

## MASTER

### Iterative learning control of a SCARA robot

van de Ven, R.J.M.

*Award date:*  
2002

[Link to publication](#)

#### **Disclaimer**

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

#### **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

# **Iterative Learning Control of a SCARA robot**

**R.J.M. v.d. Ven**

**CST 2002.10**

Research project report

Supervisor: Prof. Dr. Ir. M. Steinbuch  
Coaches: Dr. Ir. M.J.G. v.d. Molengraft  
E.A.F. Kremers M.Sc.

EINDHOVEN UNIVERSITY OF TECHNOLOGY  
DEPARTMENT OF MECHANICAL ENGINEERING  
CONTROL SYSTEMS TECHNOLOGY GROUP

Nuenen, February 2002



# Abstract

The modelling and control of an industrial SCARA robot is considered. The robot is used in the reticle handling module of a wafer scanner. During operation the robot has to perform repeated (straight-line) motions. A maximum straight-line position error of 0.2 mm is specified to prevent collision with other parts of the handling module.

The robot consists of two links and an end-effector and can move in the horizontal plane. During motion of the robot arm coupling forces and torques are present due to velocity and accelerations of the links. Each link is driven by a motor and between the motor and the link a gear train is present. On each motor an encoder is mounted for position measurement. The chain motor - transmission - load introduces dynamic effects. Furthermore, friction is dominantly present in the gear trains and the joints.

In this research an Iterative Learning Control (ILC) scheme is applied to the robot. With this control algorithm it is possible to eliminate reproducible errors of a motion on the basis of error signals. The powerful concept of ILC is illustrated by the achieved reduction of the position errors at the encoders. The maximum error at the encoders could be reduced a factor 10 to 25 when compared to PID controllers. The maximum error of the end-effector could be reduced just within the specifications.

## Samenvatting

In dit project is de modellering en de regeling van een industriële SCARA robot onderzocht. De robot wordt toegepast in de reticle handling module van een wafer scanner. Tijdens bedrijf wordt de robot gebruikt voor het herhaaldelijk uitvoeren van onder andere rechte bewegingen. De maximale positiefout tijdens een rechte beweging is gespecificeerd als 0.2 mm om botsing met andere delen van de handling module te voorkomen.

De robot is opgebouwd uit twee links en een eind-effector en kan bewegen in het horizontale vlak. Tijdens beweging van de robotarm zijn koppelingskrachten en -momenten tussen de links aanwezig als gevolg van snelheid en versnellingen van de links. Iedere link wordt aangedreven door een motor met overbrenging en op de motoren zijn encoders gemonteerd voor positiemeting. De keten motor - overbrenging - last introduceert dynamische effecten. Daarnaast is wrijving dominant aanwezig in de overbrengingen en de scharnieren.

Tijdens dit onderzoek is Iterative Learning Control (ILC) toegepast op de robot. Dit regelalgoritme is in staat om repeterende fouten weg te regelen aan de hand van foutsignalen van een beweging. Het waardevolle concept van ILC is geïllustreerd aan de hand van de reductie van de encoder positiefouten. De maximale fout kon daar een factor 10 tot 25 gereduceerd worden in vergelijking met PID regelaars. De maximale positiefout van de eind-effector kon gereduceerd worden tot net binnen de specificatie.

# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Report overview . . . . .	8
1.2	The company CCM . . . . .	9
<b>2</b>	<b>The robotic system</b>	<b>11</b>
2.1	Robot construction . . . . .	11
2.2	Robot tasks . . . . .	12
2.3	Kinematics . . . . .	12
2.3.1	Kinematic modelling as a chain of bodies . . . . .	12
2.3.2	Transformations to other coordinate frames . . . . .	14
<b>3</b>	<b>Modelling of the robot</b>	<b>17</b>
3.1	Modelling of the electrical part . . . . .	17
3.1.1	Actuator model . . . . .	18
3.1.2	Servo amplifier . . . . .	19
3.2	Modelling of the mechanical part . . . . .	19
3.2.1	Classification of dynamic effects . . . . .	19
3.2.2	Coupling torques . . . . .	20
3.2.3	4 <sup>th</sup> order model . . . . .	21
3.2.4	Friction . . . . .	22
3.2.5	Disturbances . . . . .	23
3.2.6	Measurement errors and imperfections in the robot . . . . .	23
3.2.7	The total dynamic model: an overview . . . . .	24

<b>4</b>	<b>Model verification and signal analysis</b>	<b>25</b>
4.1	Experimental determination of transfer functions . . . . .	25
4.1.1	Diagonal terms . . . . .	27
4.1.2	Comparison between models and measured FRF's . . . . .	28
4.1.3	Off-diagonal terms . . . . .	29
4.2	Rigid body torques and friction . . . . .	30
<b>5</b>	<b>Error signal analysis</b>	<b>31</b>
5.1	Experimental setup . . . . .	31
5.2	Relation between encoder signals and inductive signals . . . . .	32
5.3	Analysis of encoder signals . . . . .	33
5.4	Analysis of inductive signals . . . . .	35
<b>6</b>	<b>Controller design</b>	<b>37</b>
6.1	Control strategy . . . . .	37
6.2	Feedback controller . . . . .	37
6.2.1	Homing . . . . .	38
6.3	Data-based feedforward control . . . . .	38
6.3.1	Iterative Learning Control . . . . .	39
6.3.2	Repetitive Control . . . . .	40
6.3.3	Neural nets . . . . .	41
6.4	Iterative Learning Control . . . . .	42
6.4.1	Construction of $L$ -filter . . . . .	42
6.4.2	Convergence criterium . . . . .	43
6.4.3	ILC with robustness filter . . . . .	44
6.4.4	Construction of $Q$ -filter . . . . .	46
6.4.5	Parallel filtering . . . . .	46
6.4.6	Implementation aspects . . . . .	47
6.4.7	Results . . . . .	47
6.5	Other control methods . . . . .	50

<i>Contents</i>	5
6.6 Concluding remarks . . . . .	53
6.6.1 Implementation issues . . . . .	53
<b>7 Conclusions and recommendations</b>	<b>57</b>
7.1 Conclusions . . . . .	57
7.2 Recommendations . . . . .	58
<b>Bibliography</b>	<b>61</b>
<b>A Kinematics</b>	<b>63</b>
A.1 Kinematic modelling as a chain of bodies . . . . .	63
A.1.1 Position vectors . . . . .	63
A.1.2 Rotation . . . . .	64
A.1.3 Angular velocity . . . . .	64
A.1.4 Position, velocity and acceleration of the centre of mass . . . . .	65
A.2 Transformations to other coordinate frames . . . . .	66
A.2.1 Transformation between relative and Cartesian coordinates . . . . .	66
A.2.2 Transformation between relative and motor coordinates . . . . .	67
<b>B Rigid body dynamics</b>	<b>69</b>
B.1 Introduction . . . . .	69
B.2 Newton-Euler approach . . . . .	69
B.3 Body forces and torques . . . . .	70
B.3.1 World . . . . .	70
B.3.2 Body 1 . . . . .	71
B.3.3 Body 2 . . . . .	72
B.3.4 Body 3 . . . . .	72
B.4 Systems equations in relative coordinates . . . . .	73
B.4.1 Elimination of joint forces . . . . .	73
B.4.2 Equations for the motor torques . . . . .	73
B.5 Motion in a horizontal plane . . . . .	74
B.5.1 Interpretation of the dynamic equations . . . . .	74



<b>C ILC to axes 2 and 3</b>	<b>77</b>
<b>D Verification on another robot</b>	<b>81</b>
D.1 Encoder signals . . . . .	82
D.2 Inductive signals . . . . .	83

# Chapter 1

## Introduction

For the reticle handling in a wafer scanner a SCARA robot is used. This is a two link robot with an end-effector. The robot has to perform different straight-line motions repeatedly during operation. During operation vibrations occur when tracking a (straight-line) trajectory. The allowed vibration and tracking error are specified within a narrow band to prevent collision of the end-effector with a submodule of the reticle handler.

This brings us to the problem definition of the project:

*Is it possible to increase the straight-line accuracy of the robot to within an error band of 0.2 mm?*

From previous research [1] it was already clear that this straight-line accuracy could not be reached with only standard classical controllers, such as PID controllers. Additional friction feedforward and inverse model feedforward were necessary to increase the accuracy. A major drawback of these feedforwards is that they have to be fine-tuned for each robot individually. The maximum straight-line accuracy that was reached was approximately 0.25 mm. From this research it became also clear that the error signals show a large reproducible part. This opens a way for control strategies which aim at 'learning' of the error signal.

The main goal of the project is:

*Develop a controller which is capable of increasing the straight-line accuracy of the robot to within an error band of 0.2 mm.*

And as a subgoal of the project:

*Investigate the error signals for the typical motions of the robot and, when applicable, develop a learning control strategy.*

## 1.1 Report overview

During this master thesis project research is done to improve the tracking of the robot. The main development steps are summarized in figure (1.1), where a design flow chart is depicted.

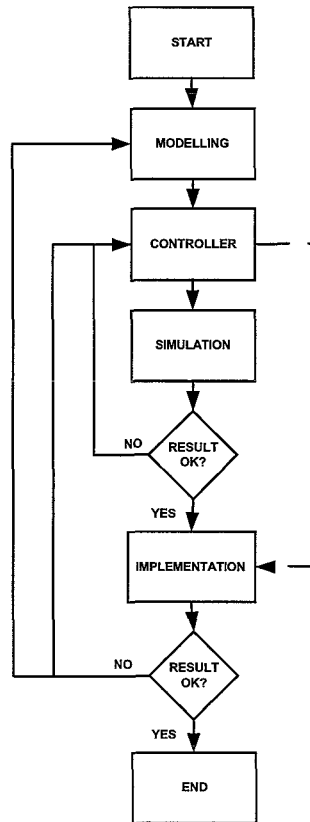


Figure 1.1: Design flow of controller design

In the figure it is seen that it can be necessary to reconsider modelling or controller design during the project. This can be necessary when results from simulation or implementation on the system are not satisfactory. The dashed line represents the direct flow from controller design to implementation. This path can be followed when simulation models are not able to simulate essential behavior of the system in a good way. For ease of readability the report is organized in a way similar to the top-bottom flow.

The chapters in the report are organized as follows:

In the second chapter a description of the robotic system will be given. The construction of the robot is investigated and the kinematics of the robot are given. It is possible to describe the kinematics in several coordinate systems and the relation between these coordinate systems is given. In appendix A the kinematic modeling is worked out more extensively.

In Chapter 3 the modelling of the robot will be described. The modelling consists of modelling of the electric part and modelling of the mechanical part of the robot. In the electric modelling the amplifiers, which supply the motors, and the electric part of the motors are described. The total dynamic model consists of several submodels: a fourth order dynamic model, a rigid body model and a friction model. All submodels are shortly described. In Appendix B the rigid body model is derived.

The relation between experimental data and the theoretically derived models will be described in the fourth chapter. In this chapter the quality of the theoretical models is investigated.

In Chapter 5 the error signals from straight-line motions of the robot are analyzed both in the time domain and in the frequency domain.

In Chapter 6 the controller design will be regarded. A strong emphasis will be put on Iterative Learning Control. At the end of the chapter the control strategy will be evaluated and some other control methods will be shortly discussed.

The report is concluded with the conclusions and recommendations.

## 1.2 The company CCM

CCM (Centre for Concepts in Mechatronics) is an independent technical research and development company with approximately 100 staff members. It is located in Nuenen, the Netherlands. Research is done on customer request and specification. Both large and small projects are carried out from conception to implementation. CCM's specialization is the application of mechatronic methods for solving technical problems. Mechanics, electronics, robotics and information technology all play a vital role in the development of new products and production systems.

CCM is co-developer of two subsystems for a wafer scanner of the company ASML. These modules are the reticle handling module and the wafer handling module. In this project the SCARA robot is investigated that is used in the reticle handling module.



## Chapter 2

# The robotic system

This chapter begins with a description of the robot and the main components of the robot. Then, the purpose of the robot is described by means of the functions it has to carry out during operation. Then, the kinematics of the robot is given in different coordinate systems.

### 2.1 Robot construction

The SCARA (Selective Compliance Articulated Robot Arm) robot is a two link robot with an end-effector. The robot with end-effector is thus regarded as a three link robot. The total arm can translate in height. For this research only motions in the horizontal plane are considered. In figure (2.1) a schematic drawing of the robot is displayed.

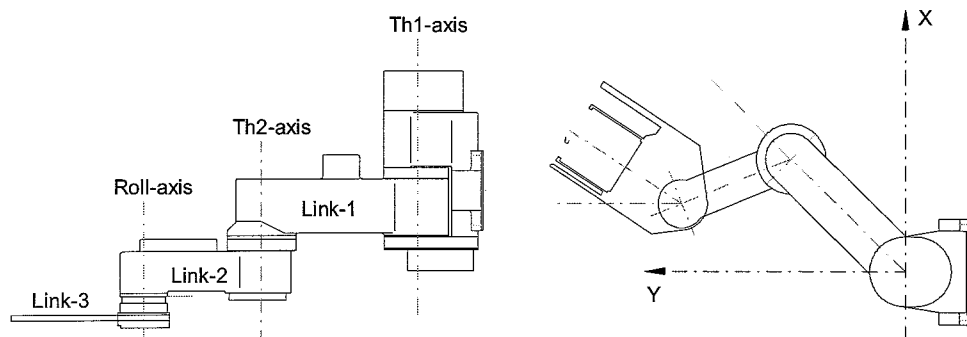


Figure 2.1: Schematic drawing of the robot, side view (left) and top view (right)

Each link is driven by a brushless AC synchronous motor. Between the motor and the

a Harmonic Drive). The gear train between the second motor and the link consists of a Harmonic Drive and a timing belt. The coupling between the third motor and the third link is established by means of a Harmonic Drive and two timing belts. On each motor an incremental encoder is mounted.

## 2.2 Robot tasks

The robot subsequently performs specific tasks during operation, like a pick and place machine. Some examples are the loading of a reticle in the library of the reticle handling module and the delivery of a reticle at the turret. The turret is a submodule of the reticle handler which passes the reticle on to the reticle stage. The tasks are visualized in figure (2.2).

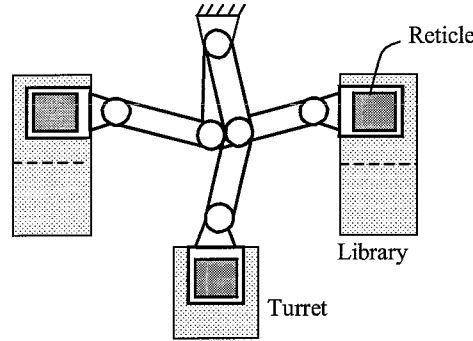


Figure 2.2: Illustration of the robot tasks

## 2.3 Kinematics

### 2.3.1 Kinematic modelling as a chain of bodies

The robot is described with an open kinematic chain. The system is modelled as three bodies  $B_1$ ,  $B_2$ ,  $B_3$  (the three links of the robot) and the real world  $B_0$ . A body is completely characterized by means of position (3 independent translational parameters) and orientation (3 independent rotational parameters). The bodies are connected with line hinges:  $H_1$ ,  $H_2$ ,  $H_3$ . The generalized coordinates,  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ , are defined as in figure (2.4). The values for the three generalized coordinates determine the configuration of the robot.

The zero configuration of the robot is given by:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The robot, modelled as an open kinematic chain, is given in figure (2.3).

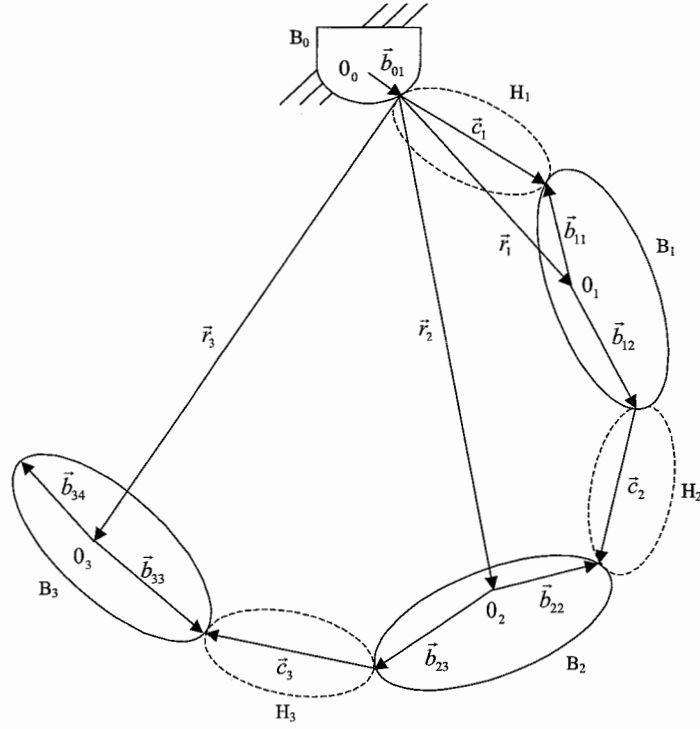


Figure 2.3: The robot modelled as an open kinematic chain

The kinematics of the robot are described by means of position vectors ( $\vec{r}$ ), body vectors ( $\vec{b}$ ) and rotation matrices. In appendix A the kinematic modelling is worked out more extensively.



### 2.3.2 Transformations to other coordinate frames

The state of the robot can be described in different coordinate systems. The coordinate systems are listed below:

reference frame	coordinates		
	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>
<i>relative coordinates</i>	$\theta_1$	$\theta_2$	$\theta_3$
<i>Cartesian coordinates</i>	$x_{TCP}$	$y_{TCP}$	$R$
<i>Sankyo coordinates</i>	$\theta_1$	$\theta_2$	$R$
<i>motor coordinates</i>	$\phi_1$	$\phi_2$	$\phi_3$

where TCP is the Tool Centre Point of the robot. In figure (2.4) the coordinates and the TCP are displayed. The motor coordinates are proportional to the relative coordinates but include the gear ratios. The Sankyo coordinates are the coordinates as described by the manufacturer of the robot. The advantage of the Sankyo coordinate system is that a straight-line motion of the end-effector can be easily described because the *ROLL* ( $R$ ) coordinate does not change during the motion while the relative coordinate does change.

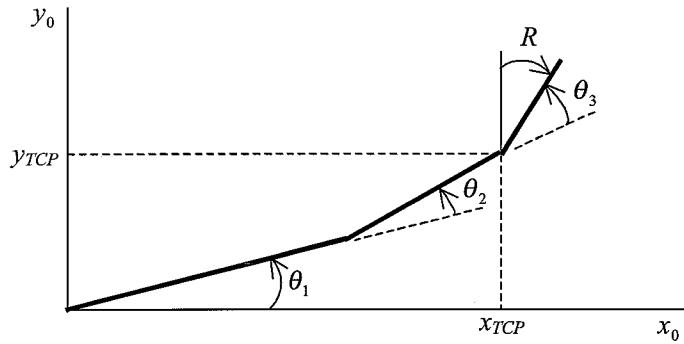


Figure 2.4: Representation of the coordinate frames

The relation between the different coordinate frames is seen in figure (2.5).

The transformation between the Cartesian coordinates and the relative coordinates is not a unique transformation. For an  $x - y$  position of the TCP two different arm configurations are possible. This is called redundancy. The transformations, visualized with an arrow in figure (2.5), are worked out in Appendix A.

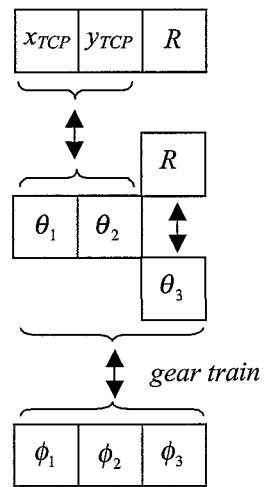


Figure 2.5: Visualization of the coordinate transformations



## Chapter 3

# Modelling of the robot

In this chapter the modelling of the robot is described. The modelling part consists of two parts: modelling of the electric part and modelling of the mechanical part of the robot. In figure (3.1) the robot (the plant) is schematically depicted in a block diagram.

