILC one should focus on feedback control designs. The methods considered here, namely, IFT and data-based loop shaping, are the viable options for feedback design.

An IFT algorithm suitable for robotics applications is derived in this chapter and experimentally evaluated on the RRR robot. This algorithm was used in combination with dynamic compensation for robot nonlinearities, which is a strategy not seen elsewhere in the robotics literature. The experimental verification shows the potential of the algorithm to provide significant performance improvements. However, these improvements cannot straightforwardly be achieved without prior knowledge about the plant dynamics. Without such knowledge, it is not easy to choose adequate design settings (e.g., reference model and weighting filters). With some knowledge available, the desired control objectives can be easily specified. To apply IFT, one should afford iterations of experiments on a given robot. In return, the IFT algorithm delivers optimal controller tuning with respect to information about the plant dynamics, contained in the observed closed-loop data. It is shown that data generated from the plant model can give different feedback tuning than data observed from the real system. This is because the outcome of the IFT design is influenced by actual behaviours encountered in the data. Thus, it seems reasonable to recommend IFT for fine tuning of even model-based controllers. Issues that warrant further researches include ensuring the closed-loop stability throughout iterations and inclusion on nonlinear dynamics in an IFT control design. The opportunity to ascertain stability would enable specification of higher demands on the control performance during IFT design. Incorporation of nonlinearities might increase effectiveness of IFT for a broader class of plants.

Iteration of experiments required by IFT may take a lot of the designer’s time. If one cannot afford a long design period, then one-shot data-based feedback tuning presented in the last part of the chapter seems as a more practical solution. This new data-based method enables simultaneous shaping of the closed-loop sensitivity and complementary sensitivity transfer functions. The performance specification is as intuitive as in model-based loop shaping control design but, similarly with IFT, specification of admissible performance objectives requires closer knowledge of the plant dynamics. In this chapter, the data-based loop-shaping was evaluated in the robot motion control problem. In this problem, the method is capable to tune controllers of given structure and complexity such that motion control of high performance is realized. However, the method is shown sensitive to deterministic disturbances (e.g., uncompensated nonlinear effects and poor coherence between excitation and response signals) present in the signals utilized during the controller design. The observed problem is resolved by substituting experimental signals with the data synthetically generated from a model. The next objective is to overcome the need for the synthetically generated data, since the target is a design based on empirically observed signals, only. For this, two possibilities can be investigated: a) design of a dedicated experiment to improve coherence between the robot input and output signals in the complete frequency range of interest, and, b) incorporating the influence of nonlinear disturbances into the data-based controller design. A stability test compatible with the considered data-based method is also needed.

Chapter 8

Conclusions and recommendations

This chapter reviews all solutions to the robot motion control problem that have been treated throughout the thesis. The solutions are discussed in light of the corresponding research questions posed in section 2.3. Values and limitations of each solution are summarized, and recommendations for future research are given.

To meet ever increasing demands on robot motion control performance, several data-driven control designs have been examined in the thesis. Their development, implementation, and evaluation have been made possible by employing different fields of theory, technology, and tools. None of the considered designs can be characterized as simplistic. On the contrary, each requires powerful software and hardware aids to be performed successfully. Fortunately, it appears that all resulting control laws can be time-efficiently derived and implemented using the standard Matlab/Simulink software and modern yet affordable hardware tools. These tools, in combination with the RRR robot, allowed the evaluation of the control designs under experimental conditions, which is very instructive for verifying their practical applicability. Because of the highly nonlinear dynamics, the RRR robot posed a significant challenge to achieving the principle objective of this thesis, namely, robot motion control of high quality. The hardware and software components of the RRR robot control system allowed verifying to what extent each design can contribute to the principle objective.

Present shortcomings of hardware and software tools useful for robot motion control designs are the lack of integration and compatibility. For instance, various Matlab toolboxes are needed to accomplish different steps of the design, i.e., rigid-body dynamic modelling, parameter estimation, data-driven modelling, derivation of model-based and data-based control laws, and real-time implementation of these laws. Incomplete compatibility of different tools and moderate integration of the whole control design process contribute to a fair amount of manual programming effort. Improved integration and compatibility, as well as user friendliness, would certainly stimulate broader application of the tools and implementation of more advanced control laws in practice.

Model-based compensation of robot nonlinearities is a powerful concept which contributes to the quality of motion control. The nonlinearity caused by friction in the robot joints can be covered with relevant parametric representations of the friction effects. A number of these representations can be found in the literature. This thesis examined performance of two parametric representations, namely, the three-sigmoidal and the LuGre friction model. Both models cover major effects in the sliding friction regime, which dominates at nonzero joint velocities. The LuGre model also captures behaviours exhibited during the pre-sliding friction regime, i.e., just before the joint motion is established. This is achieved at the cost of higher computational complexity, since friction compensation with the LuGre model involves integration of a set of nonlinear differential equations. Parameters of the three-sigmoidal and LuGre models can be estimated experimentally using straightforward procedures suggested in recent. When evaluated in motion tasks where the pre-sliding friction regime is rarely exhibited (less stick phases and not so low velocity of motion), both models have shown similar contribution to the quality of motion control. Rather fast than slow motions were of concern in this thesis, since rigorous benchmarking of the quality of a robot motion control design requires movements that impose significant dynamic loads for the robot control system. Additionally, accurate control of faster movements is appealing for industrial robot applications, since it can contribute to speed-ups in manufacturing tasks, leading to higher throughput. Because of lower computational complexity, the three-sigmoidal model can be recommended for model-based compensation of friction in robots that are supposed to perform fast motions.

Robot nonlinearities arising from the mechanical design, are described by means of rigid-body modelling. This concept is well-understood in robotics, and it is performed by a straightforward routine: given the robot geometry, a model of the rigid-body dynamics is derived automatically. In this thesis it is shown that straightforward experimental estimation of the model parameters is also possible, and it can be performed batch or online. Based on the derived dynamic model, there is a formal way to compute a robot trajectory whose experimental execution sufficiently reveals influences of the model parameters to the rigid-body dynamics. An important ingredient of batch parameter estimation is the use of a Kalman filtering technique to obtain motion coordinates, i.e., positions, velocities, and accelerations. This technique filters out the effects of disturbances and flexibilities and reconstructs only the component of the motion coordinates corresponding to the rigid-body dynamics. Based on this component of the motion coordinates, the efficient least-squares algorithm delivers estimates of the model parameters of sufficient quality. The opportunity to use this algorithm instead of more complex ones suggested in the literature, highly facilitates the estimation process. The suggested online parameter estimation using batch adaptive control is simple for implementation, but also less sensitive to noise and disturbances than standard adaptive estimation techniques. Contrary to adaptive techniques that perform estimation via nonlinear feedback control, the batch adaptive algorithm estimates parameters while delivering feedforward component of the control input. Nonlinear adaptive feedback control is more sensitive to stability problems than feedforward control using the batch adaptive algorithm. Therefore, the algorithm suggested in this thesis seems to be a more practical solution for online parameter estimation than the standard adaptive techniques, in respect of both stability robustness and robustness against noise and disturbances.

Experimental parameter estimation is performed in a straightforward fashion, but there are still possibilities for further improvements. For instance, a robot trajectory to be executed during the identification experiment is derived by solving a non-convex optimization problem. It would be advantageous if the optimization problem could be reformulated as a convex one. The data required for parameter estimation is acquired during feedback control of a nonlinear robot. There is a good understanding of bias and variance issues for models of linear dynamics identified in closed-loop, but these issues are still insufficiently clear in the case of nonlinear models. Better understanding of the bias and variance properties of nonlinear models identified under closed-loop operating conditions would contribute to the quality of estimates of the parameters describing the rigid-body robot dynamics.

The thesis confirms that comprehensive physical modelling of rigid-body dynamics and friction is insufficient to cover all dynamics of a real robot. Capturing these dynamics using a system identification approach is a viable option demonstrated in the thesis. This approach is shown to be useful in both joint and task spaces of the robot. Data-driven modelling provides linear nominal models of the additional dynamics together with linear parametric representations of the modelling uncertainty. These models and uncertainty descriptions allow loop-shaping design of feedback controllers. The loop-shaping approach provides deeper engineering insight during the design phase, since tools like frequency response, filtering, Bode and Nyquist plots can be used directly. What is very important is that the identified nominal models and uncertainty descriptions can be used in linear H_∞/μ optimal feedback control designs which enable controller tuning for the desired performance. These designs allow quantitative specification of the performance characteristics, e.g., the amount of resonance and disturbance suppression, control effort, and position accuracy. Experimental results obtained on the RRR robot verify that the specifications on the position accuracy can be practically realized.

It is shown that the task space data-driven modelling enables time-efficient identification of flexibilities of the robot structure. The advantage is a straightforward combination of the identified model with the model of the robot rigid-body dynamics with friction. The combined model is suitable for implementation in the computed-torque control law. This law transforms the robot into a regulator motion control system whose function is to suppress robot tip vibrations. The resulting regulation problem can be solved with many control methods, including methods from linear robust control theory. In this thesis, linear H_∞ feedback control is applied, leading to robust compensation of the tip vibrations.

The work indicates that the convenience of controller tuning based on data-driven models is not limited to linear feedback designs only, since it is at least applicable to one nonlinear discrete-time sliding mode control (DSMC) design. Unfortunately, it appears that if the feedback DSMC law is based only on rigid-body components of the robot dynamics, that are not covered with dynamic compensation of nonlinearity, this law can easily cause undesirable effects for the motion performance, namely, noise amplification and excitation of parasitic dynamics. A theoretical analysis has shown that these undesirable effects are a consequence of restriction in the feedback tuning, and they can not be avoided, even with careful application of loop-shaping. As a possible remedy for the observed problem, one could investigate the DSMC design based on a data-driven model which includes flexibilities. Such a design will certainly increase the order of the resulting controller. The expected merit, though, would be robust feedback control with nice and predictable transient characteristics.

The research shows that supplementing the model based compensation of robot nonlinearities with robust feedback control designs is beneficial for the quality of motion control. However, such a strategy can hardly be clarified as optimal. The formidable weakness is that the robust feedback is designed separate from the model-based compensation of the robot nonlinearities. The robust feedback should cope with limitations of the model-based nonlinear control but, in general, cannot completely correct all insufficiencies of the model-based control component. This is because the robot dynamics not covered with the dynamic compensation of nonlinearity are not linear and time invariant, but still nonlinear and time varying. Of course, nonlinearity of the additional dynamics is less profound if the implemented model of the robot rigid-body dynamics with friction is more accurate. A robust feedback design is based on a linearization of the additional dynamics, captured by data-driven modelling. The feedback control law based on linearization of actually nonlinear behaviour can hardly achieve globally optimal control performance.

Conservative stability analysis is an additional consequence of separate design of the dynamic compensation for the nonlinearity and of the robust feedback control law. The nonlinearity compensator interacts with the feedback controller and both influence stability in the closed-loop. In the thesis, stability is ascertained via data-driven models of the nominal dynamics and the description of modelling uncertainties. Very conservative uncertainty modelling has been employed to enhance guarantee for closed-loop stability. Conservative stability analysis limits the achievable motion performance. Relaxation of the stability conditions would probably allow better quality of motion control.

Ideally, the motion controller would be an entity designed in one step, leading to globally optimal control performance. Therefore, an important subject for further research should be obtaining accurate models of the actual robot dynamics and the development of control designs based on these models, covered with adequate stability analysis. Although significant progress has been made in derivation of general robot dynamic models that include flexible effects, determining accurate values of the model parameters still remains a challenge. The additional challenges are employing the general models in control designs for the desired performance, and stability analysis of such designs. Future research should provide adequate answers to these challenges.

An alternative is the use of nonlinear system identification theory and function approximators, e.g., neural-networks, splines, wavelets, etc., to capture robot dynamics. Algorithms that enable general data-driven modelling have been appearing in the robotics literature, while compatible control theory is developing, too. Nonlinear system identification will certainly play an important role in future robot motion control designs, at least as a compromise solution between physical and data-driven modelling. The nonlinear data-driven modelling could be of use to capture more versatile behaviours not covered with a model of the robot rigid-body dynamics with friction, including nonlinear effects. The resulting model would be more general than the model of a linearization of the additional dynamics. Availability of the more general model should enable a less conservative stability analysis and performance improvement.

