

Dissertation

**State Space Adaptive Control
for a Rigid Rotor
Suspended in Active Magnetic Bearings**

ausgeführt zum Zwecke der Erlangung des akademischen Grades
eines Doktors der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Fakultät für Maschinenbau

von

Dipl.-Ing. Peter Wurmsdobler
Neubaugasse 59/2/21
A-1070 Wien

Matr. Nr.: 8626655

Geboren am 14.8.1968 in Schärding

Wien, im Januar 1997

Zusammenfassung

In dieser Dissertation wird ein adaptiver Zustandsregler basierend auf einer Modellidentifikation für einen magnetgelagerten starren Rotor entwickelt und mit Hilfe von Simulationen getestet.

Ausgangspunkt für die Modellierung ist ein Experimentieraufbau zur Erprobung von adaptiven Regelalgorithmen in Verbindung mit nicht-konservativen Kreuzkopplungseffekten. Davon wird ein umfangreiches nichtlineares Modell für Simulationszwecke und zur Ableitung eines Zustandsraummodells aufgestellt. Das nichtlineare Modell inkludiert einen starren Rotor, Positions- und Stromsensoren, Digital-Analog-Wandler, Schaltverstärker mit Pulsweitenmodulatoren, und ein nichtlineares Modell des magnetischen Aktuators. Um die Ordnung des Zustandsraummodells reduzieren zu können, wird eine Stromreglerschleife um einen Arbeitspunkt eingeführt. Mit dieser Vereinfachung wird ein zeitkontinuierliches Zustandsraummodell hergeleitet und in den diskreten Bereich transformiert.

Als Kernstück der Arbeit wird ein adaptiver Zustandsregler basierend auf Modellidentifikation für den on-line Betrieb vorgestellt. Dazu wird das zeitdiskrete Zustandsraummodell in Regelungsnormalform gebracht um einerseits die Anzahl der zu schätzenden Parameter zu reduzieren, und andererseits die automatisierte Reglerauslegung zu vereinfachen. Mit einer Verwendung dieses Modells als Prädiktor in einem Kalman-Filter werden unter Zuhilfenahme der Prädiktionsfehlermethode alle Zustände des Systems, sowie alle Parameter der Systemmatrizen und der Kalman Matrix geschätzt. Ferner kommt eine spezielle Implementierung des Verfahrens zum Einsatz, die numerische Stabilität garantiert. Durch einen zusätzlichen Algorithmus können Parameteränderungen im System erkannt werden, wodurch die Schätzung der Parameter ausgelöst wird. Mit diesen on-line geschätzten Modellparametern wird dann ein Zustandsregler, wahlweise mit oder ohne zusätzlicher Integralrückführung, berechnet.

Simulationsergebnisse zeigen, daß das geschlossene System samt nichtlinearem Modell und adaptiven Zustandsregler zufriedenstellend funktioniert. Die Stromreglerkaskade, die für das verwendete Zustandsraummodell der Positionsregelung Voraussetzung ist, stellt eine genügend große Bandbreite zur Verfügung. Das Funktionieren der letzteren Schleife wurde durch Sollsprünge und Störsprungantworten getestet. Weitere Simulationen zeigen, daß der adaptive Zustandsraumregler Parameteränderungen des Systems kompensiert. In dieser Anwendung wird das sprungförmige Auftreten von nicht-konservativen Kreuzkopplungskräften angenommen.

Abstract

In this dissertation a model identification adaptive state space controller for a rigid rotor suspended in active magnetic bearings is developed and tested by means of simulation.

An experimental setup used for testing of adaptive control algorithms in conjunction with non-conservative cross-coupling effects is the starting point for the modelling process. A comprehensive nonlinear model of the entire plant including a rigid rotor, position and current sensors, analogue to digital converters, digital signal processor, digital to analogue converters, switching power amplifiers with pulse width modulators and magnetic actuators is established for simulation purposes. An internal current control loop is designed for the point of operation in order to reduce the order of the state space model. Using this simplification a linear continuous time model is derived from the nonlinear model and transformed into the discrete time domain.

The core objective is the design of a model identification adaptive controller performing under on-line conditions. A discrete time state space innovations model in controller canonical form is used by an adaptation algorithm incorporating the recursive prediction error method to estimate all entries of the system matrices, the Kalman filter matrix and all states. To provide numerical stability, a special implementation of this algorithm is introduced. In addition, an effective algorithm is proposed to detect system parameter changes. Based upon the identified linear model a state controller with or without additional integrative feedback is calculated.

Simulation results of the closed loop system consisting of the nonlinear plant model and the proposed algorithm show the successful operation of the entire control concept. The current control loop provides the necessary bandwidth to cover the frequency band of the position control loop using the state space controller. The operation of the latter loop is investigated by both step responses and reactions to additional disturbances.

Simulation runs prove that the model identification adaptive controller can cope with changes in system parameters. In the present case the sudden appearance of non-conservative cross-coupling forces, as generated by seals, for example, is assumed to change the parameters of the system.

Acknowledgements

This work has been carried out during my employment as teaching and research assistant at the Institute for Machine and Process Automation in cooperation with the Institute for Machine Dynamics and Measurement, both at the University of Technology Vienna. This project has been supported by the Austrian Science Foundation (“Fonds zur Förderung der wissenschaftlichen Forschung”, FWF).

Herewith I want to express my honest thanks to Professor H.P. Jörgl, head of the Institute for Machine and Process Automation, for granting me the freedom to work in a special field of research on the one hand and giving me the scientific support on the other hand. Living in that prosperous atmosphere makes scientific work possible.

I am most grateful to Professor H. Springer, head of the Institute for Machine Dynamics and Measurement, for providing his invaluable guiding and encouragement during the entire project.

Many thanks go to all my colleagues at the Institute for Machine and Process Automation and the Institute for Machine Dynamics and Measurement, especially to Dipl.-Ing. Martin Kozek for his patience in listening to all my explanations of problems of any kind and for his helpful hints, to Dipl.-Ing. Ensio Hokka for his profound contributions on stochastic processes and last but not least to Ing. Gottfried Höld for providing the basic means for our research work by keeping the computers running.

Finally, thanks to my parents, relatives and friends for their support.

Nomenclature

General conventions

Variable	Font	Example
Scalar	italic letter	<i>a</i>
Continuous time vector	italic, bold face letter	<i>a</i>
Discrete time vector	bold face letter	a
Scalar	italic capital letter	<i>A</i>
Continuous time matrix	italic, bold face capital letter	<i>A</i>
Discrete time matrix	bold face capital letter	A
Identity matrix	bold face capital I	I
Zero matrix or vector	bold face capital 0	0
Matrix consisting of sub-matrices ij	bold face capital	{A^(ij)}

Common operators

Laplace operator $\mathcal{L}\{.\}$

Assuming a time variant function $x(t)$ defined for $t \geq 0$, $t \in \mathbb{R}$, integrable for $(0, \infty)$, and with the limited growth $|x(t)| \leq K e^{ct}$, the Laplace operator $\mathcal{L}\{.\}$ is defined by

$$X(s) = \mathcal{L}\{x(t)\} = \int_0^{\infty} e^{-st} x(t) dt$$

with $s = \delta + j\omega$ being a complex variable and $X(s)$ the Laplace transformed function, respectively. K and c are constants [Bronstein, 1989].

z-transformation operator $\mathcal{Z}\{.\}$

Given the discrete time sequence $x(k)$ for $k \geq 0$, $k \in \mathbb{N}$, the z-transformation operator $\mathcal{Z}\{.\}$ can be defined as

$$X(z) = \mathcal{Z}\{x(k)\} = \sum_{k=0}^{\infty} x(k) z^{-k}$$

with $z = \alpha + j\beta$ being a complex variable and $X(z)$ the z-transformed function, respectively. This sequence converges for all values of z with $|z| > R$ absolutely, if the in-equation $|x(k)| < K R^k$ is satisfied. K and R are constants [Jörgl, 1994].

Expectation operator $\mathcal{E} \{ . \}$

For a discrete time, ergodic, stochastic and scalar process $x(k)$ the expectation operator $\mathcal{E} \{ . \}$ is defined as [Schlitt, 1992]

$$\mathcal{E} \{ x(k) \} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=0}^{N-1} x(k).$$

Important symbols**Symbols of the continuous time, nonlinear model**

\mathbf{u}_d	digital control voltages
\mathbf{u}_a	analogue control voltages
\mathbf{u}_s	switched voltages for amplifier
\mathbf{i}	coil currents
\mathbf{i}_m	measured coil currents
\mathbf{i}_d	sampled coil currents
\mathbf{z}_b	rotor position in bearing coordinates
\mathbf{z}_s	rotor position in sensor coordinates
\mathbf{z}_m	measured rotor position in sensor coordinates
\mathbf{z}_d	sampled rotor position in sensor coordinates
\mathbf{w}	position set points
\mathbf{v}	measurement noise
\mathbf{F}_a	actuator forces

Symbols of the electrical signal path

Φ	magnetic flux
μ	magnetic permeability
A	cross section area of magnetic path
l	length of magnetic path
B	magnetic induction
H	magnetic field strength
\mathcal{R}	magnetic reluctance
N	number of coil windings
u	voltage
i	coil current
r	Ohmic resistance

Symbols for the linear active magnetic bearing (AMB)

K_s	position stiffness of an AMB
K_i	current gain of an AMB

Symbols of the continuous time linear state space model

n_s	number of system states
n_i	number of system inputs
n_o	number of system outputs
A	continuous time system matrix
B	continuous time control matrix
C	continuous time measurement matrix
x	continuous time system states
u	continuous time control variable
y	continuous time output variable
v	continuous time measurement noise
Q	covariance matrix for measurement noise

Symbols of the discrete time linear state space model

T_s	Sampling time
ν_i	structural indices
A	discrete time system matrix
B	discrete time control matrix
C	discrete time measurement matrix
K	Kalman filter matrix
x	discrete time system states
u	discrete time control variable
y	discrete time output variable
w	discrete time set point
η	system noise
ξ	measurement noise
R_i	covariance matrices

Symbols of the adaptation algorithm

n_p	number of parameters to be estimated
p	parameter vector
P	parameter covariance matrix
L	error adaptation matrix
W	partial derivative of the estimated states with respect to the parameter vector
V_k	partial derivative of the estimated system output with respect to the parameter vector
M_k	partial derivative of the predictor with respect to the parameter vector
Ψ	gradient matrix
ε	prediction error
δ	forgetting control parameter
ρ	forgetting factor
σ^2	prediction error variance

Symbols of the state space controller

\mathbf{K}_x	controller gain
\mathbf{K}_w	feed forward gain
\mathbf{K}_I	integrative feedback gain

Polynomials and polynomial matrices for the state space controller

$P^{ii}(z)$	characteristic polynomial of the closed loop system without integrative feedback
$P_I^{ii}(z)$	characteristic polynomial of the closed loop system with integrative feedback
$\mathbf{D}(z)$	polynomial matrix of the closed loop system without integrative feedback
$\mathbf{N}(z)$	polynomial matrix determining the zeros of the closed loop transfer function
$\mathbf{P}(z)$	desired polynomial matrix of the closed loop system without integrative feedback
$\mathbf{D}_I(z)$	polynomial matrix of the closed loop system with integrative feedback
$\mathbf{P}_I(z)$	desired polynomial matrix of the closed loop system with integrative feedback
$\mathbf{G}_{cl}(z)$	reference transfer function matrix
$\mathbf{G}_{cl_I}(z)$	reference transfer function matrix with integrative feedback

Abbreviations

ADC	Analogue to Digital Converter
AMB	Active Magnetic Bearing
AR	Auto Regressive
ARMAX	Auto Regressive with Moving Average and exogenous excitation
Cor-LS	Correlation Minimisation with Least Squares
DAC	Digital to Analogue Converter
DSP	Digital Signal Processor
ELS	Extended Least Squares
GPC	Generalised Predictive Control
GLS	Generalised Least Square
IV	Instrumental Variable method
LEM	Company name acronym for current sensing module
LQR	Linear Quadratic Regulator
LS	Least Squares
MIMO	Multiple Input Multiple Output
PWM	Pulse Width Modulator
RPEM	Recursive Prediction Error Method
SISO	Single Input Single Output

Contents

Abstract	i
Acknowledgements	ii
Nomenclature	iii
1 Introduction	1
1.1 Problem	1
1.2 Related work and different approaches	2
1.2.1 Identification of a transfer function model	2
1.2.2 Identification of a state space model	4
1.3 Chosen solution	4
2 System modelling	6
2.1 Experimental setup	6
2.2 Comprehensive nonlinear model	8
2.2.1 Digital controller	9
2.2.2 Digital to analogue converter for control voltage	10
2.2.3 Pulse width modulator (PWM) with switching amplifier and current sensor	10
2.2.4 Bearing actuator model	11
2.2.5 Rotor model	15
2.2.6 Position sensors	17
2.2.7 ADC for measured rotor positions	17
2.2.8 ADCs for measured currents	18
2.3 Model linearisation	18
2.3.1 Point of operation	19

2.3.2	Current control loop	19
2.3.3	Linearised model of an electromagnet with underlying current control loop	23
2.3.4	Linearised actuator model	25
2.3.5	Linear rotor bearing model	26
3	State space adaptive control	30
3.1	Parameter and state estimation	30
3.1.1	Innovations model	31
3.1.2	The recursive prediction error method	32
3.1.3	Derivation of the recursive algorithm	32
3.1.4	Summary of the recursive algorithm	38
3.1.5	Forgetting factor	39
3.1.6	Parameter covariance	40
3.1.7	Analysis of the recursive algorithm	41
3.2	Controller design	42
3.2.1	State space controller	42
3.2.2	State space controller with integrative feedback	44
3.3	Closed loop eigenvalues	46
3.3.1	P-structure	46
3.3.2	PI-structure	47
4	Numerical results	50
4.1	Current control loop	50
4.1.1	Current controller design	51
4.1.2	Current controller implementation	53
4.1.3	Initial values for simulation	53
4.1.4	Simulation results	53
4.2	The linear actuator model	57
4.2.1	Current gain of the magnetic actuator forces	57
4.2.2	Position Stiffness of the magnetic actuator forces	59
4.3	The linear state space model	62
4.3.1	System matrices	62
4.3.2	State space controller	64

4.3.3	Predictor	66
4.4	Closed loop dynamics	68
4.4.1	Initial values for simulation	68
4.4.2	System response to set point change	68
4.4.3	System response to additional load	69
4.5	State space adaptive control	74
4.5.1	Initial values for simulation	74
4.5.2	Implementation of the state space adaptive controller	75
4.5.3	System response due to parameter change	75
4.5.4	Discussion of the numerical results	83
5	Conclusion	85
5.1	Summary	85
5.2	Outlook	86
A	Controller canonical form	94
B	Derivation of matrices \mathbf{M}_k and \mathbf{V}_k	96
B.1	Derivation of the matrix \mathbf{M}_k	97
B.2	Derivation of the matrix \mathbf{V}_k	98
C	Derivation of PI-controller	99
C.1	Closed loop transfer function	99
C.2	Simplification of closed loop transfer function	100
C.3	Controller computation for PI-structure	101
D	Simulation tools	103
D.1	Simulink block diagrams	103
D.2	MATLAB code	108
D.2.1	Data file	108
D.2.2	Estimation algorithm	113
D.2.3	Controller calculation	121
D.2.4	Controller calculation for P-structure	121
D.2.5	Controller calculation for PI-structure	122
D.2.6	Helper applications	123

Chapter 1

Introduction

The presented work is concerned with state space adaptive control of a rigid rotor suspended in active magnetic bearings. In this chapter a short presentation of the objectives, related works and the chosen way to a solution is presented.

1.1 Problem

In many active magnetic bearing applications the design of the entire control system is performed under the assumption that all system parameters are exactly known in advance and do not change during operation. For these type of plants adaptive control is not necessary. It is sufficient to design a robust controller for a limited range of operation.

In high performance turbomachinery, however, a change in system parameters, such as a sudden change of pressure in a turbo pump may easily occur. This event may lead to a change in stiffness parameters of a sealing and finally to destabilising non-conservative cross-coupling forces, for example. In this case an adaptive controller is absolutely necessary.

Another possible application for adaptive control is a milling spindle suspended in active magnetic bearings. The exchange of the milling tool results in the change of system parameters like the rotor mass or the moments of inertia. In those cases the parameter change does not destabilise the entire system, but may deteriorate its performance. Adaptive control, however, can guarantee a desired performance for a wide range of operation.

Therefore, the objectives for this work are the modelling of a rotor bearing system, the derivation of a linear discrete time model for a point of operation, the search for a proper algorithm which can cope with parameter changes of the controlled system, and the proof of the performance by means of simulation.

1.2 Related work and different approaches

Adaptive control in the discrete time domain was up to now no topic in the field of advanced control for active magnetic bearing systems. Research work primarily focuses on robust control (μ -Synthesis [Stephens and Knospe, 1996, Nonami and Ito, 1994] and H_∞ -control [Sivrioglu et al., 1996, Nonami et al., 1994]), i.e. controller design in the frequency domain. On the other hand, identification is mostly performed off-line and mainly in order to identify physical parameters of the electro-mechanic system [Herzog and Gähler, 1994, Gähler and Herzog, 1994, Gähler et al., 1996] and [Lottin and Saïdi, 1994].

In many applications, controller design is carried out in the continuous time domain to keep physical parameters transparent. Digital control is implemented in so far, as all designed controllers are transformed into the discrete time domain, because they are commonly implemented in a digital signal processor in the end. Summarising, this means that there is a need in adaptive digital control for active magnetic bearing systems.

In terms of parameter estimation several problems are involved with adaptive control for an active magnetic bearing system, which is an open loop unstable multiple input multiple output system (MIMO-system). Therefore, identification has to be performed in closed loop operation. This results in convergence problems of all estimated parameters, since the model input is correlated with the output, i.e. statistically not independent. Under certain conditions this obstacle can be circumvented [Aling, 1990, Jiang and Doraiswami, 1987, Schumann, 1982, Ng et al., 1977].

More problems result from the choice of the estimation algorithm for an open loop unstable MIMO-system. One has to consider certain constraints for an adaptive control algorithm being used for a real time implementation within a digital signal processor. No iterations should be included within one sample period, because the calculations could run into a timeout. Additionally, the algorithm should be numerically stable. This means that matrix inversions and all complex algebra should be avoided. The most important constraint is, however, that not every algorithm can be applied to an open loop unstable MIMO-system.

Nevertheless, a great variety of algorithms remain. Generally speaking, adaptive control can be based on a state space representation of the system or on a transfer function matrix model.

1.2.1 Identification of a transfer function model

The first approach in system identification is the estimation of a transfer function matrix with a simple least squares algorithm (LS-algorithm, [Isermann, 1992a]). Unfortunately, this model depends on an auto regressive (AR) model of the noise filter including all system poles. Since a magnetic bearing system is open loop unstable, such a noise model cannot be used. What remains, is to switch to a more sophisticated noise model with more degrees of freedom. An extended least squares algorithm (ELS-

algorithm, [Isermann, 1992a]) could cope with the problem, because unstable poles can be cancelled by zeros of the noise filter, if the order of the numerator polynomial is chosen high enough. Then, however, the convergence of the entire algorithm suffers from the slow convergence of the noise filter parameters and the algorithm could never follow sudden parameter changes.

More complex identification algorithms like generalised least squares (GLS) including iterations, Cor-LS or the instrumental variable method (IV, [Isermann, 1992b]) could solve the identification problem. For these algorithms there are no restrictions to the noise model or to the properties of the noise (Cor-LS), which needs not necessarily to be white. The complexity of these algorithms, however, causes problems in a real time application regarding the guarantee of a constant sampling time or numerical stability.

Another possibility is the estimation of the weighting sequence and the application of generalised predictive control (GPC) or the derivation of a transfer function in a second step. Because of the open loop instability bicausal weighting functions have to be used [Kouvraritakis and Rossiter, 1992], which makes the use of GPC difficult. Again it seems not to be reasonable to use that algorithm for two reasons. Firstly, the estimation of bicausal weighting sequences for a MIMO-system results in a very complicated algorithm which is not applicable in real time. Secondly, long weighting sequences are necessary to render the system behaviour properly which results in a huge number of parameters to be estimated.

To identify the transfer function matrix is one aspect in adaptive control. The crucial point for the identification algorithm is the subsequent controller design. Basically, there are two possibilities.

Matrix polynomial controller

For a simple single input single output system (SISO-system), the controller design can be done in a straight forward manner [Isermann, 1987a, Isermann, 1987b], if some constraints are considered like the fact that cancellation of an unstable pole must not occur. If a MIMO-system is target for a controller design, things become a little bit more complicated [Schumann, 1982]. The inversion of the transfer function matrix generates new parasitic poles resulting from the open loop zeros in form of convolutions and sums between all numerators, and additional zeros outside the unit circle. An adaptive control algorithm using the inverse model of the plant must calculate all new poles outside the unit circle to avoid pole zero cancellation. Therefore, an algorithm must take into account these effects. In principle, the problem of an automated controller design can be solved, but the implementation fails for numerical reasons.

State space controller

Once a state space representation of the identified system is found, it is easy to derive a controller, e.g. an LQR-controller. The difficulties arise out of the transformation from a transfer function matrix to state space representation and from the state estimation [Hensel, 1990]. Essential work in this field was presented in

[Nour Eldin and Heisters, 1980] and its second part in [Nour Eldin and Heisters, 1981]. A state space representation can also be retrieved from the estimated transfer function matrix or even from the estimated weighting sequence using the Hankel-matrix [Tsui and Chen, 1983].

All proposed algorithms cause an enormous amount of calculation effort or are not applicable to an open loop unstable MIMO-system and are ruled out for those reasons.

1.2.2 Identification of a state space model

Given the preceding restrictions, the estimation of a state space model becomes reasonable. A state space representation of an open loop unstable system makes controller design easier. The identification algorithm, however, must estimate both the system states and all system parameters at the same time. The system states are necessary for the state space controller. The most popular approach is the extended Kalman filter. This algorithm uses an extended state vector including both the system states and system parameters. The result is a nonlinear filter which tends to converge very slowly. Since an active magnetic bearing is a very fast system, an alternative algorithm has to be found.

1.3 Chosen solution

The *recursive prediction error method* (RPEM) presented in [Goodwin and Sin, 1984, Ljung and Söderström, 1983] incorporates both state and system parameter estimation. In [Nazaruddin, 1994, Unbehauen and Nazaruddin, 1995] an implementation of that algorithm is presented in conjunction with pole placement control. These works provide the basis material used in this thesis. Its detailed structure is explained in the following.

In Section 2.1 of Chapter 2 the experimental setup and the rotor model are presented. The rigid rotor is suspended by two active magnetic bearings with four degrees of freedom. Non-conservative cross-coupling forces, simulating a system change, are applied to the rotor in a given plane along the rotor axis by an additional magnetic actuator. For simulation purposes a comprehensive nonlinear model including a rigid rotor, position and current sensors, analogue to digital converters, digital signal processor, digital to analogue converters, switching power amplifiers with pulse width modulators and magnetic actuators is established in Section 2.2.

In Section 2.3 a continuous time linear model is derived from the nonlinear model for a point of operation. Therefore, an internal current control loop is designed for the point of operation in order to reduce the order of the resulting linear system. Under these assumptions the magnetic actuator can be treated as a negative spring regarding the rotor position, and as a gain regarding the control current. From there, a continuous time state space model for the rotor bearing system is derived in Subsection 2.3.5 and transformed into a discrete time innovations model with its system matrices in

canonical form [Schumann, 1982, Tolle, 1985]. This model is the basis for adaptive control performed on the entire system.

The third chapter, Chapter 3, is dedicated to the estimation algorithm and the controller design. The *recursive prediction error method* which can be used to identify the state space model under on-line conditions is introduced in Section 3.1. The recursive algorithm, consisting of several matrix operations, can be performed very fast, which is important for a real-time implementation. Within this algorithm a state space model and all states are calculated after each sampling time interval. To provide numerical stability, a special implementation of this algorithm is used [Bierman, 1988]. Additionally, an effective algorithm is proposed to trigger system parameter changes in order to initiate the adaptation algorithm. The last subsection, Subsection 3.1.7, deals with stability and convergence analysis.

In the second part of Chapter 3, Section 3.2, the design of a pole placement controller with and without integrative feedback is described. Since an active magnetic bearing system is open loop unstable, a controller and an observer have to be designed in advance. A deterministic approach is used for the design, because the statistical solutions like a Kalman filter depend on the knowledge of the covariance matrices of system and measurement noise, which are not known a-priori. The third part of Chapter 3, Section 3.3, deals with the closed loop analysis for the rotor bearing system under state space control in conjunction with the innovations model as state estimator.

Simulations are used to demonstrate the successful operation of the proposed algorithms. The results are presented in Chapter 4. A change of a system parameter, namely the non-conservative cross-coupling stiffness is simulated as well as a sudden appearance of an additional load. The simulation results show that the proposed identification algorithm can cope with parameter changes and the entire system can be stabilised even for high values of the non-conservative stiffness coefficients. Furthermore, changes in distortion like additional loads can be compensated by the integrative feedback loop.

The last chapter, Chapter 5, tries to draw final conclusions and gives an outlook on possible topics in the field of adaptive control for active magnetic bearing systems.

Chapter 2

System modelling

In order to apply state space adaptive control to a system, a comprehensive model has to be established for two reasons. Firstly, a precise model is needed for simulation purposes, and secondly a linear state space model can be derived from the previous one. The process of system modelling is topic of this chapter.

“A system is a potential source of data”, [Cellier, 1991].

A system is characterised by its bounds which define what belongs to it and what does not. Additionally, a system interacts with its environment through inputs which can influence the system behaviour, and its outputs which are determined by the system and can influence the environment. A system is the basic objective of scientific work. In the present case, the system under investigation is a rotor suspended in active magnetic bearings, and excited by non-conservative cross-coupling forces. This system does not exist *per se* but in specific implementations.

2.1 Experimental setup

Since no real application is accessible for testing new algorithms, an experimental setup is needed to make scientific investigations possible.

“An experiment is the process of extracting data from a system by exerting it through its inputs”, [Cellier, 1991].

To perform an experiment, it is necessary to set up a test rig. At the Institute for Machine Dynamics and Measurement, University of Technology Vienna, a test rig was built to test different algorithms for adaptive control in conjunction with non-conservative cross-coupling effects (see Fig. 2.1 and for a detailed description see [Lang, 1997]). Three actuators with backup bearings are mounted on a foundation. The outer actuators are used as active magnetic bearing, the central one as a magnetic exciter for the application of non-conservative cross-coupling forces to the rotor. In each actuator housing two proximity sensors are mounted operating as eddy current sensors. An

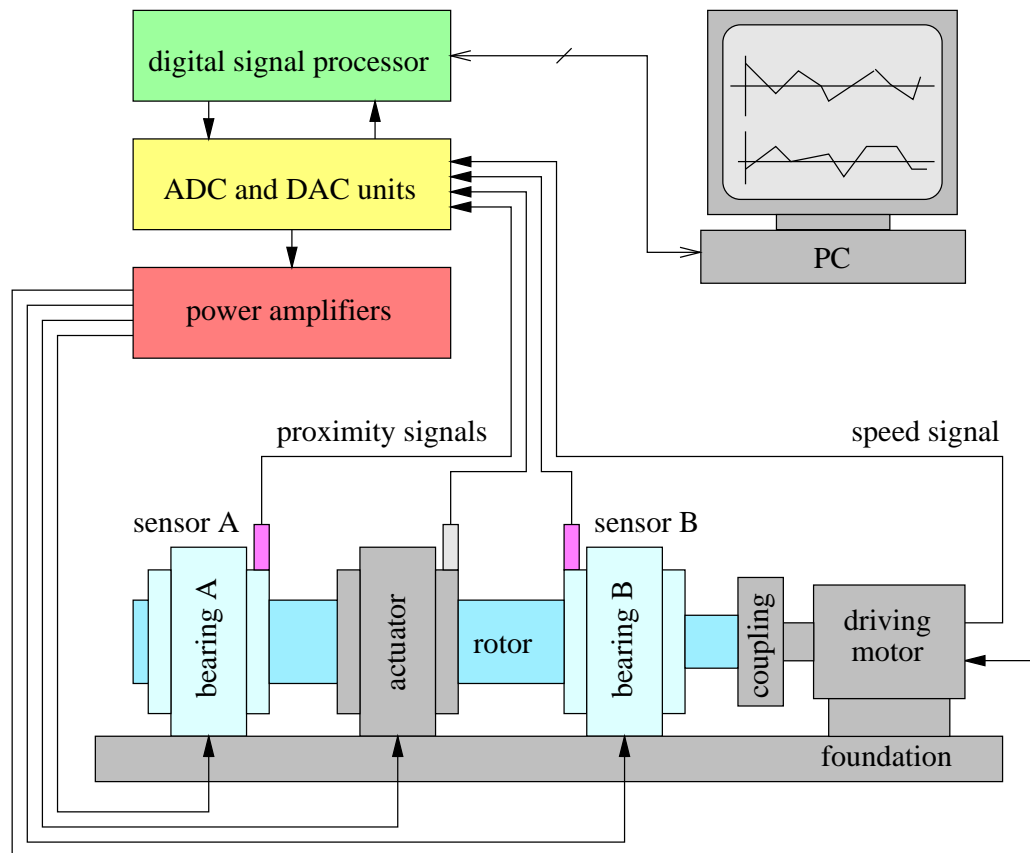


Figure 2.1: Experimental setup used for the real time implementation of adaptive control algorithms.

electric motor with a speed sensor is mounted on the foundation and drives the rigid rotor through an elastic coupling. The sensor signals are sampled by ADCs and fed into the digital signal processor (DSP) which performs the entire control task. The control variables are passed to the switching amplifiers via DACs. All actuators and the motor are driven by power amplifiers located in an electric console. The DSP-board is connected to a host computer for the sake of data acquisition and compilation of the digital signal processing code.