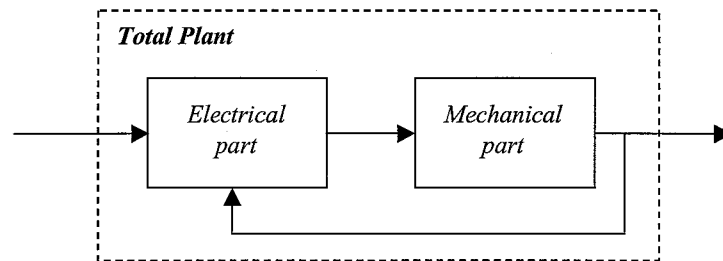


Figure 3.1: Interaction between the electrical part and the mechanical part of the plant

In this figure the mechatronic construction of the robot is seen by the interaction between the electric part and the mechanical part of the plant. As will become clear in the following sections, the models are in their turn built up from several submodels. For reasons of clarity, in some sections attention is mainly focused on only one link. The models are however valid for all three links. Only the parameters in the models will be different for each link.

### 3.1 Modelling of the electrical part

The electric part of the total system consists of the electrical part of the motors and the servo amplifiers in the Reticle Handling Robot Rack (RHRR).

### 3.1.1 Actuator model

The actuators used in the robot are brushless AC synchronous motors with four pole pairs. All windings in the motor have one common interconnection inside the motor (star-configuration). The windings in the motor (the stator part of the motor, see figure (3.2)) generate a rotating magnetic field inside the motor when connected to the three phase power supply. This rotating magnetic field generates forces on a conductor (the rotor) located in the magnetic field and thus makes the rotor to rotate. In figure (3.2) a schematic drawing is given for an AC motor with only one pole pair [20].

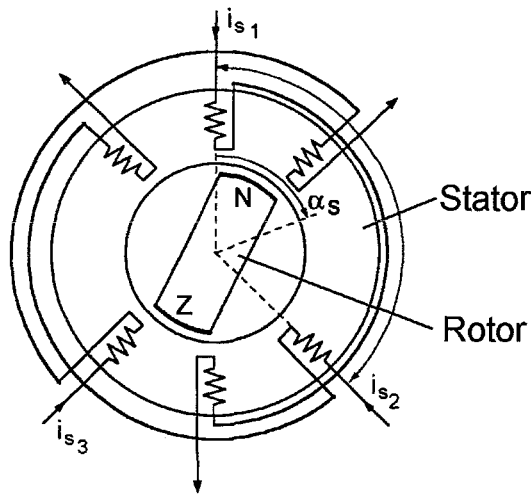


Figure 3.2: Schematic drawing of an AC motor with one pole pair

In a magnetic alignment procedure the commutation angle<sup>1</sup> of the motor is found. The encoder then uses this reference to introduce a real-time update of the commutation angle. Now a linear relation between current and torque of the motor is established. The motor can now be represented as a DC motor. The electrical part of the motor is represented by the electrical circuit displayed in figure (3.3).

The coupling between the electrical part of the motor and the mechanical part is established via the back electric magnetic force and the current through the motor:

$$U_{back-emf} = k_e \omega_m \quad (3.1)$$

$$T_m = k_m I \quad (3.2)$$

<sup>1</sup>The commutation angle can be seen as the orientation between a pole pair of the rotor and the coils in the stator in which the current through the coils can generate a maximum torque on the rotor.

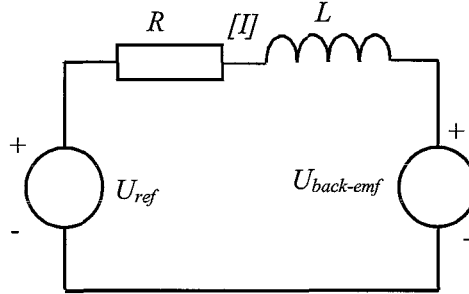


Figure 3.3: Electric part of the motor represented as a DC circuit

where  $\omega_m$  is the angular speed of the motor,  $k_e$  is the voltage constant and  $k_m$  is the motor constant. The motor constant and voltage constant are equal to each other.

### 3.1.2 Servo amplifier

The actuator is driven by a servo amplifier. The servo amplifier has a current loop which realizes that the current is proportional to torque (equation (3.2)), which is desired for the control of the robot. The obtained bandwidth of the current loop is 500 Hz.

## 3.2 Modelling of the mechanical part

### 3.2.1 Classification of dynamic effects

In the previous chapter the construction of the robot and kinematics were investigated. The construction of the robot determines for a large part the dynamic behavior. Firstly, the links are coupled. A motion of one link will also generate torques and forces on the other links. These coupling torques and forces are geometrically nonlinear and are described by means of a rigid body model. This is the contents of section 3.2.2 and appendix B. Secondly, each motor is coupled to a link by means of a gear drive. The gear drives have finite stiffness and thus also introduce dynamic effects. This is further investigated in section 3.2.3. Furthermore, the robot shows position dependent dynamics because the mass moment of inertia about a motor axis changes when the links rotate relative to each other.

Accounting for all these dynamic effects, the total dynamics of the robot can be classified into different regions according to the dominance of one of the dynamic effects. The torques that a motor 'feels' is dependent on the type of motion the robot performs. This is visualized in figure (3.4).

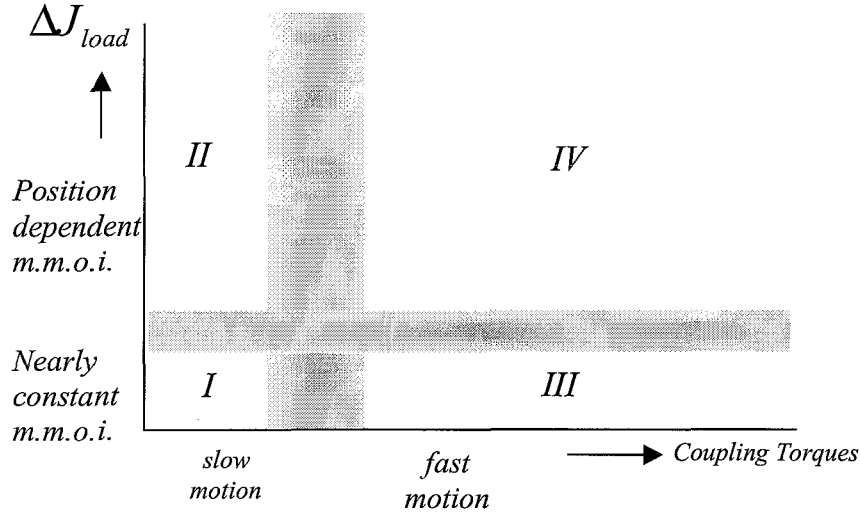


Figure 3.4: Classification of the dynamic effects according to the type of motion

The mass moment of inertia (m.m.o.i.) of the load about a motor axis is determined by the angles of the links via a kinematic relation. The coupling torques are dependent on link angles, velocities and accelerations as described in the rigid body modelling (appendix B).

In region *I* the m.m.o.i. of the load is approximately constant during motion and coupling torques are small. The dynamic description for a motor and link can then be approximated by a Linear Time Invariant (LTI) model. Typical motions which satisfy these conditions are slow rotational motions of e.g. the first link (where the second and third link do not perform a relative motion). The second region (*II*) is characterized by small coupling torques but now the m.m.o.i. of the load varies. This corresponds to slow stretching and folding of the total robot arm. For a given trajectory the model for a motor and link can be regarded as a Linear Time Varying (LTV) system. The type of motions which are typical for region *III* are fast rotational motions of the robot. In the fourth region (*IV*) all dynamic phenomena are significantly present: the inertia of the load varies and coupling torques are relatively large. Fast stretching and folding of the robot arm are typical motions which belong to this region.

### 3.2.2 Coupling torques

Coupling torques and forces between the links originate from motion of the links. To evaluate the coupling torques a rigid body analysis is performed. The coupling torques

are dependent on link positions, velocities and accelerations:

$$T_{coupling} = T_{coupling}(\theta_{1,2,3}, \dot{\theta}_{1,2,3}, \ddot{\theta}_{1,2,3}) \quad (3.3)$$

The relationship is nonlinear. In general these coupling torques are small when the link accelerations and velocities are kept low. In Appendix B the equations of motion are derived for the rigid body model.

### 3.2.3 4<sup>th</sup> order model

The attention is now focussed on only one motor coupled to a link by means of a gear drive. The geometric coupling between the links is not considered here. The link<sup>2</sup> is analyzed by means of a lumped parameter model. This part of the robot is modelled with inertias, springs, dampers and transmissions [19]. Then reduction rules<sup>3</sup> can be applied to further simplify the model and to obtain a valid model in the frequency range of interest. The lumped parameter model can be reduced to two inertias connected by a spring and a damper. The first inertia represents the motor and the second inertia represents the total load for that motor. The motor inertia is driven by a torque. The spring and the damper represent the finite stiffness of the gear train. The most simple form of a fourth order model is visualized in figure (3.5).

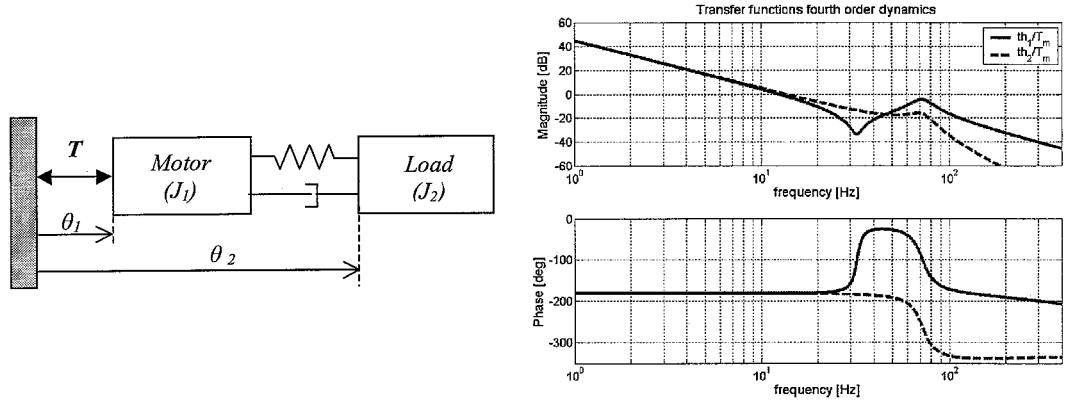


Figure 3.5: Representation of a 4<sup>th</sup> order model (left) and Bode diagrams for transfer functions (right)

This kind of model is of fourth order because each inertia acts as a double integrator. First, the inertias, spring and damper are assumed to be constant.

<sup>2</sup>In the rest of the report the word 'link' is often used to represent the total of a motor which is coupled to the 'real' link by means of a gear drive.

<sup>3</sup>Transmissions are eliminated from the lumped models by calculating equivalent inertias, spring and damper values when the transmissions are removed. Furthermore only dominant inertias, springs and dampers are taken into account.



The equations of motion for this 4<sup>th</sup> order model can be expressed in the frequency domain by using the Laplace transformation ( $s$  is the Laplace variable). The transfer functions from input  $T_m$  to the outputs  $\theta_1$  and  $\theta_2$  are given by (for simplicity damping is neglected):

$$H_1(s) = \frac{\theta_1(s)}{T_m(s)} = \frac{s^2 + \omega_a^2}{J_1 s^2 (s^2 + \omega_r^2)} \quad (3.4)$$

$$H_2(s) = \frac{\theta_2(s)}{T_m(s)} = \frac{\omega_a^2}{J_1 s^2 (s^2 + \omega_r^2)} \quad (3.5)$$

where

$$\omega_a = \sqrt{\frac{k}{J_2}} \quad (3.6)$$

$$\omega_r = \sqrt{\frac{k(J_1 + J_2)}{J_1 J_2}} \quad (3.7)$$

are the anti-resonance frequency  $\omega_a$  and resonance frequency  $\omega_r$  of the system and  $k$  is the spring stiffness. So this fourth order model will be valid for the transfer from motor torque to angle of the motor and link if all other torques are assumed to be small. Furthermore it is assumed that the inertia of the load, which the motor "feels", is constant. This is a useful approximation when the robot remains in the same configuration. When the robot on the other hand performs a motion where the robot configuration changes (the links move relative to each other) then the inertia of the load changes:

$$J_2 = J_2(\theta_2, \theta_3)$$

As can be seen from formulas (3.6) and (3.7) the anti-resonance and resonance frequencies also vary when the load inertia changes. Also the gain at frequencies below the resonance frequency varies. Beyond the resonance frequency the gain of the system will be the same for all robot configurations because the load is then decoupled.

### 3.2.4 Friction

Friction is present in the joints and in the gear train. It is difficult to assign the friction to the motor side or to the load side of each link. For simulation purposes the friction is assigned to the motor side of each link. The friction is modelled with three friction components:

1. Static friction: friction torque when there is no relative motion between two bodies
2. Coulomb friction: friction torque when two bodies move relative to each other

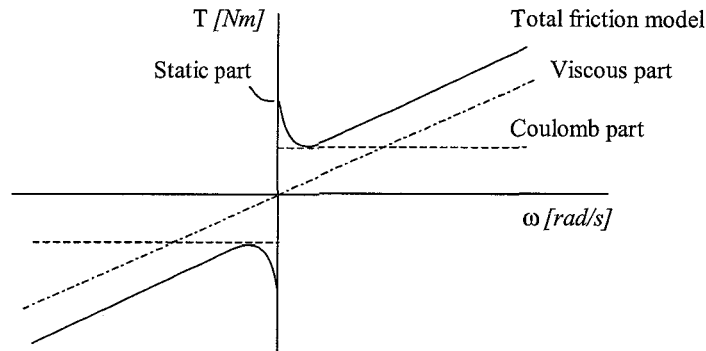


Figure 3.6: Friction model

3. Viscous friction: friction which is related to the relative speed of two bodies

The shape of the friction model is displayed in figure (3.6).

In figure (3.6) the transition from Static friction to Coulomb friction is 'smooth'. This is called the Stribeck effect.

### 3.2.5 Disturbances

Besides all torques and forces which find their origin in one of the dynamic phenomena described before (coupling torques, 4<sup>th</sup> order character and friction), also disturbances can act on the system. The nature and amplitude of these disturbances have to be verified by experiments. All phenomena which are not included in the submodels described before, are regarded as disturbances to the system. An example is unmodelled dynamics like the z-column. The robot arm is mounted on a z-column which is modelled as the fixed world. In reality the z-column is not completely fixed and thus introduces dynamic effects.

### 3.2.6 Measurement errors and imperfections in the robot

The robot has to be controlled with the information we are able to extract from the robot. The only measurement information available are the encoder signals. The encoder signals will always introduce a measurement error. The real position of a motor is thus represented by:

$$\text{output motor} = \theta_1 = \theta_{meas} + \delta\theta_1$$

Also in the gear drive and robot links imperfection can be present. The gear drive can introduce position errors, due to e.g. wear, which are not visible on the encoder

signals. The kinematic parameters of the robot, like lengths of the links, may also differ from the nominal values and thus also introduce a position error of the end-effector. Accounting for these imperfections, the output of the load is represented by:

$$\text{output load} = \theta_2 = \theta_{\text{nominal}} + \delta\theta_2$$

### 3.2.7 The total dynamic model: an overview

The separate parts of the modelling are now combined to come to a total dynamic model for a link. As a basis for the total dynamic model a fourth order model is taken, see figure (3.7).

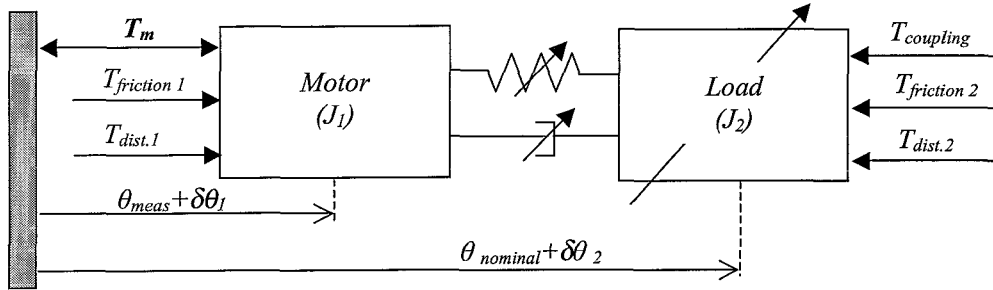


Figure 3.7: Total dynamic model of a link

The electrical part of the motor converts an input current to a torque. This motor torque  $T_m$  accelerates the motor. The motor is coupled to the load by means of a gear train. The gear train introduces flexibility and damping, which is represented by the spring and the damper. The arrows pointing through the spring, damper and the load in figure (3.7) indicate that the spring, damping and load can vary during operation. The spring is often modelled as a linear spring but this is only valid in approximation. The real spring stiffness of the Harmonic Drive is also dependent on the torque applied to the gear. Several torques act on the two inertias. Because the links of the robot are connected by means of revolute joints, a motion of one link will also generate a coupling torque on the other links. These coupling torques are indicated with  $T_{\text{coupling}}$ . Furthermore, the gear train and the joint also introduce friction:  $T_{\text{friction}}$ . Also, external disturbances can act on the motor and the load:  $T_{\text{dist.}}$ . Finally, imperfections and measurement errors are present in the system. All measurement errors and imperfections are represented by  $\delta\theta_1$  and  $\delta\theta_2$ .

## Chapter 4

# Model verification and signal analysis

In this chapter the experiments to identify the dynamics of the robot are described.

### 4.1 Experimental determination of transfer functions

The frequency response functions (FRF's) of the links are measured with SigLab [16]. This device is both a signal generator and a spectrum analyzer. By measurement of the FRF of an input to an output the linear dynamics between the input and output is measured. From the modelling of the robot it is clear that friction and coupling torques between the links are present during motion. These are nonlinear terms. To be able to measure the frequency response functions, the robot has to move to overcome the friction. The motion of the links on the other hand may not be too fast because otherwise the coupling torques will disturb the measurement. It is assumed that the linear dynamics of the robot can be modelled as three SISO systems. The transfer function matrix from input currents to motor positions is then represented by:

$$P = \begin{bmatrix} P_1(s, \underline{\theta}) & 0 & 0 \\ 0 & P_2(s, \underline{\theta}) & 0 \\ 0 & 0 & P_3(s) \end{bmatrix} \quad (4.1)$$

In the diagonal transfer functions of the first and second link a workpoint dependency is present. The off-diagonal terms in the plant matrix (4.1) are assumed to be zero or close to zero. Some critical notes about this assumption are made in section 4.1.3.

The FRF measurement of the plant now consists of two steps:

1. Determination of different arm configurations to analyze the workpoint dependency

## 2. Measurement of the SISO frequency response functions

For the first link three different arm configurations are measured: the stretched configuration, the right angle configuration and the folded up configuration (see figure (4.1)).

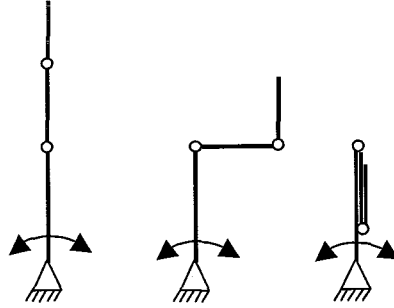


Figure 4.1: Different arm configuration for FRF measurement for the first link: stretched (left), right angle (middle) and folded (right)

For the other two axes only one configuration is measured. The mass moment of inertia about the second axis is not much affected by the arm configuration because the mass of the third link is relatively small. The diagonal sensitivity functions are measured while performing a rotation of that link with constant angular velocity and no relative motion of the other links. The measurement is carried out in the configuration as depicted in figure (4.2).

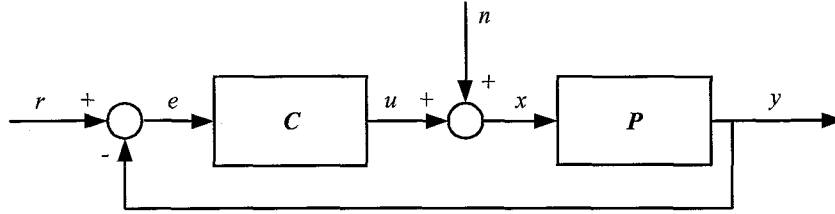


Figure 4.2: Measurement configuration for FRF measurement

A signal  $n$  is injected and the signals  $x$  and  $n$  are measured. The measured transfer function is then:

$$S(f) = \frac{S_{nx}(f)}{S_{nn}(f)} \quad (4.2)$$

The coherence function,  $\gamma_{xn}^2(f)$ , gives a measure of the quality of the measurement. It indicates the portion of the systems output power due to the input excitation and hence

the quality of the measurement. The coherence function is defined as:

$$\gamma_{xn}^2(f) = \frac{|S_{xn}(f)|^2}{S_{xx}(f)S_{nn}(f)}, \quad 0 \leq \gamma_{xn}^2(f) \leq 1$$

where  $S_{xx}(f)$  and  $S_{nn}(f)$  are the auto power spectra of the signals  $x$  and  $n$  and  $S_{xn}(f)$  is the cross power spectrum of the signals. In figure (4.3) a measurement is shown together with its coherence function. The measured sensitivity function is converted to the frequency response function of the plant.