Another alternative to the modelling approach is data-based control design. Some possibilities within this alternative have been examined in the thesis. If the robot motion task is known in advance and should be repetitively executed, then iterative learning control (ILC) appears as a powerful remedy for the insufficiencies of the implemented computed-torque control law and the stabilizing feedback. Moreover, it is shown in the thesis that sophisticated tuning of the learning controller is possible even for highly nonlinear direct-drive robots. The sophisticated tuning requires intimate knowledge of the parasitic robot dynamics that can be captured via data-driven modelling. It should be admitted, though, that the need for better insight into the actual dynamics is not a desirable requirement according to the robotics literature on ILC. On the other hand, it seems inevitable if one wants to profit from the quantitative prediction of the effect of ILC on the control performance. Acquiring information about the parasitic dynamics certainly increases the control design effort, but the profit of it is undoubtedly worth the additional effort. Use of data-driven modelling does not make ILC a model-based technique, since the model just serves as an additional aid increasing the number of merits that ILC can provide.

It is a fact that ILC produces the appropriate control command only for a single motion reference. There is little guidance at present how to modify that controller for similar trajectories. Additional questions that warrant further research include the effect of non-repeatable disturbances on the convergence of an ILC algorithm, as well as learning of non-repetitive tasks. Also, it is verified in the experimental demonstration on the RRR robot that the linear ILC design applied on the nonlinear plant may induce effects that could not be predicted. In particular, the harmonic content of the position error has been amplified in the frequency range where the ILC design was actually promising error reduction. This is a warning that further developments of ILC theory are required for a better understanding of the mutual interaction between ILC and nonlinearity in the plant dynamics.

Iterative feedback tuning (IFT) has been verified to be a useful technique for the design of robot feedback controllers that achieve adequate control performance along the prescribed motion references. If very little is known about the plant dynamics, it is difficult to achieve globally optimal performance via IFT. Still, whenever the plant dynamics is unknown, IFT is a more structural approach than tuning by trial and error, while it promises better performance than application of the tuning rules of Ziegler and Nichols. These facts recommend iterative feedback tuning of conventional feedback controllers in industrial robotics. Furthermore, the suitability of IFT for higher complexity controllers can be used as a strong argument when stimulating higher order control designs for industrial robots. Higher order designs can increase throughput of robotic manufacturing, but their wider implementation is impeded by the tuning complexity. IFT is a practical technique, almost a routine, and it can significantly facilitate higher order designs. Availability of closer knowledge of the plant dynamics enables one to more directly select admissible IFT design settings, namely, reference model and weighting filters used in the objective function. Admissible are those settings that are consistent with the limits of performance imposed by physical realizability. Additionally, a more adequate guess at the initial controller tuning is possible having some knowledge of the plant. Better knowledge thus contributes to the quality of IFT control design, since with adequate initial tuning less iterations are needed to reach a local optimum of the nonconvex objective function, while less conservative design settings allow higher control performance. The experimental evaluation of the IFT algorithm derived in this thesis on the RRR robot has shown that the outcome of IFT is optimized against actual behaviours contained in the observed closed-loop data. This observation encourages refinement of model-based controllers via IFT, in order to optimize their effectiveness along the prescribed references.

An important subject for further research related with IFT should be providing strong guarantees that the closed-loop will remain stable throughout iterations. Enhancing the stability robustness would enable less conservative controller tunings, in the sense that higher demands on the control performance could be allowed. Also, IFT has been shown to be capable of coping with some nonlinearities in the plant dynamics, but the underlying mechanism is still not well-understood. Thus, further research should better clarify the types of nonlinearities for which IFT can be effective. Eventually, an IFT algorithm which takes into account nonlinear behaviours should be derived to enhance the capability of IFT to handle nonlinear plants.

An IFT algorithm requires iteration of experiments on the robot, and thus it is not a time-efficient approach to controller design. It is especially not suitable if economical reasons do not allow the robot to stay out of the nominal usage for a longer period. In such situations, more practical are the techniques that enable feedback control design in one-shot. One of these is suggested in the thesis, and it enables offline data-based simultaneous shaping of the closed-loop sensitivity and complementary sensitivity functions. This technique offers a flexible definition of the performance specifications, namely, these are defined via frequency dependent filters, similarly as in model-based H_∞/μ optimal control designs. Such flexibility adds an important degree of freedom to the control engineer. Like with IFT, the choice of admissible performance specifications requires closer knowledge of the plant dynamics. Another similarity with IFT is that the derived data-based loop-shaping controller design method enables tuning of the feedback controller whose structure and complexity are directly selected by the control engineer and not tailored by some plant model. Thus, the engineer can prescribe controller complexity which is admissible for real-time implementation at the very beginning of the design, while the data-based method in one-shot delivers optimal tuning with respect to input/output data observed from the plant. The method has been successfully benchmarked on the RRR robot, verifying its capability to deliver feedback motion controllers of high performance.

However, the data-based loop-shaping design method has been shown sensitive to nonlinear effects contained in the signals experimentally observed from the plant, as well as on poor coherence between input and output signals. In the thesis, this sensitivity was alleviated by substituting experimental signals with data synthetically generated from high quality models of additional robot dynamics. Although effective, this remedy cannot be considered as the final one, since it still requires an intermediate plant modelling step which reduces time-efficiency of the design. Additionally, the controller tuning is optimized with respect to finite number of behaviours contained in the model, and not against experimental data that are possibly richer with behaviours. Hence, further research should provide a solution that would allow the use of experimental data. This would require an adequate circumventing the coherence problem and sensitivity to nonlinearities. Some possibilities could be the organization of a dedicated experiment to improve the coherence between the input and output signals and direct accounting for the effect of nonlinearities during the data-based design. A compromise solution could be initial controller tuning based on the synthetic data followed by experimental refinement via IFT.

An important subject for further research should be the derivation of a data-based stability test. In the thesis, stability has been ascertained based on models of the robot's additional dynamics, while the stability test compatible with the given design method should utilize only observed input/output data. An additional topic could be the definition of admissible performance specifications directly from input/output data and not mainly based on the knowledge of the plant dynamics. Finally, the given data-based control design method seems suitable for online implementation, and this option should be investigated in the future. For instance, a recursive least-squares algorithm could be used for online controller tuning, which would allow adaptation of the control action against changes in the plant dynamics. The prerequisite for this is the availability of an online data-based stability test.

Summarizing the merits and limitations of the considered control designs, it can be concluded that they are all practical solutions which can contribute to the quality of robot motion control. Instrumental of each design is recognition of the dependence between the actual robot properties and the closed-loop performance. The benefit is the capability of these designs to outperform the control design methods generally accepted in today's robotics. However, wider application of the considered designs requires that the field of robotics embraces new theories, tools, and technology, which rarely comes easy. In particular, one should afford higher complexity of the controller hardware and of the design procedures, and in addition obtain answers from the theory that resolve particular limitations of each design. The profit, though, is a robot of high performance which is more competitive on the market. Therefore, the approaches covered in this thesis can add a new value to the field of robotics, and they are worth of considering in practice.

Appendix

RRR robot kinematics and dynamics

A.1 Kinematic models in closed-form

Here, the forward kinematics model (4.2) of the RRR robot is presented in detail. For convenience, the elements of the orientation matrix ${}^0_3\mathbf{O}(\mathbf{q})$ and the Cartesian coordinates of the tip ${}^0_3\mathbf{x}(\mathbf{q})$ are denoted without '${}^0_3$', still considering that each of them is referred to the base coordinate frame '0'. The elements $o_{i,j}$ of ${}^0_3\mathbf{O}(\mathbf{q})$ are:

$$\begin{aligned}
o_{1,1} &= \cos q_1 \cos(q_2+q_3) \\
o_{1,2} &= -\cos q_1 \sin(q_2+q_3)\,, \\
o_{1,3} &= \sin q_1\,, \\
o_{2,1} &= \sin q_1 \cos(q_2+q_3)\,, \\
o_{2,2} &= -\sin q_1 \sin(q_2+q_3)\,, \\
o_{2,3} &= -\cos q_1\,, \\
o_{3,1} &= \sin(q_2+q_3)\,, \\
o_{3,2} &= \cos(q_2+q_3)\,, \\
o_{3,3} &= 0\,.
\end{aligned} \tag{A.1}$$

The elements x, y, and z of ${}^0_3\mathbf{x}(\mathbf{q})$ are:

$$\begin{aligned}
x &= \cos q_1 (a_3 \cos(q_2+q_3) + a_2 \cos q_2) + (d_2+d_3)\sin q_1\,, \\
y &= \sin q_1 (a_3 \cos(q_2+q_3) + a_2 \cos q_2) - (d_2+d_3)\cos q_1\,, \\
z &= a_3 \sin(q_2+q_3) + a_2 \sin q_2 + d_1\,.
\end{aligned} \tag{A.2}$$

The inverse kinematics model in closed-form is

$$\begin{aligned}
q_1 &= \operatorname{asin} \frac{x(d_2+d_3) + y\sqrt{x^2+y^2-(d_2+d_3)^2}}{x^2+y^2}\,, \\
q_3 &= \operatorname{atan} \frac{\pm\sqrt{1-(p_{wh}^2+p_{wv}^2-a_2^2-a_3^2)^2/(2a_2a_3)^2}}{(p_{wh}^2+p_{wv}^2-a_2^2-a_3^2)/(2a_2a_3)}\,, \\
q_2 &= \operatorname{atan} \frac{(a_2+a_3\cos q_3)p_{wv} - a_3 \sin q_3 p_{wh}}{(a_2+a_3\cos q_3)p_{wh} + a_3 \sin q_3 p_{wv}}\,,
\end{aligned} \tag{A.3a}$$

with

$$\begin{aligned}
p_{wh} &= \sqrt{[x-(d_2+d_3)\sin q_1]^2 + [y+(d_2+d_3)\cos q_1]^2}\,, \\
p_{wv} &= z - d_1\,.
\end{aligned} \tag{A.3b}$$

The elements $j_{i,j}$ of the geometric Jacobian ${}^{0}_{3}\mathbf{J}(\mathbf{q})$, referred to the base frame '0', are also denoted without '${}^{0}_{3}$':

$$\begin{aligned}
&j_{1,1} = \text{-}\sin q_1(a_2\cos q_2 + a_3\cos(q_2+q_3)) + (d_2+d_3)\cos q_1,\\
&j_{1,2} = \text{-}(a_2\sin q_2 + a_3\sin(q_2+q_3))\cos q_1,\\
&j_{1,3} = \text{-}a_3\cos q_1\sin(q_2+q_3),\\
&j_{2,1} = \cos q_1(a_2\cos q_2 + a_3\cos(q_2+q_3)) + (d_2+d_3)\sin q_1,\\
&j_{2,2} = \text{-}(a_2\sin q_2 + a_3\sin(q_2+q_3))\sin q_1,\\
&j_{2,3} = \text{-}a_3\sin q_1\sin(q_2+q_3),\\
&j_{3,2} = a_2\cos(q_2+q_3) + a_2\cos q_2,\\
&j_{3,3} = a_3\cos(q_2+q_3),\ j_{4,2} = j_{4,3} = \sin q_1,\\
&j_{5,2} = j_{5,3} = \text{-}\cos q_1,\\
&j_{6,1} = 1,\\
&j_{3,1} = j_{4,1} = j_{5,1} = j_{6,2} = j_{6,3} = 0. \qquad \text{(A.4)}
\end{aligned}$$

Notation '${}^{0}_{3}$' is avoided for the elements $j^{d}_{i,j}$ of the product ${}^{0}_{3}\dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \in \mathbb{R}^3$, too:

$$\begin{aligned}
j^{d}_{1,1} &= (\text{-}\cos q_1(a_2\cos q_2 + a_3\cos(q_2+q_3)) - (d_2+d_3)\sin q_1)\dot{q}_1\\
&\quad + \sin q_1(a_2\sin q_2 + a_3\sin(q_2+q_3))\dot{q}_2 + a_3\sin q_1\sin(q_2+q_3)\dot{q}_3,\\
j^{d}_{2,1} &= \sin q_1(a_2\sin q_2 + a_3\sin(q_2+q_3))\dot{q}_1 - \cos q_1(a_2\cos q_2 + a_3\cos(q_2+q_3))\dot{q}_2\\
&\quad - a_3\cos q_1\cos(q_2+q_3)\dot{q}_3,\\
j^{d}_{3,1} &= a_3\sin q_1\sin(q_2+q_3)\dot{q}_1 - a_3\cos q_1\cos(q_2+q_3)\dot{q}_2 - a_3\cos q_1\cos(q_2+q_3)\dot{q}_3. \qquad \text{(A.5)}
\end{aligned}$$