The presented test rig is to realise tests for adaptive control algorithms by means of the third actuator which can simulate any bearing behaviour. Before an algorithm is tested on that experimental setup, its performance must be tested and well proved by means of numerical simulation.

“Simulation is an experiment performed on a model”, [Cellier, 1991].

The major motivation for simulation lies in the fact that more situations can be tested than in a real experimental setup without destroying anything. The facilities and the quality of the simulation runs depend on the model it is based upon. Therefore, a comprehensive mathematical model is required.

it is considered to be an independent control task and therefore, it is no objective for adaptive control. In a real experimental application the simulation of the cross-coupling forces has to be separated from the adaptive control law either by a separate hardware or logically, see [Lang, 1997].

The same is true for the speed control. The rotor is simply assumed to run at a certain speed. Additionally, no unbalance is introduced in the modelling process and simulated, because the resulting vibrations mask the adaptation effects. It would be difficult to distinguish between those vibrations and possible oscillations generated by the adaptive control algorithm. Of course, simulations can be carried out including unbalance, after the adaptation algorithm has been tested in order to gain a realistic simulation.

Once the bounds, the inputs and outputs of the system are defined the knowledge about the system has to be organised and mapped onto mathematical equations. Of course, this process has to be done hierarchically by defining the inputs and outputs of all subsystems (or modules) and their interactions. All subsystems are explained in detail within the following subsections. The corresponding MATLAB/SIMULINK block diagram models can be seen in Appendix D.1. The block diagram of the entire system is shown in Fig. D.1 and in detail in Fig. D.2.

2.2.1 Digital controller

The controller is implemented in a digital signal processor (Texas Instruments TMS320C40) and executes the control laws within the sampling period $T_s = 100 \mu s$. Inputs are the measured and sampled currents of all magnets of both bearing actuators \mathbf{i}_d and the measured and sampled rotor positions in the planes of the position sensors \mathbf{z}_d . Outputs are the digital control voltages for all magnets of both actuators \mathbf{u}_d . The corresponding block diagram can be seen in Fig. 2.3.

The operation of the DSP itself is not topic of this work. The algorithm for the computation of the control voltage \mathbf{u}_d is explained in detail in Subsection 2.3.2 and Subsection 4.1.2 as for the current controller design and its implementation, respectively. The computation of the control current \mathbf{i}_c is presented in conjunction with the state space adaptive controller in Chapter 3. Its implementation is considered in Section 4.5.2.

In the presented application both the current control loop and the position control loop operate at the same sampling frequency. If a large bandwidth is required for the position control loop, e.g. for a high speed rotor, it is advisable to implement the current control loop at a higher sampling frequency, or alternatively use an analogue circuit instead.

Although the controller uses floating point numbers, it has to convert all numbers to a fixed point representation for the analogue to digital conversion and vice versa. The quantisation, however, is symbolically ascribed to the ADCs and DACs.

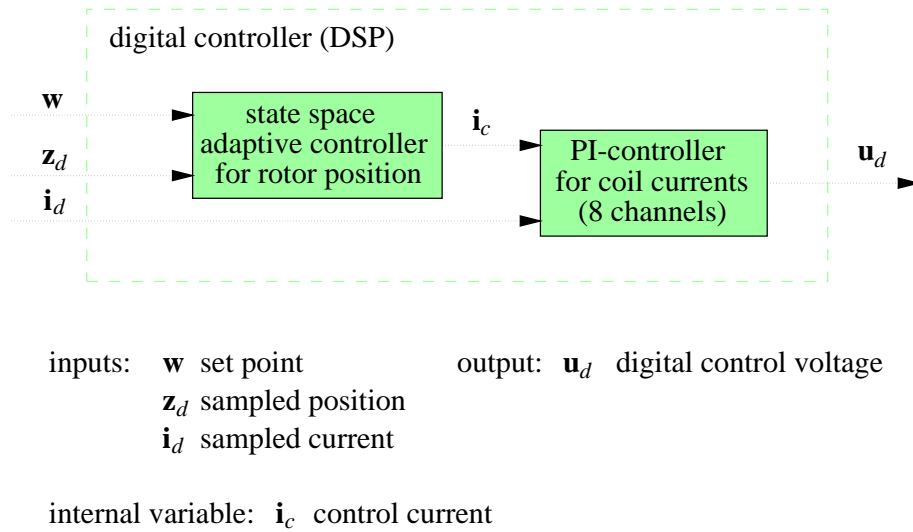


Figure 2.3: Block diagram of the digital controller with its cascaded position and current controller.

2.2.2 Digital to analogue converter for control voltage

The task of the DACs (8 channels) is to convert the fixed point numbers provided by the DSP into an analogue voltage signal. Input is the digital control voltage \mathbf{u}_d , output the analogue control voltage \mathbf{u}_a for all magnets of all bearing actuators. The DAC introduces quantisation noise and time delay which is modelled by a series of saturation, quantiser and time delay with the parameters given in Tab. 2.1. The corresponding SIMULINK block diagram is shown in Fig. D.3 in Appendix D.1.

Table 2.1: Parameters of DAC.

Bit resolution	12	-
Output voltage range	0-5	V
Quantisation	2.44	μm
Conversion delay	3	μs

2.2.3 Pulse width modulator (PWM) with switching amplifier and current sensor

Since the power losses of a linear amplifier would be too high, a switching amplifier is used in conjunction with a pulse width modulator (8 channels). Of course, this type of amplifier also introduces noise into the system at the harmonics of the switching frequency. Input to the PWM is the analogue control voltage \mathbf{u}_a for all magnets of

both bearing actuators, output is the switched voltage \mathbf{u}_s . The PWM in conjunction with the switching amplifier is modelled as a saw-tooth generator modulating the input, a comparator, and two voltage sources with the parameters given in Tab. 2.2. Of course, a current saturation has to be included as a concession to reality. The corresponding SIMULINK block diagram is shown in Fig. D.6 in Appendix D.1.

Table 2.2: Parameters of PWM and switching amplifier

Input voltage range	0-5	V
Switching frequency	60	kHz
Switching voltage U_s	75	V
Maximal current	8	A
Amplifier gain K_a	30	-

From the modelling point of view, the coil current is output of the coil when a certain voltage is applied to it assuming the voltage source to be a pure effort source (voltage is independent of the drained current). Therefore, the current output \mathbf{i} is logically assigned to the magnet model and is an input to the switching amplifier, because in a real application the current sensors are mounted on the switching amplifier boards. These sensors deliver the measured current signals \mathbf{i}_m .

2.2.4 Bearing actuator model

The rigid rotor is suspended in two active magnetic bearings. Inputs to these actuators are the switched voltages $\mathbf{u}_s = [\mathbf{u}_{sA}, \mathbf{u}_{sB}]^T$ and the rotor position in bearing coordinates $\mathbf{z}_b = [\mathbf{z}_{bA}, \mathbf{z}_{bB}]^T$. Outputs are the actuator forces $\mathbf{F}_a = [\mathbf{F}_{aA}, \mathbf{F}_{aB}]^T$ and the currents in each coil $\mathbf{i} = [\mathbf{i}_A, \mathbf{i}_B]^T$. The parameters of both actuators are identical and given in Tab. 2.3.

For the sake of simplicity, only one bearing actuator as shown in Fig. 2.4 is presented. Omitting the suffix A for bearing A and B for bearing B the inputs to one bearing are the switched voltages $\mathbf{u}_s = [u_1, u_2, u_3, u_4]$ and the rotor position in bearing coordinates $\mathbf{z}_b = [x, y]$, the outputs are the actuator forces $\mathbf{F}_a = [F_x, F_y]$ with

$$F_x = F_1 - F_3 \quad (2.1)$$

$$F_y = F_2 - F_4, \quad (2.2)$$

and four currents $\mathbf{i} = [i_1, i_2, i_3, i_4]$ of the coils. The air gap lengths between rotor and magnet poles are determined by

$$l_1 = l_0 - x, \quad (2.3) \quad l_3 = l_0 + x, \quad (2.5)$$

$$l_2 = l_0 - y, \quad (2.4) \quad l_4 = l_0 + y. \quad (2.6)$$

The corresponding SIMULINK block diagram is shown in Fig. D.4 in Appendix D.1.

Table 2.3: Parameters of an active magnetic bearing

μ_0	permeability in vacuum	$4 \pi \cdot 10^{-7}$	Vs/Am
μ_{r_s}	relative permeability of stator material	2000	-
μ_{r_r}	relative permeability of rotor material	1000	-
l_s	length of magnetic path in stator	$84 \cdot 10^{-3}$	m
l_r	length of magnetic path in rotor	$20 \cdot 10^{-3}$	m
A_s	cross section area of magnetic path in stator	$700 \cdot 10^{-6}$	m ²
A_r	cross section area of magnetic path in rotor	$420 \cdot 10^{-6}$	m ²
A_l	cross section area of magnetic path in air	A_s	m ²
B_{max_s}	maximum induction of stator material	1.3	T
B_{max_r}	maximum induction of rotor material	1.95	T
α	angle between pole shoes	$\pi/4$	rad
r	ohmic resistance in coil	0.8	Ω
N	number of coil windings = 2 pole shoes	130	-
l_0	nominal air gap	$0.5 \cdot 10^{-3}$	m
i_0	bias-current	4	A
L_0	nominal inductivity	0.01325	H
\mathcal{R}_0	nominal reluctance	$1.22 \cdot 10^6$	A/Vs

Nonlinear electromagnet model

Since all electromagnets of a bearing actuator are equal in design, only one is presented here without the suffices $j = 1, 2, 3, 4$ of their position. The input to a electromagnet is the voltage u and the air gap length l , output is the magnetic air-gap force F , which is always an attractive one. An additional output of each electromagnet is the coil current i . For one magnet the following equations for the voltage u , the magnetic flux Φ , the magnetic field strength H and the applied magneto-motive force $N i$ hold

$$N \frac{d\Phi}{dt} = u - r i, \quad (2.7)$$

$$N i = \oint H dl. \quad (2.8)$$

The magneto-motive force $N i$ can be rewritten in a discretised form

$$\oint H dl \approx \sum_i H_i l_i = H_r l_r + H_s l_s + 2 H_l l. \quad (2.9)$$

With the constitutive law for the induction $B = \mu H$ and with $\Phi = A B$ for all cross sections the magnetic circuit is described by

$$N i = \mathcal{R} \Phi, \quad (2.10)$$

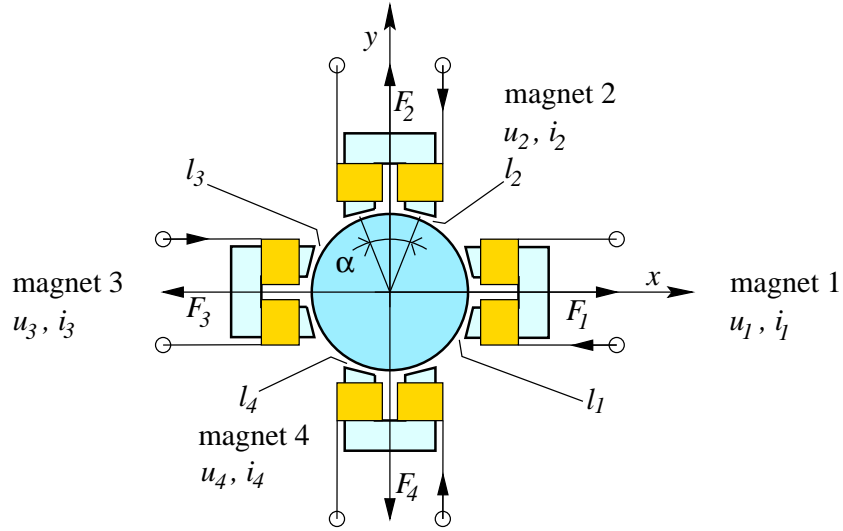


Figure 2.4: Sketch of an active magnetic bearing with four electro magnets, four voltage inputs u_j , four currents i_j and four ponderomotoric air-gap forces F_j , $j = 1, 2, 3, 4$.

where its magnetic reluctance \mathcal{R} depends on the air gap length l as

$$\mathcal{R}(l) = \frac{1}{\mu_0} \left[\frac{l_r}{A_r \mu_{r_r}} + \frac{l_s}{A_s \mu_{r_s}} + \frac{2l}{A_l} \right]. \quad (2.11)$$

Using the preceding equations, the state equations for a coil can be established with the input variable u , the modulating air gap length l , the state variable Φ , and the output variables F and i

$$\frac{d\Phi}{dt} = -\frac{r_0 \mathcal{R}(l)}{N^2} \Phi + \frac{1}{N} u, \quad (2.12)$$

$$F = \frac{\Phi^2}{\mu_0 A_l} \cos \frac{\alpha}{2}, \quad (2.13)$$

$$i = \frac{\mathcal{R}(l) \Phi}{N}. \quad (2.14)$$

For simulation purposes special attention has to be paid to the saturation of the magnetic material. Assuming the relation $B = \mu(B) H$, the permeability of the magnetic material is defined by the anhysteretic curve [Haferl, 1991, Wurmsdobler, 1992] with

$$\mu(B) = \mu_0 \left[\frac{1 - \mu_r}{\pi} \arctan K \frac{(|B| - B_{max})}{B_{max}} + \frac{1 + \mu_r}{2} \right], \quad (2.15)$$

with K being a constant determining the slope at the maximal induction B_{max} . In the presented work this constant has been chosen to be $K = 12.5$ in order to suit to a measured anhysteretic curve.

This approach guarantees $\mu \approx \mu_0 \mu_r$ for small inductions B and $\mu = \mu_0$ for high inductions B with a continuous transition. A sketch for this approach can be seen in Fig. 2.5. The anhysteretic function is implemented for both stator and rotor material.

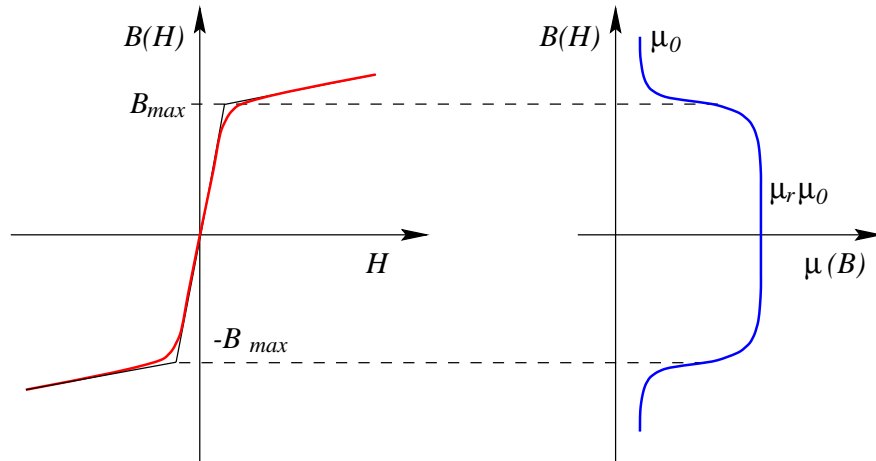


Figure 2.5: Anhysteretic curve with an induction dependent permeability $\mu(B)$.

As example, this function was computed for the stator material. The corresponding plot can be seen in Fig. 2.6

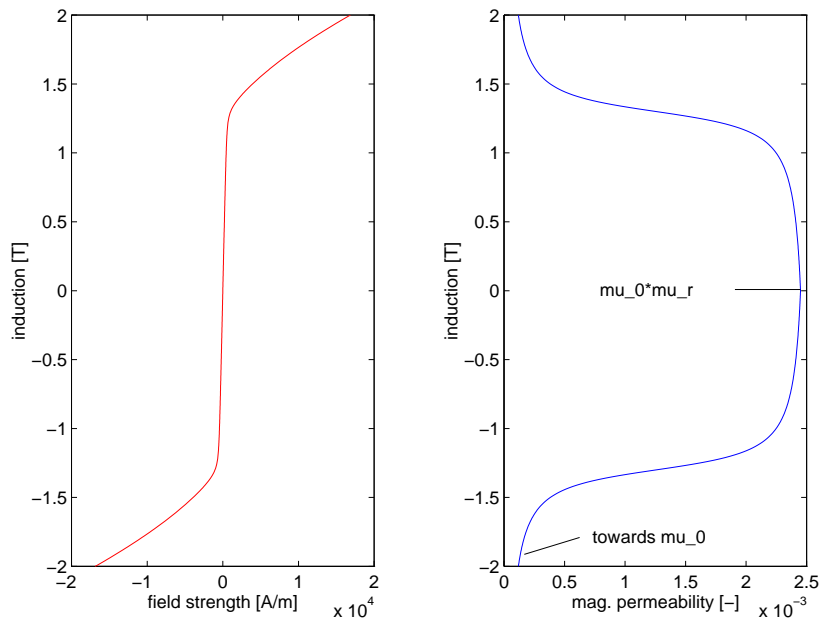


Figure 2.6: Anhysteretic curve with an induction dependent permeability $\mu(B)$ for the stator material.

Within the simulation two bearing actuators have to be implemented with four electromagnet models each. For this sake Eqn.(2.12), Eqn.(2.12) and Eqn.(2.12) have to be translated into a SIMULINK model using the definition for the magnetic reluctance $\mathcal{R}(l)$ in Eqn.(2.11). As mentioned previously, the relative permeability depends on the induction B within the simulation model. Therefore, Eqn.(2.15) has to be implemented for each electromagnet for both the stator and the rotor material. The corresponding SIMULINK block diagram is shown in Fig. D.7 in Appendix D.1.

2.2.5 Rotor model

Figure 2.7 shows a sketch of the rotor bearing system under investigation. The rigid rotor is suspended by two active magnetic bearings at positions A' and B'. Two sensors are mounted perpendicularly to the shaft axis at the positions A'' and B''. Non-conservative self-exciting forces are expected to act in the plane N'. All system parameters used for numerical simulation are shown in Tab. 2.4.

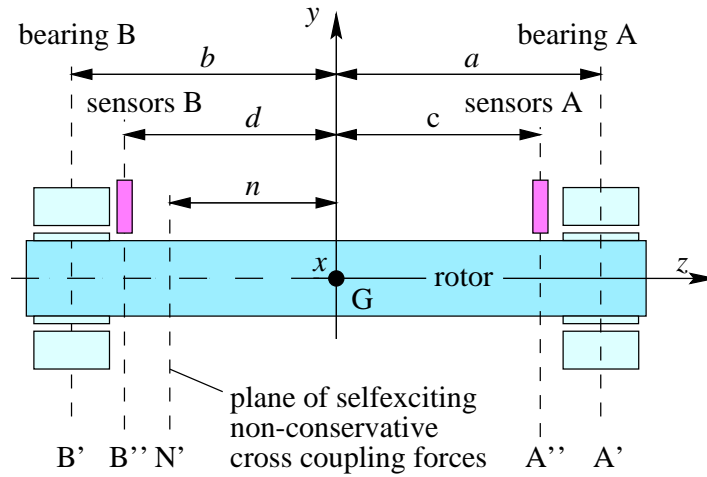


Figure 2.7: Sketch of a rigid rotor suspended by two magnetic bearings and excited by non-conservative cross-coupling forces at position N'. G denotes the centre of mass.

Inputs to the rotor are the actuator forces \mathbf{F}_a at positions A' and B', outputs are the rotor positions \mathbf{z}_s and \mathbf{z}_b in the sensor planes and in bearing planes, respectively. The second order equation of motion for a rigid rotor with its degrees of freedom expressed by the bearing coordinates $\mathbf{z}_b = [x_A, y_A, x_B, y_B]^T$ is given by

$$\mathbf{M}_b \ddot{\mathbf{z}}_b + \mathbf{G}_b \dot{\mathbf{z}}_b + \mathbf{N}_b \mathbf{z}_b = \mathbf{F}_a + \mathbf{F}_w, \quad (2.16)$$

with the input bearing force vector $\mathbf{F}_a = [F_{x_A}, F_{y_A}, F_{x_B}, F_{y_B}]^T$ and the force vector corresponding to the rotor weight expressed by bearing coordinates

$$\mathbf{F}_w = -\frac{m_r g}{a-b} \begin{bmatrix} 0 & -b & 0 & a \end{bmatrix}^T. \quad (2.17)$$

Table 2.4: Parameters of rigid rotor suspended in two active magnetic bearings

g	earth acceleration	9.81 m/s ²
m_r	rotor mass	28.7680 kg
I_a	axial mass moment of inertia	0.8632 kgm ²
I_p	polar mass moment of inertia	0.02188 kgm ²
a	distance to bearing A	0.23877 m
b	distance to bearing B	-0.24123 m
c	distance to sensor A	0.18977 m
d	distance to sensor B	-0.19233 m
n	distance to plane N'	-0.1 m
k_n	non-conservative cross-coupling stiffness	0-10 ⁷ N/m
Ω	rotor speed	20,000 rpm

All matrices are determined by a transformation from the coordinates of the centre of gravity $\mathbf{z}_c = [\phi_c, x_c, \psi_c, y_c]^T$ into bearing coordinates \mathbf{z}_b using $\mathbf{z}_b = \mathbf{T} \mathbf{z}_c$ in the form

$$\mathbf{M}_b = \mathbf{T}^{-T} \mathbf{M}_c \mathbf{T}^{-1}, \quad (2.18)$$

$$\mathbf{G}_b = \mathbf{T}^{-T} \mathbf{G}_c \mathbf{T}^{-1}, \quad (2.19)$$

$$\mathbf{N}_b = \mathbf{T}^{-T} \mathbf{T}_n^T \mathbf{N}_n \mathbf{T}_n \mathbf{T}^{-1}, \quad (2.20)$$

with the mass matrix of the rotor

$$\mathbf{M}_c = \begin{bmatrix} I_a & 0 & 0 & 0 \\ 0 & m_r & 0 & 0 \\ 0 & 0 & I_a & 0 \\ 0 & 0 & 0 & m_r \end{bmatrix}, \quad (2.21)$$

and its gyroscopic matrix

$$\mathbf{G}_c = \begin{bmatrix} 0 & 0 & I_p & 0 \\ 0 & 0 & 0 & 0 \\ -I_p & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega, \quad (2.22)$$

corresponding to \mathbf{z}_c . The geometrical transformation matrices are given by

$$\mathbf{T} = \begin{bmatrix} a & 1 & 0 & 0 \\ 0 & 0 & a & 1 \\ b & 1 & 0 & 0 \\ 0 & 0 & b & 1 \end{bmatrix}, \quad (2.23)$$

$$\mathbf{T}_n = \begin{bmatrix} n & 1 & 0 & 0 \\ 0 & 0 & n & 1 \end{bmatrix}, \quad (2.24)$$

and the matrix of the self-exciting non-conservative cross-coupling coefficients in the plane N' has the form

$$\mathbf{N}_n = \begin{bmatrix} 0 & k_n \\ -k_n & 0 \end{bmatrix}. \quad (2.25)$$

The output vector $\mathbf{z}_s = [x_{sA}, y_{sA}, x_{sB}, y_{sB}]^T$ in sensor coordinates is determined by

$$\mathbf{z}_s = \mathbf{T}_s \mathbf{z}_c = \mathbf{T}_s \mathbf{T}^{-1} \mathbf{z}_b, \quad (2.26)$$

with

$$\mathbf{T}_s = \begin{bmatrix} c & 1 & 0 & 0 \\ 0 & 0 & c & 1 \\ d & 1 & 0 & 0 \\ 0 & 0 & d & 1 \end{bmatrix}. \quad (2.27)$$

For the use in a simulation, all these equations are modelled with SIMULINK built-in functions as it can be seen in the block diagram as shown in Fig. D.9 in Appendix D.1.

2.2.6 Position sensors

The position sensors have the input $\mathbf{z} = \mathbf{z}_s + \mathbf{v}$, where \mathbf{v} is the measurement noise having a maximal deflection of $v \mu\text{m}$ and a covariance matrix $\mathbf{Q} = \mathcal{E} \{ \mathbf{v}\mathbf{v}^T \}$. \mathbf{z}_s is the rotor position vector at the sensor stations. The sensors provide the measured rotor displacements $\mathbf{z}_m = k_m \mathbf{z}$ with the parameters given in Tab. 2.5. It is assumed that the rotor rotation does not cause additional, periodic run-out errors.

Table 2.5: Parameters of position sensors

Sensor gain k_m	10^6 -
Measurement noise v	1 μm
Noise covariance \mathbf{Q}	\mathbf{I} μm^2

Usually the measurement includes a conversion from rotor displacement in meters to V and within the DSP into m again. The measurement process is short cut by the conversion constant k_m for sake of simplicity. Additionally, the gain k_m is introduced to convert the position signals \mathbf{z} from m to μm , because the simulation is carried out in m.

2.2.7 ADC for measured rotor positions

Input to the ADCs are the measured positions \mathbf{z}_m , output the sampled position values \mathbf{z}_d (4 channels). Since the position in m is transformed into V, then converted into the discrete domain, and later on converted into μm , the conversion process is simulated by a series of quantiser, time delay and saturation with the parameters in Tab. 2.6. The corresponding SIMULINK block diagram is shown in Fig. D.8 in Appendix D.1.

Table 2.6: Parameters of ADC for position sensors

Input position range	± 250 μm
Bit resolution	10 -
Quantisation for position	0.489 μm
Conversion delay	3 μs

2.2.8 ADCs for measured currents

Current sensors (LEM-modules) are located on the switching amplifier boards and provide the measured coil currents i_m of all actuators. Inputs to the ADCs (8 channels) are the measured coil currents i_m , outputs are the sampled current values i_d for all magnets of all actuators. Since the current (in A) is output in V, then converted into the discrete domain, and finally re-converted into A, the conversion process is simulated by a series of quantiser, time delay and saturation, with the parameters in Tab. 2.7, similarly to the ADCs for the position signals. The corresponding SIMULINK block diagram is shown in Fig. D.5 in Appendix D.1.

Table 2.7: Parameters of ADC for current sensors

Input current range	0 – 8 A
Bit resolution	9 -
Quantisation for current	0.0157 A
Conversion delay	3 μs

2.3 Model linearisation

The goal for the linearisation process is to derive a linear state space model of the nonlinear rotor bearing system as described above. Several steps are necessary to derive such a state space model.

1. A point of operation has to be defined for the rotor bearing system, i.e. for all energy storing state variables [Cellier, 1991]. From here, the entire system can be expected to behave linearly within a certain range of all state variables involved.
2. A current control loop is introduced. This loop guarantees the coil current to follow the control current from the point of operation within a certain bandwidth and reduces the order of the state space model.
3. Assuming the coil current as an independent input, a linear model for an electromagnet has to be derived for the point of operation.

4. A linear model for the magnetic actuator is derived. Below a certain frequency depending on the current controller the actuator can be treated as linear gain with respect to the control current and with respect to the rotor displacement.
5. A linear state space model of the rotor bearing system can be derived with control currents as input and rotor position as output.

An alternative solution is to control the rotor bearing system using a comprehensive state space controller including the states of all coils of both actuators. This approach, however, is not appropriate for adaptive control, because the number of parameters to be estimated becomes too large. Therefore, control is done by a cascaded current controller and a position controller. Note that the subsequently derived position controller always requires an underlying current control loop.

In the following subsections the presented steps lead to a linear actuator model and end up with a discrete time, linear state space model in controller canonical form.

2.3.1 Point of operation

The point of operation is defined by steady state values for all state variables involved in the rotor bearing system. In the present investigation these are the rotor position, the rotor velocity, and all fluxes of all electromagnets of both magnetic actuators defined by the bias current. This bias current is 4 A for the present application. The rotor position and velocity are zero for the point of operation.