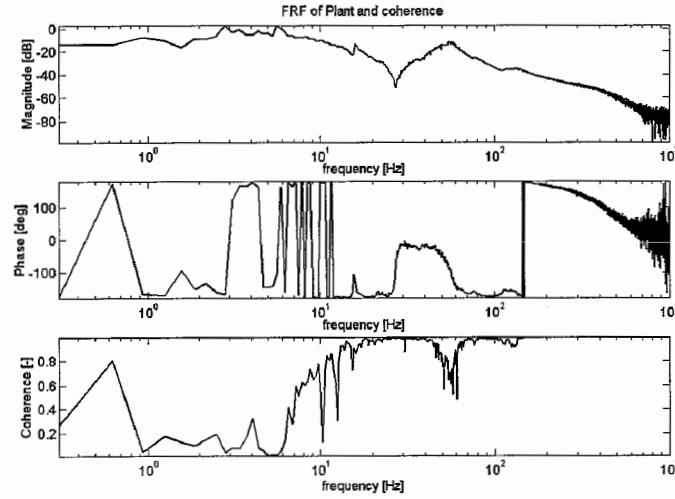


Figure 4.3: FRF measurement of the first link in the right angle configuration together with the coherence function

As can be seen in figure (4.3), the coherence is very bad below 5 Hz. This low coherence is due to nonlinearities and disturbances that act on the system. The low coherence below 5 Hz is expected to be mainly caused by friction.

#### 4.1.1 Diagonal terms

The FRF's for the first link for the different arm configurations are given in figure (4.4).

Notice the dependency of the workpoint in the FRF from current to the motor angle of the first link. The dominant anti-resonance frequency and resonance frequency are different in the different arm configurations. This is caused by the varying mass moment of inertia about the first axis, dependent on the arm configuration of the robot. The measured FRF for a fixed configuration is in line with the 4<sup>th</sup> order model as illustrated in the previous chapter.

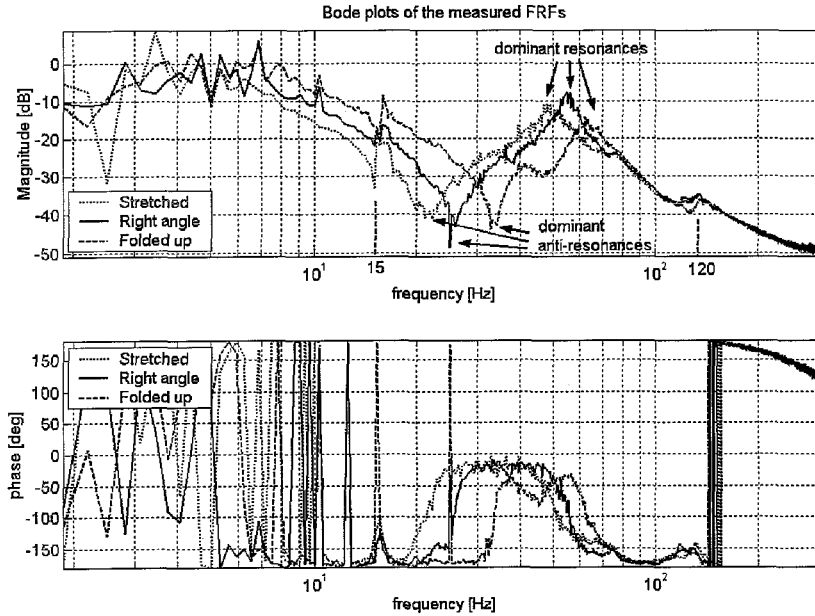


Figure 4.4: Measured FRF's for the first link in three different arm configurations

In the measurement results shown in figure (4.4) also an anti-resonance and resonance is present around 15 Hz. From further experiments it became clear that this is caused by vibrations of the z-column. The z-column is only fixed on the bottom side for this test robot and is thus not totally fixed. The anti-resonance and resonance frequencies vary several Herz with the z-position of the robot arm. It is expected that in the reticle handling module this anti-resonance and resonance are not present because there the z-column is fixed on the bottom and on the top side.

In the folded configuration an anti-resonance and resonance are present between 40 and 50 Hz. The precise origin of this is unknown. Finally there are also a small anti-resonance and resonance present around 120 Hz.

The frequency response functions for the second and third link are given in figure (4.5).

In figure (4.5) also a fit is made on the measured FRF's. These fits will be useful in the controller design part.

#### 4.1.2 Comparison between models and measured FRF's

The FRF's of the 4<sup>th</sup> order models, that are derived in the previous chapter, are compared to the measured FRF's to address the accuracy of the models. The values for the

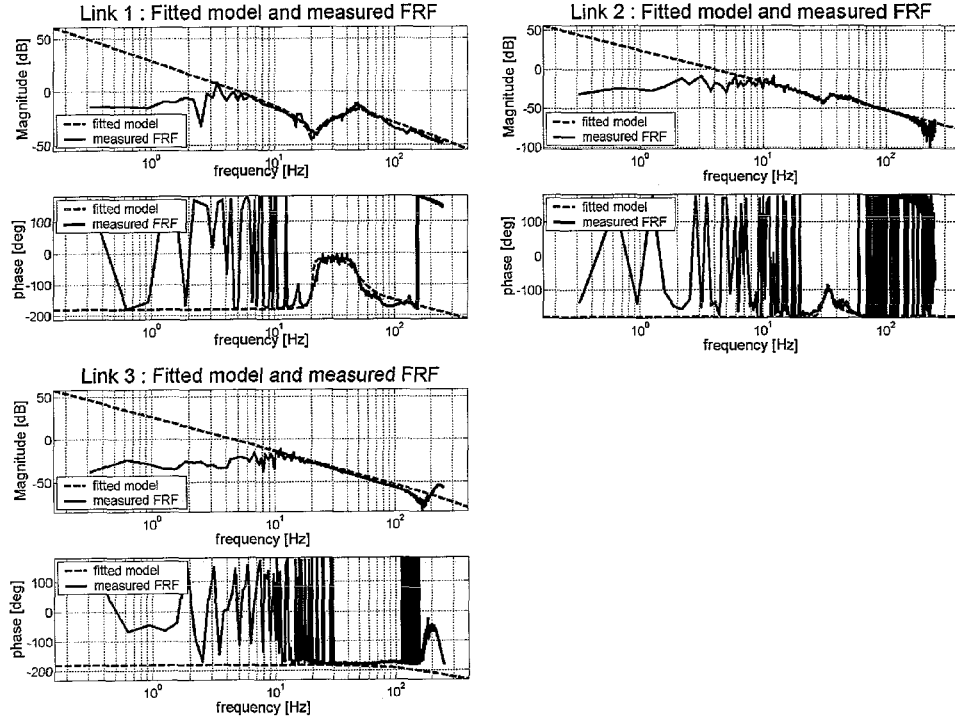


Figure 4.5: Measured FRF's together with a fit

motor inertias, load inertias and stiffnesses of the gear drives are taken from the specifications of the manufacturer and the damping coefficients in the models are fitted to the experimental data to obtain a good fit. Because the  $z$ -column is not taken into account in the modelling, the anti-resonance and resonance around 15 Hz are not present in the "fitted" model of the first link. The results are displayed in figure (4.5).

#### 4.1.3 Off-diagonal terms

The off-diagonal terms will in general not be totally zero. Because of the geometrically nonlinear coupling between the links, a torque applied to the first motor will also introduce torques and forces on the second link and third link. The effects these torques and forces cause, are dependent on the robot arm configuration. When, for example, the second link is perpendicular to the first link (right angle configuration) than an input signal to the first motor will not lead to a rotation of the second link. In the stretched and folded up configuration of the robot arm, an input torque to the first motor will also introduce vibrations in the second link. For simplicity, the off-diagonal terms in the transfer function matrix (4.1) are not taken explicitly into account in the controller design part.



## 4.2 Rigid body torques and friction

The rigid body torques and forces on a link are dependent on position, velocity and accelerations of the links and on parameters such as mass and inertia of the links and lengths of the links. The dominant parameters are well known for the robot. The setpoint contains all information about position, velocity and acceleration of each link. It is expected that the rigid body model therefore accurately describes the coupling torques. Friction on the other hand is more difficult to identify. From previous research [1] it became clear that friction is not uniform over the whole working area of the robot and can be time dependent. After a long period of standstill, friction has changed.

## Chapter 5

# Error signal analysis

Because the robot repeatedly performs the same motion tasks, as is typical for an industrial robot, it is expected that the error signals of these trajectories are quite the same for each trial and that they show a large position dependent part. If the error signals are indeed reproducible for each trial then it is possible to use control strategies which deal with these reproducible errors.

### 5.1 Experimental setup

The error signal at the end-effector can be calculated from encoder information. The straight-line error signal is a combination of the errors at the three axes. Using the kinematics, the separate encoder errors can be combined to obtain the straight-line error. When the robot performs a straight-line motion the error can also be measured with an inductive measurement system. This measurement system consists of an inductive sensor and a steel beam. The steel beam acts as the straight-line reference. During a straight-line motion, the end-effector with the inductive sensor moves along the steel beam and the straight-line error can be measured. In figure (5.1) a picture of the experimental setup is given.

The trajectory is of third degree and describes the straight-line movement from  $(x, y) = (0, 0.11)$  to  $(0, 0.685)$  m and backwards. To realize this trajectory, the setpoint for each motor is calculated by using the inverse kinematics. In figure (5.2) the setpoints for the three links are displayed in the link coordinates.



Figure 5.1: Experimental setup of the test robot

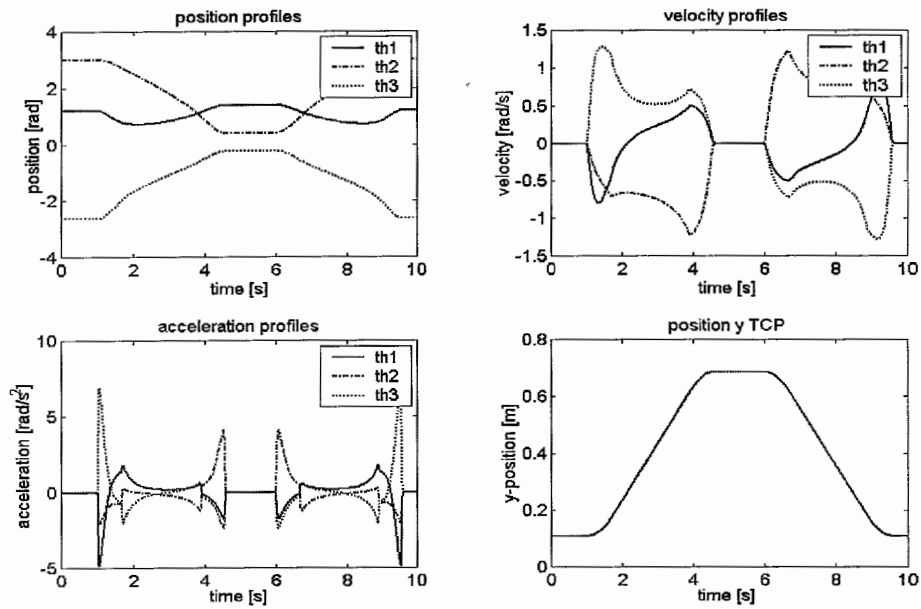


Figure 5.2: Setpoint for the three axes and y-motion of the TCP

## 5.2 Relation between encoder signals and inductive signals

When the robot is used in an industrial application only encoder signals for the motor angular positions are available. The specifications of the robot are however given in

terms of maximum error of the end-effector. In general, there will be a difference between the real error at the end-effector and the error calculated from encoder information due to the robot dynamics and the imperfections in the robot. The differences between these signals can e.g. be caused by flexibilities in the gear drives, imperfections in the gear drives or errors in the link lengths.

The encoder signals are compared with the measured end-effector error with the aid of the inductive sensor. In figure (5.3) the time-domain signals of the error are given for a typical experiment. The errors from the encoders are transformed, via the kinematics, to the error at the position of the inductive sensor.

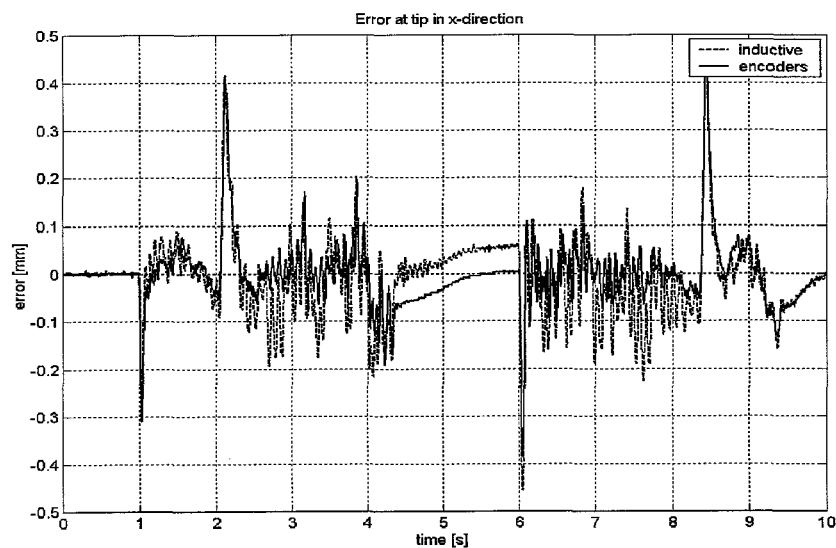


Figure 5.3: Error signals for a straight-line motion calculated from encoders and from the inductive signal

As can be seen in figure (5.3), the error signals show quite the same behavior but there is still a difference present between both signals. The difference between the two signals is a reason for concern because in the robotic system only the encoder information is available.

### 5.3 Analysis of encoder signals

The reproducible and non-reproducible parts of the encoder signals are investigated. A number of straight-line motions are made and the encoder signals are measured. These

measurements are muddled to obtain the reproducible part of the error signals.

$$e_{repr}(t) = \mu_e(t) = \frac{\sum_{i=1}^N e_i(t)}{N}, \quad N = \text{number of trials}$$

The error signal which remains after the reproducible error is subtracted from a specific trial is called the non-reproducible error (note that the non-reproducible error differs each trial).

$$e_{non-repr}(t) = e_i(t) - e_{repr}(t), \quad i = 1, 2, \dots$$

The reproducible and non-reproducible part of the error signals of each axis are shown in figure (5.4). The reproducible part of the signal is obtained after averaging over 10 trials.

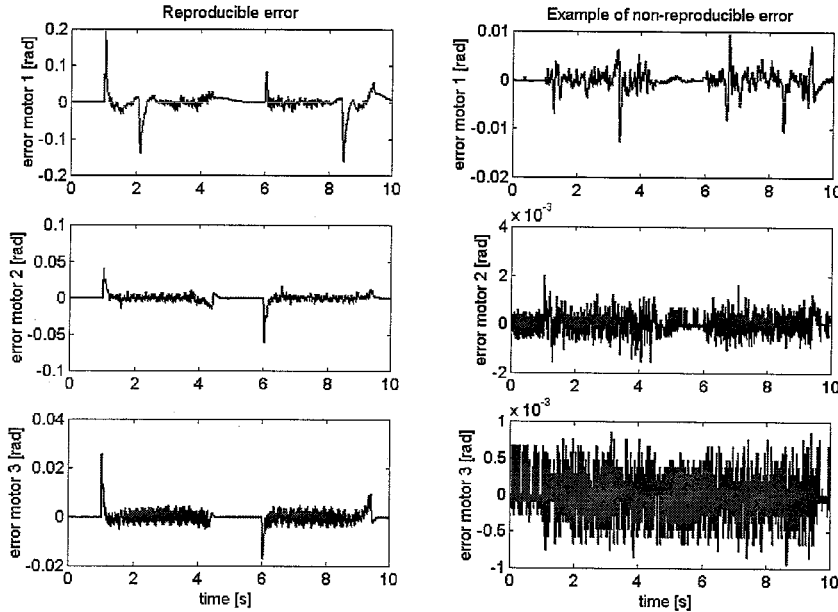


Figure 5.4: Reproducibility and non-reproducible error per axis

It is seen that only a small non-reproducible error signal remains for the error at the second and third axis. For the first axis a considerable non-reproducible error remains. Further experiments showed that for a pure rotation of the first link the same non-reproducible error remains. A probable cause of this non-reproducibility is that the robot has been used extensively in the past for experiments and the gear drive may suffer from wear. Experiments on another robot are necessary to address this problem more accurately (see Appendix D). In figure (5.5) the Power Spectral Density (PSD) is

given for the reproducible and non-reproducible signals of the first axis. In appendix C the second and third axes are analyzed.

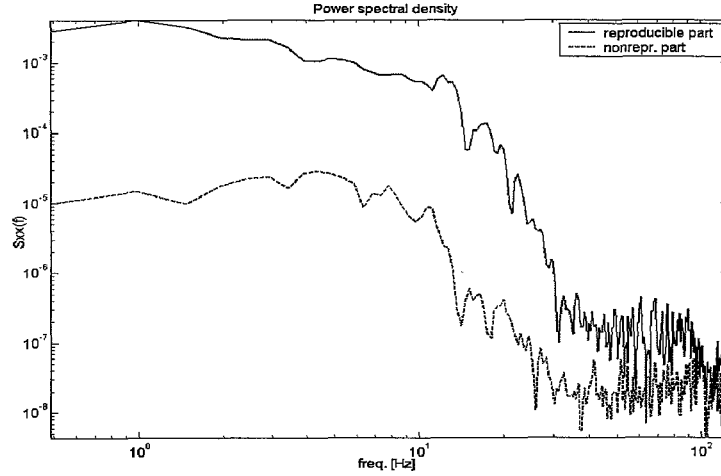


Figure 5.5: PSD of reproducible and non-reproducible error of axis 1

## 5.4 Analysis of inductive signals

In figure (5.6) the reproducible and non-reproducible part of the inductive signals is given together with the PSD of both signals.

The non-reproducible part is again small compared with the reproducible part. The non-reproducible part however shows quite a large peak at  $t \approx 3$  s. Analysis of the variance of the non-reproducible signals indeed revealed that the reproducibility at that place is bad. The robot that was used for verification (see Appendix D) did not show these large non-reproducible peaks.

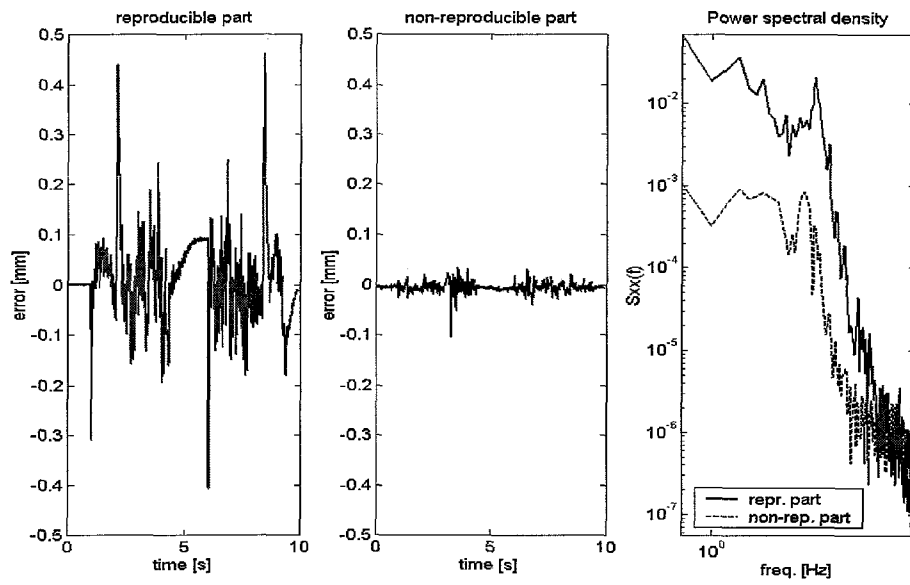


Figure 5.6: Reproducible part, non-reproducible part and PSD of inductive signals

## Chapter 6

# Controller design

### 6.1 Control strategy

The robot has to perform different straight-line movements a number of times during operation. The error signal of the end-effector has to stay within an error band of  $\pm 0.2$  mm during these motions. It is impossible to reach this accuracy with only PID controllers, separately applied to each axis, because of the nonlinear coupling torques and friction. A disadvantage of the classical control strategy with friction feedforward and inverse model feedforward is that the parameters in the feedforward signals have to be tuned for each robot and may change during operation. Furthermore, the friction feedforward had to be fine-tuned to come near to the desired accuracy. A way to overcome these problems is to use a learning control strategy. A learning controller is able to counteract reproducible errors for repetitive motions. From the error signal analysis in the previous chapter it is clear that when the error can be reduced to the level of the non-reproducible error then the desired accuracy is achieved.