A.2 Rigid-body dynamics in closed-form

Elements of the regressor representation (4.4) of the RRR robot rigid-body dynamics are provided in closed-form:

$$\begin{aligned}
&p_1 = a_2^2 m_3 + a_2 m_2(a_2 + 2x_2) - I_{xx,2} + I_{yy,2},\\
&p_2 = \text{-}2(a_2 m_2 y_2 + I_{xy,2}),\\
&p_3 = m_3(a_3 + x_3),\\
&p_4 = m_3 y_3,\\
&p_5 = a_3 m_3(a_3 + 2x_3) - I_{xx,3} + I_{yy,3},\\
&p_6 = 2(a_3 m_3 y_3 + I_{xy,3}),\\
&p_7 = 2d_2 m_2 z_2 + d_2^2 m_2 + 2(d_2+d_3)m_3 z_3 + (d_2+d_3)^2 m_3 + I_{yy,1} + I_{xx,2} + I_{xx,3},\\
&p_8 = -(d_2+d_3)m_3 y_3 - I_{yz,3},\\
&p_9 = \text{-}a_3 m_3(z_3 + d_2 + d_3) - (d_2+d_3)m_3 x_3 - I_{xz,3},\\
&p_{10} = \text{-}d_2 m_2 y_2 - I_{yz,2},\\
&p_{11} = \text{-}a_2(m_3(z_3 + d_2 + d_3) + m_2(d_2 + z_2)) - d_2 m_2 x_2 - I_{xz,2},\\
&p_{12} = (a_2^2 + a_3^2 + 2a_3 x_3)m_3 + a_2 m_2(a_2 + 2x_2) + I_{zz,2} + I_{zz,3},\\
&p_{13} = a_3 m_3(a_3 + 2x_3) + I_{zz,3},\ p_{14} = m_2(x_2 + a_2) + m_3 a_2,\ p_{15} = m_2 y_2. \qquad \text{(A.6)}
\end{aligned}$$

$$
\begin{aligned}
r_{1,1} &= \ddot{q}_1\cos^2 q_2 - \dot{q}_1\dot{q}_2\sin(2q_2)\,,\\
r_{1,2} &= 0.5\ddot{q}_1\sin(2q_2) + \dot{q}_1\dot{q}_2\cos(2q_2)\,,\\
r_{1,3} &= 2a_2[(\ddot{q}_1\cos(q_2+q_3) - \dot{q}_1\dot{q}_3\sin(q_2+q_3))\cos q_2 - \dot{q}_1\dot{q}_2\sin(2q_2+q_3)]\,,\\
r_{1,4} &= -2a_2[(\ddot{q}_1\sin(q_2+q_3) + \dot{q}_1\dot{q}_3\cos(q_2+q_3))\cos q_2 + \dot{q}_1\dot{q}_2\cos(2q_2+q_3)]\,,\\
r_{1,5} &= \ddot{q}_1\cos^2(q_2+q_3) - \dot{q}_1(\dot{q}_2+\dot{q}_3)\sin(2q_2+2q_3)\,,\\
r_{1,6} &= -0.5\ddot{q}_1\sin(2q_2+2q_3) - \dot{q}_1(\dot{q}_2+\dot{q}_3)\cos(2q_2+2q_3)\,,\\
r_{1,7} &= \ddot{q}_1\,,\\
r_{1,8} &= (\ddot{q}_2+\ddot{q}_3)\cos(q_2+q_3) - (\dot{q}_2+\dot{q}_3)^2\sin(q_2+q_3)\,,\\
r_{1,9} &= (\ddot{q}_2+\ddot{q}_3)\sin(q_2+q_3) + (\dot{q}_2+\dot{q}_3)^2\cos(q_2+q_3)\,,\\
r_{1,10} &= -\dot{q}_2^2\sin q_2 + \ddot{q}_2\cos q_2\,,\\
r_{1,11} &= \dot{q}_2^2\cos q_2 + \ddot{q}_2\sin q_2\,,\\
r_{2,1} &= 0.5\dot{q}_1^2\sin(2q_2)\,,\\
r_{2,2} &= -0.5\dot{q}_1^2\cos(2q_2)\,,\\
r_{2,3} &= 2a_2[(\ddot{q}_2+0.5\ddot{q}_3)\cos q_3 - (0.5\dot{q}_3^2+\dot{q}_2\dot{q}_3)\sin q_3 + 0.5\dot{q}_1^2\sin(2q_2+q_3)]\\
&\quad + \mathrm{g}\cdot\cos(q_2+q_3),\\
r_{2,4} &= 2a_2[-(\ddot{q}_2+0.5\ddot{q}_3)\sin q_3 - (0.5\dot{q}_3^2+\dot{q}_2\dot{q}_3)\cos q_3 + 0.5\dot{q}_1^2\cos(2q_2+q_3)]\\
&\quad - \mathrm{g}\cdot\sin(q_2+q_3),\\
r_{2,5} &= 0.5\dot{q}_1^2\sin(2q_2+2q_3)\,,\\
r_{2,6} &= 0.5\dot{q}_1^2\cos(2q_2+2q_3)\,,\\
r_{2,8} &= \ddot{q}_1\cos(q_2+q_3)\,,\\
r_{2,9} &= \ddot{q}_1\sin(q_2+q_3)\,,\\
r_{2,10} &= \ddot{q}_1\cos q_2\,,\\
r_{2,11} &= \ddot{q}_1\sin q_2\,,\\
r_{2,12} &= \ddot{q}_2\,,\\
r_{2,13} &= \ddot{q}_3\,,\\
r_{2,14} &= \mathrm{g}\cdot\cos q_2\,,\\
r_{2,15} &= -\mathrm{g}\cdot\sin q_2\,,\\
r_{3,3} &= a_2[(0.5\dot{q}_1^2+\dot{q}_2^2)\sin q_3 + \cos q_3\ddot{q}_2 + 0.5\dot{q}_1^2\sin(2q_2+q_3)] + \mathrm{g}\cdot\cos(q_2+q_3)\,,\\
r_{3,4} &= a_2[(0.5\dot{q}_1^2+\dot{q}_2^2)\cos q_3 - \ddot{q}_2\sin q_3 + 0.5\dot{q}_1^2\cos(2q_2+q_3)] - \mathrm{g}\cdot\sin(q_2+q_3)\,,\\
r_{3,5} &= 0.5\dot{q}_1^2\sin(2q_2+2q_3)\,,\\
r_{3,6} &= 0.5\dot{q}_1^2\cos(2q_2+2q_3)\,,\\
r_{3,8} &= \ddot{q}_1\cos(q_2+q_3)\,,\\
r_{3,9} &= \ddot{q}_1\sin(q_2+q_3)\,,\\
r_{3,13} &= \ddot{q}_2+\ddot{q}_3\,,\\
r_{1,12} &= r_{1,13} = r_{1,14} = r_{1,15} = r_{2,7} = r_{3,1} = r_{3,2} = r_{3,7} = r_{3,10} = r_{3,11} = r_{3,12} = r_{3,14} = r_{3,15} = 0.
\end{aligned}
\tag{A.7}
$$

A.3 Estimates of the friction parameters

The estimates of all parameters of the LuGre friction model (3.18) are presented in Table A1. The estimates of all parameters of the three-sigmoidal-friction model (3.19) are presented in Table A2.

TABLE A.1: ESTIMATES CORRESPONDING TO LUGRE FRICTION MODEL

i	1	2	3
$\alpha_{0,i}$ [Nm]	2.742	2.100	1.0683
$\alpha_{1,i}$ [Nm]	0.5916	0.420	0.5208
$v_{s,i}$ [rad/s]	0.003182	0.003	0.003
$\alpha_{2,i}$ [Nm·s/rad]	1.122684	0.420	0.3756
$\sigma_{0,i}$ [Nm/rad]	14047.2	1176	663.59
$\sigma_{1,i}$ [Nm·s/rad]	44.0916	37.92	4.4496
$\hat{J}_i$ [kg·m^2]	0.8153	0.4988	0.0730
$\hat{c}_i$ [Nm]	×	17.1317	1.8685

TABLE A.2: ESTIMATES CORRESPONDING TO THREE-SIGMOIDAL-FRICTION MODEL

i	1	2	3
$\hat{f}_{i,1}$ [Nm]	0.8682	0.7819	-0.9981
$\hat{f}_{i,2}$ [Nm]	1.5935	1.2517	2.0207
$\hat{f}_{i,3}$ [Nm]	3.1896	2.4702	0.0129
$\hat{w}_{i,1}$ [rad/s]$^{-1}$	39.3652	48.9187	36.4293
$\hat{w}_{i,2}$ [rad/s]$^{-1}$	12.9257	8.0705	20.3115
$\hat{w}_{i,3}$ [rad/s]$^{-1}$	-0.2046	0.0333	0.0412
$\hat{b}_i$ [Nm·s/rad]	1.2143	0.3697	0.4321
$\hat{J}_i$ [kgm^2]	0.8358	0.5292	0.0717
$\hat{c}_i$ [Nm]	×	16.8949	1.8860

A.4 Estimates of the base parameter set

The elements of the base parameters set estimated using the batch least-squares method are presented in Table A.3.

TABLE A.3: THE BATCH LEAST-SQUARES ESTIMATES OF THE BASE PARAMETER SET

$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{p}_5$	$\hat{p}_6$	$\hat{p}_7$	$\hat{p}_8$
0.2921	-0.0142	0.17306	0.07452	-0.0079	0.8180	-0.0116
$\hat{p}_9$	$\hat{p}_{10}$	$\hat{p}_{11}$	$\hat{p}_{12}$	$\hat{p}_{13}$	$\hat{p}_{14}$	
-0.0671	0.0062	-0.3434	0.3947	0.0440	1.5541	

TABLE A.4: THE ONLINE ESTIMATED ELEMENTS OF THE BASE PARAMETER SET AND OF FRICTION PARAMETERS IN THE EXPERIMENT WITHOUT THE LAST LINK ATTACHED

$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{p}_5$	$\hat{p}_6$	$\hat{p}_7$	$\hat{p}_8$
0.24553	-0.0106	0.0011	0.0025	0.0014	0.7857	-0.0128
$\hat{p}_9$	$\hat{p}_{10}$	$\hat{p}_{11}$	$\hat{p}_{12}$	$\hat{p}_{13}$	$\hat{p}_{14}$	
-0.0102	0.0091	-0.2752	0.3492	0.0153	1.3106	
$\hat{\alpha}_{1,1}$	$\hat{\alpha}_{1,2}$	$\hat{\alpha}_{2,1}$	$\hat{\alpha}_{2,2}$	$\hat{\alpha}_{3,1}$	$\hat{\alpha}_{3,2}$	
2.5332	0.7750	2.1398	0.2988	1.0458	0.4410	

TABLE A.5: THE ONLINE ESTIMATED ELEMENTS OF THE BASE PARAMETER SET AND OF FRICTION PARAMETERS IN THE EXPERIMENT WITH THE LAST LINK ATTACHED

$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{p}_5$	$\hat{p}_6$	$\hat{p}_7$	$\hat{p}_8$
0.2924	-0.0121	0.1840	0.0554	-0.0009	0.858504	-0.0132
$\hat{p}_9$	$\hat{p}_{10}$	$\hat{p}_{11}$	$\hat{p}_{12}$	$\hat{p}_{13}$	$\hat{p}_{14}$	
-0.0519	0.0074	-0.3288	0.4523	0.0712	1.5503	
$\hat{\alpha}_{1,1}$	$\hat{\alpha}_{1,2}$	$\hat{\alpha}_{2,1}$	$\hat{\alpha}_{2,2}$	$\hat{\alpha}_{3,1}$	$\hat{\alpha}_{3,2}$	
2.5737	0.7707	2.2274	0.2981	0.9972	0.4717	

TABLE A.6: THE ONLINE ESTIMATED ELEMENTS OF THE BASE PARAMETER SET AND OF FRICTION PARAMETERS IN THE EXPERIMENT WITH THE LAST LINK AND EXTRA MASS ATTACHED

$\hat{p}_1$	$\hat{p}_2$	$\hat{p}_3$	$\hat{p}_5$	$\hat{p}_6$	$\hat{p}_7$	$\hat{p}_8$
0.3016	-0.0419	0.3876	0.0831	0.0026	0.8308	-0.0084
$\hat{p}_9$	$\hat{p}_{10}$	$\hat{p}_{11}$	$\hat{p}_{12}$	$\hat{p}_{13}$	$\hat{p}_{14}$	
-0.0556	0.0045	-0.3157	0.6024	0.1860	1.6692	
$\hat{\alpha}_{1,1}$	$\hat{\alpha}_{1,2}$	$\hat{\alpha}_{2,1}$	$\hat{\alpha}_{2,2}$	$\hat{\alpha}_{3,1}$	$\hat{\alpha}_{3,2}$	
2.6627	0.7383	2.0063	0.3762	0.9851	0.3634	

Bibliography

[1] *MultiQ-PCI Data Acquisition System: user's guide V1.0*, Quanser Consulting, Markham, Ontario, Canada, 2000.