2.3.2 Current control loop

The position control loop basically consists of the magnetic actuator, the current sensors, the ADCs, the controller, the DACs, the PWMs and the switching power amplifier. However, the conversion delay of $3\ \mu\text{s}$ for the ADCs and DACs is negligible in comparison with the sample time of $100\ \mu\text{s}$. Therefore, not all subsystems have to be included for the linear model. Sensors, ADCs and DACs can be regarded as a unit gain with quantisation noise corrupting the output. The only subsystems remaining to be linearised are the electrical power signal path including the current measurement, the current controller, the switching power amplifier, and the electromagnets with the magnet force as output.

Linearisation of the current control loop

For two active magnetic bearings with four magnets each, in total 8 channels are necessary to be controlled. The controller for the current loop is implemented in the DSP as well, but logically separated from the position control loop.

The switching power amplifier with its PWM can be treated as gain with modulation noise depending on the input. The gain of the amplifier is determined by the ratio

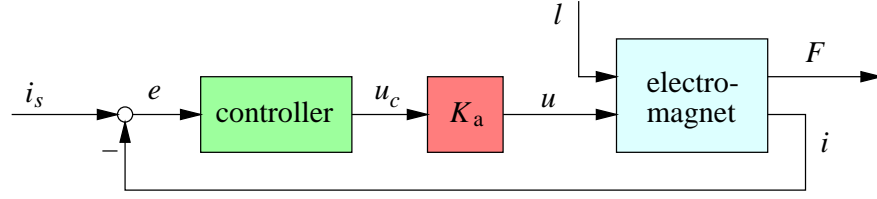


Figure 2.8: Block diagram of the current control loop with input i_s as the desired current, the air gap length l as the disturbance, the output F as the actuator force. K_a denotes the amplifier gain, e the control error, u_c the control voltage and u the input voltage to the electromagnet.

between switching voltage range and output voltage range of the DAC, in this case $150/5 = 30$.

Finally, the series DAC, PWM and switching amplifier is replaced by a gain $K_a = 30$, the series sensor and ADC by a unit gain. This means that the measured and sampled currents \mathbf{i}_d are assumed to be the true currents \mathbf{i} . The corresponding block diagram can be seen in Fig. 2.8.

For a current controller design, a linear model for the electromagnet has to be derived from the nonlinear model presented in Section 2.2.4. How the digital control voltages \mathbf{u}_d are computed from the control voltages \mathbf{u}_c , is described in Section 4.1.2.

Linearisation of an electromagnet

Given the point of operation defined by i_0 and l_0 for a centred rotor, the nonlinear Eqn.(2.12) to Eqn.(2.14) can be rewritten as

$$\frac{d\Phi}{dt} = \dot{\Phi}(\Phi(t), \mathcal{R}(l(t)), u(t)), \quad (2.28)$$

$$F = F(\Phi(t)), \quad (2.29)$$

$$i = i(\Phi(t), \mathcal{R}(l(t))) \quad (2.30)$$

and can be linearised with respect to the deviation variables $\Delta\Phi = \Phi - \Phi_0$, $\Delta F = F - F_0$, $\Delta i = i - i_0$ in the form

$$\frac{d(\Delta\Phi)}{dt} = -K_{\dot{\Phi}\Phi} \Delta\Phi + K_{\dot{\Phi}l} \Delta l + K_{\dot{\Phi}u} \Delta u \quad (2.31)$$

$$\Delta F = K_{F\Phi} \Delta\Phi \quad (2.32)$$

$$\Delta i = K_{i\Phi} \Delta\Phi + K_{il} \Delta l \quad (2.33)$$

using the partial derivatives

$$-K_{\dot{\Phi}\Phi} = \left. \frac{\partial \dot{\Phi}}{\partial \Phi} \right|_0 = -\frac{r \mathcal{R}_0}{N^2}, \quad (2.34)$$

$$K_{\dot{\Phi}l} = \left. \frac{\partial \dot{\Phi}}{\partial l} \right|_0 = \left. \frac{\partial \dot{\Phi}}{\partial \mathcal{R}} \cdot \frac{\partial \mathcal{R}}{\partial l} \right|_0 = -\frac{2r \Phi_0}{\mu_0 N^2 A_l}, \quad (2.35)$$

$$K_{\dot{\Phi}u} = \left. \frac{\partial \dot{\Phi}}{\partial u} \right|_0 = \frac{1}{N}, \quad (2.36)$$

$$K_{F\Phi} = \left. \frac{\partial F}{\partial \Phi} \right|_0 = \frac{2 \Phi_0}{\mu_0 A_l} \cos \frac{\alpha}{2}, \quad (2.37)$$

$$K_{i\Phi} = \left. \frac{\partial i}{\partial \Phi} \right|_0 = \frac{\mathcal{R}_0}{N}, \quad (2.38)$$

$$K_{il} = \left. \frac{\partial i}{\partial l} \right|_0 = \left. \frac{\partial i}{\partial \mathcal{R}} \cdot \frac{\partial \mathcal{R}}{\partial l} \right|_0 = \frac{2 \Phi_0}{\mu_0 N A_l}, \quad (2.39)$$

and the quantities for the point of operation

$$\mathcal{R}_0 = \frac{1}{\mu_0} \left[\frac{l_r}{A_r \mu_{r_r}} + \frac{l_s}{A_s \mu_{r_s}} + \frac{2l_0}{A_l} \right] \quad \text{and} \quad \Phi_0 = \frac{N i_0}{\mathcal{R}_0}. \quad (2.40)$$

Transforming Eqn.(2.31) to Eqn.(2.33) into the Laplace-domain and solving for $\Delta F(s)$ and $\Delta i(s)$ yields

$$\Delta F(s) = G_{lF}(s) \Delta l(s) + G_{uF}(s) \Delta u(s), \quad (2.41)$$

$$\Delta i(s) = G_{li}(s) \Delta l(s) + G_{ui}(s) \Delta u(s). \quad (2.42)$$

with the transfer functions

$$G_{lF}(s) = \frac{K_{F\Phi} K_{il}}{s + K_{\dot{\Phi}\Phi}}, \quad (2.43)$$

$$G_{uF}(s) = \frac{K_{F\Phi} K_{iu}}{s + K_{\dot{\Phi}\Phi}}, \quad (2.44)$$

$$G_{li}(s) = \frac{K_{i\Phi} K_{\dot{\Phi}l} + [s + K_{\dot{\Phi}\Phi}] K_{il}}{s + K_{\dot{\Phi}\Phi}}, \quad (2.45)$$

$$G_{ui}(s) = \frac{K_{i\Phi} K_{\dot{\Phi}u}}{s + K_{\dot{\Phi}\Phi}}. \quad (2.46)$$

For the sake of simplicity a continuous time PI-controller is introduced in the form

$$G_c(s) = \frac{\Delta u_c(s)}{\Delta e(s)} = K_c \frac{s + K_{\dot{\Phi}\Phi}}{s}. \quad (2.47)$$

Thus, all transfer functions for the linearised system are defined. The corresponding block diagram can be seen in Fig. 2.9.

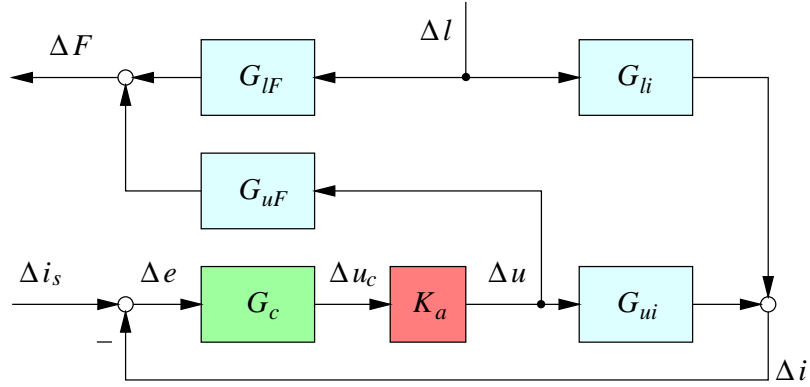


Figure 2.9: Block diagram of the linear current control loop with input Δi_s and Δl , and the output ΔF .

Current controller design

The current controller design is carried out in the continuous time domain, because the physical behaviour is more transparent than in the discrete time domain. In the present case, the zero of the proposed controller cancels the slow pole of the electromagnet. Thus, the plant transfer function results in

$$G_p(s) = G_c(s) K_a G_{ui}(s) = \frac{K_c K_a \mathcal{R}_0}{N^2 s}. \quad (2.48)$$

This means that the phase lag is theoretically -90 deg over the entire frequency range and the controller gain can be chosen arbitrarily to achieve a current control loop with unlimited bandwidth. In reality, there are restrictions like saturation and a natural phase lag due to the sampling process and conversion delay. Approximating the zero-order sample hold element by a first order delay (Pade approximation)

$$H_0(s) = \frac{1}{1 + \frac{T_s}{2} s}, \quad (2.49)$$

the open loop transfer function is rewritten as

$$G_o(s) = \frac{K_c K_a \mathcal{R}_0}{N^2 s (1 + \frac{T_s}{2} s)}. \quad (2.50)$$

The controller gain is determined for a certain phase margin and a maximum bandwidth using the frequency plots. Finally, the reference transfer function can be calculated as

$$G_{ii_s}(s) = \frac{\Delta i}{\Delta i_s} = \frac{\omega_n^2}{s^2 + 2 \zeta \omega_n s + \omega_n^2}, \quad (2.51)$$

with the relations between the undamped eigenfrequency ω_n and the damping factor ζ

$$\omega_n^2 = \frac{2 K_c K_a \mathcal{R}_0}{N^2 T_s} \quad (2.52)$$

$$2 \zeta \omega_n = \frac{2}{T_s}. \quad (2.53)$$

Note that the resulting reference transfer function is an approximation for the real system being controlled by a digital controller. Additionally, the zero order hold element is only used for control design purposes. An alternative solution is the z-transformation of the plant transfer function and the control design in the discrete time domain.

The numerical results for the controller design are presented in Section 4.1.1. The implementation is explained in detail in Subsection 4.1.2.

2.3.3 Linearised model of an electromagnet with underlying current control loop

Once the current control loop has been designed, the coil current can be assumed to be the desired one within a certain bandwidth. Therefore, the behaviour of the actuator consisting of four electromagnets must not depend on the controller properties. In the following the linearised model of an electromagnet with a current control loop is derived.

Conventional approach

In this model the output of the electromagnet is the ponderomotoric air-gap force following the law

$$F = \frac{\Phi(i, l)^2}{\mu_0 A_l} \cos \frac{\alpha}{2}, \quad (2.54)$$

with the magnetic flux

$$\Phi(i, l) = \frac{N i}{\mathcal{R}(l)}, \quad (2.55)$$

depending on the coil current $i = i_0 + \Delta i$ and the magnetic reluctance $\mathcal{R}(l)$ which is a function of the air-gap length $l = l_0 + \Delta l$. For the point of operation defined by i_0 and l_0 , the linear relation for the magnet force is

$$\Delta F = K_{lF} \Delta l + K_{iF} \Delta i, \quad (2.56)$$

with the partial derivatives

$$K_{iF} = \left. \frac{\partial F}{\partial i} \right|_0 = \frac{2 N^2 i_0}{\mathcal{R}_0^2 \mu_0 A_l} \cos \frac{\alpha}{2}, \quad (2.57)$$

$$K_{lF} = \left. \frac{\partial F}{\partial l} \right|_0 = \frac{\partial F}{\partial \mathcal{R}} \cdot \left. \frac{\partial \mathcal{R}}{\partial l} \right|_0 = -\frac{4 N^2 i_0^2}{\mathcal{R}_0^3 \mu_0^2 A_l^2} \cos \frac{\alpha}{2}, \quad (2.58)$$

being the steady state coefficients for an electromagnet with respect to the air-gap forces.

Non conventional approach

From the control point of view, the coil current cannot be assumed to be identical with the desired current, i.e. the electromagnet cannot be treated independently from the controller. This can only be the case for low frequencies, where the reference transfer function G_w behaves like a gain, or at least for a steady state. Therefore, an alternative derivation of the characteristic steady state coefficients K_{lF} and K_{iF} for an electromagnet is presented in the following.

Using the expressions $\Delta u(s) = K_a \Delta u_c(s)$ and $\Delta e(s) = \Delta i_s(s) - G_{i_s}(s) \Delta i_s(s)$ the voltage for one electromagnet becomes

$$\Delta u(s) = \frac{G_c(s) K_a}{1 + G_c(s) K_a G_{ui}(s)} \Delta i_s(s) - \frac{G_{li}(s) G_c(s) K_a}{1 + G_c(s) K_a G_{ui}(s)} \Delta l(s). \quad (2.59)$$

Finally, using this result and the above definition for $\Delta F(s)$, the magnet force can be written as a linear combination such that

$$\Delta F(s) = G_{lFc}(s) \Delta l(s) + G_{iFc}(s) \Delta i_s(s), \quad (2.60)$$

with the transfer functions

$$G_{iFc}(s) = \frac{G_{uF}(s) G_c(s) K_a}{1 + G_c(s) K_a G_{ui}(s)}, \quad (2.61)$$

$$G_{lFc}(s) = \frac{G_{lF}(s) [1 + G_c(s) K_a G_{ui}(s)] - G_{uF}(s) G_{li}(s) G_c(s) K_a}{1 + G_c(s) K_a G_{ui}(s)}. \quad (2.62)$$

Inserting the above defined transfer functions and the partial derivatives at the point of operation, these functions result in

$$G_{iFc}(s) = \frac{2 K_a K_c \Phi_0 N}{A_l \mu_0 [K_a K_c \mathcal{R}_0 + N^2 s]} \cos \frac{\alpha}{2}, \quad (2.63)$$

$$G_{lFc}(s) = -\frac{4 \Phi_0^2 [K_a K_c r \mathcal{R}_0 + N^2 [K_a K_c + r] s]}{A_l \mu_0^2 [K_a K_c \mathcal{R}_0 + N^2 s] [r \mathcal{R}_0 + N^2 s]} \cos \frac{\alpha}{2}. \quad (2.64)$$

Utilising the definitions for \mathcal{R}_0 and Φ_0 , the constant steady state coefficients for one electromagnet can be obtained from $s \rightarrow 0$ by

$$K_{iF} = G_{iFc}(s) \Big|_{s=0} = \frac{2 N^2 i_0}{\mathcal{R}_0^2 \mu_0 A_l} \cos \frac{\alpha}{2}, \quad (2.65)$$

$$K_{lF} = G_{lFc}(s) \Big|_{s=0} = -\frac{4 N^2 i_0^2}{\mathcal{R}_0^3 \mu_0^2 A_l^2} \cos \frac{\alpha}{2}. \quad (2.66)$$

It does not surprise that the coefficients are independent of the controller parameters as long as there is no steady state error between the desired current and the actual coil current. This is only the case if a PI-controller is used for the current control task.

Note that a continuous time controller was used in the derivation above. This procedure has actually no influence on the steady state behaviour. The dynamical behaviour, however, is affected by a digital implementation. If the dynamics of the digital implementation is to be included, all transfer functions have to be transformed into the z-plane in order to obtain an accurate result.

2.3.4 Linearised actuator model

In this investigation an actuator consists of four electromagnets as shown in Fig. 2.4. Outputs are the actuator forces $\mathbf{F}_a = [F_x, F_y]$ and four currents $\mathbf{i} = [i_1, i_2, i_3, i_4]$ of all coils. A current control loop is implemented for all electromagnets with the desired currents

$$i_{s_1} = i_0 + i_x, \quad (2.67) \quad i_{s_3} = i_0 - i_x, \quad (2.69)$$

$$i_{s_2} = i_0 + i_y, \quad (2.68) \quad i_{s_4} = i_0 - i_y, \quad (2.70)$$

depending on the control currents $\mathbf{i}_c = [i_x, i_y]$ in x and y directions respectively, and the bias current i_0 . From this point on, the dynamics of the current control loop is neglected, because the coil currents are expected to be equal to the desired ones ($i_{s_j} \stackrel{!}{=} i_j$ for $j = 1, 2, 3, 4$). This, however, requires an underlying current controller as a sub-task implemented in the DSP for all current loops.

Note that one has to distinguish between the desired currents i_{s_j} for $j = 1, 2, 3, 4$ which are absolute currents like the currents i_j , and the control current vector \mathbf{i}_c , which is the control variable of the position control loop assuming a linear actuator at the point of operation.

A linear model of an actuator can be derived using the assumptions for the air gap lengths defined by Eqn.(2.3) to Eqn.(2.6) in each electromagnet

$$F_x = F_1 - F_3 = K_s x + K_i i_x, \quad (2.71)$$

$$F_y = F_2 - F_4 = K_s y + K_i i_y, \quad (2.72)$$

with the current gain factor

$$K_i = 2 K_{iF} = \frac{4 N^2 i_0}{\mathcal{R}_0^2 \mu_0 A_l} \cos \frac{\alpha}{2} \quad (2.73)$$

and the position stiffness

$$K_s = 2 K_{lF} = -\frac{8 N^2 i_0^2}{\mathcal{R}_0^3 \mu_0^2 A_l^2} \cos \frac{\alpha}{2} \quad (2.74)$$

For the linear model, both bearing actuators are treated as one system with the control currents $\mathbf{i}_c = [i_{c_A}, i_{c_B}]$ as input to the bearing, and the resulting actuator forces $\mathbf{F}_a = [\mathbf{F}_{a_A}, \mathbf{F}_{a_B}]^T$ as output. The linearised model of the active magnetic bearings is then given by the linear relation

$$\mathbf{F}_a = \mathbf{K}_s \mathbf{z}_b + \mathbf{K}_i \mathbf{i}_c \quad (2.75)$$

with the stiffness matrix in bearing coordinates \mathbf{z}_b according to Fig. 2.10 $\mathbf{K}_s = K_s \mathbf{I}$, and the current gain matrix $\mathbf{K}_i = K_i \mathbf{I}$ with \mathbf{I} being the 4×4 identity matrix.

2.3.5 Linear rotor bearing model

After the linear model for an electro-magnetic actuator has been derived in the previous section, only the mechanical part remains with the rigid rotor and the active magnetic bearing as a gain with respect to the control current input and a negative stiffness coefficient with respect to the rotor position.

Sensors and ADCs can be regarded as a unit gain with quantisation noise corrupting the output. All noise of any kind generated by the system is gathered within a system noise vector for the discrete time, linear, and stochastic state space model. No specific model is provided for the disturbances. The quantisation noise is gathered within an input noise vector.

Thus, the position control loop consists basically of the magnetic actuator, the rotor, the sensors, the ADCs and the controller. The conversion delay of $3\mu\text{s}$ for the ADCs is negligible in comparison with the sample time of $100\mu\text{s}$. The conversion gain from m to μm is ascribed to the rotor model, which is linear itself. Starting with the linear continuous time model a stochastic discrete time state space model is derived in this section.

Linear continuous time model for the rotor magnetic bearing system

The rotor model itself is already linear, and the rotor weight \mathbf{F}_w can be compensated by an additional constant control current

$$\mathbf{i}_{c_w} = \mathbf{K}_i^{-1} \cdot \mathbf{F}_w. \quad (2.76)$$

Using the linear actuator model derived above, the rotor bearing equations of motion can be described by

$$\mathbf{M}_b \ddot{\mathbf{z}}_b + \mathbf{G} \dot{\mathbf{z}}_b + (\mathbf{N}_b + \mathbf{K}_s) \mathbf{z}_b = \mathbf{K}_i \mathbf{i}_c, \quad (2.77)$$

with the input vector $\mathbf{i}_c = [i_{x_A}, i_{x_B}, i_{y_A}, i_{y_B}]^T$, consisting of the control currents of four active magnetic bearing axes. Actually, the control currents \mathbf{i}_c only exist in the discrete domain as \mathbf{i}_c , but are assumed to be continuous for the derivation of the discrete state space model. The same is true for the set point vector \mathbf{w} . Figure 2.10 shows the linearised plant model.

Neglecting the transfer characteristics of the sensors, ADCs and DACs, the resulting continuous time state space model is then given in bearing coordinates in the form

$$\dot{\mathbf{x}}_b = \mathbf{A} \cdot \mathbf{x}_b + \mathbf{B} \cdot \mathbf{u}, \quad (2.78)$$

$$\mathbf{y} = \mathbf{C} \cdot \mathbf{x}_b + \mathbf{v}, \quad (2.79)$$

with the continuous time state vector $\mathbf{x}_b = [\mathbf{z}_b, \dot{\mathbf{z}}_b]^T$ (number of states $n_s = 8$), the output vector $\mathbf{y} = k_m(\mathbf{z}_b + \mathbf{v})$ (number of outputs $n_o = 4$), and the input vector $\mathbf{u} = \mathbf{i}_c$ (number of inputs $n_i = 4$). For numerical reasons a virtual sensor gain k_m converts the system states into μm . Using the output in μm yields parameters with reasonable ranges of value for both system matrices and controller parameters.

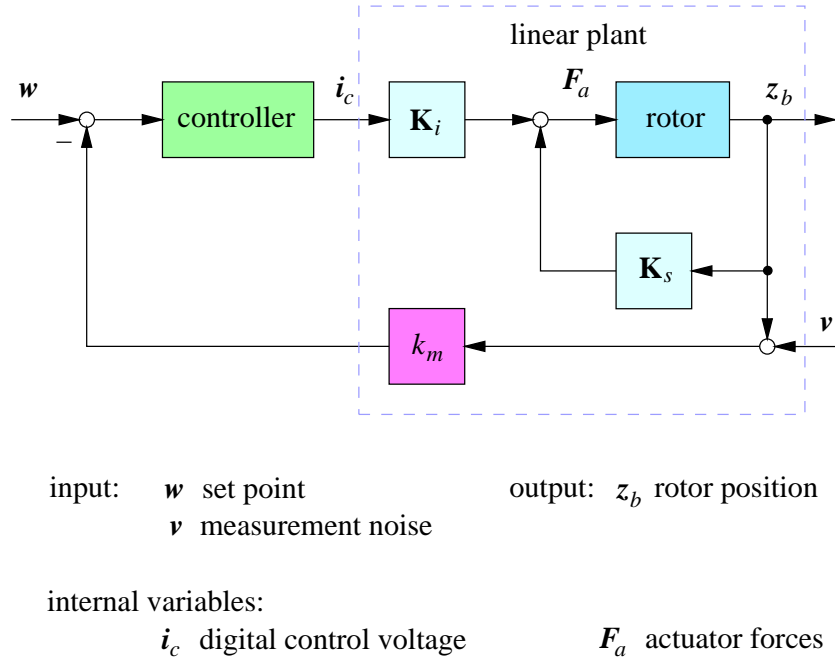


Figure 2.10: Block diagram of the linearised deterministic plant model.

Although the position is measured in the sensor planes, the proximity sensors are assumed to be collocated with the bearings for the linear model. This is justified for a rigid rotor model, because the displacements in the bearing planes z_b can easily be calculated from the sampled sensor signal z_d by the simple transformation (see Eqn.(2.26))

$$z_b = \mathbf{T} \mathbf{T}_s^{-1} z_d. \quad (2.80)$$

Similarly to \mathbf{i}_d , \mathbf{z}_d already is a discrete vector, but is assumed to have the continuous equivalent \mathbf{z}_d .

The system matrices are defined as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_b^{-1} (\mathbf{N}_b + \mathbf{K}_s) & -\mathbf{M}_b^{-1} \mathbf{G}_b \end{bmatrix}, \quad (2.81)$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_b^{-1} \mathbf{K}_i \end{bmatrix}, \quad (2.82)$$

$$\mathbf{C} = [k_m \mathbf{I} \quad \mathbf{0}]. \quad (2.83)$$

Deterministic discrete time state space model for the rotor bearing system

A transformation into a discrete time system with sampling time T_s yields the deterministic discrete state space model

$$\mathbf{x}_b(k+1) = \mathbf{\Phi} \mathbf{x}_b(k) + \mathbf{\Gamma} \mathbf{u}(k), \quad (2.84)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{x}_b(k), \quad (2.85)$$

with its system matrices of no particular structure, namely the transition matrix and the discrete time control matrix (see [Föllinger, 1990])

$$\mathbf{\Phi} = e^{\mathbf{A} \cdot T_s} \quad \text{and} \quad (2.86)$$

$$\mathbf{\Gamma} = \left[e^{\mathbf{A} \cdot T_s} - \mathbf{I} \right] \cdot \mathbf{A}^{-1} \cdot \mathbf{B}. \quad (2.87)$$

Since the discrete state space model should have a minimum number of parameters to be identified, it is transformed into a canonical form. With respect to the calculation of an adaptive controller, a controller canonical form is chosen [Tolle, 1985]. For the reason of symmetry the structural indices are $\nu_i = 2$, which means that there are four coupled second order systems [Schumann, 1982, Tolle, 1985]. With the transformation

$$\mathbf{x} = \mathbf{T} \mathbf{x}_b \quad \text{and} \quad \mathbf{x}_b = \mathbf{T}^{-1} \mathbf{x} \quad (2.88)$$

using the transformation matrix \mathbf{T} the new matrices are

$$\mathbf{A} = \mathbf{T} \mathbf{\Phi} \mathbf{T}^{-1}, \quad (2.89)$$

$$\mathbf{B} = \mathbf{T} \mathbf{\Gamma}, \quad (2.90)$$

$$\mathbf{C} = \mathbf{C} \mathbf{T}^{-1}. \quad (2.91)$$

Appendix A gives the derivation of the transformation \mathbf{T} . The entire deterministic model in canonical form can then be written as

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k), \quad (2.92)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{x}(k), \quad (2.93)$$

with the (8×1) -state vector $\mathbf{x}(k)$. The (4×1) -input vector $\mathbf{u}(k)$ is the sampled input vector $\mathbf{u}(t)$ and the (4×1) -output vector $\mathbf{y}(k)$ is the sampled vector $\mathbf{y}(t)$. The (8×8) -system matrix $\mathbf{A} = \{\mathbf{A}^{(ij)}\}$ is partitioned into sub-matrices of the form

$$\mathbf{A}^{(ij)} = \begin{bmatrix} 0 & 1 \\ -a_2^{(ij)} & -a_1^{(ij)} \end{bmatrix}, \quad (2.94)$$

$$\mathbf{A}^{(ii)} = \begin{bmatrix} 0 & 0 \\ -a_2^{(ii)} & -a_1^{(ii)} \end{bmatrix}, \quad (2.95)$$

with $i, j = 1, 2, 3, 4$. The (8×4) -control matrix $\mathbf{B} = \{\mathbf{b}^{(ij)}\}$ is composed of

$$\mathbf{b}^{(ij)} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T, \quad (2.96)$$

$$\mathbf{b}^{(ii)} = \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \quad (2.97)$$

and the (4×8) measurement matrix $\mathbf{C} = \{\mathbf{c}^{T(ij)}\}$ is

$$\mathbf{c}^{T(ij)} = \begin{bmatrix} c_2^{(ij)} & c_1^{(ij)} \end{bmatrix}. \quad (2.98)$$

The matrices are computed numerically using the transformation matrix derived in Appendix A from the transition matrix and the discrete time control matrix according to Eqn.(2.86) and Eqn.(2.87), respectively.

Stochastic state space model for the rotor bearing system

Based upon the deterministic state space model defined in Eqn.(2.92) and Eqn.(2.93), the general stochastic state space model that is widely used in systems theory is

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k) + \boldsymbol{\xi}(k), \quad (2.99)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{x}(k) + \boldsymbol{\eta}(k), \quad (2.100)$$

with the sequences of independent random noise vectors as the (8×1) -system noise $\boldsymbol{\xi}(k)$ and the (4×1) -measurement noise $\boldsymbol{\eta}(k)$ with each vector being of zero mean

$$\mathcal{E} \{ \boldsymbol{\xi}(k) \} = \mathbf{0}, \quad (2.101)$$

$$\mathcal{E} \{ \boldsymbol{\eta}(k) \} = \mathbf{0}, \quad (2.102)$$

and with the covariance matrices

$$\mathcal{E} \{ \boldsymbol{\xi}(k) \boldsymbol{\xi}^T(k) \} = \mathbf{R}_1, \quad (2.103)$$

$$\mathcal{E} \{ \boldsymbol{\eta}(k) \boldsymbol{\eta}^T(k) \} = \mathbf{R}_2, \quad (2.104)$$

$$\mathcal{E} \{ \boldsymbol{\eta}(k) \boldsymbol{\xi}^T(k) \} = \mathbf{R}_{12}. \quad (2.105)$$

Within the system noise $\boldsymbol{\xi}(k)$ all noise generated within the system is modelled without an additional input matrix. The estimation of such an input matrix would not be possible, because the disturbances cannot be measured. All noise as generated by the ADCs and the measurement noise is gathered within the measurement noise vector $\boldsymbol{\eta}(k)$. The covariance matrices \mathbf{R}_1 , \mathbf{R}_2 and \mathbf{R}_{12} are generally not known, but can be estimated online.