The strategy is to develop a PID controller for each axis first. These controllers will stabilize the links but the specifications will not be met. Then, learning controllers will be developed that will filter the remaining error signals of the axes and that produce feedforward signals. With these feedforwards the tracking error at the motors will be decreased. The remaining error signals should result in a maximum tracking error of  $\pm 0.2$  mm of the end-effector.

### 6.2 Feedback controller

The feedback controllers for the three axes are PID controllers combined with a first order low-pass filter to suppress high frequent noise. In figure (6.1) the designed feedback controller for the first axis is given together with the open loop transfer function.



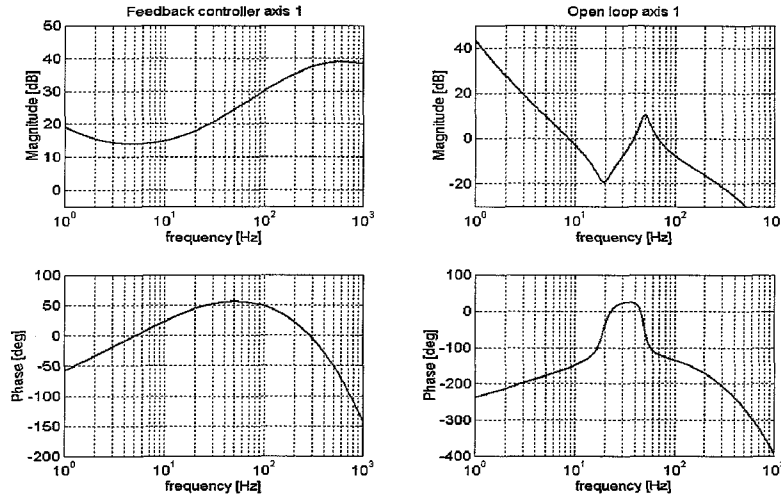


Figure 6.1: Feedback controller and open loop transfer function for axis 1

The open loop transfer function is given for the stretched configuration. The stretched configuration has the lowest bandwidth because it suffers from the anti-resonance at a lower frequency than the other arm configurations. With this feedback controller the bandwidth of the first axis is approximately 9 Hz. Stability is guaranteed for all arm configurations. For the other two axes similar PID controllers with first order low-pass filters are designed and the bandwidth of the second and third axis are respectively 30 and 40 Hz.

### 6.2.1 Homing

The feedback controllers are also used to bring the robot arm into its home position which is equal to the folded up configuration. This configuration is the point where the encoders are reset in the software. Homing is carried out simultaneously for all three axes. The home points are crossed 4 times with constant angular velocity and then averaged to obtain accurate home positions.

## 6.3 Data-based feedforward control

Feedforward control can be a valuable method to obtain a higher accuracy of a controlled system. A feedback controller just 'reacts' when an error already has occurred. When high accuracy is required one is often too late. The last two decades some new feedforward techniques are developed which can be an important supplement to the control engineers

'toolbox'. As already noted, it is very difficult to reach the desired accuracy with standard feedforward techniques because mainly the friction effects are hard to model. Therefore attention is now focused on data-based feedforward techniques because than an accurate model of e.g. friction is not required. Here a short overview is given of three data-based feedforward design techniques: Iterative Learning Control, Repetitive Control and feedforward control based on neural nets.

### 6.3.1 Iterative Learning Control

Iterative Learning Control (ILC) is a control strategy which deals with repeated motions. ILC deals with the following axioms [8]:

1. A plant repeatedly performs a specific motion that ends in a fixed duration  $T_p > 0$  [s]
2. The plant dynamics are time-invariant
3. The desired output,  $y_d(t)$   $t \in [0, T_p]$ , is given a priori
4. For each trial, the initial states are the same
5. The plant output,  $y(t)$ , is observable
6. There exists a unique input,  $u_d(t)$ , that yields  $y_d(t)$

The Iterative Learning Controller generates a feedforward signal based upon information about an error signal. The ILC scheme is depicted in figure (6.2).

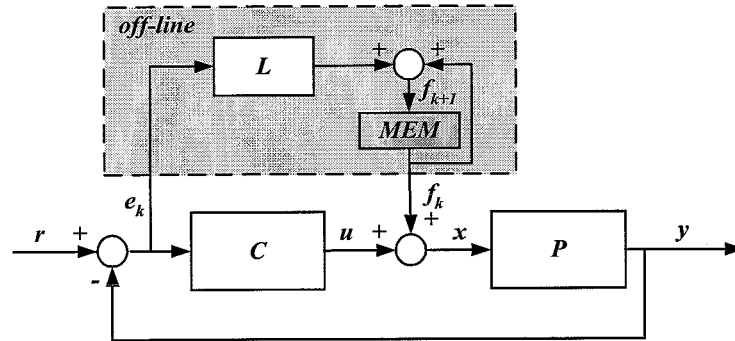


Figure 6.2: ILC scheme

To counteract non-reproducible errors and to obtain trajectory tracking during the first trials the feedback controller is included in the scheme. In a learning session a motion

is performed and the error signal is captured. After completion of the motion, the error signal is filtered with a learning filter  $L$ . The filtered signal is the feedforward signal and is stored in a memory MEM. The next time the system performs the same trajectory, the feedforward signal is implemented. When the scheme is correctly applied the feedforward signal converges after a few trials and the 'optimal' feedforward signal is found.

The update rule for the ILC controller (off-line between trials) is given by:

$$F_{k+1} = F_k + LE_k \quad (6.1)$$

where  $L$  is the learning filter. The error is given by the following relation:

$$E_k = \frac{-P}{1+PC} F_k \quad (6.2)$$

$$E_{k+1} = \frac{-P}{1+PC} F_{k+1} \quad (6.3)$$

Substituting the update rule (equation (6.1)) and equation (6.2) into equation (6.3), the following relation is found:

$$E_{k+1} = \left(1 - L \frac{P}{1+PC}\right) E_k \quad (6.4)$$

The error signal converges if:

$$E_{k+1} < E_k \quad (6.5)$$

and the convergence criterium is thus given by:

$$\left\|1 - L \frac{P}{1+PC}\right\|_{\infty} < 1 \quad (6.6)$$

### 6.3.2 Repetitive Control

The structure of the Repetitive Control (RC) scheme is similar to the ILC scheme and can be viewed as a 'continuous' Iterative Learning Controller. Repetitive Control is concerned with a periodic disturbance. The control method deals with the same axioms as ILC except for the fourth axiom. For Repetitive Control the initial states of trial  $k+1$  are equal to the end state of the  $k^{th}$  trial. Repetitive control is capable of attaining asymptotic tracking and disturbance rejection against periodic reference and disturbance signals. In the scheme (see figure (6.3)) a periodic signal generator is present e.g. 'memory loop'.

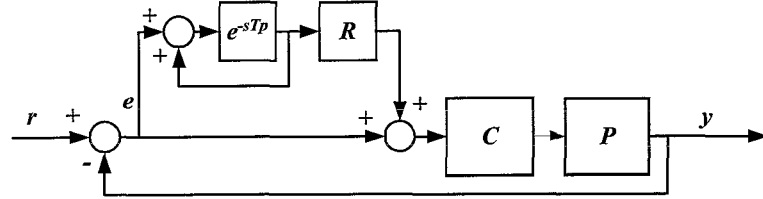


Figure 6.3: Repetitive Control scheme

The convergence criterium for this control configuration is given by:

$$\left\| 1 - R \frac{CP}{1 + CP} \right\|_{\infty} < 1 \quad (6.7)$$

Note that the stability criterium uses only amplitude information. The convergence criterium is thus some kind of small gain theorem which states that a controlled system is always stable if the loopgain is smaller than one, see e.g. [17]. The loopgain in the ILC schemes is done between the trials. In the case of modelling errors stability still has to be guaranteed and the problem can be stated in the way of the 'baby' small gain theorem, see figure (6.4).

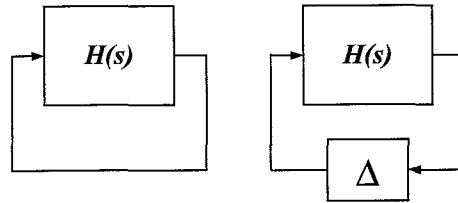


Figure 6.4: Small gain and 'baby' small gain theorem

A disadvantage of RC and ILC over standard feedforward signals, like friction feedforward and inverse model feedforward, is the trajectory dependency. A learned feedforward is only applicable to the specific trajectory.

### 6.3.3 Neural nets

This method gathers and stores information from motion operations. Using a neural network with specific structure, feedforward signals can be learned. A benefit of this method is that a feedforward signal can be computed for different motion profiles. One

of the major drawbacks of this method is that convergence proofs are difficult or even impossible to give. This can seriously hinder implementation. For a more extensive treatment of the theory and the use of neural nets or Repetitive Control the reader is referred to e.g. [8].

## 6.4 Iterative Learning Control

The robot in the reticle handling module performs a limited number of motions during operation. ILC is a method which can generate accurate feedforward signals for these motions. Moreover it enables a convergence analysis and the method in itself is quite simple. ILC can be separately applied to each link. As already mentioned, friction can show some time dependent behavior. The friction however varies so slowly in time that ILC can be applied without severe drawbacks. Because the first axis is the most challenging one, most of the attention is paid to this axis. The first axis also contributes the most to the end-effector error.

### 6.4.1 Construction of $L$ -filter

In the case of the robot, the dynamics of the first axis is workpoint dependent. This imposes some difficulties on the construction of the  $L$ -filter because now the  $L$ -filter is not univocal. We will concentrate on the construction of the  $L$ -filter for one arm configuration.

The learning filter is chosen such that it minimizes the convergence criterium (6.6):

$$L = \frac{1 + PC}{P} \quad (6.8)$$

As seen in equation (6.8), the  $L$ -filter is equal to the inverse of the Process Sensitivity. A restriction to the  $L$ -filter is that it has to be stable in order to be physically realizable. The  $L$ -filter is computed with the aid of the ZPETC-algorithm as proposed by Tomizuka [7]. The Bode plots of the Process Sensitivity and the  $L$ -filter for the first axis in folded up configuration are displayed in figure (6.5).

The controller  $C$  is equal to the feedback controller mentioned earlier. It is seen that the  $L$ -filter is an almost exact inverse of the process sensitivity. Only for high frequencies the  $L$ -filter deviates from the Process Sensitivity to obtain a proper filter. The difficulty for the first axis lies in the workpoint dependency of the transfer function. The  $L$ -filter computed for one arm configuration will not be a good inverse for the Process Sensitivity of another configuration. Applying a wrong  $L$ -filter can lead to an error growth and is thus potentially dangerous.

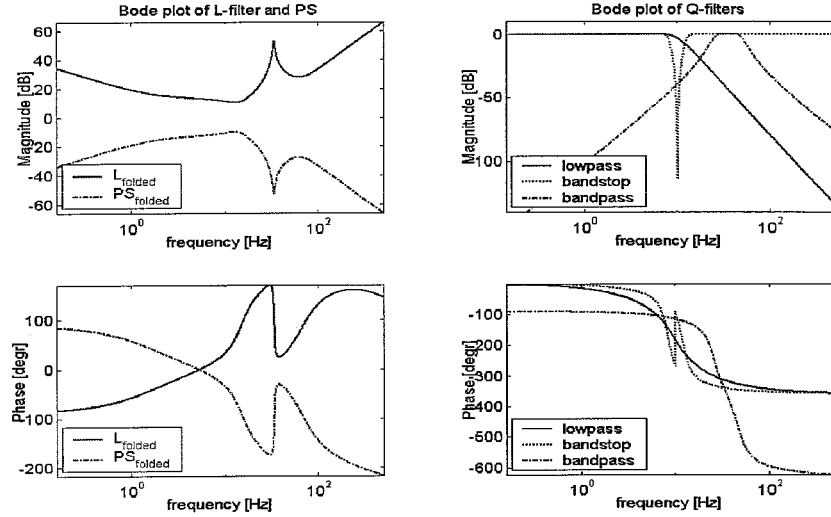


Figure 6.5: Transfer functions of Process Sensitivity and Learning filter (left) and different types of robustness filters (right)

#### 6.4.2 Convergence criterium

In the convergence criterium can be seen whether the learning process converges. The convergence criterium is given by equation (6.6). So if the convergence criterium stays within the unit circle for all frequencies then the error will decrease. If the convergence criterium is larger than one for some frequencies, then that frequencies in the error signal are amplified. For practical problems it is often desirable that the convergence criterium stays within a circle with radius smaller than one. This is because of the uncertainty in the FRF measurements of the plant. The convergence criterium is used in combination with a measured FRF to show the risk of modelling uncertainties. The results are given in figure (6.6). Here only the amplitude of the criterium is plotted.

In the left figure the stretched configuration is used as a basis for the learning filter. The criterium is plotted for the three measured FRF's for each arm configuration. In the right figure the folded up configuration is used for the learning filter. When performing a stretching motion the criterium always has to stay below one for all configurations to assure convergence. From figure (6.6) it is clear that the stretched learning filter shows convergence up to a frequency of approximately 9 Hz and beyond 35 Hz. In the case of the folded up configuration, convergence is present up to 7 Hz, between 15 and 22 Hz and beyond 42 Hz. For frequencies below 5 Hz, the criterium shows some large peaks. These are caused by nonlinear effects and disturbances in the FRF measurements below 5 Hz.

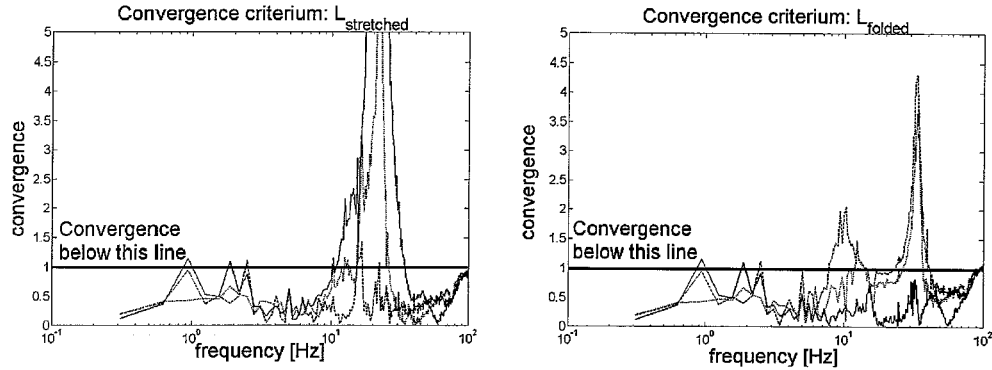


Figure 6.6: Convergence criterion when the L-filter is based on the stretched and folded up configuration. FRF folded (dot), FRF right angle (solid), FRF stretched (dash)

### 6.4.3 ILC with robustness filter

In section 6.3.1 the principle of the ILC scheme is outlined and in section 6.4.2 the convergence criterium is analyzed for the first link. In figure (6.6) it is seen that convergence is not assured over the whole frequency range up to 100 Hz. Often a robustness filter is added to the scheme when the criterium exceeds one for some frequencies. The ILC scheme with a robustness filter is given in figure (6.7).

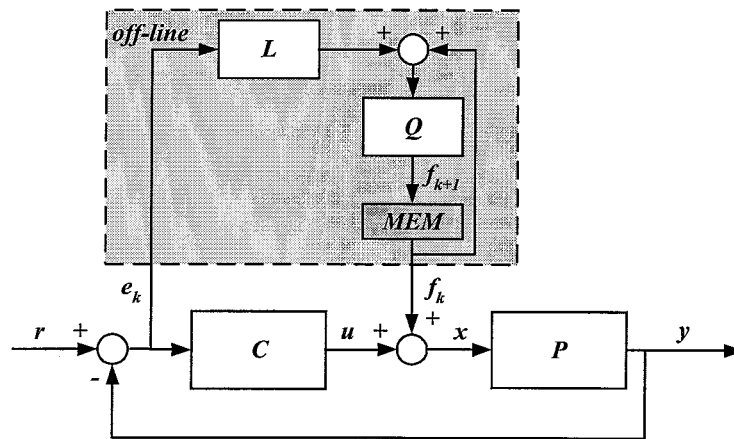


Figure 6.7: ILC scheme with robustness filter

The filter  $Q$  is a robustness filter and  $L$  is a learning filter. The update rule for the ILC

controller (off-line between trials) is now given by:

$$F_{k+1} = Q(F_k + LE_k) \quad (6.9)$$

Following the same steps as in section 6.3.1, the following relation is found:

$$E_{k+1} = Q \left( 1 - L \frac{P}{1 + PC} \right) E_k \quad (6.10)$$

and the convergence criterium is thus given by:

$$\left\| Q \left( 1 - L \frac{P}{1 + PC} \right) \right\|_{\infty} < 1 \quad (6.11)$$

For high frequencies the model is often inaccurate. Therefore the robustness filter  $Q$  is often chosen as a low-pass filter. If the model is inaccurate at other specific frequencies the  $Q$ -filter can e.g. also be chosen as a bandstop filter in combination with a low-pass filter.

A disadvantage of the application of a robustness filter is that a zero tracking error can not be achieved anymore. When learning has converged the following relation holds:

$$\lim_{k \rightarrow \infty} F_{k+1} = F_k \quad (6.12)$$

Substitution of equation (6.9) into equation (6.12) results in the following equation:

$$\lim_{k \rightarrow \infty} F_k = \frac{QL}{1 - Q} E_k \quad (6.13)$$

The relation for the error can be derived from figure (6.2):

$$E_k = \frac{1}{1 + CP} R - \frac{P}{1 + CP} F_k \quad (6.14)$$

Substituting equation (6.13) into equation (6.14) yields after some manipulations:

$$\lim_{k \rightarrow \infty} E_k = \frac{(1 - Q)}{(1 + CP)(1 - Q) + QPL} R \quad (6.15)$$

When the  $Q$ -filter has a passband unequal to one for some frequencies then a tracking error remains. Usually the  $Q$ -filter is chosen as a low-pass filter. This means that  $Q \approx 1$  for frequencies below the cut-off frequency and  $Q \approx 0$  for frequencies beyond the cut-off frequency and thus:

$$\begin{aligned} \lim_{k \rightarrow \infty} E_k &\approx 0 \quad \forall f \leq f_{cut-off} \\ \lim_{k \rightarrow \infty} E_k &\approx \frac{1}{1 + CP} R \quad \forall f > f_{cut-off} \end{aligned}$$



#### 6.4.4 Construction of Q-filter

The robustness filters are designed on the basis of the convergence criteria plots given in figure (6.6). In figure (6.5) different types of Q-filters are given. The robustness filter here is not primarily designed to cope with modelling uncertainties but to deal with the nonlinearity of the plant (varying m.m.o.i. of the load). The robustness filter is chosen as a Butterworth low-pass filter with a cut-off frequency of 9 Hz. The convergence plot for an L-filter based on the stretched configuration shows a steep increase after 9 Hz. The robustness filter is therefore chosen of seventh order to counteract this increase. When the folded up configuration is chosen as a basis for the L-filter then the robustness filter can be chosen as a combination of a bandstop filter in the range  $7 < f_{bandstop} < 15$  Hz and a low-pass filter with  $f_{low-pass} = 22$  Hz. For frequencies which are not passed by the Q-filter, convergence is assured but these frequencies are also not suppressed by the learning controller. To make the ILC scheme more robust a learning factor can be introduced after the learning filter. The system will converge slower but is made more robust for effects that should not be learned.