[2] *Optimization Toolbox: User's Guide V2*, The Mathworks, Natick, Massachusetts, 2002.

[3] *Real-Time Workshop: user's guide V5*, The Mathworks, Natick, Massachusetts, 2002.

[4] *WinCon 3.3: user's guide V1.0*, Quanser Consulting, Markham, Ontario, Canada, 2002.

[5] *Simulink Reference V5*, The Mathworks, Natick, Massachusetts, 2003.

[6] W.H.T.M. Aangenent, *Data-based LQG control*, DCT Report 2003.113, Department of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2003.

[7] C.H. An, C.G. Atkeson, and J.M. Hollerbach, *Model-based control of a robot manipulator*, The MIT Press, Cambridge, Massachusetts, 1988.

[8] S. Arimoto, *Control theory of non-linear mechanical systems: A passivity-based circuit theoretic approach*, Oxford University Press, Oxford, England, 1996.

[9] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering Operation of Robots by Learning," *J. of Robotics Systems*, Vol. 1, No. 2, pp. 123-140, 1984.

[10] B. Armstrong-Hélouvry, *Control of machines with friction*, Kluwer, Boston, Massachusetts, 1991.

[11] B. Armstrong-Hélouvry, P. Dupont, and C. Canudas de Wit, "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction," *Automatica*, Vol. 30, No. 7, pp. 1083-1138, 1994.

[12] B. Armstrong, "On Finding Exciting Trajectories for Identification Experiments Involving Systems with Nonlinear Dynamics," *Int. J. Robotics Research*, Vol. 8, No. 6, pp. 28-48, 1989.

[13] H. Asada and K. Youcef-Toumi, *Direct-drive robots: theory and practice*, MIT press, London, England, 1987.

[14] K.J. Åström and T. Hägglund, *PID controllers*, Instruments Society of America, Research Triangle Park, North Carolina, 1995.

[15] G.J. Balas, J.C. Doyle, K. Glover, A. Packard, and R. Smith, *μ-Analysis and Synthesis Toolbox*. The MathWorks, Natick, Massachusetts, 2003.

[16] S. Battilotti and L. Lanari, "Robust Control of Rigid Robots: an H_∞ Approach," *in Proc. of Europ. Control Conf.*, pp. 3462-3467, Rome, Italy, 1995.

[17] D.S. Bayard, "A Forward Method for Optimal Stochastic Nonlinear and Adaptive Control," *IEEE Trans. of. Autom. Control*, Vol. 36, No. 9, pp. 1046-1053, 1991.

[18] J.K. Bennighof, S.H.M. Chang, and M. Subramaniam, "Minimum Time Pulse Response Based Control of Flexible Structures," *J. of Guidance, Control, and Dynamics*, Vol. 16, No. 5, pp. 874-881, 1993.

[19] H. Berghuis and H. Nijmeijer, "Robust control of robots via linear estimated state feedback," *IEEE Trans. of. Autom. Control*, Vol. 39, No. 10, pp. 2159-2162, 1994.

[20] P.R. Bélanger, "Estimation of Angular Velocity and Acceleration from Shaft Encoder Measurements," *in Proc. of IEEE Int. Conf. Rob. Autom.*, pp. 585-592, Nice, France, 1992

[21] P.R. Bélanger, P. Dobrovolny, A. Helmy, and X. Zhang, "Estimation of Angular Velocity and Acceleration from Shaft-Encoder Measurements," *Int. J. Robotics Research*, Vol. 17, No. 11, pp. 1225-1233, 1998.

[22] X. Bombois, M. Gevers, G. Scorletti, and B.D.O. Anderson, “Robustness Analysis for an Uncertainty Set Obtained by Prediction Error Identification,” *Automatica*, Vol. 37, No. 10, pp. 1629-1636, 2001.

[23] P. Bondi, G. Casalino, and L. Gambraldella, “On the Iterative Learning Control Theory for Robot Manipulators,” *IEEE J. of Rob. Autom.*, Vol. 4, No. 1, pp. 14-22, 1988.

[24] W.J. Book, “Recursive Lagrangian Dynamics of Flexible Manipulator Arms,” *Int. J. Robotics Research*, Vol. 3, No. 3, pp. 87-101, 1984.

[25] O.H.Bosgra and H. Kwakernaak, *Design Methods for Control Systems*, Lecture notes for the Dutch Institute of Systems and Control, Winter term 2000-2001.

[26] R.H.Brown, S.C. Schneider, and M.G. Mulligan, “Analysis of Algorithms for Velocity Estimation from Discrete Position Versus Time Data,” *IEEE Trans. of. Ind. Elect.*, Vol. 39, No. 1, pp. 11-19, 1992.

[27] T.F. Brozenec and M.G. Safonov, “Controller Identification,” *in Proc. of Americ. Control Conf.*, pp. 2093-2097, Albuquerque, New Mexico, 1997.

[28] P.B. Brugarolas, V. Fromion, and M.G. Safonov, “Robust Switching Missile Autopilot,” *in Proc. of Americ. Control Conf.*, pp. 3665-3669, Philadelphia, Pennsylvania, 1998.

[29] B.H.M. Bukkems, D. Kostić, B. de Jager, and M. Steinbuch, “Frequency Domain Iterative Learning Control for Direct Drive Robots,” *in CD ROM Proc. of Europ. Control Conf.*, Cambridge, England, 2003.

[30] B.H.M. Bukkems, D. Kostić, B. de Jager, and M. Steinbuch, “Online Identification of a Robot using Batch Adaptive Control,” *in Proc. of 13th IFAC Symposium on System Identification SYSID*, pp. 953-958, Rotterdam, The Netherlands, 2003.

[31] F.B. Cabral and M.G. Safonov, “Fitting Controllers to Data,” *in Proc. of Americ. Control Conf.*, pp. 589-593, Philadelphia, Pennsylvania, 1998.

[32] G. Calafiore, M. Indri, and B. Bona, “Robot Dynamic Calibration: Optimal Excitation Trajectories and Experimental Parameter Estimation,” *J. of Robotics Systems*, Vol. 18, No. 2, pp. 55-68, 2001.

[33] M.C. Campi, A. Lecchini, and S.M. Savaresi, “Virtual Reference Feedback Tuning: A Direct Method for the Design of Feedback Controllers,” *Automatica*, Vol. 38, No. 8, pp. 1337-1346, 2002.

[34] M.C. Campi, A. Lecchini, and S.M. Savaresi, “An Application of the Virtual Reference Feedback Tuning Method to a Benchmark Problem,” *Europ. J. of Control*, Vol. 9, No. 1, pp. 66-76, 2003.

[35] C. Canudas de Wit and P. Lischinsky, “Adaptive Friction Compensation with Partially Known Dynamic Friction Model,” *Int. J. of Adap. Contr. and Sig. Process*, Vol. 11, No. 1, pp. 65-80, 1997.

[36] A. Casavola and E. Mosca, “H_∞ and H_2 Simple Controllers for Robotic Applications,” *Int. J. of Control*, Vol. 55, No. 2, pp. 329-341, 1992.

[37] B.S. Chen and Y.C. Chang, “Nonlinear Mixed H_2/H_∞ Control for Robust Tracking Design of Robotic Systems,” *Int. J. of Control*, Vol. 67, No. 6, pp. 837-857, 1997.

[38] B.S. Chen, T.S. Lee, and J.H. Feng, “A Nonlinear H_∞ Control Design in Robotic Systems Under Parameter Perturbation and External Disturbance,” *Int. J. of Control*, Vol. 59, No. 2, pp. 439-461, 1994.

[39] Y. Chen, “Replacing a PID Controller by a Lag-lead Compensator for a Robot-A Frequency Response Approach,” *IEEE Trans. of Rob. Autom.*, Vol. 5, No. 2, pp. 174-182, 1989.

[40] Y. Chen, Z. Gong, and C. Wen, “Analysis of a High Order Iterative Learning Control Algorithm for Uncertain Nonlinear Systems,” *Automatica*, Vol. 34, No. 3, pp. 345-353, 1998.

[41] P.I. Corke, “A Robotics Toolbox for MATLAB,” *IEEE Rob. Autom. Mag.*, Vol. 3, No. 1, pp. 24-32, 1996.

[42] F. De Bruyne, “Iterative Feedback Tuning for Internal Model Controllers,” *Control Eng. Pract.* Vol. 11, No. 9, pp. 1043-1048, 2003.

[43] F. De Bruyne and L.C. Kammer, “Iterative Feedback Tuning With Guaranteed Stabilty,” *in Proc. of Americ. Control Conf.*, pp. 3317-3321, San Diego, California, 1999.

[44] B. de Jager, “Improving the Tracking Performance of Mechanical Systems by Adaptive Extended Friction Compensation,” *Control Eng. Pract.*, Vol. 1, No. 6, pp. 1009-1018, 1993.

[45] A. De Luca, “Feedforward/Feedback Laws for the Control of Flexible Robots,” *in Proc of IEEE Int. Conf. Rob. Autom.,* pp. 233-240, San Francisco, California, 2000.

[46] A. De Luca, G. Paesano, and G. Ulivi, "A Frequency-Domain Approach to Learning Control: Implementation for a Robot Manipulator," *IEEE Trans. of Ind. Elect.*, Vol. 39, No. 1, pp. 1-10, 1992.

[47] A. De Luca and B. Siciliano, "Flexible Links," in Canudas de Wit, C. Siciliano, B. and Bastin, G. (eds.) *Theory of robot control*, Springer-Verlag, Berlin, Germany, 1996.

[48] A. De Luca and P. Tomei, "Elastic Joints," in Canudas de Wit, C., Siciliano, B., and Bastin, G. (eds.) *Theory of robot control*, Springer-Verlag, Berlin, Germany, 1996.

[49] J.C. Doyle, K. Glover, P.P. Khargonekar, and B.A. Francis, "State-space Solutions to Standard H_2 and H_∞ Control Problems," *IEEE Trans. of Autom. Control*, Vol. 34, No. 8, pp. 831-847, 1989.

[50] K. El-Awady, A. Hansson, and B. Wahlberg, "Application of Iterative Feedback Tuning to a Thermal Cycling Module," *in Proc. of 14th World Congress of IFAC*, pp.438-444, Beijing, China, 1999.

[51] W. Feng and I. Postlethwaite, "Robust Nonlinear H_∞/adaptive Control of Robot Manipulator Motion," *in Proc. of IFAC Triennial World Congress*, pp. 31-34, Sydney, Australia, 1993.

[52] G.F. Franklin, J.D.Powell, and A. Emami-Naeini, *Feedback control of dynamic systems*, Prentice-Hall, London, England, 2001.

[53] K.S. Fu, R.C. Gonzales, and C.S.G. Lee, *Robotics: control, sensing, vision, and intelligence*, McGraw-Hill, London, England, 1987.

[54] K. Furuta, M. Wongsisuwan, and H. Werner, "Dynamic Compensator Design for Discrete-time LQG Problem Using Markov Parameters," *in Proc. of IEEE Conf. Dec. Control,* pp. 96-101, San Antonio, Texas, 1993.

[55] W. Gao and J. C. Hung, "Variable Structure Control of Nonlinear Systems: A New Approach," *IEEE Trans. of Ind. Elect.*, Vol. 40, No. 1, pp. 45-55, 1993.