This model structure yields 64 parameters to be estimated, although the matrices \mathbf{A} with $n_s^2 = 64$ entries, \mathbf{B} and \mathbf{C} with $n_s n_i = 32$ and $n_s n_o = 32$ entries, respectively, have a total of 128 entries. This is due to the controller canonical form. Using this form, in the present case, only every second row of \mathbf{A} contains parameters to be estimated, i.e. only $\frac{n_s^2}{2} = 32$ parameters. The control matrix \mathbf{B} is fixed in its structure and can be omitted regarding parameter estimation, because it only contains zeros and ones. The measurement matrix has no particular structure which means that all 32 entries are parameters to be estimated.

Note, that these parameters are no longer parameters having a physical meaning, but are the result of various transformations. Additionally, the state vector \mathbf{x} is not identical with the sampled state vector \mathbf{x}_b of the original continuous time model.

Chapter 3

State space adaptive control

A regulator is a feedback mechanism by which a system is guided along a desired behaviour. The design of this regulator is commonly based upon a mathematical model of the system to be controlled. Such a model as derived in the previous chapter is treated as a true description of the system and is expected not to change in terms of its structure. The plant parameters, however, can be slowly time-variant or may be uncertain from the beginning. In these cases, adaptive control is necessary. A state space approach to adaptive control is presented in this chapter.

“... adaptive control is a special type of nonlinear feedback control in which the states of the process are separated in two categories, which change at different rates. The slowly changing states are viewed as parameters.”, [Åstrøm and Wittenmark, 1988].

This means that there is a fast feedback loop with an ordinary controller calculating the control variable from the fast states like the system output. A slow feedback loop is then wrapped around the first one adapting the regulator parameters which are the slow states according to [Åstrøm and Wittenmark, 1988] as *meta-regulator* in order to guarantee a desired closed loop behaviour if the controlled system changes (see Fig. 3.1). The controller parameters can be estimated directly from the states of the first feedback loop or indirectly via a separate controller design. In this work an indirect approach for adaptive control has been chosen, i.e. the adaptation process is separated into two parts, the identification of the system parameters and its states and the calculation of the controller parameters based upon the estimated plant parameters. For both parts the MATLAB program codes are included in Appendix D.2.

3.1 Parameter and state estimation

Identification is the extraction of parameters according to a model from measured data of a system under investigation. This process can be done on-line or off-line. Since parameters of a system can change during operation, the estimation of these parameters has to be done on-line, i.e. at each sample time when new information is available. Only

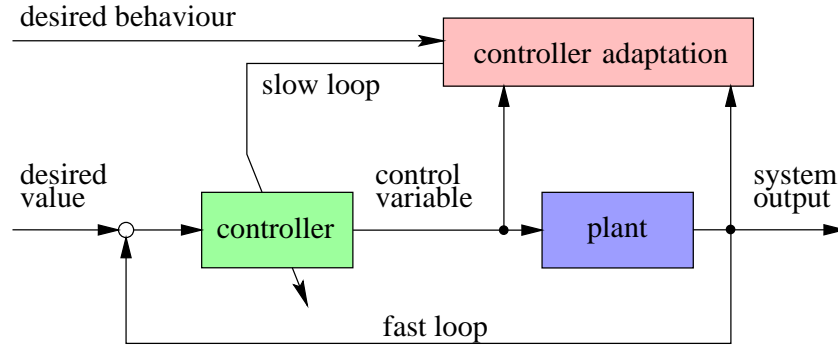


Figure 3.1: Block diagram of an adaptive control loop.

recursive algorithms can be used for that task, because its computation time must be constant. Such algorithms calculate a new estimate of the system parameters from a previous one, the current data and eventually from auxiliary variables based on a model for systems with stochastic disturbances. A basic requirement for an estimation algorithm is a predictor which predicts future outputs of a system under the assumption of known system parameters.

3.1.1 Innovations model

Based upon the stochastic state space model defined by Eqn.(2.99) and Eqn.(2.100) derived in Chapter 2, a predictor for the output $\mathbf{y}(k)$ including the Kalman matrix \mathbf{K} is introduced as [Ljung and Söderström, 1983]

$$\hat{\mathbf{x}}(k+1) = [\mathbf{A} - \mathbf{K}\mathbf{C}] \hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{K}\mathbf{y}(k), \quad (3.1)$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k). \quad (3.2)$$

At the moment the system matrices are assumed to be known, and the time invariant Kalman matrix \mathbf{K} is calculated via the Riccati equation [Ljung and Söderström, 1983] from the covariance matrices. The usual solution in adaptive control using the Kalman filter is to estimate only the parameters of the system matrix and the covariances $\mathbf{R}_1, \mathbf{R}_2$ and \mathbf{R}_{12} as defined in Eqn.(2.103), Eqn.(2.104) and Eqn.(2.105), and hereafter calculate the Kalman matrix via the time variant Riccati equation. A different approach is to directly estimate the parameters of the Kalman matrix \mathbf{K} . Then the matrices \mathbf{A} , \mathbf{K} and \mathbf{C} depend on their parameters \mathbf{p} . Eqn.(3.1) and Eqn.(3.2) can be rewritten as

$$\hat{\mathbf{x}}(k+1, \mathbf{p}) = \mathbf{A}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}) + \mathbf{B}\mathbf{u}(k) + \mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon}(k), \quad (3.3)$$

$$\mathbf{y}(k) = \mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}) + \boldsymbol{\varepsilon}(k), \quad (3.4)$$

where $\boldsymbol{\varepsilon}(k)$ corresponds to the prediction error or *innovation* with

$$\boldsymbol{\varepsilon}(k, \mathbf{p}) = \mathbf{y}(k) - \hat{\mathbf{y}}(k|\mathbf{p}), \quad (3.5)$$

and $\hat{\mathbf{y}}(k|\mathbf{p})$ is the estimated output of the system under the assumption of the parameter vector \mathbf{p} . For this reason the model described by Eqn.(3.3) and Eqn.(3.4) is known as *innovations model*. The Kalman matrix is parameterised as $\mathbf{K}(\mathbf{p}) = \{\mathbf{k}^{(ij)}\}$ with $i, j = 1, 2, 3, 4$ and

$$\mathbf{k}^{(ij)} = \begin{bmatrix} k_2^{(ij)} & k_1^{(ij)} \end{bmatrix}^T. \quad (3.6)$$

All parameters of this innovations model defined by the matrices \mathbf{A} , \mathbf{C} and \mathbf{K} are summarised within the n_p -dimensional parameter vector

$$\mathbf{p} = \left[-a_2^{(11)}, -a_1^{(11)}, \dots, c_2^{(11)}, c_1^{(11)}, \dots, k_2^{(11)}, k_1^{(11)}, \dots \right]. \quad (3.7)$$

The Kalman matrix introduces $n_s n_o = 32$ more parameters to the entire estimation algorithm, which means that there are $n_p = 96$ parameters to be estimated. Note, that the control-input matrix \mathbf{B} , as defined in Eqn.(2.96) and Eqn.(2.97), does not depend on the parameter vector, if the system is described in controller canonical form.

Based upon the innovations model a recursive algorithm has to be implemented to estimate the system parameters and the Kalman matrix under on-line conditions for a controlled system with stochastic disturbances.

3.1.2 The recursive prediction error method

The *recursive prediction error method* (RPEM, [Ljung and Söderström, 1983]) applied to the innovations model is used in this work. Figure 3.2 shows the principle of the algorithm. This algorithm can estimate both system parameters and the system states. An innovations model is used to predict the system states $\hat{\mathbf{x}}(k)$ and the output $\hat{\mathbf{y}}(k)$ based upon the system inputs $\mathbf{u}(k)$ and its parameters $\hat{\mathbf{p}}(k)$. The difference between model output $\hat{\mathbf{y}}(k)$ and the measured system output $\mathbf{y}(k)$, the prediction error $\boldsymbol{\varepsilon}(k)$, is then used for updating the system parameters $\hat{\mathbf{p}}(k)$. These parameters are later on used for the controller design. Independent inputs are the measurement noise $\mathbf{v}(k)$ and the set point $\mathbf{w}(k)$. In the following this algorithm is explained in detail.

3.1.3 Derivation of the recursive algorithm

The goal of any estimation algorithm is to keep the prediction error $\mathbf{y} - \hat{\mathbf{y}}(k|\mathbf{p})$ as small as possible in terms of its statistical properties. Thus, it is necessary to define a criterion functional to be minimised as for example

$$V(\mathbf{p}) = \mathcal{E} \{ H(\mathbf{p}, \boldsymbol{\varepsilon}) \} \rightarrow \text{Min.} \quad (3.8)$$

with the quadratic function

$$H(\mathbf{p}, \boldsymbol{\varepsilon}) = \boldsymbol{\varepsilon}^T(k, \mathbf{p}) \boldsymbol{\Lambda}^{-1} \boldsymbol{\varepsilon}(k, \mathbf{p}), \quad (3.9)$$

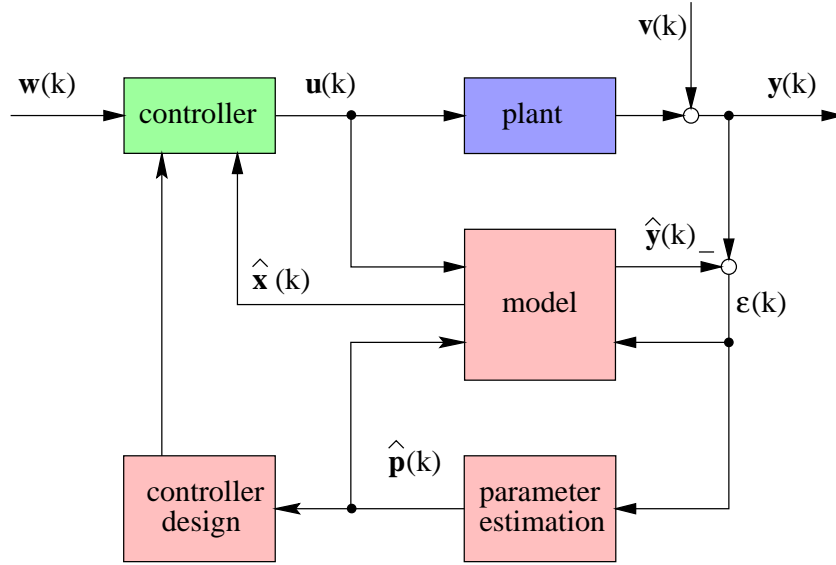


Figure 3.2: Sketch of the recursive predictive error method.

where $\mathbf{\Lambda}$ is a weighting matrix. The functional $V(\mathbf{p})$ can be minimised with respect to the parameter vector \mathbf{p} by setting its gradient equal to zero as

$$-\left[\frac{\partial}{\partial \mathbf{p}}V(\mathbf{p})\right]^T = \mathbf{0}. \quad (3.10)$$

Under the assumption that the gradient

$$-\frac{\partial}{\partial \mathbf{p}}H(\mathbf{p}, \boldsymbol{\varepsilon}(k)) = \mathbf{q}^T(\mathbf{p}, \boldsymbol{\varepsilon}(k)) \quad (3.11)$$

can be calculated for any \mathbf{p} exchanging the expectation and the derivative operator, Eqn.(3.10) can be solved in the form

$$\mathcal{E}\{\mathbf{q}(\mathbf{p}, \boldsymbol{\varepsilon}(k))\} = \mathbf{0}. \quad (3.12)$$

The problem involved in the solution of Eqn.(3.12) is that the expectation cannot be evaluated, because the distributions of \mathbf{p} and $\boldsymbol{\varepsilon}$ are not known *a priori*. Of course, there would be a solution by replacing the expectation with a mean value from an experimental sequence such that for a given \mathbf{p} , the function $\mathbf{q}(\mathbf{p}, \boldsymbol{\varepsilon}(k))$ is evaluated for a series of $\boldsymbol{\varepsilon}(k)$. If this procedure is carried out for different \mathbf{p} , a solution can be found numerically, but off-line.

A pragmatic solution is the recursive approximation [Robbins and Monroe, 1951] with

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) - \gamma(k)\mathbf{q}(\hat{\mathbf{p}}, \boldsymbol{\varepsilon}(k)), \quad (3.13)$$

where $\gamma(k)$ is a sequence of positive scalars. It has been proved that the sequence $\hat{\mathbf{p}}(k)$ converges to a solution, if certain assumptions are met as, for example, $\boldsymbol{\varepsilon}(k)$ has to be a sequence of independent random vectors [Robbins and Monroe, 1951].

A deterministic approach to a solution is the method of the steepest descent given by

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) - \gamma(k) \left[\frac{\partial}{\partial \mathbf{p}} V(\mathbf{p}) \right]^T \Bigg|_{\mathbf{p}=\hat{\mathbf{p}}(k-1)}, \quad (3.14)$$

where $\gamma(k)$ is a positive scalar, too. As it is well known, this algorithm does not converge very well to the minimum, because the first derivative becomes very small in the vicinity of the solution. Therefore, it is extended to the so-called Newton method [Ljung and Söderström, 1983], where the gradient is modified such that the algorithm results in

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) - \left[\frac{\partial^2}{\partial \mathbf{p}^2} V(\mathbf{p}) \right]^{-1} \cdot \left[\frac{\partial}{\partial \mathbf{p}} V(\mathbf{p}) \right]^T \Bigg|_{\mathbf{p}=\hat{\mathbf{p}}(k-1)}, \quad (3.15)$$

replacing the weighted first derivative by the product of the inverse second derivative and the first derivative of the criterion functional. This gives a better approximation in the vicinity of the solution, especially if the criterion is quadratic. The same solution can be obtained, if the criterion functional $V(\mathbf{p})$ is approximated by a second order Taylor series, and differentiated with respect to \mathbf{p} afterwards in order to solve Eqn.(3.10) for the estimated vector $\hat{\mathbf{p}}$.

The stochastic Newton method

Merging the Newton method and the scheme for the recursive calculation of the expectation in Eqn.(3.13), Eqn.(3.15) can be rewritten in the form

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) + \gamma(k) [\bar{V}''(\hat{\mathbf{p}}(k-1), \boldsymbol{\varepsilon}(k))]^{-1} \cdot \mathbf{q}(\hat{\mathbf{p}}(k-1), \boldsymbol{\varepsilon}(k)) \quad (3.16)$$

where $\bar{V}''(\hat{\mathbf{p}}(k-1), \boldsymbol{\varepsilon}(k))$ is the approximation of the second derivative of $V''(\mathbf{p})$ with respect to the parameter vector \mathbf{p} based upon all observations up to k . The vector for the given criterion can be written for k as

$$\mathbf{q}(\hat{\mathbf{p}}, \boldsymbol{\varepsilon}(k)) = - \left[\frac{\partial}{\partial \mathbf{p}} H(\mathbf{p}, \boldsymbol{\varepsilon}(k)) \right]^T \Bigg|_{\mathbf{p}=\hat{\mathbf{p}}(k)} = -\boldsymbol{\Psi}(k, \hat{\mathbf{p}}) \boldsymbol{\Lambda}^{-1} \boldsymbol{\varepsilon}(k). \quad (3.17)$$

with the gradient matrix $\boldsymbol{\Psi}$ as the partial derivative of the estimation error as

$$\boldsymbol{\Psi}(k, \hat{\mathbf{p}}) = - \left(\frac{\partial}{\partial \mathbf{p}} \boldsymbol{\varepsilon}(k) \right)^T \Bigg|_{\mathbf{p}=\hat{\mathbf{p}}(k)} = \left(\frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{y}}(k, \mathbf{p}) \right)^T \Bigg|_{\mathbf{p}=\hat{\mathbf{p}}(k)}. \quad (3.18)$$

The Newton search direction

The matrix $\bar{V}''(\hat{\mathbf{p}}(k-1), \boldsymbol{\varepsilon}(k))$ in Eqn.(3.16) has to be computed for each sample time interval and has to be positive definite to guarantee a stable convergence. Therefore,

the second derivative $V''(\mathbf{p})$ with respect to the parameter vector \mathbf{p} is approximated by

$$\begin{aligned} \frac{\partial^2}{\partial \mathbf{p}^2} V(\mathbf{p}) &= \mathcal{E} \{ \Psi(k, \mathbf{p}) \Lambda^{-1} \Psi^T(k, \mathbf{p}) \} \\ &+ \mathcal{E} \left\{ \left[\frac{\partial^2}{\partial \mathbf{p}^2} \boldsymbol{\varepsilon}^T(k, \mathbf{p}) \right] \Lambda^{-1} \boldsymbol{\varepsilon}(k, \mathbf{p}) \right\} \\ &\approx \mathcal{E} \{ \Psi(k, \mathbf{p}) \Lambda^{-1} \Psi^T(k, \mathbf{p}) \} \end{aligned} \quad (3.19)$$

with the assumption that the last term in Eqn.(3.19) is equal to zero at the true solution. Replacing the expectation in this equation with a weighted sample mean and evaluating this sample mean with the knowledge up to k based upon the estimated parameter vector $\mathbf{p} = \hat{\mathbf{p}}(k-1)$, the approximation for the second derivative of $V(\mathbf{p})$ with respect to the parameter vector \mathbf{p} is according to [Ljung and Söderström, 1983]

$$\mathbf{R}(k) = \frac{1}{k} \sum_{h=1}^k \gamma_1(h, k) \Psi(h, \hat{\mathbf{p}}) \Lambda^{-1} \Psi^T(h, \hat{\mathbf{p}}) + \gamma_2(k) \mathbf{R}_0, \quad (3.20)$$

with an initial matrix \mathbf{R}_0 and $\gamma_1(h, k)$ and $\gamma_2(k)$ being weighting sequences, respectively. If these weighting coefficients are chosen properly, the matrix $\mathbf{R}(k)$ can be computed recursively such that

$$\mathbf{R}(k) = \mathbf{R}(k-1) + \gamma(k) [\Psi(k) \Lambda^{-1} \Psi^T(k) - \mathbf{R}(k-1)], \quad (3.21)$$

Using this algorithm for the estimation of $\bar{V}''(\hat{\mathbf{p}}(k-1), \boldsymbol{\varepsilon}(k))$, the recursive parameter estimation algorithm can be carried out in the form

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) + \gamma(k) \mathbf{R}^{-1}(k) \Psi(k, \hat{\mathbf{p}}) \Lambda^{-1} \boldsymbol{\varepsilon}(k, \hat{\mathbf{p}}(k-1)). \quad (3.22)$$

The weighting sequence $\gamma(k)$ finally turns out to be a function depending on the so-called forgetting factor $\rho(k)$. This factor and the weighting sequence $\gamma(k)$ determine the speed of convergence.

Simplification

The matrix Λ is chosen as the identity matrix \mathbf{I} , i.e. it is assumed to be constant and therefore only scales the algorithm. Since the inverse matrix $\mathbf{R}^{-1}(k)$ has to be calculated for the parameter update, the matrix

$$\mathbf{P}(k) = \gamma(k) \mathbf{R}^{-1}(k) \quad (3.23)$$

is introduced. Using Eqn.(3.21) the previous equation can then be rewritten as

$$\mathbf{P}(k) = \gamma(k) \left[\mathbf{R}(k-1) + \gamma(k) [\Psi(k) \Lambda^{-1} \Psi^T(k) - \mathbf{R}(k-1)] \right]^{-1}. \quad (3.24)$$

Applying Eqn.(3.23) for $k - 1$ and using the matrix inversion Lemma for the sum of matrices [Ljung and Söderström, 1983], the recursive calculation of the parameter vector and the matrix \mathbf{P} yields

$$\mathbf{P}(k) = \frac{1}{\rho(k)} [\mathbf{P}(k-1) - \mathbf{L}(k) \boldsymbol{\Psi}^T(k) \mathbf{P}(k-1)], \quad (3.25)$$

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) + \mathbf{L}(k) \boldsymbol{\varepsilon}(k), \quad (3.26)$$

with the auxiliary matrix

$$\mathbf{L}(k) = \mathbf{P}(k-1) \boldsymbol{\Psi}(k) [\mathbf{I} + \boldsymbol{\Psi}^T(k) \mathbf{P}(k-1) \boldsymbol{\Psi}(k)]^{-1}, \quad (3.27)$$

and the so-called forgetting factor

$$\rho(k) = \frac{\gamma(k-1)}{\gamma(k)} [1 - \gamma(k)]. \quad (3.28)$$

The calculation of the factor $\rho(k)$ is treated Subsection 3.1.5 in detail. The matrix \mathbf{P} is later on referred to as covariance matrix, because it corresponds to the covariance of the estimated parameters. The choice of the starting value for this matrix is considered in Subsection 3.1.6.

Derivation of the gradient matrix

The only expression which is left to be determined is the gradient matrix $\boldsymbol{\Psi}$ (see also [Nazaruddin, 1994]). Utilising the definition for $\hat{\mathbf{y}}(k, \mathbf{p})$ the gradient matrix can be obtained for a parameter vector \mathbf{p} as

$$\boldsymbol{\Psi}^T(k, \mathbf{p}) = \frac{\partial}{\partial \mathbf{p}} [\mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p})]. \quad (3.29)$$

In order to calculate the gradient matrix, the matrices

$$\mathbf{W}(k, \mathbf{p}) = \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{x}}(k, \mathbf{p}), \quad (3.30)$$

$$\mathbf{V}_k(\mathbf{p}, \hat{\mathbf{x}}) = \frac{\partial}{\partial \mathbf{p}} [\mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}] \quad (3.31)$$

are introduced. Then the partial derivative using the product derivative theorem yields

$$\begin{aligned} \boldsymbol{\Psi}^T(k, \mathbf{p}) &= \frac{\partial}{\partial \mathbf{p}} [\mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p})]. \\ &= \mathbf{C}(\mathbf{p}) \frac{\partial}{\partial \mathbf{p}} [\hat{\mathbf{x}}(k, \mathbf{p})] + \frac{\partial}{\partial \mathbf{p}} [\mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}], \\ &= \mathbf{C}(\mathbf{p}) \mathbf{W}(k, \mathbf{p}) + \mathbf{V}_k(\mathbf{p}, \hat{\mathbf{x}}(k, \mathbf{p})). \end{aligned} \quad (3.32)$$

Applying Eqn.(3.30) to $k + 1$ and using the innovations model output for $\hat{\mathbf{x}}(k + 1, \mathbf{p})$ from Eqn.(3.3) the auxiliary matrix results in

$$\begin{aligned}\mathbf{W}(k + 1, \mathbf{p}) &= \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{x}}(k + 1, \mathbf{p}) \\ &= \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{A}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}) + \mathbf{B} \mathbf{u}(k) + \mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon}(k) \right].\end{aligned}\quad (3.33)$$

The previous equation can be expressed as

$$\begin{aligned}\mathbf{W}(k + 1, \mathbf{p}) &= \mathbf{A}(\mathbf{p}) \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{x}}(k, \mathbf{p}) + \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{A}(\mathbf{p}) \hat{\mathbf{x}} \right] \\ &\quad + \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{B} \mathbf{u} \right] \\ &\quad + \mathbf{K}(\mathbf{p}) \frac{\partial}{\partial \mathbf{p}} \boldsymbol{\varepsilon}(k, \mathbf{p}) + \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon} \right].\end{aligned}\quad (3.34)$$

Using the definition for the gradient matrix $\boldsymbol{\Psi}$ in Eqn.(3.18) the fourth term of the previous equation can be written as

$$\begin{aligned}\mathbf{K}(\mathbf{p}) \frac{\partial}{\partial \mathbf{p}} \boldsymbol{\varepsilon}(k, \mathbf{p}) &= \mathbf{K}(\mathbf{p}) [-\boldsymbol{\Psi}] \\ &= \mathbf{K}(\mathbf{p}) \left[-\mathbf{C}(\mathbf{p}) \mathbf{W}(k, \mathbf{p}) - \mathbf{V}_k(\mathbf{p}, \hat{\mathbf{x}}(k, \mathbf{p})) \right] \\ &= -\mathbf{K}(\mathbf{p}) \mathbf{C}(\mathbf{p}) \mathbf{W}(k, \mathbf{p}) - \mathbf{K}(\mathbf{p}) \mathbf{V}_k(\mathbf{p}, \hat{\mathbf{x}}(k, \mathbf{p})).\end{aligned}\quad (3.35)$$

Inserting this solution into Eqn.(3.34) yields

$$\begin{aligned}\mathbf{W}(k + 1, \mathbf{p}) &= \mathbf{A}(\mathbf{p}) \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{x}}(k, \mathbf{p}) \\ &\quad + \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{A}(\mathbf{p}) \hat{\mathbf{x}} \right] + \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{B} \mathbf{u} \right] + \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon} \right] \\ &\quad - \mathbf{K}(\mathbf{p}) \mathbf{C}(\mathbf{p}) \mathbf{W}(k, \mathbf{p}) - \mathbf{K}(\mathbf{p}) \mathbf{V}_k(\mathbf{p}, \hat{\mathbf{x}}(k, \mathbf{p}))\end{aligned}\quad (3.36)$$

and finally with some simplifications and the definition for $\mathbf{W}(k, \mathbf{p})$, the result is

$$\begin{aligned}\mathbf{W}(k + 1, \mathbf{p}) &= \left[\mathbf{A}(\mathbf{p}(k)) - \mathbf{K}(\mathbf{p}(k)) \mathbf{C}(\mathbf{p}(k)) \right] \mathbf{W}(k, \mathbf{p}) \\ &\quad + \mathbf{M}_k - \mathbf{K}(\mathbf{p}(k)) \mathbf{V}_k.\end{aligned}\quad (3.37)$$

with the matrix

$$\mathbf{M}_k(\hat{\mathbf{p}}, \mathbf{x}, \mathbf{u}, \boldsymbol{\varepsilon}) = \frac{\partial}{\partial \mathbf{p}} \left[\mathbf{A}(\mathbf{p}) \mathbf{x} + \mathbf{B} \mathbf{u} + \mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon} \right].\quad (3.38)$$

Finally, the gradient matrix is calculated according to Eqn.(3.32) for $k + 1$ as

$$\boldsymbol{\Psi}(k + 1) = \mathbf{W}^T(k + 1, \mathbf{p}) \mathbf{C}^T(\mathbf{p}(k)) + \mathbf{V}_k^T(\mathbf{p}(k), \hat{\mathbf{x}}(k + 1)).\quad (3.39)$$

The matrices $\mathbf{M}_k = \mathbf{M}_k(\mathbf{p}, \hat{\mathbf{x}}(k), \boldsymbol{\varepsilon}(k))$ and $\mathbf{V}_k = \mathbf{V}_k(\mathbf{p}, \hat{\mathbf{x}}(k))$ are model specific and have to be calculated only once in terms of their structure. A derivation for the magnetic bearing system is given in Appendix B.

3.1.4 Summary of the recursive algorithm

The implemented algorithm can be summarised by the following equations with the estimated parameter vector $\hat{\mathbf{p}}$. These equations have to be evaluated at each sample time interval.

1. Compute the estimation error $\boldsymbol{\varepsilon}$ from the measured output and the previous estimate such that

$$\boldsymbol{\varepsilon}(k) = \mathbf{y} - \hat{\mathbf{y}}(k|\hat{\mathbf{p}}(k-1)). \quad (3.40)$$

2. Compute the auxiliary matrix

$$\mathbf{L}(k) = \mathbf{P}(k-1) \boldsymbol{\Psi}(k) \left(\mathbf{I} + \boldsymbol{\Psi}^T(k) \mathbf{P}(k-1) \boldsymbol{\Psi}(k) \right)^{-1}. \quad (3.41)$$

3. Then compute the new parameter estimate $\hat{\mathbf{p}}$ such that

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) + \mathbf{L}(k) \boldsymbol{\varepsilon}(k), \quad (3.42)$$

4. and update the covariance matrix

$$\mathbf{P}(k) = \frac{1}{\rho(k)} \left(\mathbf{P}(k-1) - \mathbf{L}(k) \boldsymbol{\Psi}^T(k) \mathbf{P}(k-1) \right). \quad (3.43)$$

5. For the use with the state space controller compute the estimated states

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}(\hat{\mathbf{p}}(k)) \hat{\mathbf{x}}(k) + \mathbf{B} \mathbf{u}(k) + \mathbf{K}(\hat{\mathbf{p}}(k)) \boldsymbol{\varepsilon}(k), \quad (3.44)$$

6. and the estimated outputs $\hat{\mathbf{y}}$ for the next cycle

$$\hat{\mathbf{y}}(k+1) = \mathbf{C}(\hat{\mathbf{p}}(k)) \hat{\mathbf{x}}(k+1). \quad (3.45)$$

7. The computation of the auxiliary matrix

$$\begin{aligned} \mathbf{W}(k+1, \hat{\mathbf{p}}(k)) &= \left(\mathbf{A}(\hat{\mathbf{p}}(k)) - \mathbf{K}(\hat{\mathbf{p}}(k)) \mathbf{C}(\hat{\mathbf{p}}(k)) \right) \mathbf{W}(k, \hat{\mathbf{p}}(k-1)) \\ &\quad + \mathbf{M}_k - \mathbf{K}(\hat{\mathbf{p}}(k)) \mathbf{V}_k, \end{aligned} \quad (3.46)$$

8. yields the gradient matrix

$$\boldsymbol{\Psi}(k+1) = \mathbf{W}^T(k+1, \hat{\mathbf{p}}) \mathbf{C}^T(\hat{\mathbf{p}}(k)) + \mathbf{V}_k^T(k+1). \quad (3.47)$$

The initialisation of this recursive algorithm is discussed in Section 3.1.7 and in Section 4.5.1 with respect to its numerical implementation.