#### 6.4.5 Parallel filtering

The ILC theory described before is developed for linear time invariant systems. However the robot is a nonlinear system. For slow stretching and folding the robot suffers from varying fourth order dynamics (region II in figure (3.4)). Using the ILC theory based on a linear plant assumption, the performance of the ILC is severely limited. Normally the robustness filter is used to account for model uncertainty. In our case the robustness filter is used to account for the model nonlinearity. One has to use a robustness filter with a very low cut-off frequency to guarantee stability. In this way ILC becomes very conservative. A way to partly solve this problem is to use two parallel learning filters: one based on the stretched configuration and one based on the folded configuration. The scheme is displayed in figure (6.8). The convergence criterium then becomes:

$$\|Q_1(1 - L_1PS) + Q_2(1 - L_2PS)\|_\infty < 1 \quad (6.16)$$

where PS is the Process Sensitivity. Using the triangle inequality, the criterium states that convergence is assured if addition of the two separate convergence criteria is smaller than one. Accounting for the fact that we are dealing with complex vectors, vectors can partly cancel out when phase differs more than 90 degrees. In fact parallel filtering is the discrete case of the application of a scheduling technique to the learning filter. For this robot the parallel learning scheme did not reduce the error much because the L-filter based on the stretched configuration as well as the L-filter based on the folded up configuration do not show convergence for frequencies between 9 and 15 Hz.

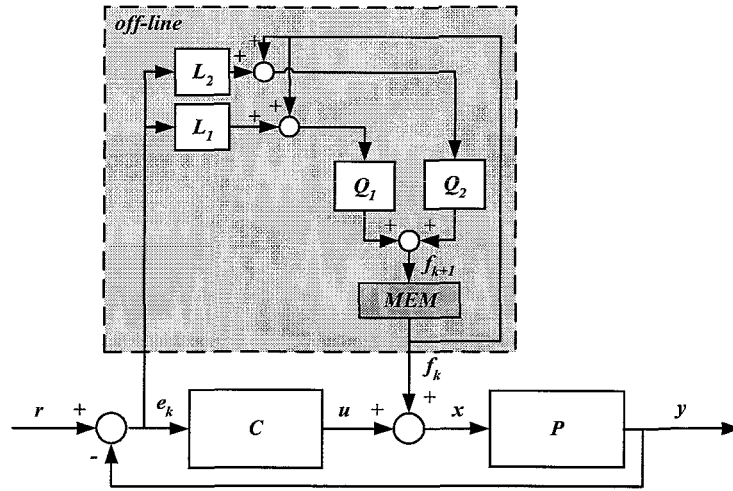


Figure 6.8: Configuration for parallel learning configuration

#### 6.4.6 Implementation aspects

The robot is controlled via a digital signal processor (dSpace DS 1103). In the MATLAB SIMULINK environment [11] block models are made and built with the Real Time Workshop. Using the MLIB/MTRACE interface libraries [14], the experiment can be handled in real time. For each trial the error signals from the encoders are captured and then filtered off-line with learning filters and robustness filters. The filtering with the learning filter is done by using the standard MATLAB function `DLSIM.M`. Filtering with the robustness filter is performed with `FILTFILT.M` (Signal Processing Toolbox). This filtering routine will filter the signal and filters it again, but then backwards. In this way phase delay will be cancelled. Also a learning factor is introduced. This will slow down the convergence process (so more trials are needed) but this makes the convergence more robust. The generated feedforward signals are then written to look-up tables. The setpoint and feedforward signal start simultaneously the next trial.

#### 6.4.7 Results

First the results are given for the error signals of the encoders. Then, the error reduction at the real output, the end-effector, is given.

In figure (6.9) the error signals of the first motor for a typical learning session are displayed.

In the first three trials the error has almost converged to its final level. The friction component has been removed. Large disturbances which are not reproducible deterio-

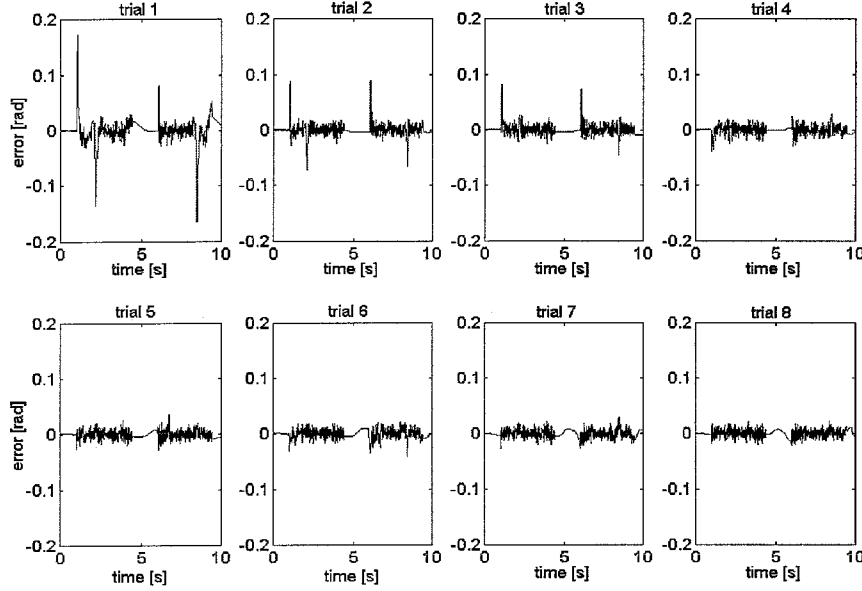


Figure 6.9: Error signals of axis 1 for a learning session

rate the convergence of the learning controller because these are learned the next trial while they should not be learned. Spectral analysis shows the strength of the learning controller. A Power Spectral Density plot of the error signals before and after learning is given in figure (6.10).

The frequencies of the error signal are decreased to the non-reproducible level for frequencies up to 9 Hz. From this figure it is also clear that the error at the encoder can still be significantly reduced if the learning controller is capable of reducing frequencies up to a higher value. The convergence of the error signal norms is shown in figure (6.11) where the 2-norm and  $\infty$ -norm are plotted versus the trial number.

The error norms can become worse for a trial but in the end they converge. In appendix C the results are shown for an iterative learning control scheme applied to the second and third axes. In figure (6.12) the obtained feedforward signal after learning is compared with a friction feedforward and inverse model feedforward. The shape of the learned feedforward is similar to the standard model feedforwards (friction feedforward and inverse model feedforward). There are however some differences. The increase of the learned feedforward at the place where friction changes from direction, is limited by the cut-off frequency of the low-pass filter. The learned feedforward also shows some anticausal anticipation, i.e. the feedforward starts before the setpoint starts. Furthermore the learned feedforward shows a low frequent ripple.

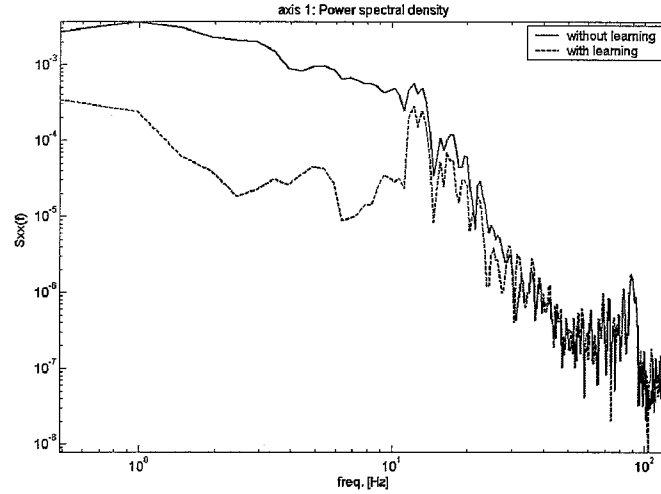
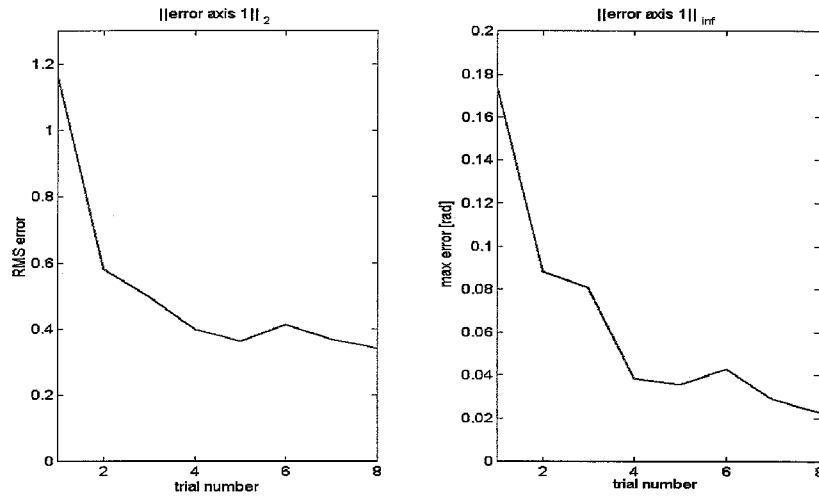


Figure 6.10: PSD of error signal from axis 1 before and after learning

Figure 6.11: 2-norm and  $\infty$ -norm for a learning session of axis 1

The real output to be minimized is the error at the end-effector. In figure (6.13) the error at the end-effector is given without learning and after 8 learning trials.

The error caused by friction, which is for a great part deductible to the first axis, is clearly reduced. During motion still a considerable error remains. In the PSD plot

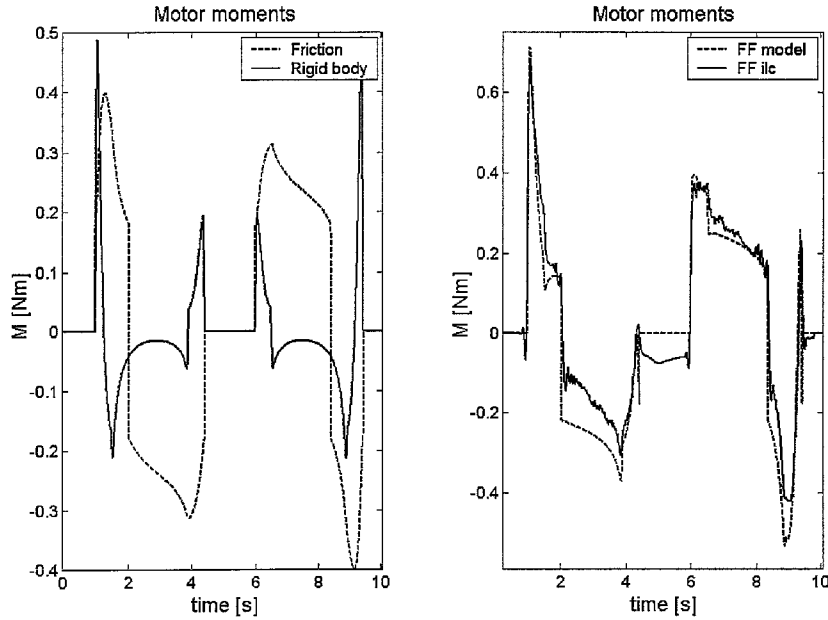


Figure 6.12: Friction feedforward and inverse rigid body model feedforward (left), combined friction feedforward and inverse rigid body model feedforward and ILC feedforward (right)

it is seen that the frequency content after learning is only marginally less than before learning. Also frequencies up to 9 Hz which are suppressed well at the motor sides are only slightly reduced. The remaining error is thus for a large part invisible at the encoder side.

In figure (6.14) the straight-line error signals calculated from encoders and measured with the inductive sensor are given. After learning has converged there is hardly any correlation between the signals anymore. In the following section this is further investigated.

## 6.5 Other control methods

As can be seen in figure (6.14) Iterative Learning Control alone is capable of reducing the error at the end-effector just within the specifications. There is however hardly any correlation between the error signal calculated from encoders and the inductive signal. In other words, accurate control of the motors does not guarantee accurate control of the end-effector.

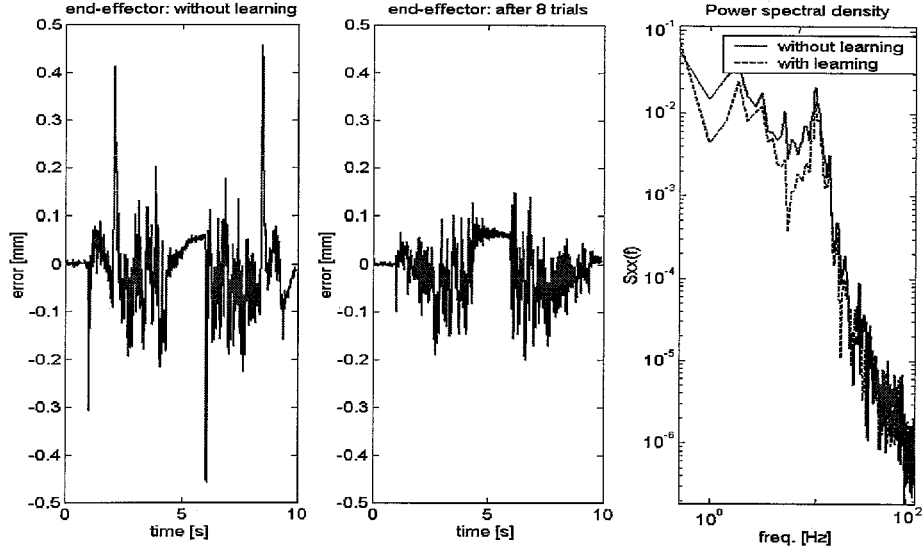


Figure 6.13: Error signal of end-effector before and after learning together with the PSD before and after learning

It is assumed that ILC has almost completely removed the contribution of the coupling torques and friction to the error. The peaks in the error signals, caused by friction and coupling torques, are clearly reduced. So only flexibility (originating from the fourth order dynamics of the system), disturbances and inaccuracies are assumed to be still considerably present. If it is possible to reconstruct the real output from the encoder measurements then the error signal could probably be further minimized.

In figure (6.15) an observer scheme is displayed which is applied to the encoder signals to reconstruct the real tip error. The observer is based on a Kalman filter. The estimated system matrices are indicated with a hat.

To evaluate the quality of reconstruction the following steps are taken. Learned feedforward signals are applied to the second and third axis to control these axes very accurately so that the tip error is mainly caused by the first axis. The motor signal of the first axis is then fed into the estimated linear model. Tracking of the error is established by feeding back the innovation signal:  $y - \hat{y}$  (real motor error - estimated motor error). The tracking can be further improved by also applying a low frequent feedforward to the first link which cancels the nonlinearities due to friction and coupling torques. This is possible because a 'natural' decoupling is present in the frequency domain between vibrations arising from nonlinear phenomena and linear phenomena. The nonlinear phenomena, friction and coupling torques, are mainly low frequent (frequencies  $< 6$  Hz) for the straight-line movement of 200 mm/s. The frequencies which describe the flexible

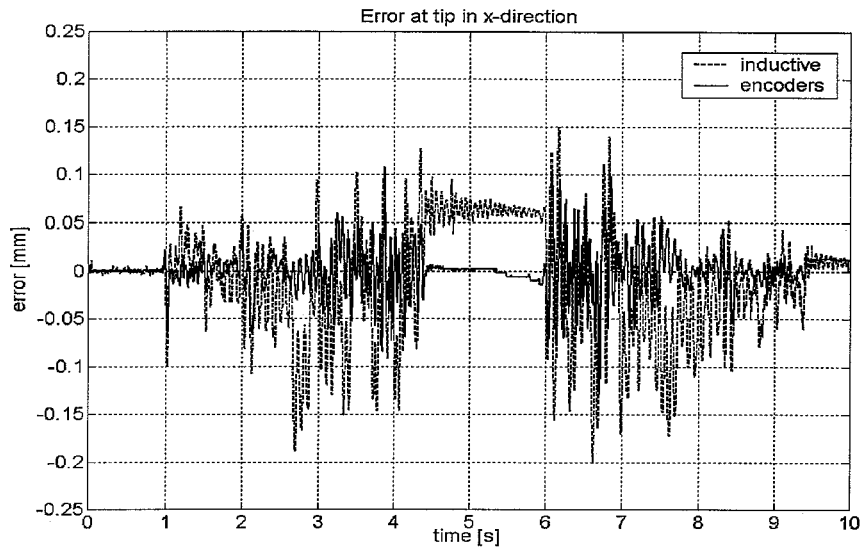


Figure 6.14: Error at end-effector calculated from encoders and from the inductive signal

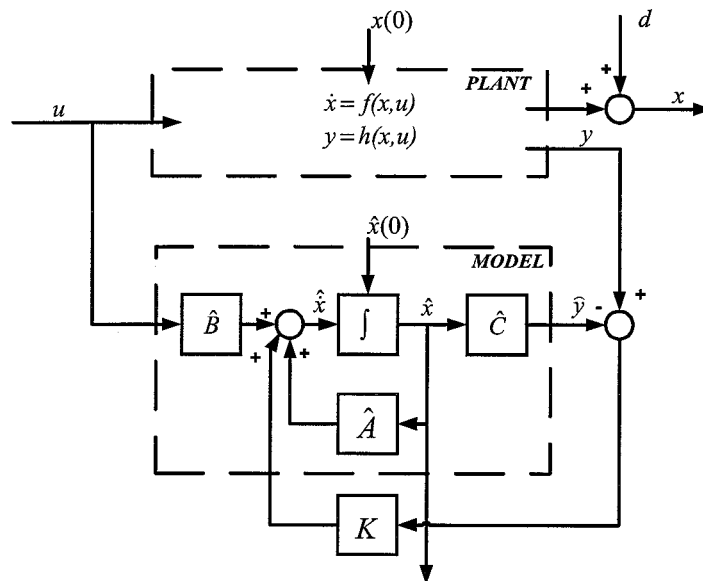


Figure 6.15: Linear observer scheme

behavior are of higher frequency content (frequencies between approximately 10 and 40 Hz).

Experiments however showed that it is impossible to reconstruct the tip position accurately. Effects like inaccuracy in the gear drive and hysteresis can not be predicted and are at this accuracy level dominantly present. Setpoints with different straight-line velocities revealed that a vibration in the error signal of the end-effector is present which is proportional to the angular velocity of the motor. This makes it impossible to design a controller which is based on the reconstructed tip error. Faster setpoints are expected to cause problems in reaching the desired accuracy. Vibrations at the anti-resonance frequencies of the motor side will be excited more severely while they are hardly visible at the encoders.

## 6.6 Concluding remarks

Now the different control methods are compared and implementation aspects are discussed. With only PID controllers applied to each link, the robot arm can be controlled but the specifications are not met. The maximum straight-line error at the end-effector is approximately 0.50 mm. This error occurs at the start of a motion in the stretched configuration.

In [1] the application of friction feedforward and inverse model feedforward is treated. With these feedforwards already a large error reduction was achieved. The maximum error at the end-effector could be reduced a factor two to 0.25 mm. However, these feedforwards had to be fine-tuned to reach this accuracy and it is questionable if this accuracy can be held during operation. Furthermore the tuning of different robots is a time consuming task and sensitive to model errors.

With the ILC approach the error could be further reduced to 0.20 mm. Again this error occurred during the start of the motion in the stretched configuration. With this accuracy level also other parts of the motion trajectory become critical. The inaccuracies in the gear drives and effects like hysteresis cause an error with an amplitude of approximately  $50\text{ }\mu\text{m}$  at the end-effector over the entire trajectory. This error is independent of the applied setpoint. The ILC algorithm is however more flexible than hand tuned feedforwards. Once the algorithm is developed, feedforward signals can be generated for each robot by performing some trials.

### 6.6.1 Implementation issues

Some steps have to be taken to successfully implement an Iterative Learning Controller. First, feedback controllers have to be designed for all links which stabilize the links for all robot arm configurations and which are robust. A learning controller can be implemented on each link and can compute feedforward signals for the specific trajectories



of the robot. The ILC controller also has to be 'robust' in the sense that it computes feedforward signals which assure convergence of the error signals. There are two possible ways to deal with 'robustness' of the ILC algorithm. The convergence criterium can be sharpened by requiring for all possible arm configurations:

$$\|Q(1 - LPS)\|_{\infty} < \alpha, \alpha < 1 \quad (6.17)$$

The factor  $\alpha$  is introduced to account for deviations from the nominal plant model. The deviations are e.g. unmodelled dynamics and uncertainty in the nominal plant model. The passband of the robustness filter  $Q$  can be chosen such that the criterium (6.17) is fulfilled. This method is rather ad lib and it is hard to give an upper bound for allowable perturbations of the plant.

Another method is to take deviations from the nominal plant model explicitly into account. The convergence criterium (equation (6.11)) can be viewed in an  $H_{\infty}$  framework, see e.g. [17]. In figure (6.16) this analogy is displayed [9].