[56] W. Gao, Y. Wang, and A. Homaifa, "Discrete-time Variable Structure Control Systems," *IEEE Trans. of. Ind. Elect.*, Vol. 42, No. 2, pp. 117-122, 1995.

[57] M. Gautier and W. Khalil, "On the Identification of the Inertial Parameters of Robots," *in Proc. of IEEE Conf. Dec. Control*, pp. 2264-2269, Austin, Texas, 1988.

[58] M. Gautier and W. Khalil, "Exciting Trajectories for the Identification of Base Inertial Parameters of Robots," *Int. J. Robotics Research*, Vol. 11, No. 4, pp. 362-375, 1992.

[59] A. Gelb, *Applied Optimal Estimation*, The M.I.T. Press, London, England, 1996.

[60] G. Golo and Č. Milosavljevic, "Two-phase Triangular Wave Oscillator Based on Discrete-time Sliding Mode Control," *Electronic Letters*, Vol. 32, No. 2, pp. 117-122, 1997.

[61] G. Golo and Č. Milosavljevic, "Robust Discrete-time Chattering Free Sliding Mode Control," *Syst. & Control Letters*, Vol. 41, No. 1, pp. 19-28, 2000.

[62] G.H. Golub and C.F. van Loan, *Matrix computations*, John Hopkins Univ. Press, London, England, 1996.

[63] D.M. Gorinevski, A.V. Lensky, and E.I. Savbitov, "Feedback Control of a One-link Flexible Manipulator With a Gear Train," *J. of Robotics Systems*, Vol. 8, No. 5, pp. 659-676, 1991.

[64] D. Gorinevsky, D. Torfs, and A.A. Goldenberg, "Learning Approximation of Feedforward Dependence on the Task Parameters: Experiments in Direct-drive Manipulator Tracking," *in Proc. of Americ. Control Conf.*, pp. 883-887, Seattle, Washington, 1995.

[65] G.O. Guardabassi and S.M. Savaresi, "Virtual Reference Direct Design Method: An Off-line Approach to Data-based Control System Design," *IEEE Trans. of Autom. Control*, Vol. 45, No. 5, pp. 954-959, 2000.

[66] G.O. Guardabassi and S.M. Savaresi, "Approximate Linearization via Feedback," *Automatica*, Vol. 37, No. 1, pp. 1-15, 2001.

[67] K. Guglielmo and N. Sadegh, "Theory and Implementation of a Repetitive Robot Controller with Cartesian Trajectory Description," *J. of Dyn. System Meas. and Control*, Vol. 118, No. 1, pp. 15-21, 1996.

[68] S. Gunnarsson, V. Collignon, and O. Rousseaux, "Tuning of a Decoupling Controller for a 2×2 System Using Iterative Feedback Tuning," *Control Eng. Pract.*, Vol. 11, No. 9, pp. 1035-1041, 2003.

[69] S. Gunnarsson, O. Rousseaux, and V. Collignon, "Iterative Feedback Tuning Applied to Robot Joint Controllers," *in Proc. of 14th World Congress of IFAC*, pp. 438-444, Beijing, China, 1999

[70] K. Hamamoto, T. Fukuda, and T. Sugie, "Iterative Feedback Tuning of Controllers for a Two-mass-spring System With Friction," *Control Eng. Pract.*, Vol. 11, No. 9, pp. 1061-1068, 2003.

[71] K. Hamamoto and T. Sugie, "Iterative Learning Control for Robot Manipulators Using The Finite Dimensional Input Subspace," *IEEE Trans. of Rob. Autom.*, Vol. 18, No. 4, pp. 632-635, 2002.

[72] R.H.A. Hensen, G.Z. Angelis, M.J.G. van de Molengraft, A.G. de Jager, and J.J. Kok, "Grey-box Modeling of Friction: An Experimental Case Study," *Europ. J. of Control*, Vol. 6, No. 3, pp. 258-267, 2000.

[73] R.H.A. Hensen, M.J.G. van de Molengraft, and M. Steinbuch, "Frequency Domain Identification of Dynamic Friction Model Parameters," *IEEE Trans. of Control Sys. Technology*, Vol. 10, No. 2, pp. 191-196, 2002.

[74] H. Hjalmarsson, "Efficient Tuning of Linear Multivariable Controllers Using Iterative Feedback Tuning," *Int. J. of Adap. Contr. and Sig. Process.*, Vol. 13, No. 7, pp. 553-572, 1999.

[75] H. Hjalmarsson, "Iterative Feedback Tuning - An Overview," *Int. J. of Adap. Contr. and Sig. Process.*, Vol. 16, No. 5, pp. 373-395, 2002.

[76] H. Hjalmarsson and M.T. Cameron, "Iterative Feedback Tuning of Controllers in Cold Rolling Mills," *in Proc. of 14th World Congress of IFAC*, pp. 445-450, Beijing, China, 1999.

[77] H. Hjalmarsson, M. Gevers, S. Gunnarsson, and O. Lequin, "Iterative Feedback Tuning: Theory and Applications," *IEEE Control Sys. Mag.*, Vol. 18, No. 4, pp. 26-41, 1998.

[78] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, "A Convergent Iterative Restricted Complexity Control Design Scheme," *in Proc. of IEEE Conf. Dec. Control*, pp. 1735-1740, Lake Buena Vista, Florida, 1994.

[79] H. Hjalmarsson and S.M. Veres, "Robust Loopshaping Using Iterative Feedback Tuning," *in Proc. of Europ. Control Conf.*, pp. 2046-2051, Porte, Portugal, 2001.

[80] J. M. Hollerbach, "Understanding Manipulator Control by Synthesizing Human Handwriting," pp. 311-332, in *Artificial Intelligence: an MIT Perspective*, MIT Press, Cambridge, Massachusetts, 1979.

[81] J. M. Hollerbach, "An Oscillation Theory of Handwriting," *Biological Cybernetics*, Vol. 39, pp. 139-156, 1981.

[82] R. Horowitz, "Learning Control of Robot Manipulators," *J. of Dyn. System., Meas. and Control*, Vol. 115, pp. 402-411, 1993.

[83] R. Horowitz, W. Messner, and J. B. Moore, "Exponential Convergence of a Learning Controller for Robot Manipulators," *IEEE Trans. of Autom. Control*, Vol. 36, No. 7, pp. 890-894, 1991.

[84] J.Y. Hung, W. Gao, and J.C. Hung, "Variable Structure Control: A Survey," *IEEE Trans. of Ind. Elect.*, Vol. 40, No. 1, pp. 2-22, 1993.

[85] J. B. Jonker, "A Finite Element Dynamic Analysis of Flexible Manipulators," *Int. J. Robotics Research*, Vol. 9, No. 4, pp. 59-74, 1990.

[86] M. Jun and M.G. Safonov, "Automatic PID Tuning: An Application of Unfalsified Control," *Proc. of Int. Symp. Comp. Aided Control Sys. Des.*, pp. 328-333, Kohala Coast-Island of Hawai'i, Hawai'i, 1999.

[87] L.C. Kammer, F.D. De Bruyne, and R.R. Bitmead, "Iterative Feedback Tuning via Minimization of the Absolute Error," *in Proc. of IEEE Conf. Dec. Control*, pp. 4619-4624, Phoenix, Arizona, 1999.

[88] A. Karimi, L. Mikovi, and D. Bonvin, "Iterative Correlation-based Controller Tuning with Application to a Magnetic Suspension System," *Control Eng. Pract.*, Vol. 11, No. 9, pp. 1069-1078, 2003.

[89] S.J. Kim and I.J. Ha, "A Frequency-Domain Approach to Identification of Mechanical Systems with Friction," *IEEE Trans. of Autom. Control*, Vol. 46, No. 6, pp. 888-893, 2001.

[90] M. Koot, *Identification and control of the RRR-robot*, DCT Report No. 2001.48, Department of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2001.

[91] M. Koot, D. Kostić, B. de Jager, and M. Steinbuch, "A Systematic Approach to Improve Robot Motion Control," *in Proc. of Conf. on Mechantronics*, pp. 1377-1386, Enschede, The Netherlands, 2002.

[92] D. Kostić, B. de Jager, and M. Steinbuch, "Experimentally Supported Control Design for a Direct Drive Robot," *in Proc. of IEEE Int. Conf. Rob. Autom.*, pp. 186-191, Glasgow, Scotland, 2002.

[93] D. Kostić, B. de Jager, and M. Steinbuch, "Robust Attenuation of Direct-Drive Robot-Tip Vibrations," *in Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 2206-2212, Lausanne, Switzerland, 2002.

[94] D. Kostić, B. de Jager, and M. Steinbuch, "Control Design for Robust Performance of a Direct-drive Robot," *in Proc. of IEEE Int. Conf. Control Appl.*, pp. 1448-1453, Istanbul, Turkey, 2003.

[95] D. Kostić, B. de Jager, and M. Steinbuch, "Data-based Design of High-performance Motion Controllers," *accepted for Americ. Control Conf.*, Boston, Massachusetts, 2004.

[96] D. Kostić, B. de Jager, M. Steinbuch, and R. Hensen, "Modeling and Identification for High-performance Robot Control: An RRR-Robotic Arm Case Study," *IEEE Trans. of Control Sys. Technology*, Vol. to appear 2004.

[97] D. Kostić, R. Hensen, B. de Jager, and M. Steinbuch, "Modeling and Identification of an RRR-robot," *in Proc. of IEEE Conf. Dec. Control,* pp. 1144-1149, Orlando, Florida, 2001.

[98] D. Kostić, R. Hensen, B. de Jager, and M. Steinbuch, "Closed-form Kinematic and Dynamic Models of an Industrial-like RRR Robot," *in Proc. of IEEE Int. Conf. Rob. Autom.*, pp. 1309-1314, Washington D.C.,2002.

[99] R.L. Kosut, "Unfalsified Nonlinear Adaptive Control," *in Proc. of IEEE Conf. Dec. Control,* pp. 1243-1247, Sydney, Australia, 2000.

[100] K. Kozlowski, *Modeling and identification in robotics*, Springer, London, England, 1998.

[101] K.R. Kozlowski and P. Dutkiewicz, "Experimental Identification of Robot and Load Dynamics," *in Proc. of 13th Triennial IFAC World Congress*, pp. 397-402, San Francisco, California, 1996.

[102] I. Kruise, "*Modelling and Control of a Flexible Beam and Robot Arm,*" PhD Thesis, University of Twente, Enschede, The Netherlands, 1990.

[103] H. Kwakernaak, "Robust Control and H_∞-optimization-Tutorial Paper," *Automatica*, Vol. 29, No. 2, pp. 255-273, 1993.

[104] H. Kwakernaak and R. Sivan, *Modern signals and systems*, Englewood Cliffs, Prentice-Hall, New Jersey, 1991.

[105] V. Lampaert, J. Swevers, and F. Al-Bender, "Modification of the Leuven Integrated Friction Model Structure," *IEEE Trans. of Autom. Control*, Vol. 47, No. 4, pp. 683-687, 2002.

[106] F. Lange and G. Hirzinger, "Learning Accurate Path Control of Industrial Robots with Joint Elasticity," *in Proc. of IEEE Int. Conf. Rob. Autom.*, pp. 2084-2089, Detroit, Michigan, 1999.

[107] A. Lecchini, "Virtual Reference Feedback Tuning: a new direct data-based method for the design of feedback controllers," Universita di Brescia, Italy, 2001.

[108] A. Lecchini, M.C. Campi, and S.M. Savaresi, "Virtual Reference Feedback Tuning for Two Degree of Freedom Controllers," *Int .J. of Adap. Contr. and Sig. Process.*, Vol. 16, No. 5, pp. 355-371, 2002.

[109] D.J. Leith and W.E. Leithead, "Survey of Gain-scheduling Analysis and Design," *Int. J .of Control*, Vol. 73, No. 11, pp. 1001-1025, 2000.

[110] O. Lequin, M. Gevers, M. Mossberg, E. Bosmans, and L. Triest, "Iterative Feedback Tuning of PID Parameters: Comparison With Classical Tuning Rules," *Control Eng. Pract.*, Vol. 11, No. 9, pp. 1023-1033, 2003.

[111] O. Lequin, M. Gevers, and L. Triest, "Optimizing the Settling Time With Iterative Feedback Tuning," *in Proc. of 14th World Congress of IFAC*, pp. 433-437, Beijing, China, 2004.