3.1.5 Forgetting factor

Special attention has to be paid to the so-called forgetting factor $\rho(k)$ in Eqn.(3.43). This factor determines the rate of change for the covariance matrix and can be compared to a gain for an ordinary control loop. If this factor is too small ($\rho(k) \ll 1$), the covariance matrix increases too fast, if it is too large (close to 1), the algorithm reacts too slowly to parameter changes. Therefore, the forgetting factor needs to be controlled separately by a statistical value, which gives the change of the variance of the estimation error. Similarly to [Basseville and Beneviste, 1984] and [Basseville, 1983] the parameter

$$\delta(k) = \frac{\hat{\sigma}_\varepsilon^2(N_1, k) - \hat{\sigma}_\varepsilon^2(N_2, k)}{\hat{\sigma}_\varepsilon^2(N_2, k)}, \quad (3.48)$$

is introduced with $N_1 \leq N_2$ and $\hat{\sigma}_\varepsilon^2(N_i, k)$ as estimate for the prediction error variance with a variable memory N_i . The estimates for the variances are calculated recursively as

$$\hat{\sigma}_\varepsilon^2(N_i, k) = \frac{N_i - 2}{N_i - 1} \hat{\sigma}_\varepsilon^2(N_i, k - 1) + \frac{1}{n_o N_i} \boldsymbol{\varepsilon}^T(k) \boldsymbol{\varepsilon}(k) \quad (3.49)$$

with n_o being the number of outputs. The estimated value for $\delta(k)$ is filtered recursively by

$$\hat{\delta}(k) = k_\delta \hat{\delta}(k - 1) + (1 - k_\delta) \delta(k) \quad (3.50)$$

with k_δ determining the rate of change.

The idea behind this algorithm is, that under the assumption of a stationary process, the variance of the estimation error in a steady state should be stationary as well. This assumption is called the hypothesis H_0 [Nazaruddin, 1994]. The alternate hypothesis H_1 is that parameters of the controlled system have changed which causes higher estimation errors. Statistically speaking, H_0 assumes $\delta(k)$ to have zero mean with a certain variance. Under the assumption of an uniform distribution, this probability density function maps $\delta(k)$ to the likelihood of H_0 . A threshold δ_H decides between both hypotheses by [Basseville and Beneviste, 1984]

$$\hat{\delta}(k) \underset{H_1}{\overset{H_0}{\lesseqgtr}} \delta_H. \quad (3.51)$$

If $\hat{\delta}(k)$ triggers the threshold δ_H , e.g. 20% of the mean variance, then the hypothesis H_1 is more likely than H_0 . As long as hypothesis H_1 is more likely than H_0 , the consequences are:

1. Reset the forgetting factor $\rho(k)$ to a smaller value ρ_0 ,
2. Add a white noise signal of certain deflection to the set point.

The first action causes the increase of the covariance matrix on the one hand, and weight the estimated values in its memory less than the innovation on the other hand.

This makes the algorithm converge to the true parameters faster. With the second action the closed loop system is excited by an uncorrelated signal. Therefore, the signal to noise ratio increases and accelerate the convergence.

If $\hat{\delta}(k)$ falls beyond the threshold δ_H again, $\rho(k)$ follows a law

$$\rho(k) = k_\rho \rho(k-1) + (1 - k_\rho) \rho_\infty, \quad (3.52)$$

with ρ_∞ the final value and k_ρ determining the rate of change. Then, assuming that the new parameters are already found, the covariance decreases to a minimum value.

3.1.6 Parameter covariance

Since one has to guarantee that the covariance matrix in Eqn.(3.43) remains positive definite, a special algorithm for the covariance update has been used, based upon the factorisation of the covariance matrix $\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ with \mathbf{U} as upper triangular matrix with all diagonal elements equal to 1 and \mathbf{D} a diagonal matrix [Ljung and Söderström, 1983]. Since the system under investigation is a MIMO-system and the proposed factorisation algorithm is only applicable to SISO-systems, all prediction errors are considered to be independent observations [Bierman, 1988]. The matrices \mathbf{U} and \mathbf{D} are updated separately by the following algorithm for all outputs $h = 1 \dots n_o$, which yields a column \mathbf{L}_h for the auxiliary matrix \mathbf{L} .

For $h = 1 \dots n_o$ outputs do the following [Bierman, 1988]:

1. Compute the vectors

$$\mathbf{f} = \mathbf{U}^T(k-1) \boldsymbol{\Psi}_h(k), \quad (3.53)$$

$$\mathbf{g} = \mathbf{D}(k-1) \mathbf{f}, \quad (3.54)$$

$$\boldsymbol{\beta}_0 = \rho(k). \quad (3.55)$$

2. For $j = 1 \dots n_p$, with n_p being the number of parameters in the parameter vector \mathbf{p} do

- (a) Compute the vectors

$$\boldsymbol{\beta}_j = \boldsymbol{\beta}_{j-1} + \mathbf{f}_j \mathbf{g}_j, \quad (3.56)$$

$$\mathbf{D}_{jj}(k) = \frac{\boldsymbol{\beta}_{j-1}}{\boldsymbol{\beta}_j \rho(k)} \mathbf{D}_{jj}(k-1), \quad (3.57)$$

$$\boldsymbol{\nu}_j = \mathbf{g}_j, \quad (3.58)$$

$$\boldsymbol{\mu}_j = -\frac{1}{\boldsymbol{\beta}_{j-1}} \mathbf{f}_j. \quad (3.59)$$

- (b) For $i = 1 \dots j-1$ compute

$$\mathbf{U}_{ij}(k) = \mathbf{U}_{ij}(k-1) + \boldsymbol{\nu}_i \boldsymbol{\mu}_j, \quad (3.60)$$

$$\boldsymbol{\nu}_i = \boldsymbol{\nu}_i + \mathbf{U}_{ij}(k-1) \boldsymbol{\nu}_j. \quad (3.61)$$

3. Calculate one column for the auxiliary matrix

$$\mathbf{L}_h = \frac{1}{\beta_{n_p}} [\nu_1, \dots, \nu_{n_p}]^T. \quad (3.62)$$

In the preceding equations Ψ_h symbolises the h^{th} column of the gradient matrix Ψ . \mathbf{f}_j symbolises the j^{th} entry of the column vector \mathbf{f} as example for \mathbf{g} , β , ν and μ . \mathbf{U}_{ij} is the ij^{th} entry of the upper triangular matrix \mathbf{U} , and \mathbf{D}_{jj} the diagonal entry at index jj of the diagonal matrix \mathbf{D} .

Eqn.(3.43) and Eqn.(3.41) can therefore be replaced by the above equations. This algorithm keeps the covariance matrix positive definite. If the covariance becomes small anyhow, the diagonal matrix \mathbf{D} is increased by $c\mathbf{I}$ with $c > 0$.

3.1.7 Analysis of the recursive algorithm

The prediction error method algorithm was derived in the previous section for a discrete time state space model in controller canonical form. In the following a short analysis of that algorithm is given.

Initial conditions

All recursively computed matrices and vectors have to be initialised once. In most off-line applications, these elements are set equal to zero at the beginning, except for the parameter covariance matrix. This matrix should be large, e.g. $\mathbf{P}(0) = 1000\mathbf{I}$, in order to guarantee a large variance of the parameter vector when the algorithm is started. This makes the estimates converge very fast to the true parameters. For the rotor bearing system, however, which is open loop unstable, the choice is more complex and the initial values have to be found by means of trial and error as shown in Section 4.5.1.

Stability of the estimation algorithm

The most interesting problem about the estimation algorithm is its stability. As proved in [Ljung and Söderström, 1983], the stability of the estimation algorithm is given, if

$$\left| \text{eig} \left(\mathbf{A}(\mathbf{p}) - \mathbf{K}(\mathbf{p}) \mathbf{C}(\mathbf{p}) \right) \right| < 1, \quad (3.63)$$

i.e. if the predictor based upon the innovations model defined by Eqn.(3.3) is stable.

Therefore, the eigenvalues of the predictor have to be calculated each time step, when a new estimate has been calculated with a projection algorithm described in the following.

1. Choose a factor $0 \leq \kappa \leq 1$,
2. Compute the differential update of the parameter vector $\hat{\mathbf{p}}$ as

$$\Delta \hat{\mathbf{p}}(k) = \mathbf{L}(k) \boldsymbol{\varepsilon}(k), \quad (3.64)$$

3. As long as the predictor is unstable do with $i = 0, 1, \dots$

$$\hat{\mathbf{p}}(k) = \hat{\mathbf{p}}(k-1) + \kappa^i \Delta \hat{\mathbf{p}}(k), \quad (3.65)$$

until the predictor is stable. Since κ remains as design parameter and the aforementioned algorithm does not guarantee a constant computation time, κ can be chosen equal to zero as well. This means that an observation is not used if it causes the predictor to be unstable and the parameter vector is only updated if the resulting predictor is stable.

Convergence of the estimation algorithm

In Section 3.1.3, the prediction error method was derived as a recursive computation of the system parameter vector based upon a criterion functional of the prediction error. Basically, a negative gradient multiplied by a strictly positive matrix adjusts the parameter vector until a minimum is reached, i.e.: “*The estimate will converge with probability one to a local minimum of $V(\mathbf{p})$ as t approaches infinity*”, [Ljung and Söderström, 1983]. An extensive derivation for the convergence of the algorithm presented can be found in [Ljung and Söderström, 1983].

3.2 Controller design

Once the parameters of the system are estimated, the state space controller can be computed based upon the identified model with the assumption that the system parameters are estimated properly (*certainty equivalence principle*, [Goodwin and Sin, 1984, Isermann et al., 1992]).

In this work a pole placement controller is chosen, because stability has to be guaranteed from the beginning based upon a precalculated model. Another reason for pole placement is that active magnetic bearing systems are viewed as mechanical systems with eigenfrequencies and a certain bandwidth. By means of pole placement specific dynamic characteristics can be applied to the controlled system.

There are two possibilities for a state space controller, simple state feedback with feed-forward gain, or state feedback with additional integrative feedback. Both concepts are presented in the following sections.

3.2.1 State space controller

The control law for a state space controller is generally defined as

$$\mathbf{u}(k) = \mathbf{K}_w \mathbf{w}(k) - \mathbf{K}_x \mathbf{x}(k). \quad (3.66)$$

The controller gain $\mathbf{K}_x = \{\mathbf{k}_x^{T(ij)}\}$ is parameterised with $i, j = 1, 2, 3, 4$ by

$$\mathbf{k}_x^{T(ij)} = \begin{bmatrix} k_{x_2}^{(ij)} & k_{x_1}^{(ij)} \end{bmatrix}, \quad (3.67)$$

and the feed forward gain matrix is parameterised as $\mathbf{K}_w = \{k_w^{(ij)}\}$ with $i, j = 1, 2, 3, 4$. A block diagram of the presented controller is shown in Fig. 3.3.

The poles of the closed loop system are determined by the closed loop system matrix $\mathbf{A}_{cl} = \mathbf{A} - \mathbf{B}\mathbf{K}_x = \{\mathbf{A}_{cl}^{(ij)}\}$ with its sub-matrices

$$\mathbf{A}_{cl}^{(ij)} = \begin{bmatrix} 0 & 1 \\ a_2^{(ij)} - k_{x_2}^{(ij)} & a_1^{(ij)} - k_{x_1}^{(ij)} \end{bmatrix}, \quad (3.68)$$

$$\mathbf{A}_{cl}^{(ii)} = \begin{bmatrix} 0 & 0 \\ a_2^{(ii)} - k_{x_2}^{(ii)} & a_1^{(ii)} - k_{x_1}^{(ii)} \end{bmatrix}, \quad (3.69)$$

and the system matrices \mathbf{A} and \mathbf{B} being in controller canonical form. The coefficients of the control matrix can be easily calculated in the form [Nazaruddin, 1994]

$$k_{x_m}^{(ij)} = a_m^{(ij)}, \quad (3.70)$$

$$k_{x_m}^{(ii)} = a_m^{(ii)} + p_m^{(ii)}, \quad (3.71)$$

with $m = 1, 2$ and $i, j = 1, 2, 3, 4$. The coefficients $p_m^{(ii)}$ are derived from the desired polynomial for each subsystem of order $\nu_i = 2$ by

$$P^{(ii)}(z) = z^2 + p_1^{(ii)}z + p_2^{(ii)}. \quad (3.72)$$

Here, an advantage of a model description in controller canonical form becomes obvious. All parameters of the state space controller can be calculated directly from the state space model using simple algebra.

The output of the controlled system is

$$\mathbf{Y}(z) = \mathbf{G}_{cl}(z) \mathbf{W}(z), \quad (3.73)$$

with the reference transfer function matrix

$$\mathbf{G}_{cl}(z) = \mathbf{C} (\mathbf{I}z - \mathbf{A}_{cl})^{-1} \mathbf{B} \mathbf{K}_w. \quad (3.74)$$

This transfer function matrix contains the closed loop system matrix $\mathbf{A}_{cl} = \{\mathbf{A}_{cl}^{(ij)}\}$ with $i, j = 1, 2, 3, 4$ and its diagonal sub-matrices using Eqn.(3.70) to Eqn.(3.71)

$$\mathbf{A}_{cl}^{(ii)} = \begin{bmatrix} 0 & 1 \\ -p_2^{(ii)} & -p_1^{(ii)} \end{bmatrix}. \quad (3.75)$$

The feed forward matrix \mathbf{K}_w can then be calculated from the steady state condition $\mathbf{G}_{cl}(z)|_{z=1} = \mathbf{I}$ by

$$\mathbf{K}_w = \left(\mathbf{C} (\mathbf{I}z - \mathbf{A}_{cl})^{-1} \mathbf{B} \right)^{-1} \Big|_{z=1} \quad (3.76)$$

The inverse of $(\mathbf{I} - \mathbf{A}_{cl})$ can be calculated in advance. Only Eqn.(3.76) has to be evaluated at every sample time interval. If the set point is equal to zero, the feed forward gain can be omitted at all.

3.2.2 State space controller with integrative feedback

The control law for a state space controller with integrative feedback is defined as

$$\mathbf{u}(k) = \mathbf{u}_I(k) - \mathbf{K}_x \mathbf{x}(k), \quad (3.77)$$

$$\mathbf{u}_I(k+1) = \mathbf{u}_I(k) + \mathbf{K}_I \mathbf{e}(k), \quad (3.78)$$

$$\mathbf{e}(k) = \mathbf{w}(k) - \mathbf{y}(k). \quad (3.79)$$

with the integrative feedback gain matrix $\mathbf{K}_I = \{k_I^{(ij)}\}$ and $i, j = 1, 2, 3, 4$.

Usually, the set point error is integrated first, and the resulting integral is weighted by an integrative gain [Jörgl, 1994]. In the presented control concept, an additional control variable is integrated with a the weighted set point error as input. The implementation of the integrator using the backwards difference method [Ogata, 1987] in Eqn.(3.78) is used for the sake simplicity. A block diagram of the presented controller is shown in Fig. 3.4.

Using the transformation into the z -domain, Eqn.(3.78) yields

$$\mathbf{U}_I(z) = \frac{1}{z-1} \mathbf{K}_I \mathbf{E}(z). \quad (3.80)$$

With the assumption that all system parameters and all states are estimated properly, the deterministic state space model used for controller design is

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k), \quad (3.81)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{x}(k). \quad (3.82)$$

Applying the z -transformation for Eqn.(3.77), Eqn.(3.81) and Eqn.(3.82), the transformed output of the controlled system with simple state feedback results in

$$\mathbf{Y}(z) = \mathbf{C} \left(z\mathbf{I} - \mathbf{A} + \mathbf{B} \mathbf{K}_x \right)^{-1} \mathbf{B} \mathbf{U}_I(z). \quad (3.83)$$

Inserting the z -transformed Eqn.(3.79) and Eqn.(3.80), the output remains as function depending on $\mathbf{W}(z)$ and $\mathbf{Y}(z)$ such that

$$\mathbf{Y}(z) = \mathbf{N} \mathbf{D}^{-1} \frac{1}{z-1} \mathbf{K}_I (\mathbf{W}(z) - \mathbf{Y}(z)), \quad (3.84)$$

with the convention making use of the controller canonical form [Keuchel, 1988]

$$\mathbf{N} \mathbf{D}^{-1} = \mathbf{C} \left(z\mathbf{I} - \mathbf{A} + \mathbf{B} \mathbf{K}_x \right)^{-1} \mathbf{B}. \quad (3.85)$$

A derivation of the above convention is given in Appendix C.1.

Matrix \mathbf{D} is a diagonal matrix and can be interpreted as the characteristic polynomial matrix of the closed loop system without integrative feedback

$$\mathbf{D} = \text{diag } P^{(ii)}(z) \quad \text{with } i = 1, 2, 3, 4, \quad (3.86)$$

with the same polynomials as defined in Eqn.(3.72). The matrix \mathbf{N} depends on the measurement matrix \mathbf{C} such that $\mathbf{N} = \{\mathbf{N}^{(ij)}\}$ with

$$\mathbf{N}^{(ij)} = \begin{bmatrix} c_2^{(ij)} & c_1^{(ij)} z \end{bmatrix}. \quad (3.87)$$

Solving Eqn.(3.84) for $\mathbf{Y}(z)$ yields

$$\mathbf{Y}(z) = \mathbf{G}_{cl_I}(z)\mathbf{W}(z) \quad (3.88)$$

with the reference transfer function matrix

$$\mathbf{G}_{cl_I}(z) = \mathbf{N}\mathbf{D}_I^{-1}\mathbf{K}_I \quad (3.89)$$

and with the closed loop polynomial matrix

$$\mathbf{D}_I = (z - 1)\mathbf{D} + \mathbf{K}_I\mathbf{N}. \quad (3.90)$$

A derivation for the preceding equation is given in Appendix C.2.

The essence of the controller design is that the matrix \mathbf{D}_I is set equal to a desired polynomial matrix \mathbf{P}_I , which is a diagonal matrix with the elements $P_I^{(ii)}(z)$ of order $\nu_i + 1$ such as

$$\mathbf{P}_I = \text{diag } P_I^{(ii)}(z) \quad \text{with } i = 1, 2, 3, 4 \quad (3.91)$$

with the desired polynomials for the closed loop system with integrative feedback

$$P_I^{(ii)}(z) = z^3 + p_{I_1}^{(ii)}z^2 + p_{I_2}^{(ii)}z + p_{I_3}^{(ii)}. \quad (3.92)$$

This matrix has to be chosen in advance. The gain for the integrative feedback path \mathbf{K}_I can be determined from Eqn.(3.89) for the stationary case with the desired closed loop polynomial matrix \mathbf{P}_I from the steady state condition $\mathbf{G}_{cl_I}(z)|_{z=1} = \mathbf{I}$ by

$$\mathbf{K}_I = \mathbf{P}_I(z)\mathbf{N}^{-1}(z)\Big|_{z=1}. \quad (3.93)$$

What remains is the computation of the polynomials in \mathbf{D} , or the desired polynomials $P^{(ii)}(z)$ of the closed loop system without integrative feedback, i.e. heuristically speaking, how to chose the polynomials $P^{(ii)}(z)$ in order to achieve the closed loop polynomials $P_I^{(ii)}(z)$ based upon the integrative feed back gain \mathbf{K}_I . Eqn.(3.90) contains the necessary conditions.

Without claim on generality, the following solution derived in Appendix C.3 is proposed

$$p_1^{ii} = p_{I_1}^{ii} + 1, \quad (3.94)$$

$$p_2^{ii} = \sum_{j=1}^4 c_2^{ji} k_w^{(ij)} - p_{I_3}^{ii}. \quad (3.95)$$

With the thus computed polynomials for the state space controller without integrative feedback, the matrix \mathbf{K}_x can be determined according to Eqn.(3.70) and Eqn.(3.71).

3.3 Closed loop eigenvalues

Using the *certainty equivalence principle* the estimated parameters are assumed to be true [Goodwin and Sin, 1984, Isermann et al., 1992]. Therefore, the estimation algorithm, which is the *meta-controller* with the slow states, is omitted in this subsection. Only the fast states are objective for the stability analysis for each simple state feedback and state space control with additional integrative feedback.

3.3.1 P-structure

The recursive equations for the controlled system, the predictor and the controller are given by the following equations

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k), \quad (3.96)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{x}(k) + \mathbf{v}(k), \quad (3.97)$$

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{A}} \hat{\mathbf{x}}(k) + \mathbf{B} \mathbf{u}(k) + \hat{\mathbf{K}} \boldsymbol{\varepsilon}(k), \quad (3.98)$$

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{C}} \hat{\mathbf{x}}(k), \quad (3.99)$$

$$\mathbf{u}(k) = \mathbf{K}_w \mathbf{w}(k) - \mathbf{K}_x \hat{\mathbf{x}}(k), \quad (3.100)$$

$$\boldsymbol{\varepsilon}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k). \quad (3.101)$$

Fig. 3.3 shows a block diagram of the entire linear control system.

The above equations can be rewritten in the form

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}} \bar{\mathbf{x}}(k) + \bar{\mathbf{B}} \bar{\mathbf{u}}(k), \quad (3.102)$$

with the new system states and inputs

$$\bar{\mathbf{x}}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{u}}(k) = \begin{bmatrix} \mathbf{w}(k) \\ \mathbf{v}(k) \end{bmatrix}, \quad (3.103)$$

and the system matrices

$$\bar{\mathbf{A}} = \left[\begin{array}{c|c} \mathbf{A} & -\mathbf{B} \mathbf{K}_x \\ \hline \hat{\mathbf{K}} \mathbf{C} & \hat{\mathbf{A}} - \mathbf{B} \mathbf{K}_x - \hat{\mathbf{K}} \hat{\mathbf{C}} \end{array} \right], \quad (3.104)$$

$$\bar{\mathbf{B}} = \left[\begin{array}{c|c} \mathbf{B} \mathbf{K}_w & \mathbf{0} \\ \hline \mathbf{B} \mathbf{K}_w & \hat{\mathbf{K}} \end{array} \right]. \quad (3.105)$$

Utilising the equivalence of the “hatted” matrices and the true ones, ($\hat{\mathbf{A}} = \mathbf{A}$, $\hat{\mathbf{C}} = \mathbf{C}$) the separation principle [Jörigl, 1994] holds and

$$\det(z\mathbf{I} - \bar{\mathbf{A}}) = \underbrace{\det(z\mathbf{I} - \mathbf{A} + \mathbf{B} \mathbf{K}_x)}_{\text{controller}} \cdot \underbrace{\det(z\mathbf{I} - \mathbf{A} + \mathbf{K} \mathbf{C})}_{\text{predictor}}. \quad (3.106)$$

The eigenvalues then are the ones from the predictor which are inside the unit circle, and the eigenvalues determined by the control design.

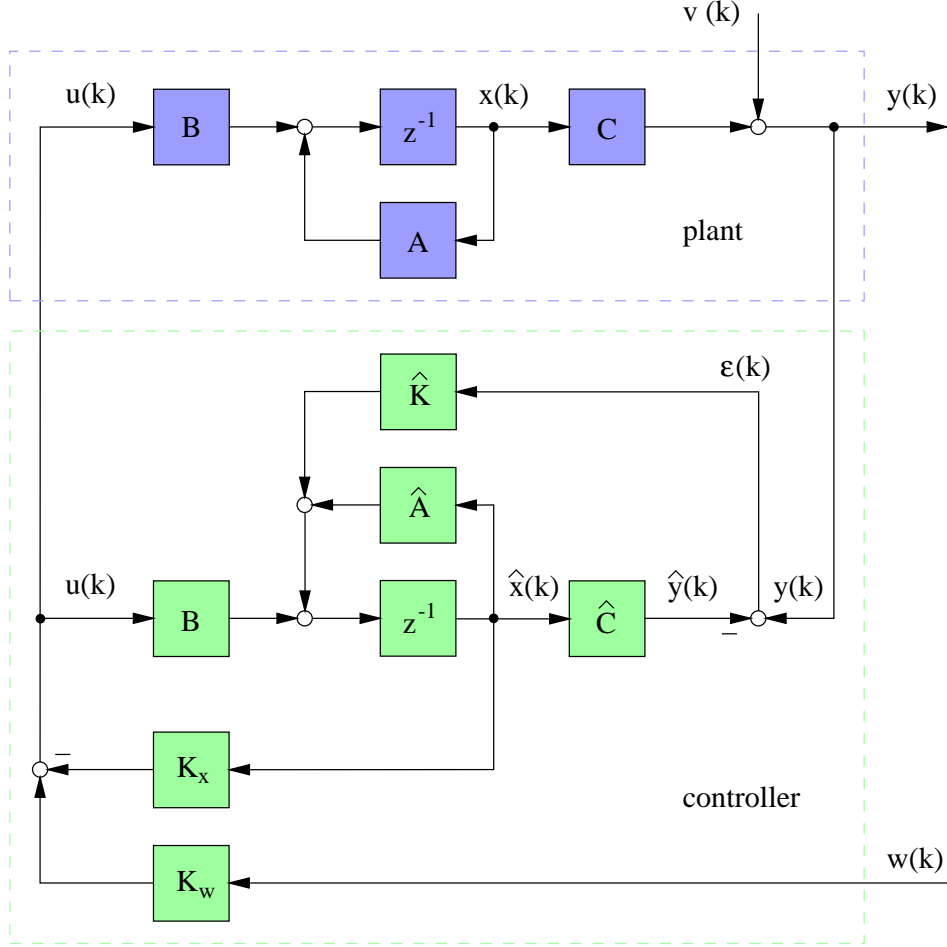


Figure 3.3: The block diagram of the system controlled by a conventional state space controller.

3.3.2 PI-structure

The recursive equations for the controlled system, the predictor and the controller with integrative feedback are given by the following equations as

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k), \quad (3.107)$$

$$\mathbf{y}(k) = \mathbf{C} \mathbf{x}(k) + \mathbf{v}(k), \quad (3.108)$$

$$\hat{\mathbf{x}}(k+1) = \hat{\mathbf{A}} \hat{\mathbf{x}}(k) + \mathbf{B} \mathbf{u}(k) + \hat{\mathbf{K}} \boldsymbol{\varepsilon}(k), \quad (3.109)$$

$$\hat{\mathbf{y}}(k) = \hat{\mathbf{C}} \hat{\mathbf{x}}(k), \quad (3.110)$$

$$\mathbf{u}(k) = \mathbf{u}_I(k) - \mathbf{K}_x \hat{\mathbf{x}}(k), \quad (3.111)$$

$$\mathbf{u}_I(k+1) = \mathbf{u}_I(k) + \mathbf{K}_I \mathbf{e}(k), \quad (3.112)$$

$$\mathbf{e}(k) = \mathbf{w}(k) - \mathbf{y}(k), \quad (3.113)$$

$$\boldsymbol{\varepsilon}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k). \quad (3.114)$$

Fig. 3.4 shows a block diagram of the entire linear control system with integrative feedback.