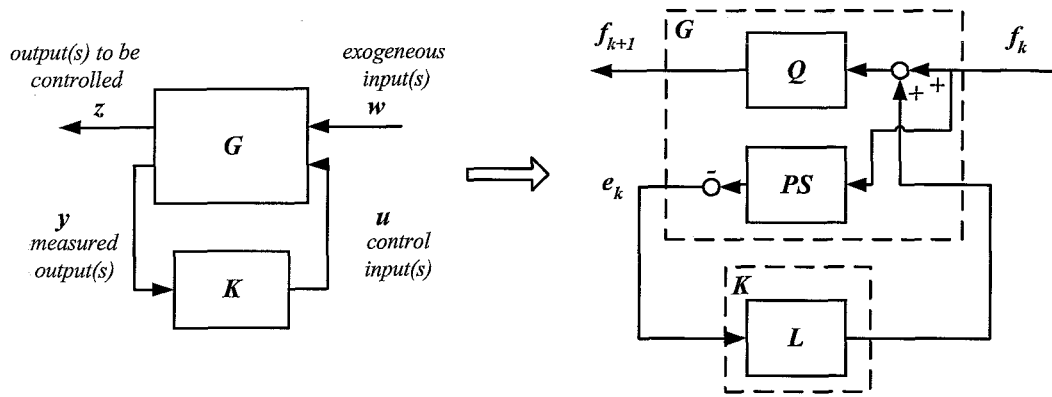


Figure 6.16: Standard plant (left) and convergence criterium in standard plant configuration (right)

In this configuration the deviations from the model can be incorporated in the model of the Process Sensitivity ( $PS$ ) and thus into the standard plant  $G$ . When the model deviations are grouped into the  $\Delta$ -block, the augmented plant is represented in figure (6.17).

The convergence criterium can now be stated into an  $H_{\infty}$  control problem in which a learning filter is designed while model deviations are taken explicitly into account. The position dependent inertia of the load can be taken into account by modelling it e.g. as a parameter uncertainty. For a more thorough treatment of this approach is referred to [9].

When the error has converged after a few trials, the generated feedforward signals can be used as standard feedforward signals. On the background an observer has to trigger

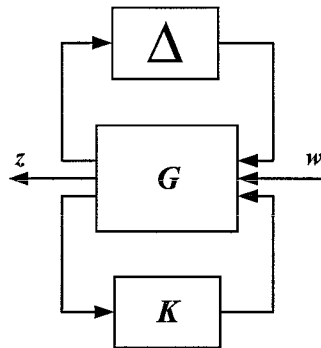


Figure 6.17: Augmented plant with pulled out uncertainty block

the error signals during operation of the robot. If after some time an error signal increases above some specified level, the learning algorithm can be evoked to recompute a feedforward signal. Additional security measures will be necessary to evaluate the error signals in the proper sense. It is e.g. not wanted to learn a disturbance which occurred only once and is thus not reproducible.



## Chapter 7

# Conclusions and recommendations

### 7.1 Conclusions

- The maximum straight-line error, measured at the end-effector, is just within specification. Verification measurements on another robot yielded better results and the error could be reduced far within the specified error band of 0.2 mm.
- For the robot with the given measurement situation it is difficult to increase the straight-line accuracy with control actions any further. At this accuracy level, the inaccuracies can for a large extent be deducted to the imperfections in the gear drive.
- Iterative Learning Control is a powerful method when the plant can be accurately described by a linear model. This is demonstrated by the error reduction which could be achieved for the second and third axes.
- The Iterative Learning Control scheme is combined with a parallel filtering technique to deal with the smooth nonlinearity due to varying fourth order dynamics. The advantage of this approach for this robot was however negligible.
- Faster setpoints are expected to cause problems to stay within the desired accuracy band because the flexibilities in the robot are excited more severely. No proper control actions can be taken to counteract these vibrations because it is impossible to reconstruct the end-effector position.

## 7.2 Recommendations

- Improve the measurement situation

An effective way is to place the encoders on the load side of each link. Especially the placement of the encoder of the first link on the load side will be beneficial. It will probably be more difficult to place the encoders on the load side of each link than on the motor side of the links but the advantages are:

1. The bandwidth of the feedback controller for the first link can probably be increased. The feedback controller is not limited by the varying anti-resonance frequency of this link.
2. The Iterative Learning Controller is probably capable of obtaining a higher 'bandwidth'. Convergence can probably be assured for higher frequencies.
3. The whole train of inaccuracies between the measured output and the end-effector becomes shorter. Inaccuracies in the gear drive, which cause the largest error, are visible on the encoder.

- Change of robot construction

If it is impossible to place the encoders on the load side, the problem of the dominant low frequent anti-resonance of the first link could have been avoided in the design stage. Examples are a more accurate gear drive and a stiffer gear drive. A stiffer gear drive places the anti-resonance of the transfer function from motor current to motor angle to a higher frequency. The bandwidth of the feedback controller and learning controller can then be increased.

- If faster setpoints are required it is recommended to design these setpoints such that the accelerations at the start and end of a motion are not increased. The most critical parts of the trajectory are at the start and end of a motion where friction and high coupling torques are present. The speed in the middle of the straight-line trajectory can probably be increased without introducing much error amplification.

# Acknowledgments

This report is the result from my Master Thesis project in the period from April 2001 until February 2002. This Master Thesis project is carried out at the development company CCM in Nuenen, the Netherlands. During this project I was able to combine theory with practice from which I gained a lot of insight into the field of control engineering. I would like to thank Erik Kremers and John Hazenberg from the company CCM for their supervision. I would also like to thank Hans Sluiters, projectleader of the Reticle Handling project, who made this research possible and Henk Huisman for his useful comments on the report. Furthermore I thank Maarten Steinbuch and René van de Molengraft for their supervision from the Dynamics and Control Technology group at TU/e.

Nuenen, February 2002



# Bibliography

- [1] E.A.F. Kremers, *Robodyn*, MSc. Thesis, CCM: Robodyn-final-010115, January 2000.
- [2] J. Hazenberg, *EPS Reticle Handling Robot*, Internal report CCM/ASML, December 1999.
- [3] S.G.M. Hendriks, *Iterative Learning Control on the H-drive*, MSc. Thesis Eindhoven University of Technology, November 2000.
- [4] M. Steinbuch and A.J.J. v.d. Weiden, *Repetitive and Learning Control*, Lecture notes of PATO course, April 2000.
- [5] A. De Luca, G. Paesano and G. Ulivi, *A Frequency-Domain Approach to Learning Control: Implementation for a Robot Manipulator*, IEEE Transactions on Industrial Electronics, vol 39, February 1992.
- [6] J. Moon, T. Doh and M.J. Chung, *An Iterative Learning Control Scheme for Manipulators*, IEEE, Proc. IROS 97.
- [7] M. Tomizuka, *Zero phase error tracking algorithm for digital control*, Journal of Dynamic Systems, Measurement and Control, March 1987.
- [8] W.J.R. Velthuis, *Learning Feed-Forward Control*, Phd. Thesis, The University of Twente 2000.
- [9] D. d. Roover, *Motion Control of a Wafer Stage*, Phd Thesis Delft University of Technology, 1997.
- [10] G.F. Franklin, J.D. Powell and M. Workman, *Digital Control of Dynamic Systems*, Addison Wesley Inc., 3<sup>rd</sup> edition, 1997.
- [11] The Mathworks Inc, *SIMULINK, Dynamic System Simulation for MATLAB*, Using Simulink Version 3, The Mathworks Inc, January 1999.
- [12] A. Sauren, *Multibody dynamics*, Lecture notes Eindhoven University of Technology, September 1991.



- [13] A. d. Kraker, *A numerical-experimental approach in structural dynamics*, Lecture notes Eindhoven University of Technology, August 1999.
- [14] dSPACE, *MATLAB-dSPACE Interface Libraries*, May 1999.
- [15] dSPACE, *ControlDesk: Experiment Guide*, September 1999.
- [16] DSP Technology Inc, *SigLab V3.0 Measurement Software*, DSP Technology Inc, Terrace, 2000.
- [17] A. Damen and S. Weiland, *Robust Control*, Lecture notes Eindhoven University of Technology, November 2000.
- [18] A. Isidori, *Nonlinear control systems*, Springer-Verlag, Berlin, 3<sup>rd</sup> edition, 1995.
- [19] A.M. Rankers, *Machine dynamics in mechatronic systems*, Phd. Thesis, Philips Electronics N.V., 1997.
- [20] A. Vandenput, *Electromechanica*, Lecture notes Eindhoven University of Technology.

## Appendix A

# Kinematics

The kinematics can be described in four different coordinate frames: relative coordinates, Cartesian coordinates, motor coordinates and coordinates as given by the manufacturer: Sankyo Seiki Ltd. First, the kinematics are described in relative coordinates. Then, the kinematics are described in other coordinate frames using transformations between these coordinate frames.

### A.1 Kinematic modelling as a chain of bodies

In the following sections, the symbols are defined as in figure (2.3). The vectors  $\vec{b}$  are body vectors and  $\vec{c}$  are vectors of the connecting elements. The vectors  $\vec{r}$  are position vectors from the origin to the point 0 in a body. Usually, the position vectors point from origin to the centre of mass of a body.

#### A.1.1 Position vectors

For the positions the following relations hold:

$$B^0 : \quad \vec{r}_0 = \vec{b}_{01} \quad (\text{A.1})$$

$$B^1 : \quad \vec{r}_1 = -\vec{b}_{11} + \vec{c}_1 \quad (\text{A.2})$$

$$B^2 : \quad \vec{r}_2 = \vec{r}_1 + \vec{b}_{12} + \vec{c}_2 - \vec{b}_{22} \quad (\text{A.3})$$

$$B^3 : \quad \vec{r}_3 = \vec{r}_2 + \vec{b}_{23} + \vec{c}_3 - \vec{b}_{33} \quad (\text{A.4})$$

The elements which connect the bodies are line hinges. The relation for the position vectors in the hinges is given by:

$$\vec{c}_i = \underline{\vec{e}}_i^T \begin{bmatrix} 0 \\ 0 \\ z_i \end{bmatrix}, \quad i = 1, 2, 3 \quad (\text{A.5})$$

If the z-axis is also operational then  $\vec{c}_1 = z_1(t)$ . The global basis can then e.g. be located on  $(x, y, z) = (0, 0, 0)$ . Here, all  $z_i$  are taken zero. The body related vectors  $\vec{b}$  are described in their local basis.

### A.1.2 Rotation

The orientation is described by means of rotation tensors:

$$B_0 : \mathbf{R}_0 = \mathbf{I} \quad (\text{A.6})$$

$$B_1 : \mathbf{R}_1 = \mathbf{C}_1 \quad (\text{A.7})$$

$$B_2 : \mathbf{R}_2 = \mathbf{C}_2 \cdot \mathbf{C}_1 \quad (\text{A.8})$$

$$B_3 : \mathbf{R}_3 = \mathbf{C}_3 \cdot \mathbf{C}_2 \cdot \mathbf{C}_1 \quad (\text{A.9})$$

Where the orientation of a vector basis  $\vec{e}$  relative to a vector basis  $\vec{e}_0$  is described by means of an orthogonal rotation tensor  $\mathbf{R}$ :

$$\underline{\vec{e}}^T = \mathbf{R} \cdot \underline{\vec{e}_0}^T \quad (\text{A.10})$$

The matrix representation of the rotation tensor is given by:

$$\text{tensor} \rightarrow \text{matrix} : \underline{\mathbf{A}} = \underline{\vec{e}} \cdot \mathbf{A} \cdot \underline{\vec{e}}^T \quad (\text{A.11})$$

$$\text{matrix} \rightarrow \text{tensor} : \mathbf{A} = \underline{\vec{e}}^T \underline{\mathbf{A}} \underline{\vec{e}} \quad (\text{A.12})$$

The rotation matrices are given by:

$$\underline{\mathbf{C}}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad i = 1, 2, 3 \quad (\text{A.13})$$

This results in the following components of the position vectors (the vector from the global basis to the desired point) where all components of the body vectors are expressed in their local basis:

$$\begin{aligned} \underline{r}_1 &= \underline{r}_1(\theta_1) &= -\underline{\mathbf{C}}_1(\theta_1)\underline{b}_{11} \\ \underline{r}_2 &= \underline{r}_2(\theta_1, \theta_2) &= \underline{r}_1 + \underline{\mathbf{C}}_1(\theta_1)(\underline{b}_{12} - \underline{\mathbf{C}}_2(\theta_2)\underline{b}_{22}) \\ \underline{r}_3 &= \underline{r}_3(\theta_1, \theta_2, \theta_3) &= \underline{r}_2 + \underline{\mathbf{C}}_1(\theta_1)\underline{\mathbf{C}}_2(\theta_2)(\underline{b}_{23} - \underline{\mathbf{C}}_3(\theta_3)\underline{b}_{33}) \end{aligned} \quad (\text{A.14})$$

### A.1.3 Angular velocity

All angular velocities of the bodies point in the  $\vec{e}_0$  z-direction. The angular velocities of the bodies are (the angular velocity of a body is the same for all points in the rigid body, the absolute velocity can however differ from point to point):

$$\vec{\omega}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix}^T \underline{\vec{e}}_0, \quad \vec{\omega}_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{bmatrix}^T \underline{\vec{e}}_0, \quad \vec{\omega}_3 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix}^T \underline{\vec{e}}_0 \quad (\text{A.15})$$

#### A.1.4 Position, velocity and acceleration of the centre of mass

The position, velocity and acceleration of the centre of mass of the three links are given by ( $d$  represents the distance from a line hinge to the centre of mass of the body):

$$\begin{aligned}\vec{r}_{M1} &= \begin{bmatrix} d_1 \cos(\theta_1) \\ d_1 \sin(\theta_1) \\ z_1 \end{bmatrix} \vec{e}_0 \\ \dot{\vec{r}}_{M1} &= \begin{bmatrix} -d_1 \sin(\theta_1) \dot{\theta}_1 \\ d_1 \cos(\theta_1) \dot{\theta}_1 \\ \dot{z}_1 \end{bmatrix} \vec{e}_0 \\ \ddot{\vec{r}}_{M1} &= \begin{bmatrix} -d_1 \left( \cos(\theta_1) \dot{\theta}_1^2 + \sin(\theta_1) \ddot{\theta}_1 \right) \\ d_1 \left( -\sin(\theta_1) \dot{\theta}_1^2 + \cos(\theta_1) \ddot{\theta}_1 \right) \\ \ddot{z}_1 \end{bmatrix} \vec{e}_0\end{aligned}$$

$$\begin{aligned}\vec{r}_{M2} &= \begin{bmatrix} \cos(\theta_1)(d_1 + d_2) + d_3 \cos(\theta_1 + \theta_2) \\ \sin(\theta_1)(d_1 + d_2) + d_3 \sin(\theta_1 + \theta_2) \\ \dot{z}_1 + \dot{z}_2 \end{bmatrix} \vec{e}_0 \\ \dot{\vec{r}}_{M2} &= \begin{bmatrix} -\sin(\theta_1)(d_1 + d_2) \dot{\theta}_1 - d_3 \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \\ \cos(\theta_1)(d_1 + d_2) \dot{\theta}_1 + d_3 \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \\ z_1 + z_2 \end{bmatrix} \vec{e}_0 \\ \ddot{\vec{r}}_{M2} &= \begin{bmatrix} -(d_1 + d_2) \left( \cos(\theta_1) \dot{\theta}_1^2 + \sin(\theta_1) \ddot{\theta}_1 \right) - \\ d_3 \left( \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 + \sin(\theta_1 + \theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2) \right) \\ (d_1 + d_2) \left( -\sin(\theta_1) \dot{\theta}_1^2 + \cos(\theta_1) \ddot{\theta}_1 \right) + \\ d_3 \left( -\sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 + \cos(\theta_1 + \theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2) \right) \\ \ddot{z}_1 + \ddot{z}_2 \end{bmatrix} \vec{e}_0\end{aligned}$$

$$\begin{aligned}
\vec{r}_{M3} &= \begin{bmatrix} (d_1 + d_2) \cos(\theta_1) + (d_3 + d_4) \cos(\theta_1 + \theta_2) + d_5 (\cos(\theta_1 + \theta_2 + \theta_3)) \\ (d_1 + d_2) \sin(\theta_1) + (d_3 + d_4) \sin(\theta_1 + \theta_2) + d_5 (\sin(\theta_1 + \theta_2 + \theta_3)) \\ z_1 + z_2 + z_3 \end{bmatrix} \vec{e}_0 \\
\dot{\vec{r}}_{M3} &= \begin{bmatrix} -(d_1 + d_2) \sin(\theta_1) \dot{\theta}_1 - (d_3 + d_4) \sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) \\ -d_5 \sin(\theta_1 + \theta_2 + \theta_3) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ (d_1 + d_2) \cos(\theta_1) \dot{\theta}_1 + (d_3 + d_4) \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2) + \\ d_5 \cos(\theta_1 + \theta_2 + \theta_3) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3) \\ \dot{z}_1 + \dot{z}_2 + \dot{z}_3 \end{bmatrix} \vec{e}_0 \\
\ddot{\vec{r}}_{M3} &= \begin{bmatrix} -(d_1 + d_2) \left( \cos(\theta_1) \ddot{\theta}_1^2 + \sin(\theta_1) \ddot{\theta}_1 \right) - \\ (d_3 + d_4) \left( \cos(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 + \sin(\theta_1 + \theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2) \right) - \\ d_5 \left( \cos(\theta_1 + \theta_2 + \theta_3) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + \sin(\theta_1 + \theta_2 + \theta_3) (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3) \right) \\ (d_1 + d_2) \left( -\sin(\theta_1) \dot{\theta}_1^2 + \cos(\theta_1) \ddot{\theta}_1 \right) + \\ (d_3 + d_4) \left( -\sin(\theta_1 + \theta_2) (\dot{\theta}_1 + \dot{\theta}_2)^2 + \cos(\theta_1 + \theta_2) (\ddot{\theta}_1 + \ddot{\theta}_2) \right) + \\ d_5 \left( -\sin(\theta_1 + \theta_2 + \theta_3) (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 + \cos(\theta_1 + \theta_2 + \theta_3) (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3) \right) \\ \ddot{z}_1 + \ddot{z}_2 + \ddot{z}_3 \end{bmatrix} \vec{e}_0
\end{aligned}$$

## A.2 Transformations to other coordinate frames

### A.2.1 Transformation between relative and Cartesian coordinates

The relative coordinates and the Cartesian coordinates are related via a polar coordinate description. It follows that  $r = r(\theta_2)$  and  $\psi = \psi(\theta_1, \theta_2)$ , see figure (A.1). From figure (A.1) also the redundancy is apparent.

The transformation between the first two relative coordinates and the Cartesian coordinates is then given by:

$$\begin{bmatrix} x_{TCP} \\ y_{TCP} \end{bmatrix} = \begin{bmatrix} L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin(\theta_1) + L_2 \sin(\theta_1 + \theta_2) \end{bmatrix}$$

where  $L$  represents the length of the links (measured from line hinge to line hinge). The inverse transformation, from the Cartesian coordinates of the TCP to the relative coordinates, is given by (there are two solutions possible due to the redundancy):

$$\begin{bmatrix} \theta_2 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \pm \left( -\arccos\left(\frac{L_1^2 + L_2^2 - r^2}{2L_1L_2}\right) + \pi \right) \\ \psi - \arcsin\left(\frac{L_2 \sin(\pi - \theta_2)}{r}\right) \end{bmatrix}$$

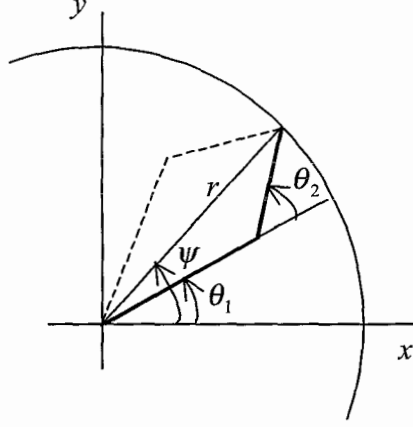


Figure A.1: Relation between polar coordinates and relative coordinates

where  $r = \sqrt{x_{TCP}^2 + y_{TCP}^2}$  and  $\psi = \begin{cases} \arctan(\frac{y_{TCP}}{x_{TCP}}) & \text{if } x_{TCP} \geq 0 \\ \arctan(\frac{y_{TCP}}{x_{TCP}}) + \pi & \text{if } x_{TCP} < 0 \end{cases}$

The redundancy can mathematically be removed by always choosing the smallest angle for  $\theta_1$ . The relation between the roll angle  $R$  and the relative coordinates is given by:

$$R = \begin{bmatrix} -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} + \frac{\pi}{2}$$

### A.2.2 Transformation between relative and motor coordinates

The relation between the relative coordinates and the motor coordinates is given by:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \begin{bmatrix} N_{11} & 0 & 0 \\ 0 & N_{22} & N_{23} \\ 0 & 0 & N_{33} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

where the gear ratios are given by the numbers  $N$ :

$$N_{11} = 161, N_{22} = \frac{3621}{23}, N_{33} = \frac{3234}{31}, N_{23} = \frac{1}{49}$$

This transformations is linear. The inverse transformation is computed by simply inverting the transformation matrix.