[112] S.K. Lin, "Minimal Linear Combinations of the Inertial Parameters of a Manipulator," *IEEE Trans. of Rob. Autom.*, Vol. 11 , No. 3, pp. 360-373, 1995.

[113] L. Ljung, *System Identification: theory for the user,* Upper Saddle River, Prentice-Hall, 1999.

[114] S. López-Linares, A. Konno, and M. Uchiyama, "Vibration Suppression Control of 3D Flexible Robots Using Velocity Inputs," *J. of Robotics Systems* Vol. 14[12], pp. 823-837, 1997.

[115] Z.H. Luo and D.X. Feng, "Nonlinear Torque Control of a Single-Link Flexible Robot," *J. of Robotics Systems*, Vol. 16, No. 1, pp. 25-35, 1999.

[116] S.E. Lyshevski and M.G. Safonov, "Intelligent Motion Control for Electromechanical Servos Using Evolutionary Learning and Adaptation Mechanisms," *in Proc. of Americ. Control Conf.*, pp. 2840-2845, Arlington, Virginia, 2001.

[117] Y. Maeda and R.J.P. De Figueiredo, "Learning Rules for Neuro-controller via Simultaneous Perturbation," *IEEE Trans. of on Neural Networks*, Vol. 8, No. 5, pp. 1119-1130, 1997.

[118] Y. Maeda, K. Kawaguchi, K. Kumon, and T. Inoue, "Neuro-controller Using Simultaneous Perturbation for a Flexible Arm System," *in Proc. of 26th Ann. IEEE Conf. of Indust. Elect. Soc.*, Vol. 1, pp. 22-28, 2000.

[119] Y. Maeda, "Real-time Control and Learning Using Neuro-controller via Simultaneous Perturbation for Flexible Arm System," *in Proc. of Americ. Control Conf.*, Vol. 4, pp. 2583-2588, Anchorage, Alaska, 2002.

[120] F. Matsuno and Y. Sakawa, "A Simple Model of Flexible Manipulators with Six Axes and Vibration Control by Using Accelerometers," *J. of Robotics Systems*, Vol. 7[4], pp. 575-597, 1990.

[121] H. Mayeda, K. Yoshida, and K. Osuka, "Base Parameters of Manipulator Dynamic Models," *IEEE Trans. of Rob. Autom.*, Vol. 6, No. 3, pp. 312-321, 1990.

[122] G.K. McMillan, *Tuning and control loop performance,* Research Triangle Park: Instruments Society of America, 1983.

[123] Č. Milosavljević, "General Conditions for the Existence of a Quasi-sliding Mode on the Switching Hyperplane in Discrete Variable Structure Systems," *Autom. Remote Control*, Vol. 46, pp. 307-314, 1985.

[124] T. Mita and E. Kato, "Iterative Control and its Application to Motion Control of Robot Arm - A Direct Approach to Servo-Problems," *in Proc. of IEEE Conf. Dec. Control,* pp. 1393-1398, Ft. Lauderdale, Florida, 1985.

[125] K.L. Moore, "Iterative Learning Control - An Expository Overview," *Applied and Computational Controls, Signal Processing, and Circuits*, Vol. 1, No. 1, pp. 151-214, 1998.

[126] K.L. Moore, M. Dahleh, and S.P. Bhattacharyya, "Iterative Learning Control: A Survey and New Results," *J. of Robotics Systems*, Vol. 9, No. 5, pp. 563-594, 1992.

[127] T. Namerikawa, F. Matsumura, and M. Fujita, "Robust Trajectory Following for an Uncertain Robot Manipulator Using H_∞ Synthesis," *in Proc. of Europ. Control Conf.*, pp. 3474-3479, Rome, Italy, 1995.

[128] J. Nethery and M.W. Spong, "Robotica: A Mathematica Package for Robot Analysis," *IEEE Rob. Autom. Mag.*, Vol. 1, No. 1, pp. 13-20, 1994.

[129] H. Nijmeijer and A. Van der Schaft, *Nonlinear dynamical control systems,* Berlin, Springer, 1991.

[130] M. Norrlöf, *Iterative Learning Control: Analysis, Design, and Experiments,* Department of Electrical Engineering, Linköpings Universiteit, Linköping, Sweden, 2000.

[131] M. Norrlöf and S. Gunnarsson, "Experimental Comparison of Some Classical Iterative Learning Control Algorithms," *IEEE Trans. of Rob. Autom.*, Vol. 18, No. 4, pp. 636-641, 2002.

[132] K. Ohishi, "Manipulator Motion Control Based on a Robust Acceleration Controller," *Advanced Robotics*, Vol. 7, No. 6, pp. 547-557, 1993.

[133] M. Olsen and H.G. Petersen, "A New Method for Estimating Parameters of a Dynamic Robot Model," *IEEE Trans. of Rob. Autom.*, Vol. 17, No. 1, pp. 95-100, 2001.

[134] H. Olsson, K.J. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky, "Friction Models and Friction Compensation," *Europ. J. of Control*, Vol. 4, No. 3, pp. 176-195, 1998.

[135] H. Onishi and Y. Maeda, "Tracking Control for Robot Arm Using Neural Network with Simultaneous Perturbation Learning Rule," *in Proc. of 41st SICE Annual Conference,* Vol.5, pp. 3188-3191, Osaka, Japan, 2002.

[136] T.A.E. Oomen. *The RRR-robot feedback control design via iterative feedback tuning,* DCT Report 2003.94, Department of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 2003.

[137] R. Ortega and M.W. Spong, "Adaptive Motion Control of Rigid Robots: A Tutorial," *Automatica*, Vol. 25, No. 6, pp. 877-888, 1989.

[138] B. Paden and R. Panja, "A Globally Asymptotically Stable 'PD+' Controller for Robot Manipulators," *Int. J. of Control*, Vol. 47, No. 6, pp. 1697-1712, 1988.

[139] M.Q. Phan, R.K. Lim, and R.W. Longman, *Unifying Input-output and State-space Perspectives of Predictive Control*, Technical report No. 3044, Princeton University, Princeton, New Jersey, 1998.

[140] R. Pintelon and J. Schoukens, *System identification: a frequency domain approach,* IEEE Press, Piscataway, New Jersey, 2001.

[141] R. Pintelon, P. Guillaume, Y. Rolain, J. Schoukens, and H. Van Hamme, "Parametric Identification of Transfer Functions in the Frequency Domain-A Survey," *IEEE Trans. of Autom. Control*, Vol. 39, No. 11, pp. 2245-2260, 1994.

[142] V. Potkonjak, "Contribution to the Dynamics and Control of Robots Having Elastic Transmissions," *Robotica*, Vol. 6, No. 1, pp. 63-69, 1988.

[143] V. Potkonjak, M. Popović, M. Lazarević, and J. Sinanović, "Redundancy Problem in Writing: From Human to Anthropomorphic Robot Arm," *IEEE Trans. of Sys. Man Cyb., Part B: Cybernetics*, Vol. 28, No. 6, pp. 790-805, 1998.

[144] V. Potkonjak, S. Tzafestas, D. Kostić, G. Djordjević, and M. Rašić, "Illustrating Man-machine Motion Analogy in Robotics - The Handwriting Problem," *IEEE Rob. Autom. Mag.*, Vol. 10, No. 1, pp. 35-46, 2003.

[145] L.R. Ray, A. Ramasubramanian, and J. Townsend, "Adaptive Friction Compensation Using Extended Kalman-Bucy Filter Friction Compensation," *Control Eng. Pract.*, Vol. 9, No. 2, pp. 169-179, 2001.

[146] H.A. Razavi and T.R. Kurfess, "Real Time Force Control of a Grinding Process Using Unfalsification and Learning Concept," *in Proc. of IEEE Conf. Dec. Control*, pp. 2682-2685,. Orlando, Florida, 2000.

[147] M.C. Readman, *Flexible Joint Robots,* CRC Press, Boca Raton, Florida, 1994.

[148] M.G. Safonov and T.C. Tsao, "The Unfalsified Control Concept and Learning," *IEEE Trans. of Autom. Control*, Vol. 42, No. 6, pp. 843-847, 1997.

[149] H.G. Sage, M.F. de Mathelin, G. Abba, J.A. Gangloff, and E. Ostertag, "Nonlinear Optimization of Robust H_∞ Controllers for Industrial Robot Manipulators," *in Proc. of IEEE Int. Conf. Rob. Autom., pp.* 2352-2357, Albuquerque, New Mexico, 1997.

[150] H.G. Sage, M.F. de Mathelin, and E. Ostertag, "Robust Control of Robot Manipulators: A Survey," *Int. J. of Control*, Vol. 72, No. 16, pp. 1498-1522, 1999.

[151] C.K. Sanathanan and J. Koerner, "Transfer function synthesis as a ratio of two complex polynomials," *IEEE Trans. of Autom. Control*, Vol. 8, No. 1, pp. 56-58, 1963.

[152] E. Schrijver, *Improved Robot Tracking Control for Laser Welding*, Drebbel Institute for Mechatronics, University of Twente, Enschede, The Netherlands, 2002.

[153] L. Sciavicco and B. Siciliano, *Modeling and control of robot manipulators,* McGraw-Hill, London, England, 1996.

[154] M. Simaan and P.L. Love, "Knowledge-based Detection of Out-of-control Outputs in Process Control," Vol. 1, pp. 128-129, *in Proc. of IEEE Conf. Dec. Control*, Honolulu, Hawaii, 1990.

[155] R.E. Skelton and G. Shi, "The Data-Based LQG Control Problem," *in Proc. of IEEE Conf. Dec. Control,* pp. 1447-1452, Lake Buena Vista, Florida, 1994.

[156] R.E. Skelton and G. Shi, "Markov Data-Based LQG Control," *J. of Dyn. System., Meas. and Control*, Vol. 122, No. 3, pp. 551-559, 2000.

[157] J.J.E. Slotine and W. Li, "Theoretical issues in adaptive manipulator control," *in Proc. of 5th Yale Workshop on Application of Adaptive Systems*, pp. 252-258, New Haven, USA, 1987.

[158] J.J.E. Slotine and W. Li, *Applied nonlinear control,* Prentice-Hall, Upper Saddle River, New Jersey, 1991.

[159] Y.D. Song, A.T. Alouani, and J.N. Anderson, "Robust Path Tracking Control of Industrial Robots: An H_∞ Approach," *in Proc. of IEEE Int. Conf. Control Appl.*, pp. 25-30, Dayton, Ohio, 1992.

[160] J.C. Spall, "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," *IEEE Trans. of Autom. Control*, Vol. 37, No. 3, pp. 332-341, 1992.

[161] J.C. Spall, "Stochastic Search, Optimization, and the Simultaneous Perturbation Algorithm - A tutorial," *in Proc. of Int. Conf. on Integration of Knowledge Intensive Multi-Agent Systems,* pp. 676-677, Cambridge, Massachusetts, USA, 2003.

[162] J.C. Spall and J.A. Cristion, "A Neural Network Controller for Systems with Unmodeled Dynamics with Applications to Wastew ater Treatment," *IEEE Trans. of Sys., Man, Cyb., Part B: Cybernetics*, Vol. 27, No. 3, pp. 369-375, 1997.

[163] J.C. Spall and J.A. Cristion, "Model-free Control of Nonlinear Stochastic Systems with Discrete-time Measurements," *IEEE Trans. of Autom. Control*, Vol. 43, No. 9, pp. 1198-1210, 1998.

[164] J.C. Spall, "Adaptive Stochastic Approximation by the Simultaneous Perturbation Method," *IEEE Trans. of Autom. Control*, Vol. 45, No. 10, pp. 1839-1853, 2000.

[165] M.W. Spong, F.L. Lewis, and C.T. Abdallah, *Robot control: dynamics, motion planning, and analysis,* IEEE Press, Piscataway, New Jersey, 1993.

[166] M. Steinbuch and M.L. Norg, "Advanced motion control," *Europ. J. of Control*, Vol. 4, No. 4, pp. 278-293, 1998.

[167] W.L. Stout and M.E. Sawan, "Application of H-infinity Theory to Robot Manipulator Control," *in Proc. of IEEE Int. Conf. Control Appl.*, pp. 148-153, Dayton, Ohio, 1992.