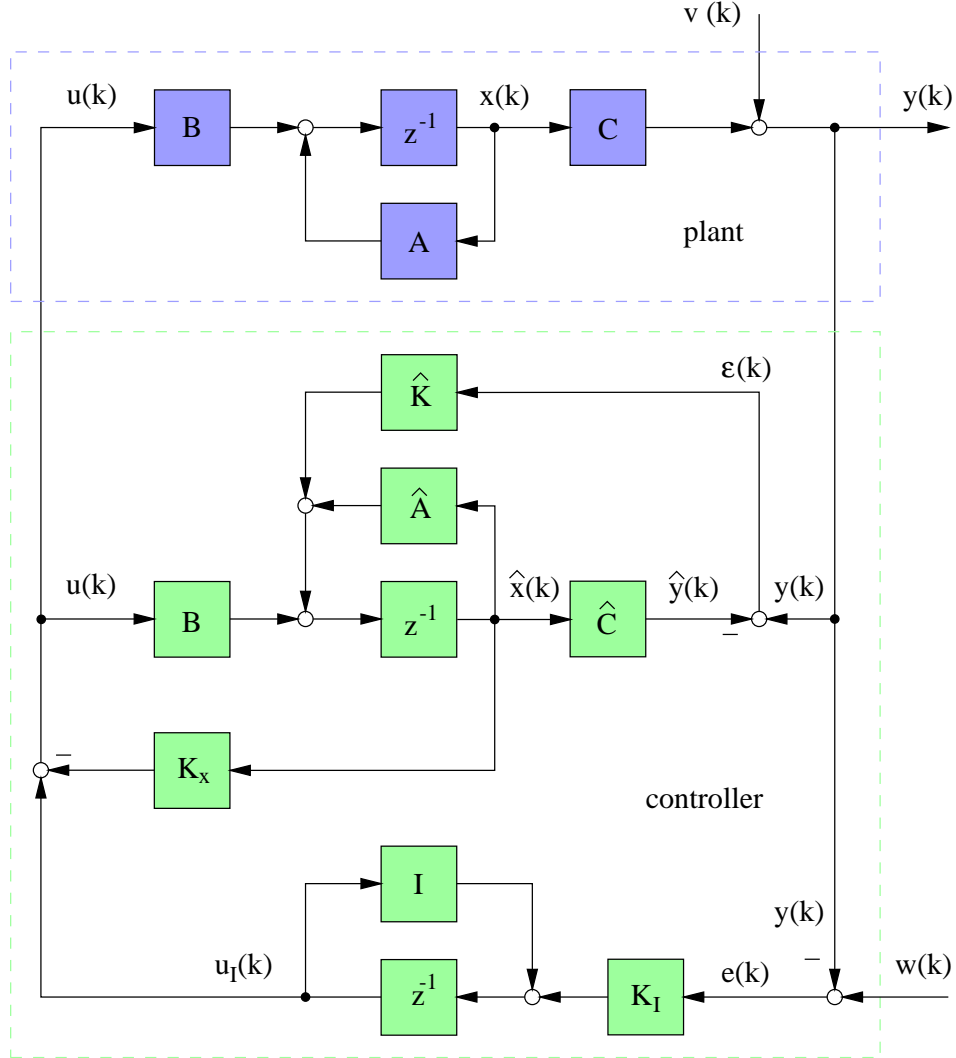


Figure 3.4: The block diagram of the system controlled by a state space controller with additional integrative feedback.

The above equations can be rewritten in the form

$$\bar{\mathbf{x}}(k+1) = \bar{\mathbf{A}} \bar{\mathbf{x}}(k) + \bar{\mathbf{B}} \bar{\mathbf{u}}(k), \quad (3.115)$$

with the new system states and inputs

$$\bar{\mathbf{x}}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}_I(k) \\ \hat{\mathbf{x}}(k) \end{bmatrix} \quad \text{and} \quad \bar{\mathbf{u}}(k) = \begin{bmatrix} \mathbf{w}(k) \\ \hat{\mathbf{v}}(k) \end{bmatrix}, \quad (3.116)$$

and the system matrices

$$\bar{\mathbf{A}} = \left[\begin{array}{c|c|c} \mathbf{A} & \mathbf{B} & -\mathbf{B}\mathbf{K}_x \\ \hline -\mathbf{K}_I\mathbf{C} & \mathbf{I} & \mathbf{0} \\ \hline \hat{\mathbf{K}}\mathbf{C} & \mathbf{B} & \hat{\mathbf{A}} - \mathbf{B}\mathbf{K}_x - \hat{\mathbf{K}}\hat{\mathbf{C}} \end{array} \right], \quad (3.117)$$

$$\bar{\mathbf{B}} = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{K}_I & -\mathbf{K}_I \\ \hline \mathbf{0} & \hat{\mathbf{K}} \end{array} \right]. \quad (3.118)$$

Utilising the equivalence of the “hatted” matrices and the true ones, ($\hat{\mathbf{A}} = \mathbf{A}$, $\hat{\mathbf{C}} = \mathbf{C}$) the separation principle [Jörgl, 1994] holds and

$$\det(z\mathbf{I} - \bar{\mathbf{A}}) = \underbrace{\det(z\mathbf{I} - \tilde{\mathbf{A}})}_{\text{controller}} \cdot \underbrace{\det(z\mathbf{I} - \mathbf{A} + \mathbf{K}\mathbf{C})}_{\text{predictor}}, \quad (3.119)$$

and the extended system matrix with integrative feedback

$$\tilde{\mathbf{A}} = \left[\begin{array}{c|c} \mathbf{A} - \mathbf{B}\mathbf{K}_x & \mathbf{B} \\ \hline -\mathbf{K}_I\mathbf{C} & \mathbf{I} \end{array} \right]. \quad (3.120)$$

The eigenvalues of the entire system are the eigenvalues of the predictor, and the eigenvalues for the state space controlled system with additional integrative feedback.

Chapter 4

Numerical results

This chapter is concerned with the numerical implementation of the control concept presented in the previous chapter, i.e. state space adaptive control. The simulation of the entire control concept is carried out using a nonlinear model as described in detail in Chapter 2 with the simulation parameters given in Appendix D.2. The nonlinear model and the control algorithms are implemented by the numerical simulation tool MATLAB/SIMULINK [MATLAB, 1992, SIMULINK, 1992].

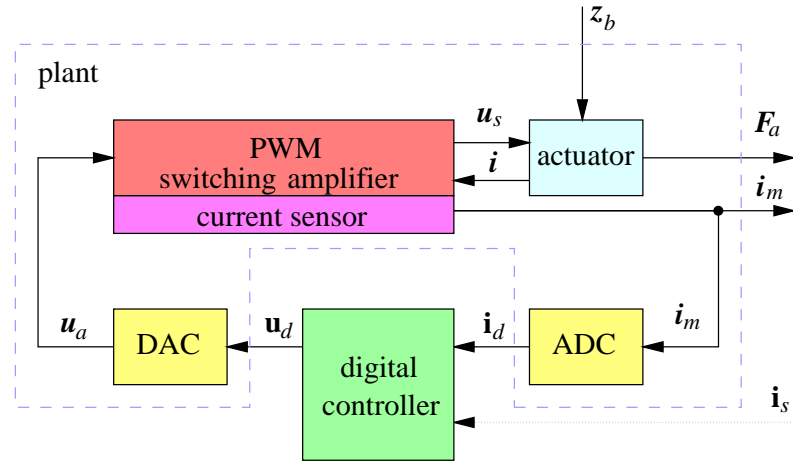
The estimation algorithm, however, was derived for a linear state space model based on a linear actuator model. Therefore, this simplification has to be justified by means of simulation, before the state space adaptive control concept is implemented.

4.1 Current control loop

In order to derive the linear state space model for the rotor bearing system used for the controller design in Chapter 3 the current control loop was introduced in Section 2.3.2. This is a necessary condition in order to treat the magnetic actuator as a gain with respect to the control current and as a spring with a negative coefficient with respect to the rotor displacement. However, this simplification is only justified, if the bandwidth of the current control loop covers the frequency band of the position control loop. Therefore, the bandwidth of the current control loop has to be tested.

In Chapter 2 the entire nonlinear model of a rigid rotor suspended in active magnetic bearings was depicted by Fig. 2.2. For the simulation of the current control loop, only the electric signal path is under investigation as depicted in Fig. 4.1. The nonlinear model was implemented as a MATLAB/SIMULINK model with its block diagrams shown in Appendix D.1. The parameters used for simulation are given in Tab. 2.1 for the digital to analogue converter, in Tab. 2.2 for the pulse width modulator, in Tab. 2.3 for the actuator, and in Tab. 2.7 for the analogue to digital converter.

For the sake of the controller design, a linear model of the current control loop was derived in Section 2.3.2. In the following, the numerical results for the controller design of only one channel are presented.



inputs: \mathbf{i}_s desired current output: \mathbf{F}_a actuator forces
 z_b rotor position \mathbf{i}_m measured current

internal variables:

\mathbf{u}_d digital control voltage \mathbf{i} current
 \mathbf{u}_a analogue control voltage \mathbf{i}_m measured current
 \mathbf{u}_s switched voltage \mathbf{i}_d sampled current

Figure 4.1: Block diagram of the current control loop with the inputs “rotor position” and “current set point”, the output “actuator force” and “measured current”, and all internal variables and subsystems. This loop controls 8 current channels.

4.1.1 Current controller design

In Section 2.3.2 the open loop transfer function was derived for the current control loop as

$$G_o(s) = \frac{K_c K_a \mathcal{R}_0}{N^2 s (1 + \frac{T_s}{2} s)},$$

and with all known parameters inserted

$$G_o(s) = K_c \frac{2.17005 \cdot 10^3}{s (1 + 0.00005 s)}.$$

The controller gain K_c is determined using the frequency plot as shown in Fig. 4.2 with a phase margin of 70 deg which corresponds to an aperiodic transition after a set point change. The high value for the phase margin is chosen, because in an experimental implementation additional time delay or saturation decreases the phase margin anyway. For a large phase margin stability can be guaranteed for a wide range of operation.

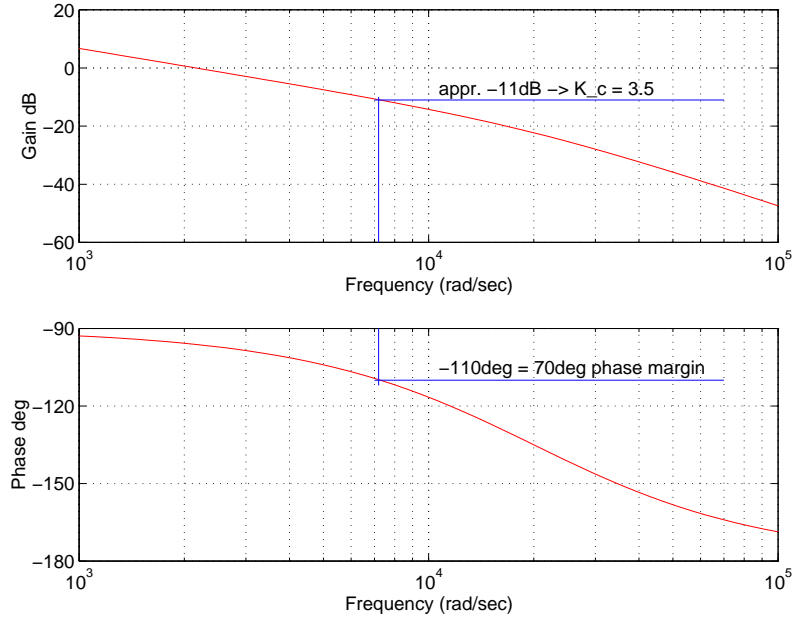


Figure 4.2: Bode plot of the open current loop transfer function with the controller gain $K_c = 3.5$.

The controller gain has been found to be $K_c = 3.5$ for a phase margin of 70 deg. For this second order system the closed loop transfer function can be calculated as

$$G_{ii_s}(s) = \frac{i_m}{i_s} = \frac{1.519 \cdot 10^8}{s^2 + 2 \cdot 10^4 s + 1.519 \cdot 10^8},$$

with the undamped closed loop eigenfrequency $\omega_n = 12,325$ 1/s and a damping factor of $\zeta = 0.81$. For this frequency and the corresponding damping factor the time delay for a unit step response can be calculated as $320 \mu\text{s}$ for a linear system. Using a bode plot for the closed loop system, the -3 dB limit can be found for a frequency of $10,367$ 1/s or a bandwidth of approximately $1,650$ Hz.

The current controller, designed in the analogue time domain, has to be transformed into the discrete time domain for an implementation as a digital controller. Using the Tustin transformation

$$s = \frac{2}{T_s} \frac{1 - z^{-1}}{1 + z^{-1}},$$

the digital controller operating at the sampling time T_s can be presented as

$$G_{c_d}(z) = \frac{b_0 + b_1 z^{-1}}{1 - z^{-1}}.$$

with the coefficients

$$\begin{aligned} b_0 &= \frac{K_c T_s r \mathcal{R}_0}{N^2 2} + K_c, \\ b_1 &= \frac{K_c T_s r \mathcal{R}_0}{N^2 2} - K_c. \end{aligned}$$

4.1.2 Current controller implementation

As already mentioned, the controller is implemented digitally in the DSP for 8 channels, i.e. for 2 bearing actuators in this investigation. Inputs to the control loops are the sampled coil currents \mathbf{i}_d , outputs of all control loops are the control variables \mathbf{u}_d . The controller for one channel is implemented in a recursive form such that

$$u_c(k) = u_c(k-1) + b_0 e(k) + b_1 e(k-1)$$

with the set point error at time k as $e(k) = i_s(k) - i_d(k)$ with $i_d(k)$ being the measured and sampled coil current. The controller output u_c is shifted by 2.5 V to meet the range of the DACs of 0-5 V with $u_d = u_c + 2.5$ V. The parameters of the resulting current controller are presented in Tab. 4.1.

Table 4.1: Parameters of current controller.

T_s	sampling time	100	μs
K_c	Controller gain	3.5	-
b_0	controller coefficient for z^0	3.5101	-
b_1	controller coefficient for z^{-1}	-3.4899	-

4.1.3 Initial values for simulation

For a simulation run, all initial states have to be chosen in advance. The simulation is started from a precalculated point of operation determined by $i_0 = 4$ A, and the rotor position set equal to zero ($\mathbf{z}_b = \mathbf{0}$ m), i.e. an air gap length for each electromagnet of $l_0 = 500$ μm . All further initial variables can be calculated upon these values following the equations in Section 2.3.

The current control loop starts with all internal and state variables like the initial flux Φ_0 and the control voltage u_c initialised properly. The initial control variable u_c can be computed as $u_{c_0} = r i_0$ V, the initial flux as $\Phi_0 = 0.4254 \cdot 10^{-3}$ Vs.

A fact which has to be mentioned is that the voltage output is $u_d = 2.5$ V for a control voltage of $u_c = 0$ V, because the range of the DAC is 0-5 V. Additionally, for the point of operation defined by $i_0 = 4$ A, the mean voltage for the coil is $u_0 = r \cdot i_0 = 3.2$ V, or prior to the amplifier, $u_{c_0} = \frac{u_0}{K_a} = 0.1067$ V. This means that the digital control voltage for the point of operation is $u_{d_0} = u_{c_0} + 2.5 = 2.607$ V.

4.1.4 Simulation results

In order to determine the bandwidth of the current control loop, simulation runs have been carried out. This is possible by both a step response and frequency analysis.

Current step response

Using all previous parameters, a current step response was simulated for one channel only. From the point of operation of 4 A current step responses to 4.25 A, 4.5 A and 5 A were simulated. The corresponding current plots can be seen in Fig. 4.3 and the plots of the control voltage in Fig. 4.4.

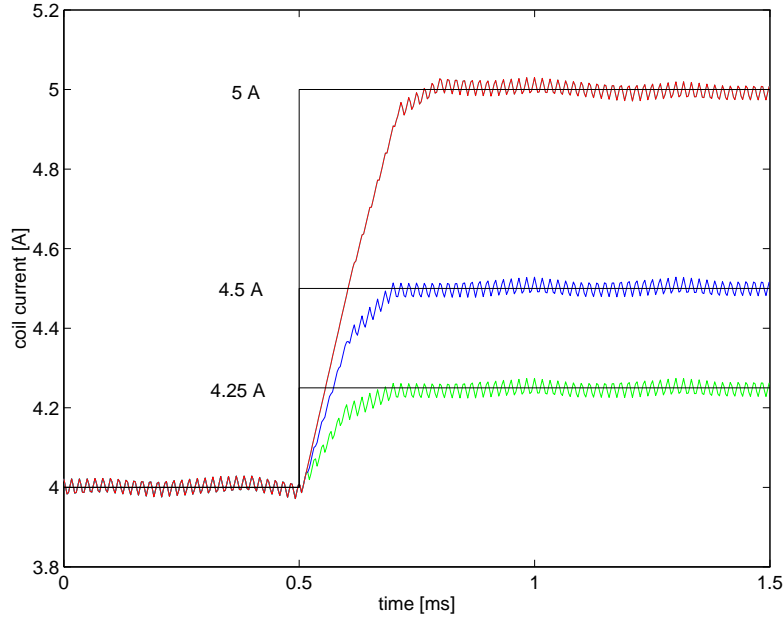


Figure 4.3: Step response of the controlled current after a set point change from 4 A to 4.25 A, 4.5 A, and 5 A.

As it can be seen in Fig. 4.3, it takes approximately three sample intervals, i.e. about $300 \mu\text{s}$ until the set point is reached, which corresponds well to the predicted value. If the step is smaller than approximately $\frac{2.5}{K_c} = 0.71 \text{ A}$, then the system undergoes no saturation and the control loop is linear. If the set point change is higher than 0.71 A , the control voltage runs into saturation and the time consumed to reach the set point is determined by the slope

$$\left. \frac{di}{dt} \right|_{\max} = \frac{\text{switching voltage } U_s}{\text{initial inductivity } L_0} = 5.4 \cdot 10^3 \text{ A/s}$$

such that

$$\Delta T = \left(\left. \frac{di}{dt} \right|_{\max} \right)^{-1} \Delta i \approx 185 \cdot 10^{-6} \cdot \Delta i \text{ s}$$

This means that for a step of 1 A , it takes about $185 \mu\text{s}$ until the set point is reached which corresponds well to the simulation result in Fig. 4.3. The saturation of the control voltage can be seen in Fig. 4.4.

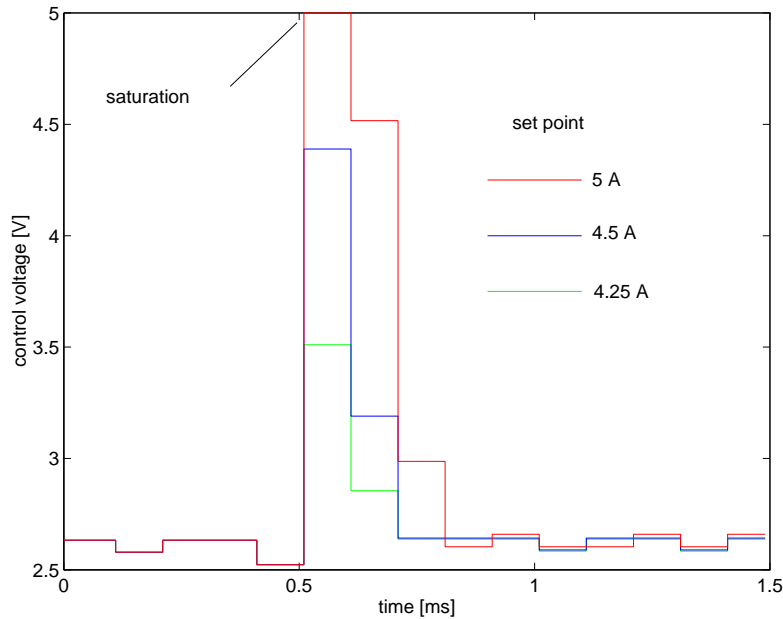


Figure 4.4: Step response of the control voltage after a set point change from 4 A to 4.25 A, 4.5 A, and 5 A.

Since a PI-controller is implemented in a digital version, no set point error remains, but the control current is corrupted by a large ripple due to the switching amplifier operating at a frequency of 60 kHz. The ripple could only be reduced if the switching frequency is increased, or if the switching voltage is decreased. In the latter case, the dynamics suffers from a small current slope.

In addition, the current signals contaminated by the large ripple contain frequencies beyond the Nyquist frequency. This effect is probably due to the presence of nonlinear elements as the quantisers (ADC, and DAC) and to an aliasing effect. Therefore, a low pass filter should be used to cancel the switching frequency and all harmonics. It is advisable to implement this filter as an analogue filter. Unfortunately this filter causes an additional phase lag which is the reason why it was not implemented in this simulation. A better solution is to sample the current at a higher frequency and apply down sampling techniques on the resulting digital signal. This yields a signal at the original sampling frequency of higher quality.

Frequency response

Here, the set point for the current control loop is a sinusoidal signal of 0.25 A, 0.5 A and 1 A amplitude with variable frequency at the point of operation of 4 A. In Fig. 4.5 the frequency plot can be seen. The frequency range of this plot is limited by the Nyquist frequency, i.e. half the sampling frequency of 10 kHz. The bandwidth of the current control loop depends on the amplitude as can be expected for a nonlinear behaviour due to the saturation of the control voltage. For an amplitude range up to 0.71 A, the

−3dB limit is about at 1.7 kHz, which corresponds well to the predicted bandwidth determined by Eqn.(2.52).

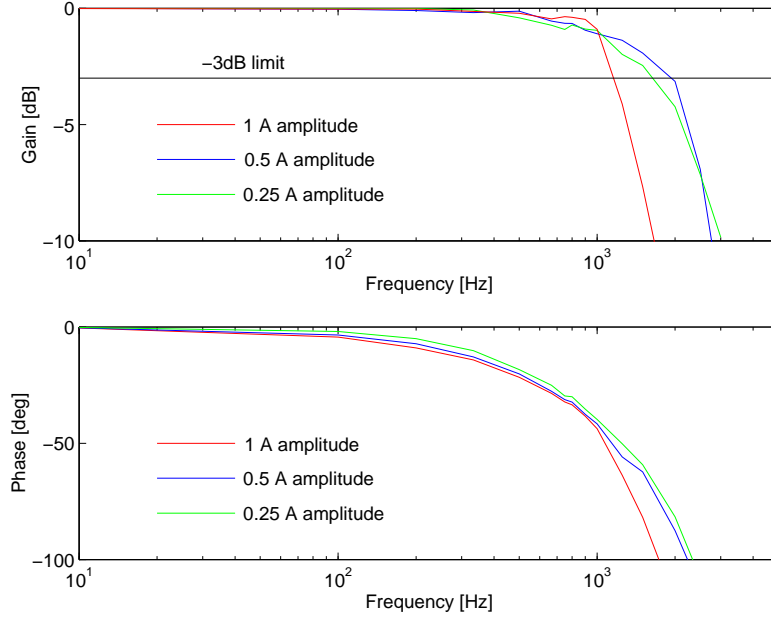


Figure 4.5: Frequency response of the closed current control loop excited by a sinusoidal signal of amplitude 0.25 A, 0.5 A, and 1 A.

For larger current amplitudes \bar{A} the limit frequency depends on the maximal current slope available such that

$$\omega_s \bar{A} = \left. \frac{di}{dt} \right|_{\max}.$$

The maximum frequency the controlled variable can follow is

$$f_s = \frac{\omega_s}{2\pi} \approx \frac{1}{\bar{A}} 865 \text{ Hz}.$$

Up to now only the amplitude was investigated to determine the bandwidth of the current control loop. However, as can be seen in Fig. 4.5, the phase lag is approximately 90 deg for a frequency of 1.7 kHz. Considering the phase lag, the real bandwidth is limited to 250 Hz with a maximum phase lag of 10 deg. Below this frequency the response transfer function can be treated as a gain with respect to the control current.

4.2 The linear actuator model

Once the current controller loop has been tested, the actuator forces and the typical coefficients for an active magnetic bearing can be computed using simulation as well. Using the derivation in Section 2.3.4, these coefficients for an active magnetic bearing with linearised characteristics can be calculated by

$$K_i = \frac{4 N^2 i_0}{\mathcal{R}_0^2 \mu_0 A_l} \cos \frac{\alpha}{2},$$

$$K_s = -\frac{8 N^2 i_0^2}{\mathcal{R}_0^3 \mu_0^2 A_l^2} \cos \frac{\alpha}{2}.$$

Inserting the initial values for the given actuator of Tab. 2.3 yields the coefficients for the active magnetic bearing as given in Tab. 4.2.

Table 4.2: Parameters of the linearised active magnetic bearing.

K_i	current gain	190.04 N/A
K_s	position stiffness	$-1.4138 \cdot 10^6$ N/m

The predicted values for the coefficients can then be compared with the values determined by means of simulation.

4.2.1 Current gain of the magnetic actuator forces

The rotor is held in a fixed position $\mathbf{z}_b = \mathbf{0}$ m at the point of operation defined by the bias current. If the control current \mathbf{i}_c is varied in one direction, the actuator acts as a gain with a certain coefficient, generally referred to as force current gain. The actuator forces $\mathbf{F}_a = [F_x, F_y]$ can be obtained by

$$\mathbf{F}_a = \mathbf{K}_i \mathbf{i}_c$$

with the control currents $\mathbf{i}_c = [i_x, i_y]$ in x and y direction. \mathbf{K}_i is diagonal matrix of proper dimension with the diagonal entries K_i , the current gain factor.

An underlying current control loop is required for all coils of the actuator. The reference signals of each current control loop is computed according to Eqn.(2.67) to Eqn.(2.70) given in Section 2.3.4 from the control current \mathbf{i}_c .

Current step response of the magnetic actuator force

A step response simulation for one actuator was carried out for a control current i_x of 0.25 A, 0.5 A, and 1 A in x -direction. For reasons of symmetry, the response in y -direction is similar and therefore not simulated. The corresponding plot can be

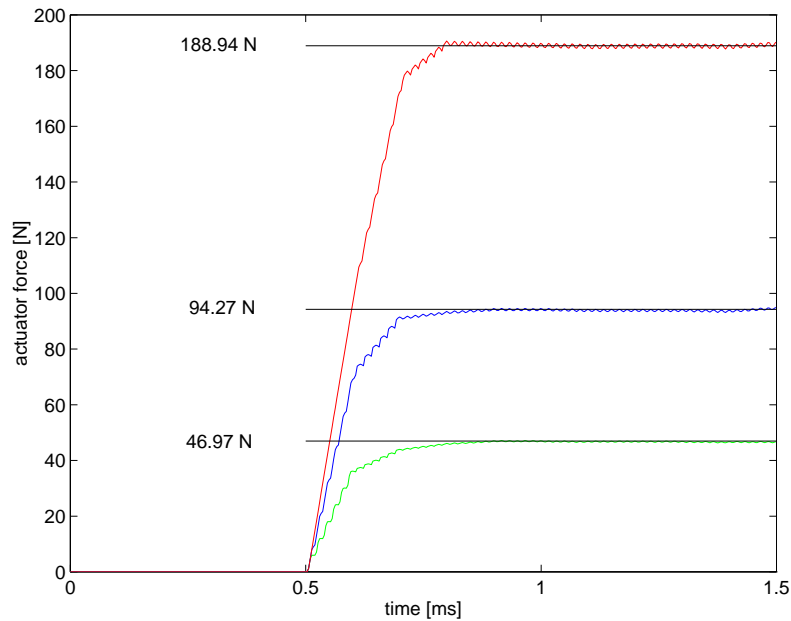


Figure 4.6: Step response of the actuator force for a control current steps of 0.25 A, 0.5 A, and 1 A at $t = 0.5$ ms.

seen in Fig. 4.6. In comparison with the precalculated values, the current gain is $K_i = 188.94$ N/A for a step from $i_x = 0$ A to $i_x = 1$ A, which is about 0.6 % below the precalculated value. For smaller steps, e.g. of $i_x = 0.25$ A, the resulting gain is approximately $K_i = 187.90$ N/A, for a step of $i_x = 0.5$ A, the resulting gain is approximately $K_i = 188.54$ N/A. Thus, the current factor is almost invariant of the magnitude of the step applied. In addition, the value determined by means of simulation is slightly smaller than the precalculated one. This is probably due to the fact that within the simulation the relative permeability of all magnetic materials depends on the flux itself. Therefore, the real flux is a little smaller than the computed flux for $\mu = \mu_0 \mu_r = \text{constant}$.

Current frequency response of the magnetic actuator force

If the actuator is excited by a sinusoidal control current, the actuator force is sinusoidal as well with a certain gain and a certain phase lag. A plot with the sinusoidal control current and the resulting actuator force can be seen in Fig. 4.7. The current gain K_i can be obtained by the slope of the main axis of the ellipsis for the excitation frequency $\rightarrow 0$. It can be seen that this gain decreases with increasing amplitude of the control current. This is due to the limited bandwidth of the current control loop. More obviously, the increasing phase lag can be seen in the phase plot in Fig. 4.7. The consequence is that the actuator can only be treated as a gain up to a frequency of about 250 Hz.

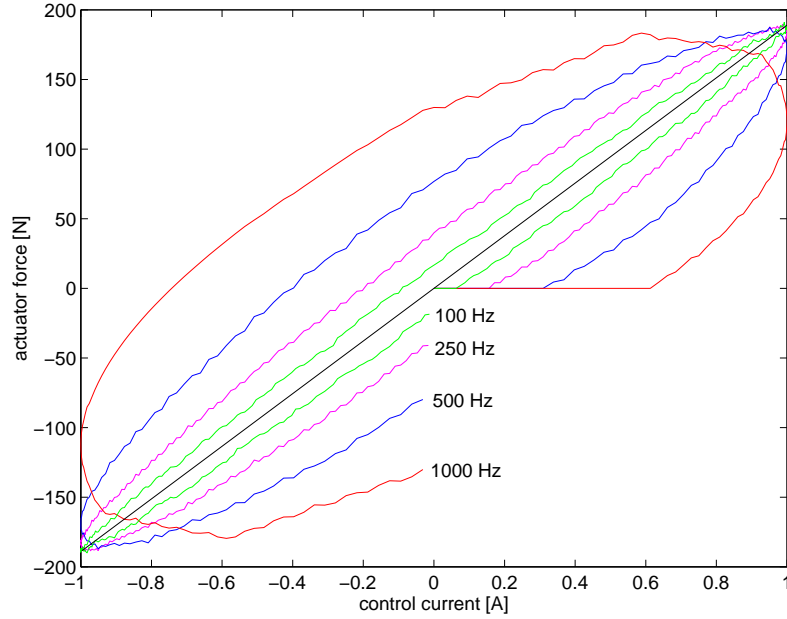


Figure 4.7: Plot of the sinusoidal excitation of control current at a amplitude of 1 A versus the resulting actuator forces at 100 Hz, 250 Hz, 500 Hz, and 1000 Hz.