## Appendix B

# Rigid body dynamics

### B.1 Introduction

In this appendix the dynamic equations are derived. The equations of motion are derived to obtain insight in the forces and torques which play a role in the dynamic behavior of the system. The modelling is kept as simple as possible by not including too many bodies in the rigid body modelling.

### B.2 Newton-Euler approach

The dynamic equations will be derived by using the Newton-Euler approach. The Newton-Euler approach is chosen because this leads to a good understanding of the present forces and torques and the motion these forces and torques cause. The approach consists of 5 steps:

1. Choose reference coordinate frames
2. Isolate each body from its surrounding and draw all forces and torques on it  
note: all forces and torques have to be divided into applied loads and connection loads
3. Calculate the forces and torques relative to the centre of mass per body
4. Use Newton's second law and the angular momentum equations:

$$\vec{F} = m\ddot{\vec{r}}_M \quad (\text{B.1})$$

$$\begin{aligned} \vec{M}_M &= \dot{\vec{L}}_M \\ &= \vec{\omega} \times (\mathbf{J}_M \cdot \vec{\omega}) + \mathbf{J}_M \cdot \dot{\vec{\omega}} = \dot{H}_M + \mathbf{J}_M \cdot \dot{\vec{\omega}} \end{aligned} \quad (\text{B.2})$$



where  $m$  is the mass of the link,  $\vec{r}_M$  is the position vector from the global basis to the centre of mass of the link and  $J_M$  is the rotational inertia of the link about its centre of mass.

5. Combine the equations for all bodies. We obtain  $2nb$  ( $nb$  = number of bodies) vector differential equations
  - often one is not interested in the connecting loads. These can be eliminated from the differential equations.

### B.3 Body forces and torques

In this section, the loads on the three bodies are investigated. For the figures the following relations hold (see figure (2.3)):

$$\vec{d}_1 = \vec{b}_{11} - \vec{c}_1, \vec{d}_2 = \vec{b}_{22}, \vec{d}_3 = \vec{b}_{23} - \vec{c}_2, \vec{d}_4 = \vec{b}_{33}, \vec{d}_5 = \vec{b}_{34} - \vec{c}_3$$

where the body vectors point to the centre of mass of the links. The resulting torques of the actuators are described by  $\vec{F}_{TOT1}$ ,  $\vec{M}_{TOT1}$ ,  $\vec{M}_{TOT2}$ ,  $\vec{M}_{TOT3}$ .  $\vec{F}_{TOT1}$  describes the force of the z-motor and  $\vec{M}_{TOT}$  is the torque of a motor.

#### B.3.1 World

The real world can be seen as a body with infinite mass and moment of inertia. It is capable of taking all forces and torques acting on it without showing any acceleration. The forces and torques which act on the real world are drawn in figure (B.1).

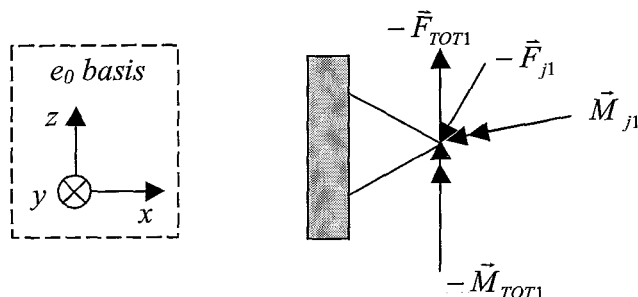


Figure B.1: Forces and moments on the world

The force  $\vec{F}_j$  is the force on the joint and  $\vec{M}_j$  is the moment on the joint.

### B.3.2 Body 1

The forces and torques on body 1 are depicted in figure (B.2).

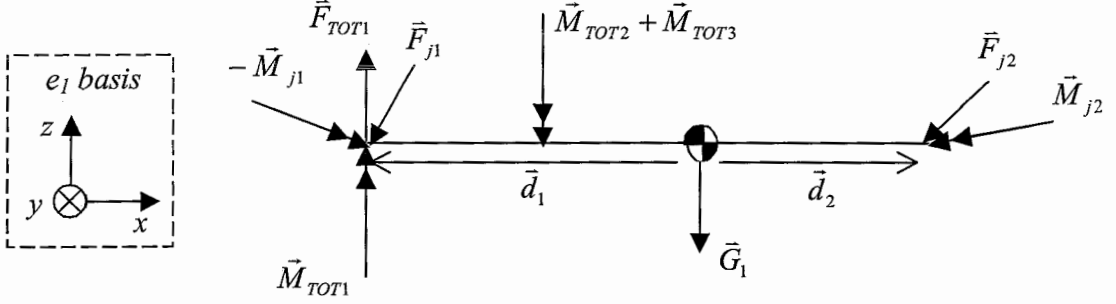


Figure B.2: Forces and moments on body 1

The basis is chosen as a right hand orientated basis with positive directions as given in figure (B.2). Because the motors and gear trains are not considered as bodies, the torque at the place of the motors is equal to the torques at the joints of body 2 and body 3.

The resulting force and torque about the centre of mass:

$$\vec{F} = \sum \vec{F}_{con.} + \sum \vec{F}_{appl.} \quad (B.3)$$

$$= (-\vec{F}_{j1} + \vec{F}_{j2}) + (-\vec{G}_1 + \vec{F}_{TOT1}) \quad (B.4)$$

$$\vec{M}_M = \sum (\vec{M}_{con.} + \vec{b} \times \vec{F}_{con.}) + \sum (\vec{M}_{appl.} + \vec{b} \times \vec{F}_{appl.}) \quad (B.5)$$

$$= (-\vec{M}_{j1} + \vec{M}_{j2} + \vec{d}_1 \times \vec{F}_{j1} + \vec{d}_2 \times \vec{F}_{j2} - \vec{d}_1 \times \vec{F}_{TOT1}) + \quad (B.6)$$

$$(\vec{M}_{TOT1} - \vec{M}_{TOT2} - \vec{M}_{TOT3}) \quad (B.7)$$

Because the gravity points through the centre of gravity, the torque is zero.

Newtons second law:

$$\begin{aligned} \vec{F} &= m \ddot{\vec{r}}_M \\ -\vec{F}_{j1} + \vec{F}_{j2} - \vec{G}_1 + \vec{F}_{TOT1} &= m_1 \ddot{\vec{r}}_{M1} \end{aligned}$$

and the impulse moment equation:

$$\begin{aligned} \vec{M}_M &= \dot{\vec{L}}_M \\ &= \dot{H}_M + \mathbf{J}_M \cdot \dot{\vec{\omega}} \\ &= \mathbf{J}_{M1} \cdot \dot{\vec{\omega}}_1 \end{aligned}$$

### B.3.3 Body 2

The forces and torques of body 2 are depicted in figure (B.3).

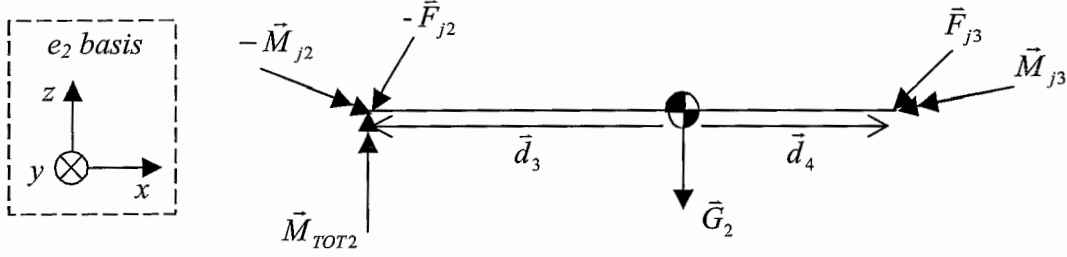


Figure B.3: Forces and moments on body 2

Newtons second law and the impulse moment equation:

$$\begin{aligned} -\vec{F}_{j2} + \vec{F}_{j3} - \vec{G}_2 &= m_2 \ddot{\vec{r}}_{M2} \\ -\vec{M}_{j2} + \vec{M}_{j3} + \vec{d}_3 \times \vec{F}_{j2} + \vec{d}_4 \times \vec{F}_{j3} + \vec{M}_{TOT2} &= \mathbf{J}_{M2} \cdot \dot{\vec{\omega}}_2 \end{aligned}$$

### B.3.4 Body 3

The forces and torques for body 3 are depicted in figure (B.4).

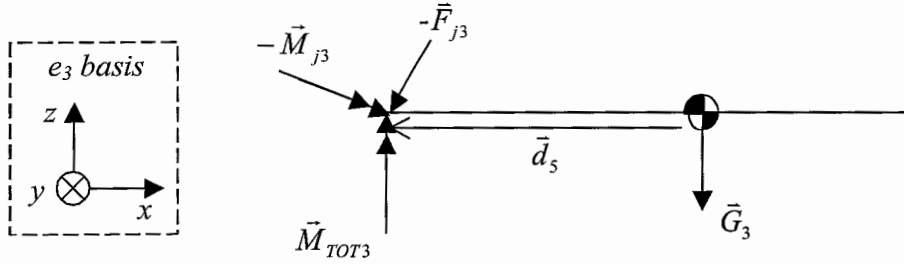


Figure B.4: Forces and moments on body3

Newtons second law and the impulse moment equation:

$$\begin{aligned} -\vec{F}_{j3} - \vec{G}_3 &= m_3 \ddot{\vec{r}}_{M3} \\ -\vec{M}_{j3} + \vec{d}_5 \times \vec{F}_{j3} + \vec{M}_{TOT3} &= \mathbf{J}_{M3} \cdot \dot{\vec{\omega}}_3 \end{aligned}$$

## B.4 Systems equations in relative coordinates

Combining the equations of the separate bodies leads to the following system in matrix notation (B.8).

$$\begin{bmatrix} -\vec{F}_{j1} + \vec{F}_{j2} + \vec{F}_{TOT1} \\ -\vec{F}_{j2} + \vec{F}_{j3} \\ -\vec{F}_{j3} \end{bmatrix} - \begin{bmatrix} \vec{G}_1 \\ \vec{G}_2 \\ \vec{G}_3 \end{bmatrix} = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \ddot{\vec{r}}_{M1} \\ \ddot{\vec{r}}_{M2} \\ \ddot{\vec{r}}_{M3} \end{bmatrix} \quad (\text{B.8})$$

$$\begin{bmatrix} \vec{M}_{TOT1} - \vec{M}_{TOT2} - \vec{M}_{TOT3} \\ \vec{M}_{TOT2} \\ \vec{M}_{TOT3} \end{bmatrix} + \begin{bmatrix} -\vec{M}_{j1} + \vec{M}_{j2} + \vec{d}_1 \times \vec{F}_{j1} + \vec{d}_2 \times \vec{F}_{j2} - \vec{d}_1 \times \vec{F}_{TOT1} \\ -\vec{M}_{j2} + \vec{M}_{j3} + \vec{d}_3 \times \vec{F}_{j2} + \vec{d}_4 \times \vec{F}_{j3} \\ -\vec{M}_{j3} + \vec{d}_5 \times \vec{F}_{j3} \end{bmatrix} = \begin{bmatrix} J_{M1} & 0 & 0 \\ 0 & J_{M2} & 0 \\ 0 & 0 & J_{M3} \end{bmatrix} \begin{bmatrix} \dot{\vec{\omega}}^1 \\ \dot{\vec{\omega}}^2 \\ \dot{\vec{\omega}}^3 \end{bmatrix} \quad (\text{B.9})$$

All these equations are expressed relative to the global coordinate system. The position vectors and body vectors therefore need also be expressed in the global coordinate system.

### B.4.1 Elimination of joint forces

We are interested in the motion of the system under the influence of the applied forces and torques. Therefore the reaction forces are eliminated out of equations (B.8):

$$\begin{bmatrix} \vec{F}_{j1} \\ \vec{F}_{j2} \\ \vec{F}_{j3} \end{bmatrix} = \begin{bmatrix} -\vec{G}_1 - \vec{G}_2 - \vec{G}_3 - m_1 \ddot{\vec{r}}_{M1} - m_2 \ddot{\vec{r}}_{M2} - m_3 \ddot{\vec{r}}_{M3} + \vec{F}_{TOT1} \\ -\vec{G}_2 - \vec{G}_3 - m_2 \ddot{\vec{r}}_{M2} - m_3 \ddot{\vec{r}}_{M3} \\ -\vec{G}_3 - m_3 \ddot{\vec{r}}_{M3} \end{bmatrix} \quad (\text{B.10})$$

where  $\ddot{\vec{r}}_M = \ddot{\vec{r}}_M(\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3)$ . The equations for  $\vec{r}_M$  and their derivatives are given in Appendix A.

### B.4.2 Equations for the motor torques

The joint forces (B.10) can be substituted in the momentum equations (B.9). Joint torques about the z-axis are zero because a radial bearing is not capable of taking a torque in z-direction. Furthermore we are only interested in torques about the z-direction because the joint torques and the acceleration torques around the x and y direction are equal to each other (because the radial acceleration in x and y direction is zero).

## B.5 Motion in a horizontal plane

Now, we will restrict our attention to a motion in the horizontal plane. Forces in z-direction and torques about the x- and y-axis are not considered anymore (so all joint torques). The resulting equations for the motor torques are:

$$\begin{aligned}\vec{M}_{TOT1} &= J_{M1}(\theta_2, \theta_3)\dot{\vec{\omega}}_1 - \left( \vec{d}_1 \times \vec{F}_{j1} + \vec{d}_2 \times \vec{F}_{j2} \right)_z + \\ &\quad \vec{M}_{TOT2} + \vec{M}_{TOT3} \\ \vec{M}_{TOT2} &= J_{M2}(\theta_3)\dot{\vec{\omega}}_2 - \left( \vec{d}_3 \times \vec{F}_{j2} + \vec{d}_4 \times \vec{F}_{j3} \right)_z \\ \vec{M}_{MOT3} &= J_{M3}\vec{\omega}_3 - \left( \vec{d}_5 \times \vec{F}_{j3} \right)_z\end{aligned}$$

All equations are now expressed in the global basis. The equations are 3 coupled, non-linear  $2^{nd}$  order differential equations in the independent coordinates  $\theta_1, \theta_2, \theta_3$ . By given motor torques and given initial conditions  $\theta_1(t=0), \theta_2(t=0), \theta_3(t=0), \dot{\theta}_1(t=0), \dot{\theta}_2(t=0), \dot{\theta}_3(t=0)$ , the independent coordinates can be solved by means of (numerical) integration.

### B.5.1 Interpretation of the dynamic equations

One can identify two parts in the equations for the motor torques: a linear part ( $J_M\dot{\vec{\omega}}$ ) and a part containing the acceleration profiles of the centre of mass of the bodies (this is included in the joint forces). This is a nonlinear part because the equations for the accelerations are nonlinear. When rotational velocities are low then the second part can also be approximated as a linear part

The torque the motor for the second axis has to deliver is visualized in figure (B.5). Here all components are expressed in the local basis for better understanding.

Note that only the joint forces in the local y-direction give a torque about the z-axis of the body.

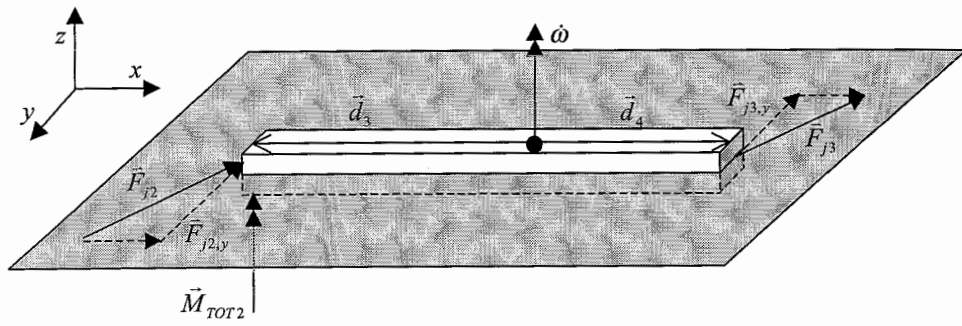


Figure B.5: Forces and torques acting on body 2 for a motion in the horizontal plane



## Appendix C

### ILC to axes 2 and 3

In this appendix the ILC scheme for the second and third axes and results are outlined. The approach as mentioned in chapter 6 is followed. From figure (5.4) it is seen that the non-reproducible part of the error signals for the second and third axes are very small. First, the frequency content of the reproducible part and non-reproducible part for both axes is given in figure (C.1).

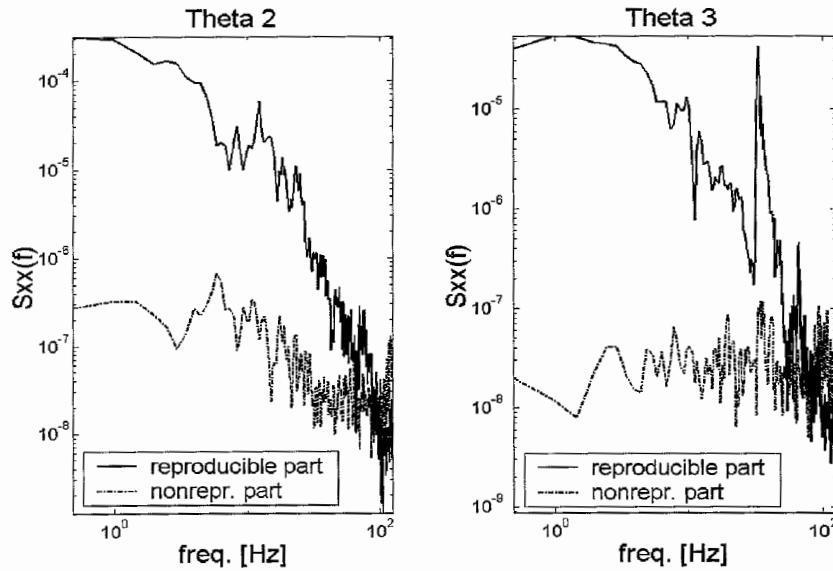


Figure C.1: PSD of reproducible and non-reproducible errors for axes 2 and 3

In the power spectral density plot of the reproducible error signal of the third axis an increase is seen around 30 Hz. The origin of this increase is unknown. Then, the convergence criterium for the learning filters is analyzed. Again the learning filters are



designed as the inverse of the process sensitivity of the linear parts of the models. The second and third axes suffer hardly from varying fourth order dynamics and therefore the learning filters are valid for all arm configurations. In figure (C.2) the convergence plots are given.

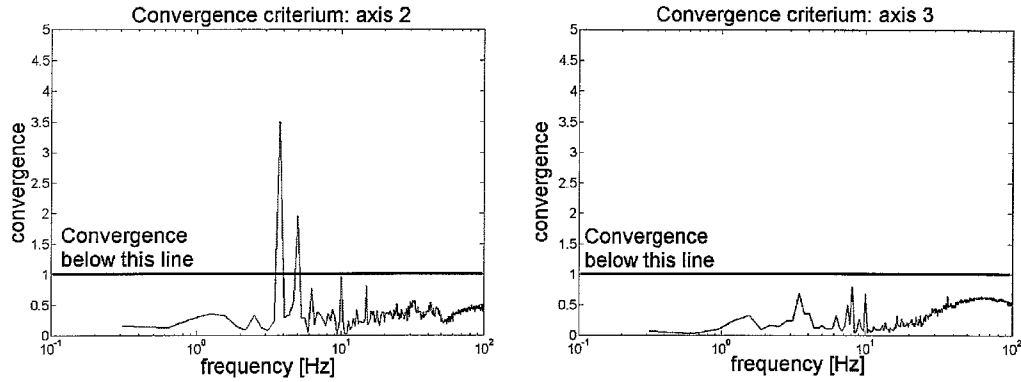


Figure C.2: Convergence criteria for axes 2 and 3

For the third axis convergence is assured over the whole frequency range of interest (up to 100 Hz). In the convergence criterium for the second axis some peaks are present with an amplitude larger than one. These peaks occur at low frequencies and the peaks are probably caused by friction in the measurement of the FRF of the second axis. The robustness filters  $Q_{\text{axis } 2}$  and  $Q_{\text{axis } 3}$  are chosen as third order low-pass filters with a cut-off frequency of 80 Hz. During implementation no problems were encountered and in the figures (C.3) and (C.4) a learning session is seen for both axes.