[168] D. Šurdilović and M. Vukobratović, "One Method for Efficient Dynamic Modeling of Flexible Manipulators," *Mech. and Mach. Theory*, Vol. 31, No. 3, pp. 297-315, 1996.

[169] J. Swevers, F. Al-Bender, C.G. Ganseman, and T. Prajogo, "An Integrated Friction Model Structure with Improved Presliding Behavior for Accurate Friction Compensation," *IEEE Trans. of Autom. Control*, Vol. 45, No. 4, pp. 675-686, 2000.

[170] J. Swevers, C. Ganseman, D.B. Tukel, J. de Schutter, and H. van Brussel, "Optimal Robot Excitation and Identification," *IEEE Trans. of Rob. Autom.*, Vol. 13, No. 5, pp. 730-740, 1997.

[171] P. Tomei, "Nonlinear H_∞ Disturbance Attenuation for Robots With Flexible Joints," *Int. J. Robust and Nonlinear Control*, Vol. 5, No. 4, pp. 365-373, 1995.

[172] M. Tomizuka, "Zero Phase Error Tracking Algorithm for Digital Control," *J. of Dyn. System., Meas. and Control*, Vol. 109 , No. 65, pp. 68, 1987.

[173] R.L. Tousain, J.C. Boissy, M.L. Norg, M. Steinbuch, and O.H. Bosgra, "Suppressing Non-periodically Repeating Disturbances in Mechanical Servo Systems," *in Proc. of IEEE Conf. Dec. Control*, pp. 2541-2542, Tampa, Florida, 1998.

[174] T.C. Tsao and M.G. Safonov, "Unfalsified Direct Adaptive Control of a Two-link Robot Arm," *Int. J. of Adap. Contr. and Sig. Process*, Vol. 15, No. 3, pp. 319-334, 2001.

[175] V.I. Utkin, *Sliding modes in control optimization,* Springer-Verlag, Berlin, Germany, 1992.

[176] A.M. van Beek, *Design of an Industrial-like Test Facility for Nonlinear Robot Control,* WFW report 98.014, Faculty of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands, 1998.

[177] A.M. van Beek, *RRR-robot: instruction manual*, WFW report 98.012, Faculty of Mechanical Engineering, Technische Universiteit Eindhoven, Eindhoven, The Netherlands,1998.

[178] B. van Beek and B. de Jager, "An Experimental Facility for Nonlinear Robot Control," *in Proc. of IEEE Int. Conf. Control Appl.*, pp. 668-673, Hawai'i, USA, 1999.

[179] B. van Beek and B. de Jager, "RRR-Robot Design: Basic Outlines, Servo Sizing, and Control," *in Proc. of IEEE Int. Conf. Control Appl.*, pp. 36-41, Hartford, USA. 1997.

[180] S. Veres and H. Hjalmarsson, "Tuning for Robustness and Performance Using Iterative Feedback Tuning," *in Proc. of IEEE Conf. Dec. Control,* pp. 4682-4687, Las Vegas, Nevada, 2002.

[181] M. Vukobratović and V. Potkonjak, *Dynamics of manipulation robots: theory and application,* Berlin: Springer-Verlag, 1982.

[182] B.R. Woodley, J.P. How, and R.L. Kosut, "Direct Unfalsified Controller Design - Solution Via Convex Optimization," *in Proc. of Americ. Cont. Conf.*, pp. 3302-3306, 1999. San Diego, California, 1999.

[183] M. Yasuda, T. Osaka, and M. Ikeda, "Feedforward Control of a Vibration Isolation System for Disturbance Suppression," *in Proc. of IEEE Conf. Dec. Control,* Vol. 2, pp. 1229-1233, Kobe, Japan, 1996.

[184] S. Yurkovich, A.P. Tzes, I. Lee, and K.L. Hillsley, "Control and System Identification of a Two-Link Flexible Manipulator," *in Proc. of IEEE Int. Conf. Rob. Autom.*, pp. 1626-1631, Cincinnati, Ohio,1990.

[185] K. Zhou, J.C. Doyle, and K. Glover, *Robust and Optimal Control,* Upper Saddle River: Prentice Hall, 1996.

[186] J.G. Ziegler and N.B. Nichols, "Optimum Settings for Automatic Controllers," *Trans. of the ASME,* Vol. 64, pp. 759-768, 1942.

Summary

Data-driven robot motion control design

There is a lasting challenge to increase the velocity of robot motions, while preserving a prescribed accuracy of robot-tip positioning or executing the desired tip trajectory. As the requirements for the velocity and accuracy are increasing, robustness against uncertainty in the robot dynamics and against disturbances must be maintained. High demands on control performance and robustness have led to the development of various motion control approaches.

Advanced control makes use of a robot dynamic model. This model is implemented in the computed-torque control law to compensate for nonlinear dynamic couplings between the robot axes. The robot dynamics not covered with this compensation are the responsibility of feedforward/feedback controllers. These controllers can be designed in a number of ways.

This thesis examines various feedforward/feedback designs of robot motion controllers that are used with the computed-torque control law. The common feature of all these designs is close accounting for the actual dynamics of a robot under control, namely, they allow the specific dynamics of the given robot to influence the form of the resulting controller. By incorporating these specific dynamics into the control design, one can design a controller which is able to compensate for more versatile plant behaviours, which in turn leads to performance improvement. Treating the given robot as a physical system with individual properties is the key reasoning in the thesis. This reasoning is the major departure from the dominant concept in robotics to perform control design just based on general models of the robot dynamics. General models cannot represent the full complexity of the true dynamics, and, consequently, their practical effects on the motion performance are limited. The control designs treated in this thesis tend to reduce conservatism of the dominant robotics control design strategies.

The quality of the model implemented in the computed-torque control law and the quality of the accompanying feedforward/feedback design both influence the performance of robot motion control. Sufficient quality of the model is achieved by comprehensive modelling of the robot kinematics and rigid-body dynamics with friction, as well as by sophisticated estimation of the model parameters. Both off-line and on-line approaches to parameter estimation are investigated. Control performance improvement via model-based compensation of nonlinearities is experimentally verified on a direct-drive robotic set-up.

Control designs supplementing the model-based compensation of nonlinearities can be divided into indirect and direct ones. Indirect designs utilize parametric representations of additional robot dynamics not covered by the nominal compensation. Suitable parametric representations are obtained by black-box identification of these additional dynamics, which is the strategy also known as data-driven modelling. Indirect designs in both joint and task spaces of the robot are studied. In the joint space: 1) design of conventional PD (proportional, derivative) feedback controllers, 2) H_∞ feedback design within disturbance-based control design cycles, 3) feedback design for robust performance using μ-synthesis, 4) design of a chattering-free discrete-time sliding-mode feedback controller. A problem of active vibration suppression is studied in the task space. These vibrations arise at the tip of the robot, and are caused by robot structural flexibility. The design of a robust vibration compensator is presented.

Direct control designs belong to the emerging field of data-based control. Coarsely speaking, data-based controllers are designed without explicitly making use of parametric models, but are merely based on input/output signals observed from the plant. Several data-based techniques are investigated: 1) iterative learning control design, 3) control design via iterative feedback tuning, and 3) data-based shaping of feedback controllers for the desired performance.

The control performances achieved using different control strategies are experimentally compared with respect to the design complexity, motion accuracy, and robustness against parasitic dynamics and disturbance effects. Guidelines for selection of a design, suitable for a given robot motion control problem, are proposed.

Samenvatting

Data-gestuurd regelen van robotbewegingen

Het blijft een uitdaging om de snelheid van robotbewegingen te verhogen, met behoud van de voorgeschreven positienauwkeurigheid of de uitvoering van een voorgeschreven baan van de punt van de robot. Terwijl de eisen voor snelheid en nauwkeurigheid strenger worden, moet de robuustheid tegen verstoringen en onzekerheden in de robot dynamica gehandhaafd blijven. Strenge eisen aan regelprestaties en robuustheid hebben tot de ontwikkeling van verschillende regelstrategieën geleid.

Geavanceerde regelaars maken gebruik van een dynamisch model van de robot. Dit model wordt geïmplementeerd in de computed-torque regelwet ter compensatie van niet-lineaire dynamische koppelingen tussen de robot assen. De robot dynamica die niet gecompenseerd wordt door deze regelwet, valt onder de verantwoordelijkheid van voorwaartse koppelingen en terugkoppelingen. Deze laatstgenoemde regelaars kunnen op verschillende manieren worden ontworpen.

In dit proefschrift zijn verschillende voorwaartse koppelingen en terugkoppelingen voor robotbewegingen onderzocht, die gebruikt worden in combinatie met de computed-torque regelwet. De gemeenschappelijke eigenschap van deze ontwerpen is dat rekening gehouden is met de werkelijke dynamica van een geregelde robot. Ze zorgen er namelijk voor dat de specifieke dynamica van een bepaalde robot van invloed is op de vorm van de resulterende regelaar. Door deze specifieke dynamica mee te nemen in het ontwerp van de regelaar, kan men een regelaar ontwerpen die in staat is om voor een veelzijdiger machinegedrag te compenseren, hetgeen leidt tot een verbetering van de regelprestaties. De belangrijkste strategie in dit proefschrift is het feit dat de gegeven robot beschreven is als een fysiek systeem met individuele eigenschappen. Met deze methode wordt afstand gedaan van het gebruikelijke concept in de robotica, waarin het regelaarontwerp slechts gebaseerd is op algemene modellen van de robot dynamica. Deze algemene modellen zijn niet in staat om de volledige complexiteit van de werkelijke dynamica te beschrijven, waardoor hun praktische effect op de prestaties van de beweging beperkt is. De behandelde regelaarontwerpen zijn er op gericht, het conservatisme van de dominante regelstrategieën in de robotica te verminderen.

Zowel de kwaliteit van het geïmplementeerde model in de computed-torque regelwet als de kwaliteit van de bijbehorende voorwaartse koppeling en terugkoppeling, zijn van invloed op de prestaties van de regelaar. Een model van voldoende kwaliteit wordt gerealiseerd door zowel een uitgebreide modellering van de kinematica van de robot en rigid-body dynamica met wrijving, als door een verfijnde schatting van de modelparameters. Er zijn methoden onderzocht om zowel off-line als on-line parameters te schatten. Het verbeteren van de regelprestaties, door op basis van een model de niet-lineariteiten te compenseren, is experimenteel geverifieerd op een direct aangestuurde robot opstelling.

Regelaarontwerpen die model gebaseerde compensatie van niet-lineariteiten aanvullen, kunnen verdeeld worden in indirecte en directe ontwerpen. Indirecte ontwerpen maken gebruik van een parametrische representatie van de resterende dynamica van de robot, die niet gecompenseerd wordt door de computed-torque regelwet. Bruikbare parametrische representaties zijn verkregen door een black-box identificatie van deze dynamica. Deze strategie staat bekend als datagestuurde modellering. Indirecte ontwerpen in zowel de configuratieruimte als de taakruimten van de robot zijn onderzocht. In de configuratieruimte: 1) het ontwerp van conventionele PD (proportioneel, differentiërend) terugkoppelingen, 2) het ontwerp van H_∞ terugkoppelingen binnen de cycli van verstoringsgebaseerd regelaarontwerp, 3) ontwerp van een terugkoppeling voor robuuste prestaties met behulp van μ-synthese, 4) ontwerp van trillingsvrije discrete tijd schakelvlak regelaars. Een probleem betreffende actieve trillingsonderdrukking in de taakruimte is bestudeerd. Deze trillingen ontstaan bij de punt van de robot en worden veroorzaakt door flexibiliteiten in de robot. Het ontwerp van een robuuste trillingscompensator is gepresenteerd.

Directe regelaarontwerpen behoren tot het opkomende gebied van data-gebaseerd regelen. Deze regelaars worden ontworpen zonder expliciet gebruik te maken van parametrische modellen, alleen ingangs- en uitgangssignalen van het betreffende systeem worden gebruikt in het regelaarontwerp. Verschillende data-gebaseerde technieken zijn onderzocht: 1) iteratief lerend regelaarontwerp, 2) regelaarontwerp via het iteratief instellen van de regelaarparameters en 3) data-gebaseerd ontwerp van terugkoppelingen voor de gewenste prestatie.