4.2.2 Position Stiffness of the magnetic actuator forces

The rotor displacement and thus the change of the air gaps within the several electromagnets can be viewed as disturbance for the current control loop (see Fig. 2.8 and Fig. 2.9). This loop controls the current in all electromagnets at the point of operation for varying rotor positions. If the position of the rotor changes, the actuator reacts as a negative spring with a specific coefficient at the point of operation. The actuator forces $\mathbf{F}_a = [F_x, F_y]$ can be obtained by

$$\mathbf{F}_a = \mathbf{K}_s \mathbf{z}_b$$

with the rotor position $\mathbf{z}_b = [x, y]$ and the diagonal matrix \mathbf{K}_s with K_s the position stiffness as diagonal entry. This position coefficient for an actuator can be calculated by a steady state simulation as well as by frequency analysis.

Displacement step response of the magnetic actuator force

A step response for one actuator was carried out for a rotor displacement in one direction x of $25 \mu\text{m}$, $50 \mu\text{m}$ and $100 \mu\text{m}$. The corresponding plots can be seen in Fig. 4.8. The mean values for all step responses can be used to compute the position stiffness which in this case, depends on the magnitude of the step. For a step of $10 \mu\text{m}$, the resulting stiffness is $1.40 \cdot 10^6 \text{ N/m}$ which is quite close to the predicted value of $1.4138 \cdot 10^6 \text{ N/m}$. For a step of $50 \mu\text{m}$ the position stiffness is $1.4258 \cdot 10^6 \text{ N/m}$, and for

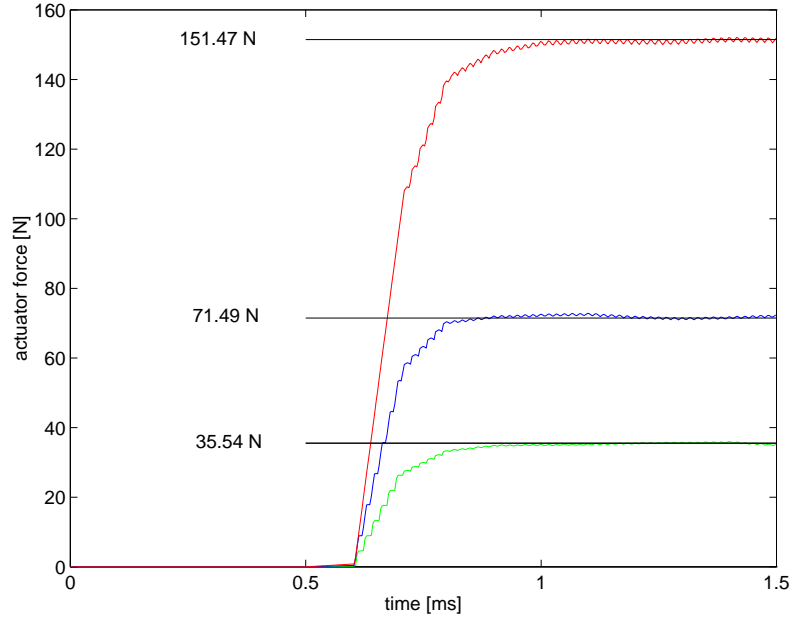


Figure 4.8: Step response of the controlled actuator for rotor displacements of $10\ \mu\text{m}$, $50\ \mu\text{m}$, and $100\ \mu\text{m}$ at $t = 0.5\ \text{ms}$.

a step of $100\ \mu\text{m}$ it is $1.5090 \cdot 10^6\ \text{N/m}$. This means that the stiffness increases with increasing rotor displacement. Therefore, simulations have been carried out for various steps of different magnitudes. The corresponding plot can be seen in Fig. 4.9. The pro-

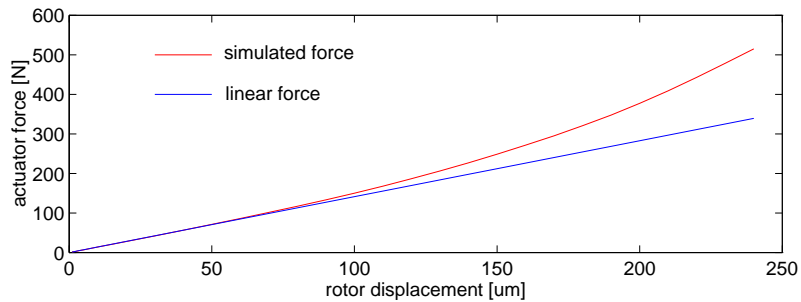


Figure 4.9: Actuator force as function of the rotor displacement for a given point of operation defined by $i_0 = 4\ \text{A}$.

gressive behaviour can be explained by the dependence of the position stiffness on the air gap length which directly influences the magnetic flux. If the current is controlled to be constant, the flux increases whenever the reluctance decreases.

In Fig. 4.8 it can be observed that it takes approximately one sampling interval until the actuator force increases. As mentioned before, the varying rotor displacement and thus the varying air gap within the magnetic path can be viewed as disturbance for

the current control loop. The entire energy of the magnetic path is stored in the flux. If the rotor is moved in a stepwise manner, the flux does not change abruptly, i.e. the actuator force does not change abruptly either. Since an ideal voltage source is assumed, the step of the rotor displacement does not introduce a voltage, but changes the rate of the flux and the current instantaneously to keep the energy constant. After the current control loop has managed to keep the current at the point of operation, the entire energy has changed according to the definition of the flux depending on the current and the air gap. Thus, the resulting force increases as fast as the current loop can control the coil current.

Frequency response of the magnetic actuator force in terms of the rotor displacement

If the actuator is excited by a sinusoidal signal of the rotor displacement, the actuator forces is sinusoidal as well with a certain gain and a certain phase lag. A phase plane plot can be seen in Fig. 4.10. The resulting stiffness decreases with increasing

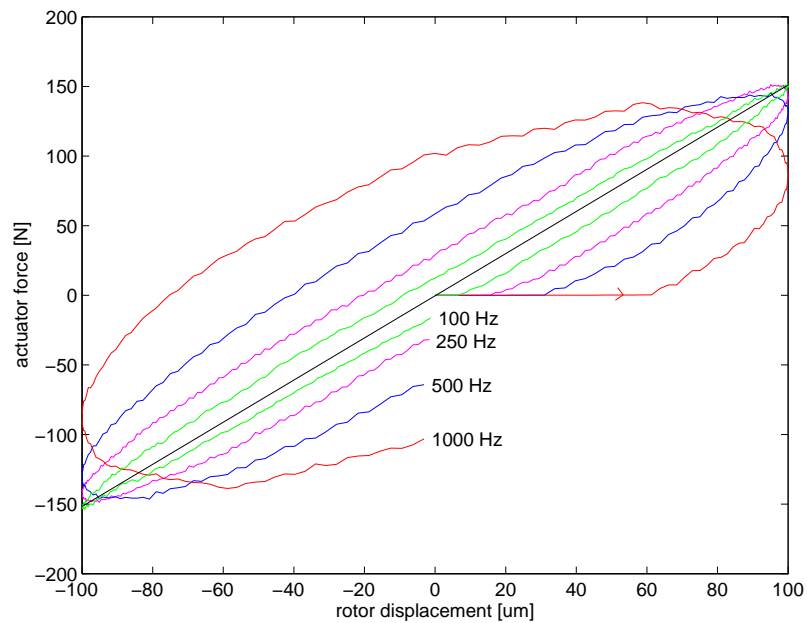


Figure 4.10: Plot of the sinusoidal excitation of the rotor displacement at a amplitude of $100 \mu\text{m}$ versus the resulting actuator forces at 100 Hz, 250 Hz, 500 Hz and 1000 Hz.

frequency starting at approximately $1.52 \cdot 10^6 \text{ N/m}$. Additionally, the phase lag increases according to the current control loop which generates the elliptic shape of the force to displacement curve.

In this section it has been shown that the actuator controlled by a current control loop can be treated as a gain up to a frequency of approximately 250 Hz. Thus, the

derivation of a linear state space model is justified. This model can then be used for a state space controller, as long as the frequency band of the resulting closed loop of the position controller is below the frequency band of the the current control loop.

4.3 The linear state space model

After the linearity of the actuator has been shown in the previous section by means of simulation, the deterministic linear state space plant model can be established. This model is used for the state space adaptive control concept. Note that for this state space model an internal current control loop is always required.

To keep the rotor in a centred position, the steady state load has to be compensated by an additional control current such that

$$\mathbf{i}_{c_w} = \mathbf{K}_i^{-1} \cdot \mathbf{F}_w,$$

with the rotor weight components in bearing coordinates

$$\mathbf{F}_w = [0, -141.8302, 0, -140.3839,]^T \text{ N.}$$

This is necessary to provide the disturbances of zero mean which is an assumption for both the adaptive control concept and the predictor. Heuristically speaking, the predictor cannot render the reaction of the system to be observed, if there is no model available describing how the disturbances act on the system. The resulting steady state control current

$$\mathbf{i}_{c_w} = [0, -0.7463, 0, -0.7387]^T \text{ A}$$

is simply added to the dynamic control current in order to compensate the steady state load. Note that the point of operation is shifted regarding the steady state coil currents which affects the current gain and the position stiffness.

4.3.1 System matrices

The continuous time state space model was derived in Chapter 2 in the form

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u}, \\ \mathbf{y} &= \mathbf{C} \cdot \mathbf{x} + \mathbf{v}, \end{aligned}$$

with the state vector $\mathbf{x} = [\mathbf{z}_b, \dot{\mathbf{z}}_b]^T$ and the output vector $\mathbf{y} = k_m (\mathbf{z}_b + \mathbf{v})$. A virtual sensor gain k_m converts the system states from m into μm for numerical reasons. The concentrated measurement noise is assumed to have a maximum deflection of $1 \mu\text{m}$. Inserting all rotor parameters, an initial non-conservative cross-coupling stiffness parameter $k_n = 0 \text{ N/m}$ and the coefficients for the active magnetic bearing, the system

matrices are (in the following numbers consistently SI-units are applied)

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1.4252e3 & 0.0 & -0.4519e3 & 0.0 & 0 & 0 & 0 & 0 \\ 0.0 & 1.4252e3 & 0.0 & -0.4519e3 & 0 & 0 & 0 & 0 \\ -0.4519e3 & 0.0 & 1.4446e3 & 0.0 & 0 & 0 & 0 & 0 \\ 0.0 & -0.4519e3 & 0.0 & 1.4446e3 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 19.1573 & 0.0 & -6.0748 & 0.0 \\ 0.0 & 19.1573 & 0.0 & -6.0748 \\ -6.0748 & 0.0 & 19.4173 & 0.0 \\ 0.0 & -6.0748 & 0.0 & 19.4173 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1e6 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Using a sampling time of $T_s = 100 \mu\text{s}$ and the transformation \mathbf{T} as derived in Appendix A, the discrete time state space model in controller canonical form results in

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C} \mathbf{x}(k), \end{aligned}$$

with the state vector is $\mathbf{x}(k)$. The input vector $\mathbf{u}(k)$ is the sampled input vector $\mathbf{u}(t)$ and the output vector $\mathbf{y}(k)$ is the sampled vector $\mathbf{y}(t)$. The system matrices are given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.0 & 2.0014 & 0.0 & 0.0 & 0.0 & -0.0005 & 0.0 & 0.0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0.0 & 0.0 & -1.0 & 2.0014 & 0.0 & 0.0 & 0.0 & -0.0005 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0.0 & -0.0005 & 0.0 & 0.0 & -1.0 & 2.0014 & 0.0 & 0.0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0.0 & 0.0 & 0.0 & -0.0005 & 0.0 & 0.0 & -1.0 & 2.0014 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{C} = \begin{bmatrix} 0.096 & 0.096 & 0.0 & 0.0 & -0.030 & -0.030 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.096 & 0.096 & 0.0 & 0.0 & -0.030 & -0.030 \\ -0.030 & -0.030 & 0.0 & 0.0 & 0.097 & 0.097 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.030 & -0.030 & 0.0 & 0.0 & 0.097 & 0.097 \end{bmatrix}.$$

Note that some entries of the state space matrices are not necessarily equal to zero for numerical reasons which is indicated by the number 0.0 instead of simply 0. The latter is used, if the controller canonical form requires a zero entry.

4.3.2 State space controller

Since the system is open-loop unstable, a controller has to be designed in advance. This has been done according to the nominal system without destabilising cross-coupling effects of the rotor. The poles of the closed loop system have been placed at $z_i = 0.97$, or in the continuous complex plane to $s_i = -288$ 1/s, which would meet the eigenvalue for a mass-spring-system with a negative stiffness factor of the magnetic bearing.

Inputs to the controller are the estimated system states $\hat{\mathbf{x}}(k)$ provided by the predictor, and the rotor positions $\mathbf{y}(k)$ determined by equation Eqn.(2.80) and Eqn.(2.79). Output of the position controller is the control current $\mathbf{u} = \mathbf{i}_c$ which is used to compute the reference signals of each current control loop according to Eqn.(2.67) to Eqn.(2.70) as presented in Section 2.3.4 for one actuator.

P-structure

The characteristic polynomial for the closed loop system can be easily calculated with all system poles at $z_i = 0.97$ as

$$P^{(ii)}(z) = z^2 - 1.9382 z + 0.9392.$$

The control matrix \mathbf{K}_x for the control law

$$\mathbf{u}(k) = -\mathbf{K}_x \mathbf{x}(k) + \mathbf{K}_w \mathbf{w}(k)$$

can then be computed using Eqn.(3.70) to Eqn.(3.71) as

$$\mathbf{K}_x = \begin{bmatrix} -0.061 & 0.063 & 0.0 & 0.0 & 0.0 & -0.001 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.061 & 0.063 & 0.0 & 0.0 & 0.0 & -0.001 \\ 0.0 & -0.001 & 0.0 & 0.0 & -0.061 & 0.063 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.001 & 0.0 & 0.0 & -0.061 & 0.063 \end{bmatrix}.$$

Using Eqn.(3.76) the feed forward gain can be computed as

$$\mathbf{K}_w = \begin{bmatrix} 0.0055 & 0.0 & 0.0017 & 0.0 \\ 0.0 & 0.0055 & 0.0 & 0.0017 \\ 0.0017 & 0.0 & 0.0055 & 0.0 \\ 0.0 & 0.0017 & 0.0 & 0.0055 \end{bmatrix}.$$

Since the controller design is carried out for the nominal system without cross-coupling effects, the stability region for this controller design can be shown by a root locus plot with the non-conservative stiffness coefficient k_n as parameter. The resulting plot can be seen in Fig. 4.11 with k_n starting at zero for the nominal system and increasing up to 10^7 N/m. If $k_n = 0$ N/m, all 8 poles are identically located at $z_i = 0.97$. If k_n is larger than $4.2 \cdot 10^6$ N/m, the system becomes unstable unless the controller is adapted to the new parameters.

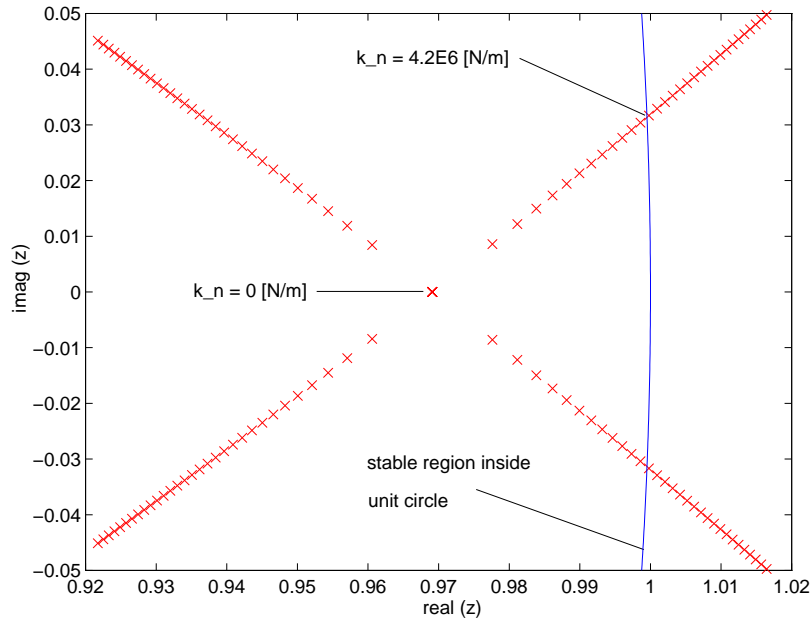


Figure 4.11: Root locus of the closed loop system with varying non-conservative stiffness parameter k_n from 0 N/m to 10^7 N/m, and with constant controller parameters. The closed loop system starting at $z_i = 0.97$ is unstable for $k_n > 4.2 \cdot 10^6$ N/m.

PI-structure

The characteristic polynomial of the closed loop system with additional integrative feedback can be easily calculated with all system poles at $z_i = 0.97$ as

$$P_I^{(ii)}(z) = z^3 - 2.9073 z^2 + 2.8175 z - 0.9101.$$

The control law for the PI-structure is given by

$$\begin{aligned}\mathbf{u}(k) &= \mathbf{u}_I(k) - \mathbf{K}_x \mathbf{x}(k), \\ \mathbf{u}_I(k+1) &= \mathbf{u}_I(k) + \mathbf{K}_I \mathbf{e}(k), \\ \mathbf{e}(k) &= \mathbf{w}(k) - \mathbf{y}(k).\end{aligned}$$

Using Eqn.(3.93) the controller matrix \mathbf{K}_I for the initial system is

$$\mathbf{K}_I = 10^{-3} \begin{bmatrix} 0.1709 & 0.0 & 0.0535 & 0.0 \\ 0.0 & 0.1709 & 0.0 & 0.0535 \\ 0.0535 & 0.0 & 0.1687 & 0.0 \\ 0.0 & 0.0535 & 0.0 & 0.1687 \end{bmatrix}.$$

The coefficients of the polynomial for the closed loop system without integrative feedback have been determined using Eqn.(3.94) and Eqn.(3.95). For the nominal system which consists of four identical subsystems, these coefficients determine the new polynomial for the system without integrative feedback

$$P^{(ii)}(z) = z^2 - 1.9073z + 0.9101.$$

With these coefficients the controller matrix \mathbf{K}_x can be obtained from Eqn.(3.70) and Eqn.(3.71) as

$$\mathbf{K}_x = \begin{bmatrix} -0.090 & 0.094 & 0.0 & 0.0 & 0.0 & -0.001 & 0.0 & 0.0 \\ 0.0 & 0.0 & -0.090 & 0.094 & 0.0 & 0.0 & 0.0 & -0.001 \\ 0.0 & -0.001 & 0.0 & 0.0 & -0.090 & 0.094 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & -0.001 & 0.0 & 0.0 & -0.090 & 0.094 \end{bmatrix}.$$

The controller design is carried out for the nominal system without cross-coupling effects. A root locus with the non-conservative stiffness coefficient k_n as parameter shows the stability margin, which can be seen in Fig. 4.12. k_n starts at zero with all system poles located at $z_i = 0.97$. If k_n is larger than $9.4 \cdot 10^6$ N/m, the system becomes unstable unless the controller is adapted to the new parameters.

4.3.3 Predictor

After the state space model is assigned, the predictor used for the state space adaptive control has to be calculated. Additionally, this predictor is necessary to compute the estimated system states for the state space controller following the law

$$\begin{aligned}\hat{\mathbf{x}}(k+1, \mathbf{p}) &= \mathbf{A}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}) + \mathbf{B} \mathbf{u}(k) + \mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon}(k), \\ \hat{\mathbf{y}}(k) &= \mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}), \\ \boldsymbol{\varepsilon}(k, \mathbf{p}) &= \mathbf{y}(k) - \hat{\mathbf{y}}(k|\mathbf{p}).\end{aligned}$$

Inputs to the predictor are the rotor positions $\mathbf{y}(k)$ determined by equation Eqn.(2.80) and Eqn.(2.79). Outputs are the estimated system states $\hat{\mathbf{x}}(k)$ which are inputs to the state space controller.

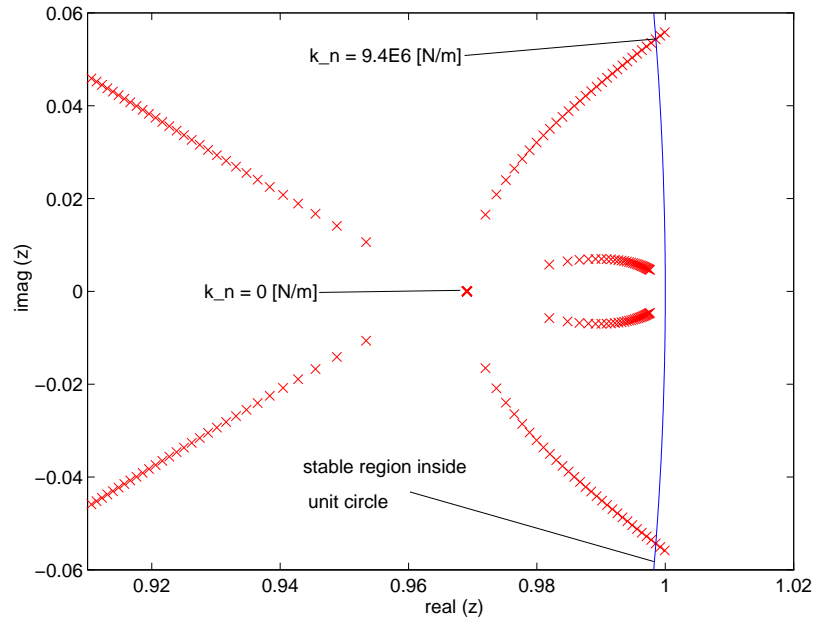


Figure 4.12: Root locus of the closed loop system with varying non-conservative stiffness parameter k_n from 0 N/m to 10^7 N/m and constant controller parameters. The closed loop system starting at $z_i = 0.97$ is unstable for $k_n > 9.4 \cdot 10^6$ N/m.

Since the system to be controlled is open loop unstable and the statistical properties of the system and measurement noise are not known in advance, the Kalman matrix has to be designed in advance, too. This deterministic approach can be done using pole placement for the state space model.

Similarly to the state space model in controller canonical form, a model can be derived in observer canonical form [Tolle, 1985]. If this model is used, the observer design can be carried out using the same procedure as for the controller design with all matrices transposed. The only design parameters are the poles of the initial predictor. Choosing these poles to be $z_i = 0.90$, which corresponds to a predictor being four times faster than the closed loop system, the resulting initial Kalman matrix becomes

$$\mathbf{K} = \begin{bmatrix} 1.1332 & 0.0 & 0.3535 & 0.0 \\ 1.1993 & 0.0 & 0.3717 & 0.0 \\ 0.0 & 1.1332 & 0.0 & 0.3535 \\ 0.0 & 1.1993 & 0.0 & 0.3717 \\ 0.3535 & 0.0 & 1.1181 & 0.0 \\ 0.3717 & 0.0 & 1.1834 & 0.0 \\ 0.0 & 0.3535 & 0.0 & 1.1181 \\ 0.0 & 0.3717 & 0.0 & 1.1834 \end{bmatrix}.$$

4.4 Closed loop dynamics

Once the controller and the predictor have been designed, the closed loop dynamics can be simulated in order to verify the desired behaviour of the controlled rotor bearing system. Additionally, simulations should show whether the linearisation of the actuator is justified and the linear state space controller can cope with both set point changes and disturbances like additional loads.

4.4.1 Initial values for simulation

Before a simulation run can be started the initial values for the controller and for all system variables have to be defined and chosen properly in advance.

All current control loops start at the point of operation defined by the bias current $i_0 = 4$ A with all state variables like the initial flux Φ_0 and the control voltage u_c initialised properly for each channel similarly to Section 4.1.3.

The steady state force acting on the rotor, like the rotor weight, is compensated by a steady state magnetic force using the current control loops as described previously. Since no external forces act on the rotor, the rotor can be assumed to stay in a centred position and behave according to the derived state space model. The initial state vector is set equal to zero for this position as $\mathbf{x}(t = 0) = \mathbf{0}$. The same is true for the estimated state vector $\hat{\mathbf{x}}(k = 0) = \mathbf{0}$ with $\mathbf{u}(k = 0) = \mathbf{0}$ and $\boldsymbol{\varepsilon}(k = 0) = \mathbf{0}$.

4.4.2 System response to set point change

The closed loop system poles were placed at $z_i = 0.97$ or $s_i = -2881/s$ for the controller with and without integrative feedback. In the following, system responses to a set point change are simulated for a P-structured and a PI-structured controller.

P-structure

A set point change is applied to the controlled system with $100 \mu\text{m}$ in each direction in both bearing planes, i.e. $\mathbf{W} = [100, 100, 100, 100]^T \mu\text{m}$ (see Fig. 3.3). The resulting time history for the rotor displacements and the control currents can be seen in Fig. 4.13 and Fig. 4.14.

As mentioned previously, a steady state control current is added to the control current with $\mathbf{i}_{c_w} = \mathbf{K}_i^{-1} \cdot \mathbf{F}_w$. The current gain K_i is computed for the point of operation defined by $i_0 = 4$ A. Since the point of operation is shifted due to the additional load, a mismatch between the assumed system and the true system is generated with respect to the estimated coefficients for the active magnetic bearing. This results in the deviation of the rotor from the centre. Additionally, the controller parameters calculated from the estimated current gain and position stiffness cause a position error in y -direction.

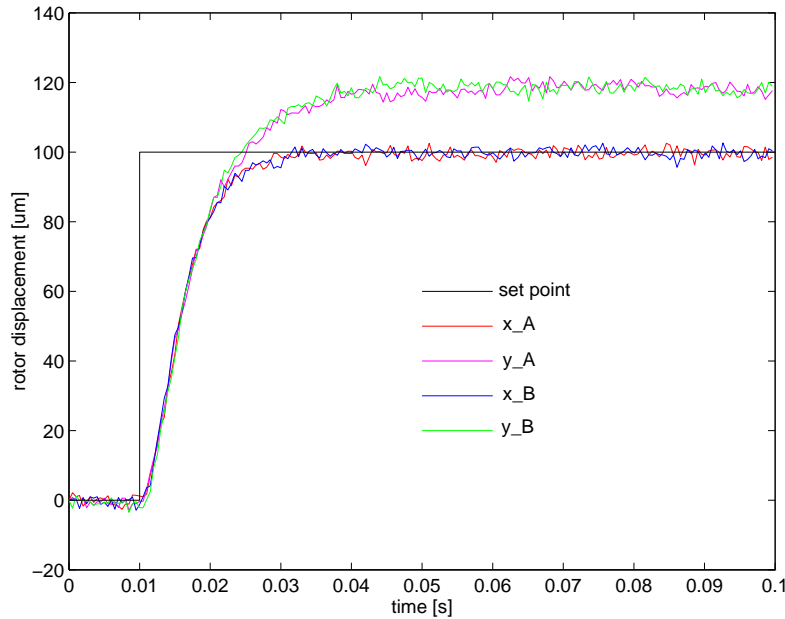


Figure 4.13: Time history of the rotor displacements after a set point change from zero to $\mathbf{W} = [100, 100, 100, 100]^T \mu\text{m}$ at time $t = 0.01 \text{ s}$ (P -structured controller).

PI-structure

A set point change is applied to the controlled system with $100 \mu\text{m}$ in each direction in both bearing planes, i.e. $\mathbf{W} = [100, 100, 100, 100]^T \mu\text{m}$ (see Fig. 3.4). In this case an integrative feedback changes the steady state behaviour as shown in Fig. 4.15 for the time history of the rotor displacements and in Fig. 4.16 for the time history of the control currents, respectively. Note that no set point error remains if integrative feedback is added to the controller.

4.4.3 System response to additional load

In order to investigate the closed loop system behaviour with respect to external disturbances, an additional load is applied to the rotor, in this case 100 N in x -direction on both bearing A and bearing B.