The maximum values of the error signals after learning are 2.5 mrad for axis 2 and 2.0 mrad for the third axis. The errors thus almost converged to the non-reproducible part (see figure (5.4)). The 2-norm and  $\infty$ -norm for the learning session are given in figure (C.5).

The error norms decrease almost monotonically and after 5 trials they stabilize around their final values. The power spectral densities of the error signals before and after learning together with the power spectral density of the generated feedforward signals are displayed in figure (C.6).

From the figures (C.5) and (C.6) the real strength of the ILC scheme is clear. For a plant which can be accurately described by a linear model, the reproducible errors can be 'learned' away. An error reduction over a wide frequency range is possible. In this case the cut-off frequency of the robustness filter is not limited by the varying fourth order dynamics.

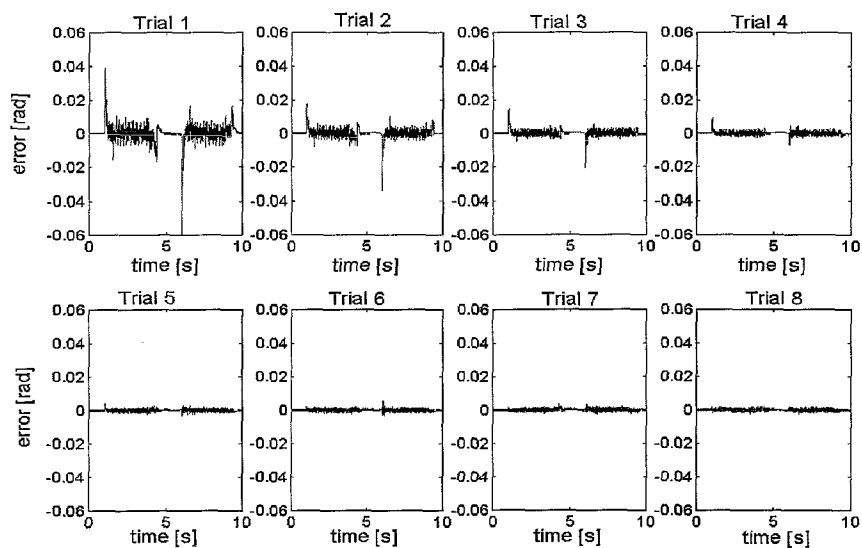


Figure C.3: Error signals of axis 2 for a learning session

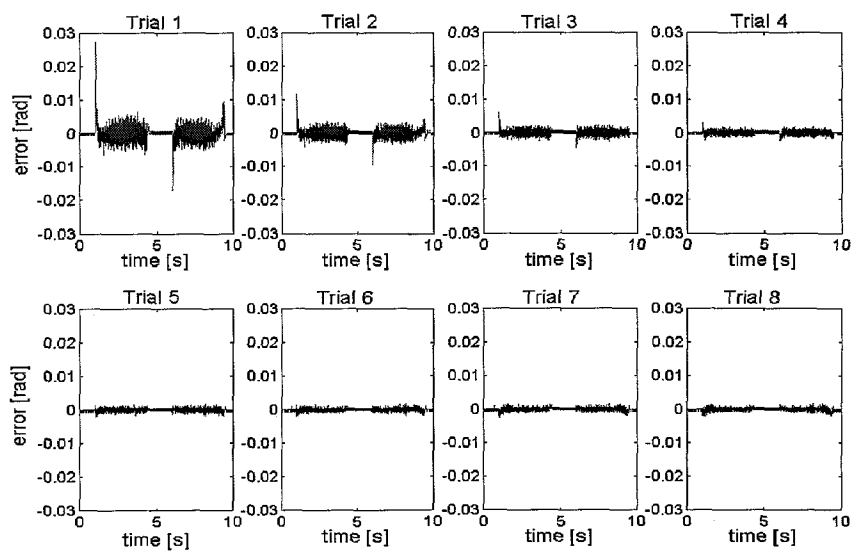


Figure C.4: Error signals for axis 3 for a learning session

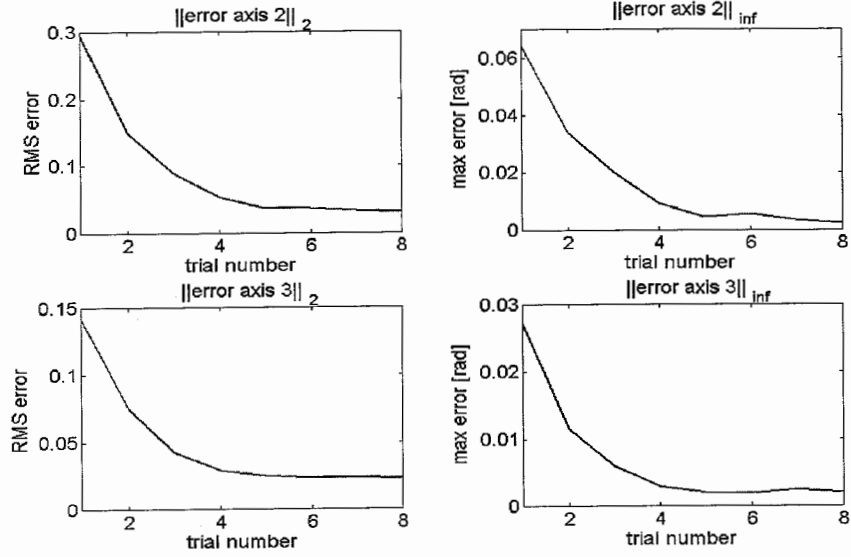
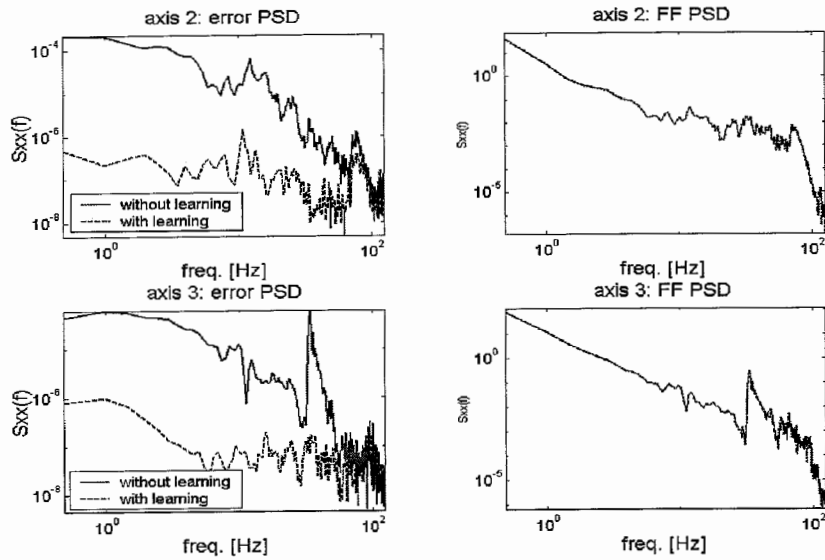
Figure C.5: 2-norm and  $\infty$ -norm for a learning session of axes 2 and 3

Figure C.6: PSD of error signals before and after learning and PSD of generated feed-forward signals

## Appendix D

### Verification on another robot

Besides the test robot also some experiments were done on another robot which is used in a reticle handling module. Some results are displayed in this appendix. The FRF's of the robot links are assumed to be equal to the test robot. So the same learning filters are used as for the test robot. Some minor differences may be present which may lead to slightly different error signals.

The robot is completely integrated in the reticle handling module. However, there was still enough space to place the steel beam for the end-effector measurement (see figure (D.1)).

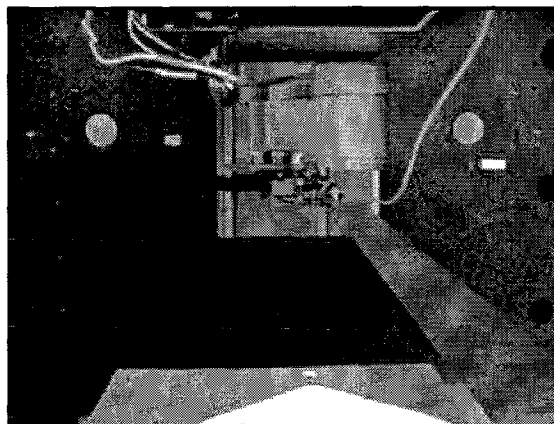


Figure D.1: Experimental setup for measurement on the robot in the reticle handling module

For each robot the manufacturer gives an offset for the second axis. This offset can be seen as a calibration value for the home position of the robot. For this robot the original offset for the second axis was changed to obtain better results for the inductive

measurement. The original offset caused a drift from the zero error line during forward and backward motions. The offset was approximately 10% in error.

## D.1 Encoder signals

The reproducibility of the encoder signals is investigated for the three axes. The results for the second and third axes are comparable to the results of the test robot. The reproducibility of the first axis is better than the test robot. For the test robot some peaks occurred with a maximum value of approximately 20 mrad. For this robot the non-reproducible part is a considerably lower than the test robot: approximately 8 mrad.

A learning session is applied to the robot. The folded configuration is used for the learning filter and a low-pass robustness filter is used with a cut-off frequency of 15 Hz. For this robot, the cut-off frequency of the robustness filter could be pushed to a higher frequency than for the test robot without losing convergence. This could be due to the fixed z-column on the top side but an FRF measurement should be performed to verify this. In figure (D.2) the results of the learning session for the first axis are shown.

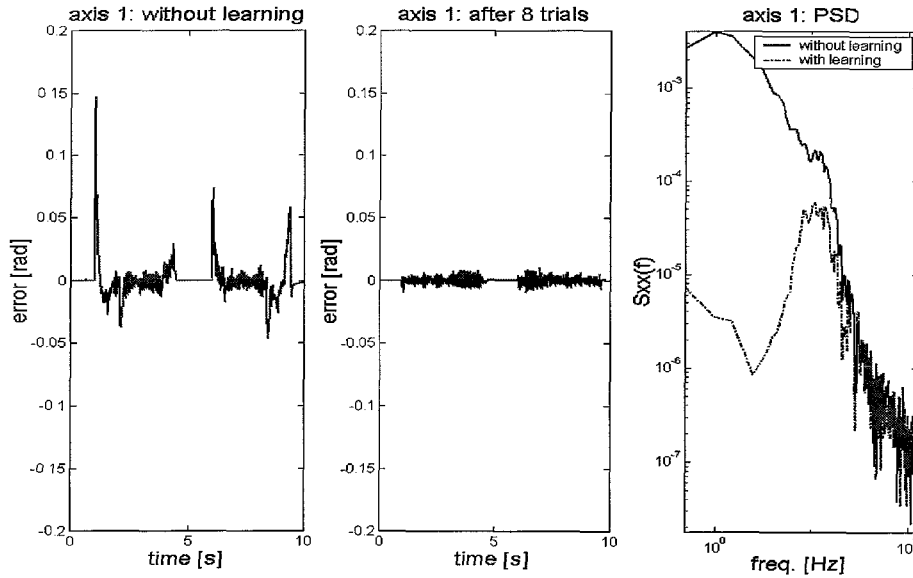


Figure D.2: Error signals of axis 1 before and after learning and a PSD of the error signals

The error monotonically decreased to a maximum value of approximately 10 mrad at the motor. The results of the learning session for the second and third axes are given

in figures (D.3) and (D.4). The cut-off frequencies of the low-pass robustness filters for the axes were respectively 30 and 50 Hz.

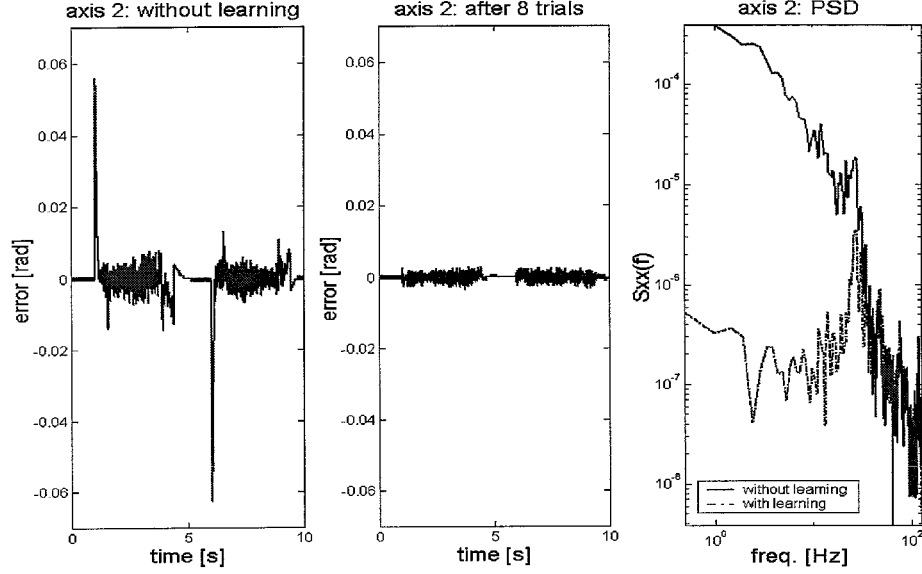


Figure D.3: Error signals of axis 2 before and after learning together with a PSD of the error signals

The maximum errors for the second and third axes after learning are both 3.0 mrad at the motor side. These maximum errors are close to the maximum non-reproducible errors.

## D.2 Inductive signals

The error signals at the encoders show a strong reduction. Now, the effect at the end-effector is evaluated. In figure (D.5) the error signal at the end-effector is given for the learning session.

In figure (D.5) it can be seen that the error reduces but after approximately 4 trials the error does not reduce any further. The end-effector is indeed controlled within the specified error band of 0.2 mm but still a considerable error remains. A power spectral density plot illustrates this, see figure (D.6).

The reduction over the whole frequency range is only marginal. Even frequencies which should be reduced, up to 10 Hz, are not reduced very much. The robot is controlled

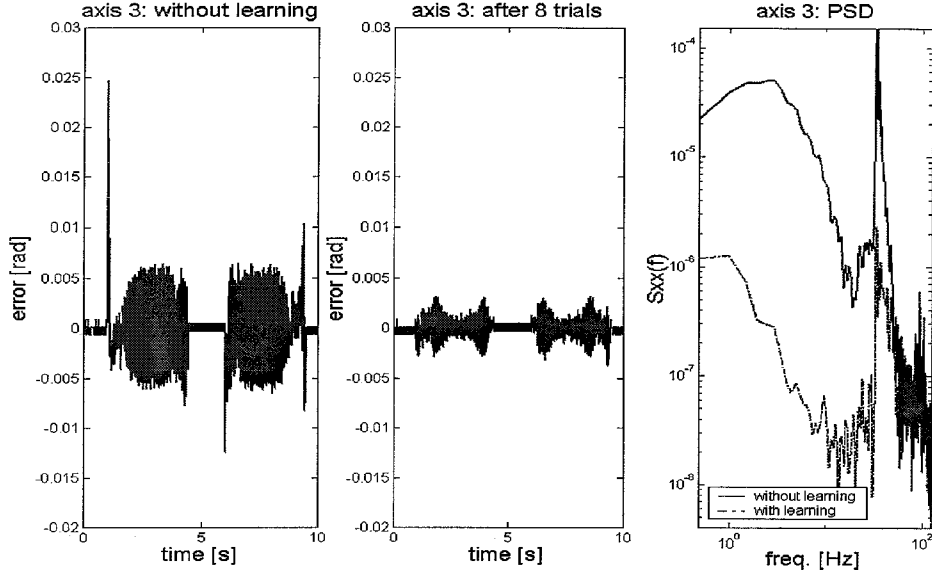


Figure D.4: Error signals of axis 2 before and after learning together with a PSD of the error signals

so accurately that other phenomena like gear inaccuracy, limit the further reduction of the error signals at the end-effector. This is further illustrated in figure (D.7).

In figure (D.7) the end-effector error is compared with the error resulting from the encoders. The encoder signals are transformed via the kinematics to give the error at the end-effector. The error from the inductive measurement shows hardly any correlation with the error signal from the encoders.

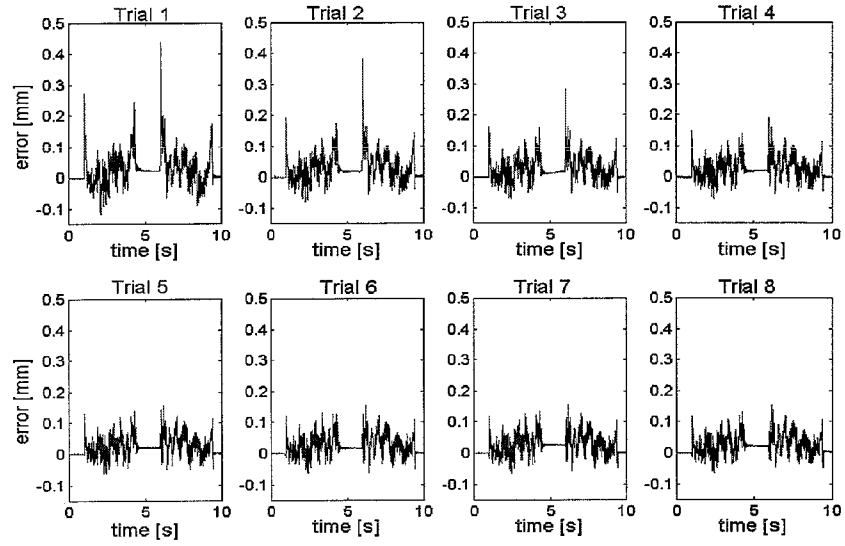


Figure D.5: Error signals at the end-effector for a learning session

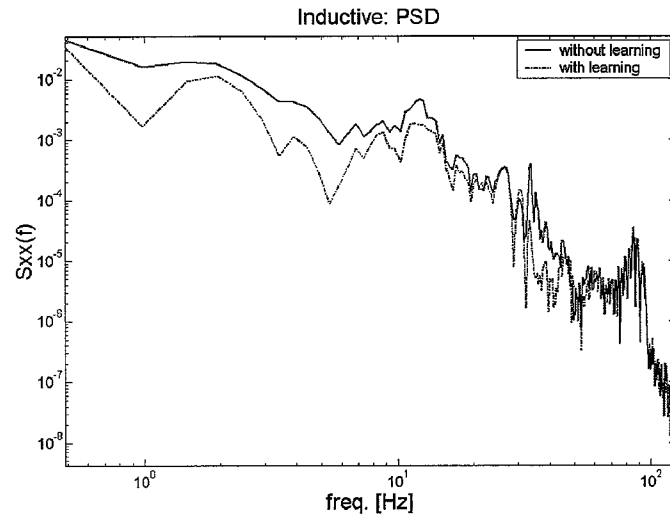


Figure D.6: PSD of error at end-effector



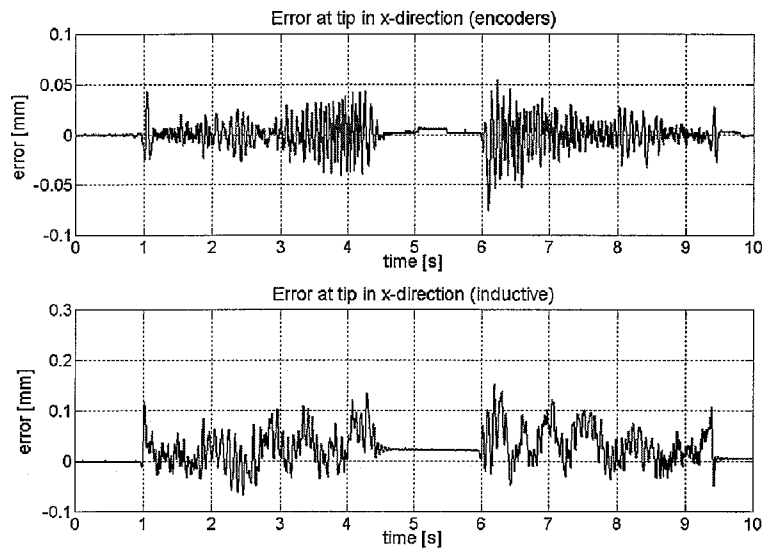


Figure D.7: Comparison between error signal from encoders and measured at the end-effector