De prestaties van de regelaar, verkregen met de verschillende regelstrategieën, zijn met behulp van experimenten vergeleken met betrekking tot ontwerpcomplexiteit, bewegingsnauwkeurigheid en robuustheid tegen parasitaire dynamica en verstoringen. Daarnaast zijn er voorstellen gedaan voor richtlijnen om een ontwerp te selecteren, dat geschikt is voor een gegeven regelprobleem van een robotbeweging.

Резиме

Пројектовање управљања кретањем робота на бази мерених података

Убрзавање кретања робота уз очување задате тачности позиционирања његовог врха, те извршења жељење просторне трајекторије, представљају вечите изазове у роботици. Упркос све захтевнијим брзинама и тачностима кретања, неопходно је очувати неопходну робусност управљања у односу на присутне неодређености у динамици робота те поремећаје који утичу на кретање. Високи захтеви у погледу перформансе управљања и робусности мотивисали су развој различитих метода управљања кретањем робота.

Напредне методе управљања користе модел динамике робота. Модел се имплементира у инверзном динамичком закону управљања како би омогућио компензацију нелинеарних динамичких спрезања између зглобова робота. За динамике робота које нису компензиране на бази модела, надлежни су контролери директног управљања (feedfroward control) и повратне спреге (feedback control). Ови се контролери могу пројектовати на више начина.

Докторска дисертација представља резултате истраживања различитих метода пројектовања контролера директног управљања и повратне спреге, који се могу користити при управљању роботским кретањем. Све обухваћене методе блиско воде рачуна о практичним динамикама управљаног робота, наиме, ове методе омогућавају да и веома специфичне динамике имају утицаја на карактеристике резултујућег контролера. Узимањем у обзир специфичних динамика могуће је развити контролер способан да компензује најразличитија понашања робота и тако унапредити перформансу кретања. Кључно полазиште ове дисертације је да при пројектовању контролера кретања, робот треба третирати као реалан физички систем са свим засебним карактеристикама. Овакво схватање битно одступа од концепта који је доминантан у роботици, а по коме је за пројектовање управљања довољно узети у обзир само опште моделе динамике робота. Општи модели не могу у потпуности да обухвате све аспекте стварне динамике робота и зато управљачки закони који се базирају на тим моделима у пракси имају ограничене ефекте на перформансу кретања. Методи пројектовања управљања који су обухваћени овом дисертацијом намењени су умањењу конзервативности метода који су доминанто прихваћени у роботици.

Перформанса кретања зависи од квалитета модела имплементираног у инверзном динамичком закону управљања, као и од квалитета примењеног закона директног управљања и/или управљања повртаном спрегом. У овој дисертацији, задовољавајући квалитет модела остварен је свеобухватним моделирањем роботске кинематике и круте динамике са трењем, те брижљивом естимацијом параметара ових модела. Испитане су и активне (on-line) и пасивне (off-line) методе параметарске естимације. На роботу који је погоњен direct-drive моторима извршена је експериментална верификација унапређења перформансе кретања када се нелинеарна динамика робота компензује на бази модела.

Номиналној компензацији нелинеарности динамике робота на бази модела, придодају се управљачки закони који се могу пројектовати на дирекан и индиректан начин. Индирекнто пројектовање користи параметарске репрезентације (моделе) спефичичних динамика робота, тј., оних динамика које нису обухваћене номиналном компензацијом. Адекватне параметарске

репрезентације могу се добити идентификацијом специфичних динамика, што је приступ познат под називом моделирање на бази мерених података (data-driven modelling). Дисертација разматра индиректне методе пројектовања у простору зглобова робота, као и у простору врха робота. У простору зглобова: 1) пројектовање конвенционалних PD (пропорционално–диференцијалних) контролера повратне спреге, 2) итеративно H_∞ пројектовање контролера повратне спреге са експерименталним тестирањем контролера на поремећаје, 3) пројектовање контролера повратне спреге применом μ-синтезе ради постизања робусности перформансе, 4) пројектовање дискретног контролера повратне спреге променљиве структуре са клизним режимом без chattering ефекта. Проблем активног пригушења вибрација разматран је у простору врха робота. Вибрације врха настају као последица еластичности структуре робота. У дисертацији је описано пројектовање контролера за робустну компензацију вибрација.

Директне методе пројектовања управљачких закона припадају све актуелнијој области која се назива управљање на бази мерених података (data-based control). У општем случају, приликом пројектовања data-based контролера не користе се параметарски модели објекта, већ само улазно/излазни сигнали добијени мерењима на објекту. У дисертацији је анализирано неколико data-based техника: 1) пројектовање управљања на бази итеративног учења (iterative learning control), 3) итеративно подешавање контролера повратне спреге (iterative feedback tuning), и 3) data-based пројектовање контролера повратне спреге ради постизања жељене перформансе управљања.

Извршено је експериментално поређење управљачке перформансе остварене применом различитих техника управљања. Као критеријуми поређења, узети су у обзир захтевност пројектовања управљачког закона, тачност кретања и робусност у односу на паразитне динамике и поремећаје. Дате су и препоруке за избор технике пројектовања управљачког закона која је најпогоднија у датом проблему управљања роботским кретањем.

Acknowledgments

Herewith I would like acknowledge a number of associates, colleagues and friends for their valuable support.

First of all, it is my great pleasure to thank my promoter Maarten Steinbuch and my co-promotor Bram de Jager for their instructive and fair guidance, as well as for their contribution to the work presented in this thesis. I find collaboration with them a great privilege, since as supervisors they keep high working spirit and allow a candidate enough freedom for individual action. I also thank my promoter Okko Bosgra, Henk Nijmeijer and Ben Jonker for their useful comments that helped me to improve the contents of this thesis.

I am grateful to Ron Hensen for his support and for the interesting and clarifying conversations we had on friction modelling and identification. Thanks to Edo Aneke for instructive discussions on robot modelling and identification. I need to mention Rens Kodde and Pieter Nuij for their support in preparing experiments during my research. I also appreciate Harrie van de Loo and Peter Hamels for precious assistance on the experimental set-up. Also many thanks to Lia Neervoort and Corlien van Dam for being friendly and cooperative when resolving different administrative issues.

Special acknowledgment deserves Goran Golo, since without his precious and generous support on many occasions my journey towards PhD would be much different and certainly less efficient.

As particularly valuable part of my PhD program I find the collaboration with students from the Dynamics and Control Technology Group: Michiel Koot, Björn Bukkems, Arjen den Hamer, Wouter Aangenent, Geert van der Zalm, and Tom Oomen. Their contributions had great impact on my work, and herewith I thank them all.

I want to thank my colleagues from the Dynamics and Control Technology Group for nice atmosphere and pleasant time we had at the group. In particular, I thank to Rogier Hesseling, Ana Cuellar, Alejandro Rodriguez, Nenad Mihajlović, Wilbert Dijkhof, Ronald Faassen, Devi Putra, Kiril Rangelov, Alexei Pavlov, Rene van Rooij, Apostolos Doris, Nathan van de Wouw, Rob Fey, Frans Veldpaus, and Sascha Pogromsky.

Greetings to Iuliana Rotariu and to my neighbours Alex and Chrissi.

Also, special greetings go to Serbian colony in The Netherlands: Nataši, Bratislavu, Mirjani, Dušanu, Sanji, Vojkanu, Maji, Milanu, Mariji, Darku, and Draganu.

Although thousands kilometres away, kumovi Aca and Micko, Marta, Icko, Šone and Vučko are the true friends, and many thanks for all their honesty, support and love.

I am grateful to all my family and especially to my mother Dušanka, who I miss a lot, and to my father Ranko and sister Biljana who have been encouraging me through all these years.

Most of all, I thank my lovely wife Milena for wonderful love, support and encouragement. She makes me happy and proud man, who enjoys the life and brightly looks into the future.

Curriculum Vitae

September 6th, 1969	Born in Priština, Serbia.
1984-1986	High School "May 9th", Niš, Serbia.
1986-1988	High School "Bora Stanković", Niš, Serbia.
1988-1989	Military service.
1989-1994	B.Sc. studies in Automatic Control in Electrical Engineering (10 semester study program), Faculty of Electronic Engineering, University of Niš, Serbia. Graduation project: Sliding mode control of induction machines with squirrel-cage rotors.
1994-1996	Research associate within the Group for Automatic Control, Faculty of Electronic Engineering, University of Niš, Serbia.
1996-2000	Teaching and Research Assistant within the Group for Automatic Control, Faculty of Electronic Engineering, University of Niš, Serbia. M.Sc. studies in robotics. M.Sc. thesis: Application of biomechanical analogies in robotics.
2000-2004	Ph.D. candidate within the Dynamics and Control Technology Group, Department of Mechanical Engineering, Technische Universiteit Eindhoven, The Netherlands. Ph.D. Project: Data-based control of motion systems.
September 19th, 2002	Presented the Howard Kaufman Memorial award for the best student paper at the IEEE Conference on Control Applications, Glasgow, UK, 2002.

Propositions

accompanying the thesis

Data-driven Robot Motion Control Design

1. The handwriting task is interesting for industrial robots since it is a good representative of complex dynamically demanding industrial tasks.
   In: V. Potkonjak, S. Tzafestas, D. Kostić, G. Djordjević, and M. Rašić, “Illustrating Man-machine Motion Analogy in Robotics - The Handwriting Problem,” *IEEE Rob. Autom. Mag.*, Vol. 10, No. 1, pp. 35-46, 2003.

2. I don’t know if our scientific equations correspond with reality, because I don’t know what reality is. All that matters to me is that theory predicts the results obtained by the measurements.
   by Steve Hawking, physician

3. It is common experience in industrial control design that a mathematical description of the plant to be controlled is not available and that undertaking a modelling study is considered too costly and time-consuming. In these cases, one would like to come up with a controller directly from measurements coming from the plant without going through a modelling phase.
   In: A. Lecchini, M.C. Campi, and S.M. Savaresi, “Virtual Reference Feedback Tuning for Two Degree of Freedom Controllers,” *Int .J. of Adap. Contr. and Sig. Process.*, Vol. 16, No. 5, pp. 355-371, 2002.

4. The possibility to quantitatively predict motion performance is not typical in Iterative Learning Control of robot manipulators.

5. Iterative Feedback Tuning is a powerful strategy to facilitate higher complexity control designs for industrial robots.

6. The suggested data-based (DB) loop-shaping method is capable to tune controllers of given structure and complexity such that motion control of high performance is realized. The next objective should be adaptation of the method to extend its application from off-line controller design to an online DB control strategy. The expected benefit is an adaptive feature of the DB controller, which realizes the desired control performance despite time-varying plant dynamics and disturbance effects.
   This thesis: Chapter 7

7. An important subject for further research should be the derivation of a data-based (DB) stability test. Such test should ascertain if the DB controller is stabilizing for the plant, while utilizing only input/output data observed from the plant.
   This thesis: Chapter 8

8. The pioneer experiments in data-based robot motion control presented in this thesis, provide new and instructive information about the practical merits and difficulties of the considered control methods.
   This thesis: Chapter 7

9. Essentially, we feel that stability analysis should not be a necessary aspect in building all controllers since that would prevent the solution of many real-world problems. In practice, systems can often be monitored for anomalous behavior and sometimes shut down or converted to a default control if instabilities are a threat.
   In: J.C. Spall and J.A. Cristion, “Model-free Control of Nonlinear Stochastic Systems with Discrete-time Measurements,” *IEEE Trans. of Autom. Control*, Vol. 43, No. 9, pp. 1198-1210, 1998.

10. After all, it is not difficult to imagine that it may not be long before biologists construct a “perfect personal robot”−a real human, cloned and genetically engineered with all attributes of a perfect “battler” (a worker, a soldier?) despite all ethical, moral, and sociological problems that may exist.
    In: T. Fukuda, R. Michelini, V. Potkonjak, S. Tzafestas, K. Valavanis, and M. Vukobratović, “How Far Away is "Artificial Man",” *IEEE Rob. Autom. Mag.*, Vol. 8, No. 1, pp. 66-73, 2001.

11. No argument makes me believe unless my feelings say that it could be true.

12. Controller tuning is as bringing up a child: parents are pushing through their ideals, while the child only adjusts to circumstances.

13. The journey of a thousand miles must begin with a single step.
    by Lao Tzu, philosopher

14. The secret to creativity is knowing how to hide your sources.
    by Albert Einstein, physician

15. Computers are useless. They can only give you answers.
    by Pablo Picasso, painter

Dragan Kostić.
Eindhoven, 8th March 2004.