P-structure

The additional load appears at $t = 0.01 \text{ s}$ with 100 N for each bearing in x -direction. The time history of the rotor displacements and the control currents can be seen in Fig. 4.17 and Fig. 4.18, respectively.

The load of 100 N in x -direction causes a mean displacement of approximately $165 \mu\text{m}$.

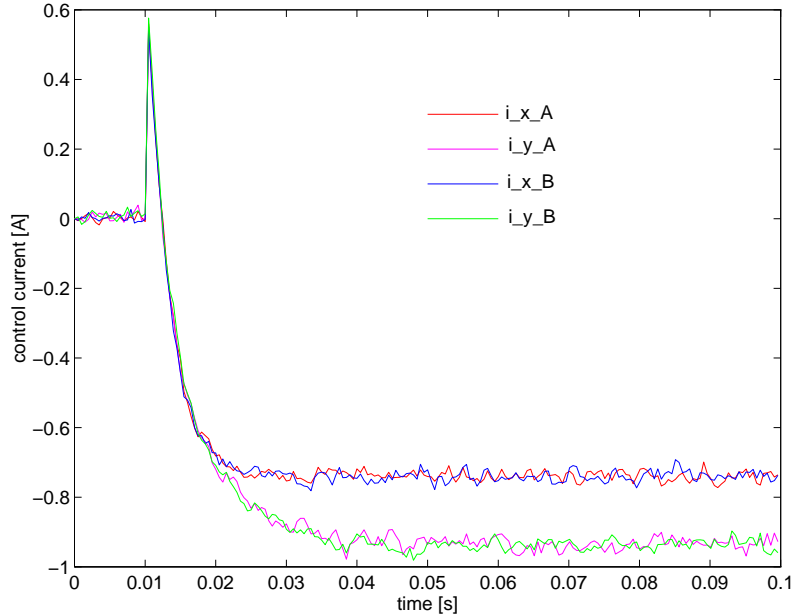


Figure 4.14: Time history of the control currents after a set point change from zero to $\mathbf{W} = [100, 100, 100, 100]^T \mu\text{m}$ at time $t = 0.01$ s (P -structured controller).

This yields a closed loop stiffness for the active magnetic bearing under state space control of $6 \cdot 10^6$ N/m or a flexibility of $1.65 \cdot 10^{-6}$ m/N.

Note that in y -direction there is a slight deviation from the centre position, although the rotor weight is compensated by a steady state current. The reason for the deviation is the difference between the calculated stiffness factors for the point of operation and the real stiffness coefficients which depend on the new point of operation determined by the steady state currents. This effect appears independently from the additional load and can only be compensated by an iterative solution for the current gain of the magnetic actuator which depends on the steady state load, e.g. rotor weight. This load influences the steady state currents which are computed from the current gain.

PI-structure

A load of 100 N for each bearing is applied to the rotor at $t = 0.01$ s in x -direction. The corresponding time history of the rotor displacements and the control currents can be seen in Fig. 4.19 and Fig. 4.20, respectively.

As shown in Fig. 4.19 the advantage of an additional integrative feedback becomes obvious. Despite the additional load no deviation from the centred rotor position remains. This behaviour can be interpreted as an infinite gain of the actuator at a zero frequency and distinguishes a rotor bearing system using active magnetic bearings from ordinary rotor bearing systems.

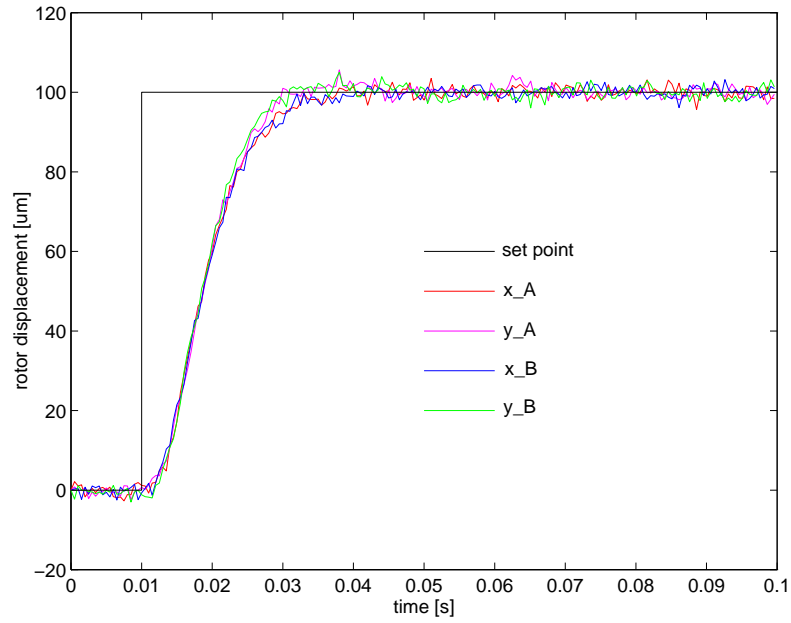


Figure 4.15: Time history of the rotor displacements after a set point change from zero to $\mathbf{W} = [100, 100, 100, 100]^T \mu\text{m}$ at time $t = 0.01 \text{ s}$ (PI-structured controller).

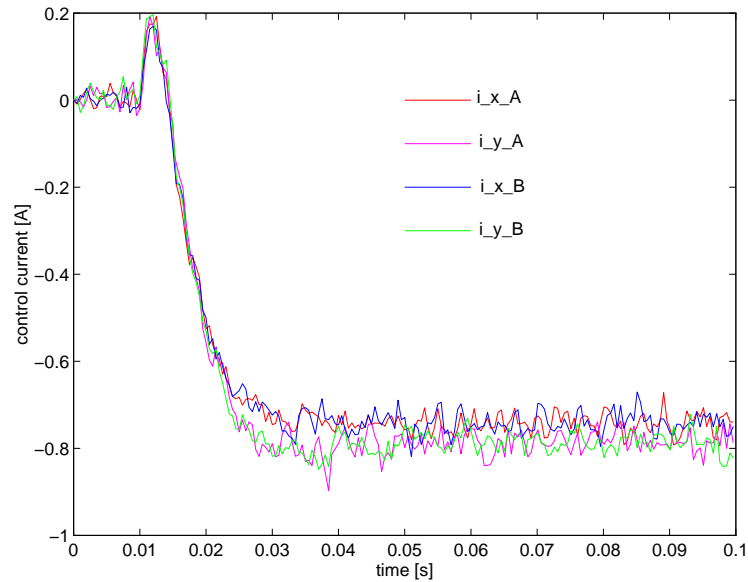


Figure 4.16: Time history of the control currents after a set point change from zero to $\mathbf{W} = [100, 100, 100, 100]^T \mu\text{m}$ at time $t = 0.01 \text{ s}$ (PI-structured controller).

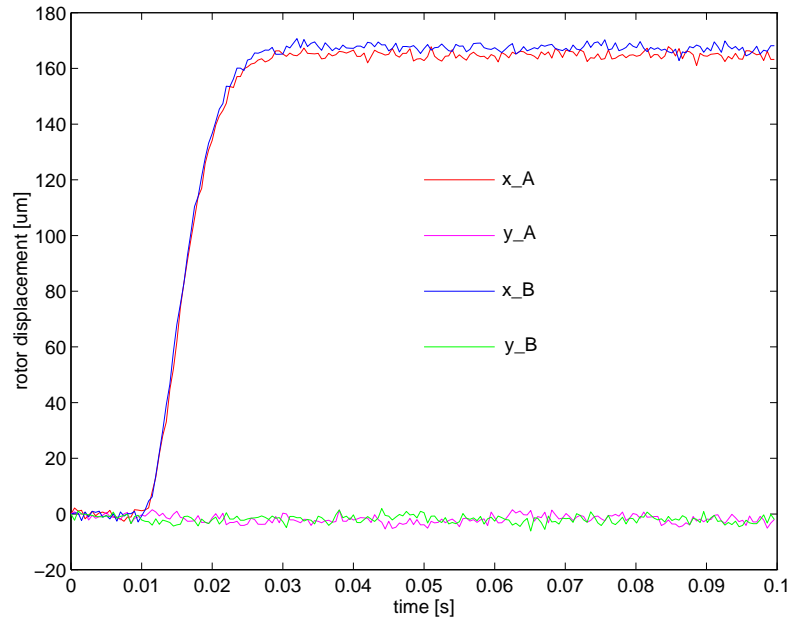


Figure 4.17: Time history of the rotor displacements after an additional load of 100 N exerted to each bearing at time $t = 0.01$ s (P -structured controller).

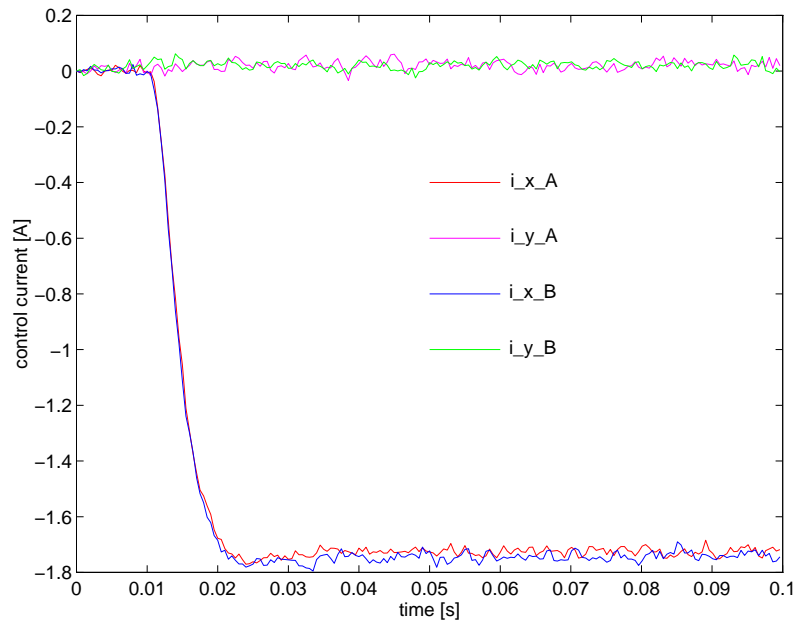


Figure 4.18: Time history of the control currents after an additional load of 100 N exerted to each bearing at time $t = 0.01$ s (P -structured controller).

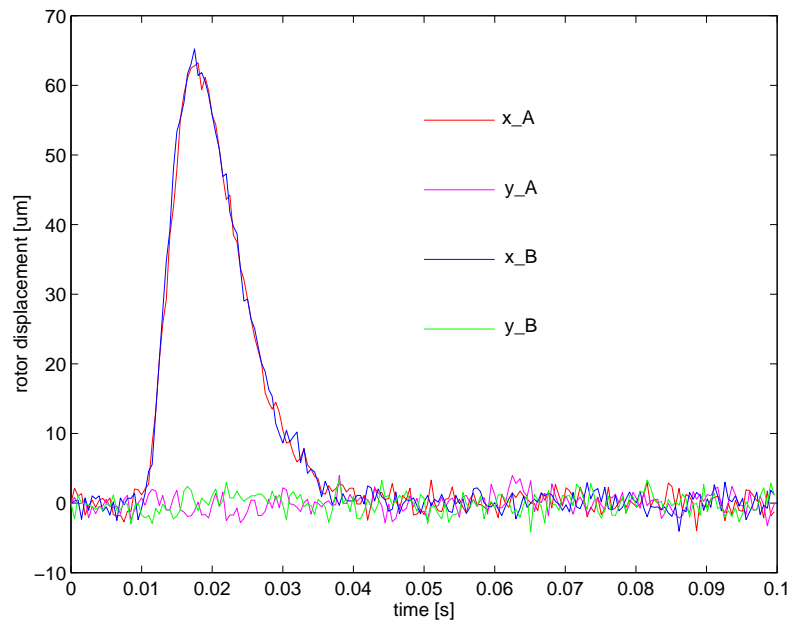


Figure 4.19: Time history of the rotor displacements after an additional load of 100 N exerted to each bearing at time $t = 0.01$ s (PI-structured controller).

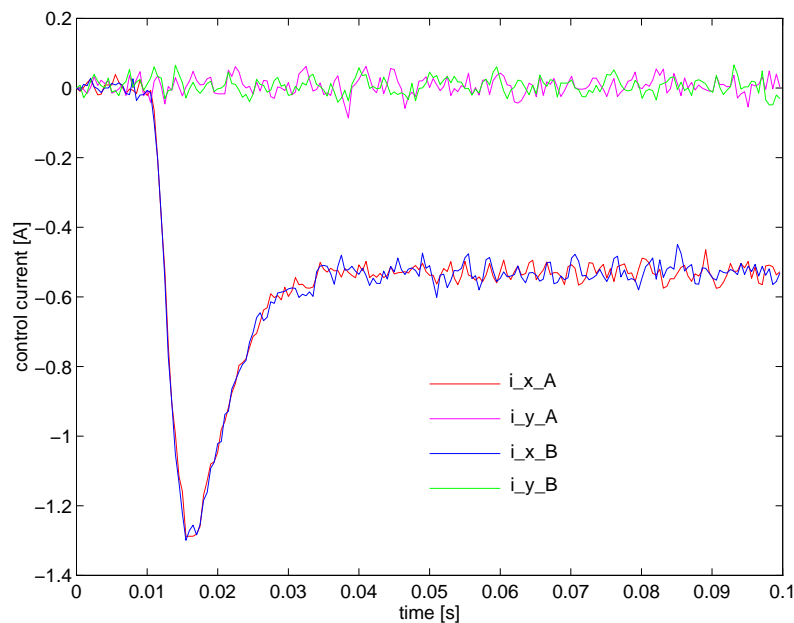


Figure 4.20: Time history of the control currents after an additional load of 100 N exerted to each bearing at time $t = 0.01$ s (PI-structured controller).

4.5 State space adaptive control

The major objective of the presented work is the state space adaptive control for a rigid rotor suspended in two active magnetic bearings. After it has been shown that a state space controller based on a linear discrete time model of the plant is able to control the system, the adaptive control algorithm developed in Chapter 3 is implemented using a nonlinear plant model as described in detail in Chapter 2 with the block diagrams shown in Appendix D.1.

In order to simulate a change in system parameters, the sudden appearance of non-conservative cross-coupling forces as generated by seals of rotating machinery rotors is assumed. These forces change the system matrix by changing the skew-symmetric cross-coupling terms of the stiffness matrix in a given rotor plane. The step is considered to be the worst case in turbomachinery application, e.g. a sudden pressure loss or leakage in sealings leads to a parameter change in a linear model. Actually, the effect of the cross-coupling forces develops slowly, and an abrupt change in the non-conservative stiffness parameter can be considered as a worst case scenario.

In the following all initial values necessary for both simulation and for a real time implementation are considered.

4.5.1 Initial values for simulation

Before a simulation run can be started the initial values for the adaptive control and for all system variables have to be defined and chosen properly in advance.

Initial system states

For the simulation of a parameter change all initial state values are set equal to the initial values for a set point change as described in the previous section, namely $\mathbf{x}(t = 0) = \mathbf{0}$, $\hat{\mathbf{x}}(k = 0) = \mathbf{0}$, $\mathbf{u}(k = 0) = \mathbf{0}$, and $\boldsymbol{\varepsilon}(k = 0) = \mathbf{0}$.

Initial parameters and covariance matrix

In contrast to most applications of adaptive control, the initial parameters in the parameter vector \mathbf{p} are assumed to be the true ones with a rather small covariance \mathbf{P} . Usually, the initial parameters are chosen equal to zero with a high covariance. If the plant to be identified is open loop stable and the identification is performed in an open loop mode, this approach suits very well and the parameters to be estimated converge fast. Since an unstable plant has to be identified in closed loop, the choice of the initial parameters is more complex.

A parameter error may cause instability from the beginning, because the controller parameters depend on the initial parameters. If these are false, the control action is wrong as well. Therefore, the initial parameters for the estimation algorithm are set

equal to the *a priori* values, derived from the linear state space model in controller canonical form.

A large covariance has similar effects on wrong system parameters, since the parameters change very quickly. Therefore, the initial covariance is set to a rather small value of $\mathbf{P} = 10^{-8} \cdot \mathbf{I}$. The covariance is only increased if a parameter change is triggered by using the forgetting factor.

Initial forgetting factor

The forgetting factor $\rho(k)$ determines the change of the covariance matrix, i.e. its increase or decrease. The initial value is chosen to be $\rho(k=0) = 1$ which corresponds to practically no forgetting, or heuristically speaking, the initial parameters are believed to be the true ones.

4.5.2 Implementation of the state space adaptive controller

The control algorithm is coded with the numerical engineering tool MATLAB with its functions given in Appendix D.2 [MATLAB, 1992]. The algorithm includes the parameter estimation algorithm, the computation of the controller parameters based upon the estimated system parameters, the predictor, and the state space controller. All functions are executed at each sampling period by the simulation program.

Since the state space adaptive controller is implemented separately from the simulation model, the porting to a real application should be easier. The nonlinear model itself is executed at a integration frequency of 1 MHz using the Runge-Kutta algorithm [SIMULINK, 1992]. This yields a simulation step size of 1 μ s, which is very small, but necessary in order to render the behaviour of the switching amplifiers. Otherwise the zero crossings are not hit exactly which results in additional parasitic noise.

4.5.3 System response due to parameter change

After the initial values have been set, the rotor, suspended on two active magnetic bearings, is simulated in the MATLAB/SIMULINK environment. Although the rotor runs at a speed of 20,000 rpm, no unbalance forces are considered, because the unbalance response conceal instability. This is done only to show whether the control concept works or not. The so controlled rotor bearing system is then exposed to a parameter change in order to test the adaptive control concept. Simulation runs have been carried out for both a simple state space controller and a state space controller with additional integrative feedback.

P-structure

The closed loop system controlled by a simple state space controller is investigated with respect to a parameter change at $t = 0.01$ s applied to the system by a sudden

appearance of cross-coupling forces with a skew-symmetric non-conservative stiffness factor of $k_n = 6 \cdot 10^6$ N/m at a given plane along the rotor axis determined by the parameter n given in Tab. 2.4.

Immediately after the parameter change the system is unstable in terms of its structure, but the displacements increase slowly. When a certain signal to noise ratio is achieved, the parameter change is recognised by the algorithm. In other words, if the variance of the prediction error increases and thus the forgetting control factor $\delta(k)$ presented in Eqn.(3.48) passes a certain threshold, the parameters are assumed to have changed. Therefore, the forgetting factor is set to a smaller value, in this case $\rho_0 = 0.999$. The corresponding plots can be seen in Fig. 4.21.

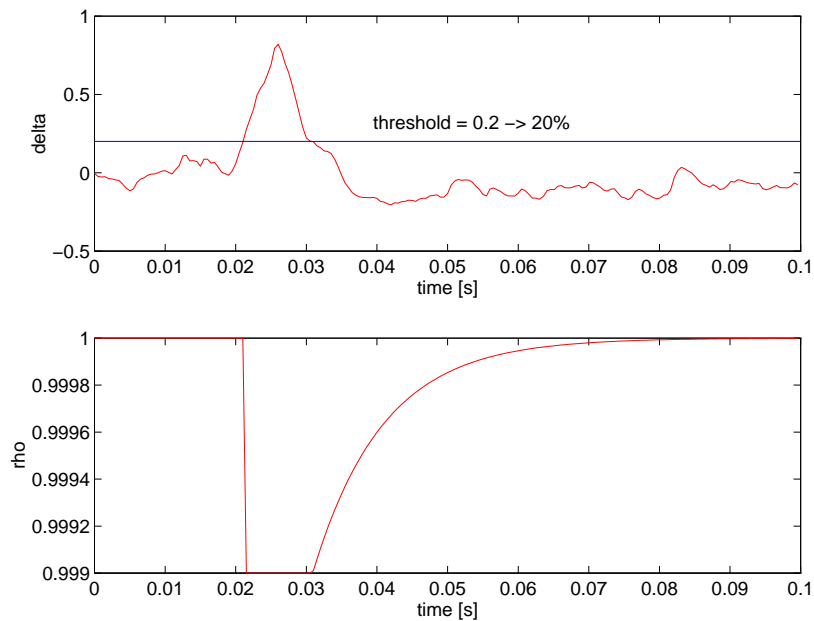


Figure 4.21: Time history of the forgetting control variable δ and the forgetting factor ρ , after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (P -structured controller).

In addition to the reset of $\rho(k)$, a white noise signal with the arbitrarily chosen maximum deflection of $100 \mu\text{m}$ is added to the set point. This generates a ripple on the control variable, because the set point is fed through the feed forward matrix \mathbf{K}_w , and excites the entire system. Since this part of the control variable is not correlated to the estimation error generated by the measurement noise, the signal to noise ration increases which makes the algorithm converge faster. A plot of the set point can be seen in Fig. 4.22. The burst signal is added to the set point as long as $\rho(k)$ remains reset, i.e. as long as $\delta(k)$ is larger than the threshold.

Parallel to the identification, the controller parameters are calculated according to Eqn.(3.70) and Eqn.(3.71). It can be observed that the feedback gain matrix of the

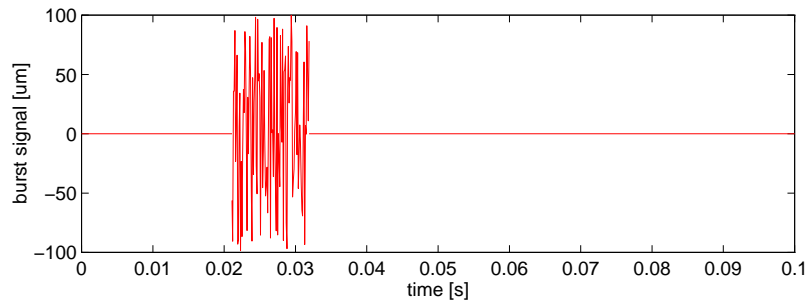


Figure 4.22: Time history of the white noise burst signal added to the set point, when a parameter change is triggered (P -structured controller).

state space controller becomes non-symmetric to the same extent as the parameter of the non-conservative stiffness does. This effect can easily be seen in Eqn.(3.70). These non-symmetric entries in the controller matrix cause a force, which directly compensates the non-conservative forces of the system. Due to the adaptation of the controller the rotor bearing system can recover from a destabilising parameter change.

The time history of the rotor displacements and the control currents can be seen in Fig. 4.23 and Fig. 4.24, respectively.

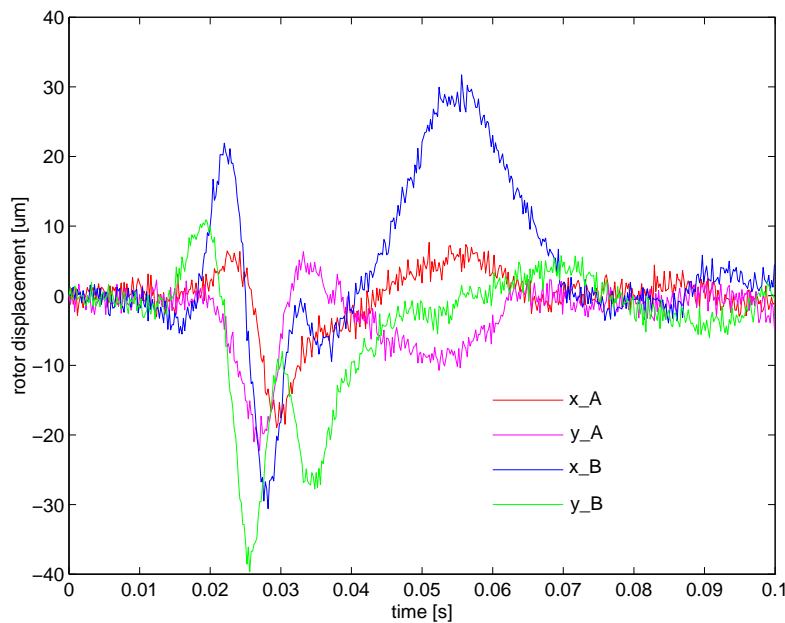


Figure 4.23: Time history of the rotor displacements after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (P -structured controller).

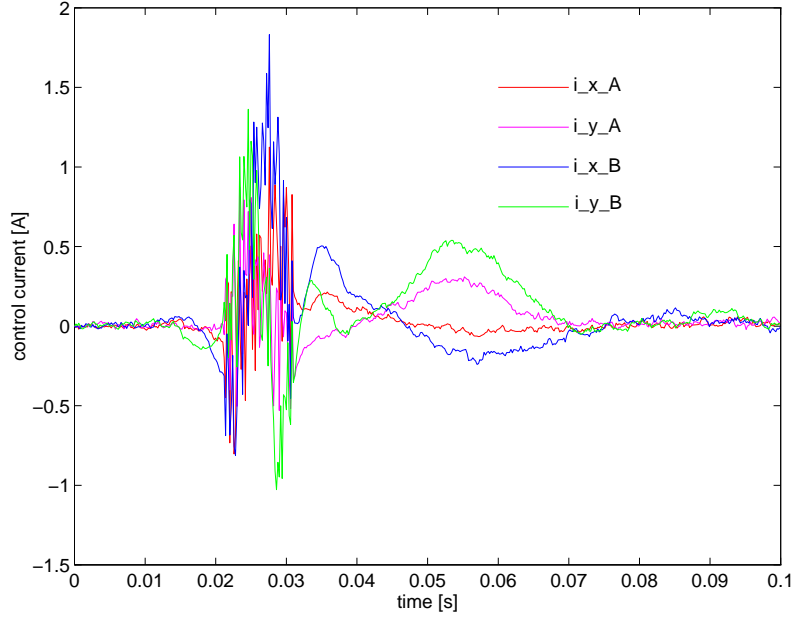


Figure 4.24: Time history of the control currents after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (P -structured controller).

In total $n_p = 96$ parameters are adapted, but only few of them change significantly. This is due to the fact that not every parameter is affected by the change of the non-conservative stiffness factor to the same extent. Note that the parameters in the parameter vector \mathbf{p} depend on discrete time state space matrices, and have neither a specific unit nor a unique physical meaning.

Investigations with parameter variations have shown, whether a physical parameter influences a parameter in the discrete time state space model. It turned out that one physical parameter can influence more discrete model parameters and vice versa. But the trace of certain parameters as for example rotor mass or the skew-symmetric non-conservative stiffness parameter could be followed. The rotor mass mainly affects the sub-matrices in the diagonal of the system matrix \mathbf{A} , i.e. the parameters a_m^{ii} with $m = 1, 2$ and $i = 1, 2, 3, 4$. The non-conservative stiffness parameter influences some non-symmetric entries of the the system matrix \mathbf{A} .

The parameter $\mathbf{p}(1)$, is equal to $-a_2^{11}$ of the system matrix \mathbf{A} and depends mainly on the rotor mass. Therefore, it is obvious that this parameter does not change significantly. The time history of this parameter can be seen in Fig. 4.26 in Fig. 4.25.

A second parameter, $\mathbf{p}(8)$ is equal to $-a_1^{14}$, a non-symmetric entry of the system matrix \mathbf{A} which is of course affected by the skew-symmetric non-conservative stiffness parameter. The time history of this parameter can be seen in Fig. 4.26.

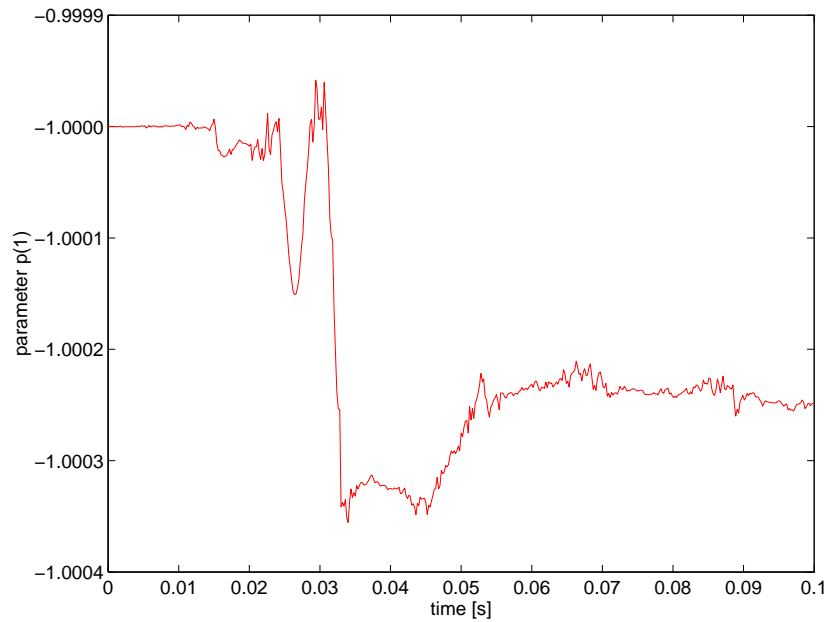


Figure 4.25: Time history of a sample system parameter $\mathbf{p}(1)$ after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (P -structured controller).

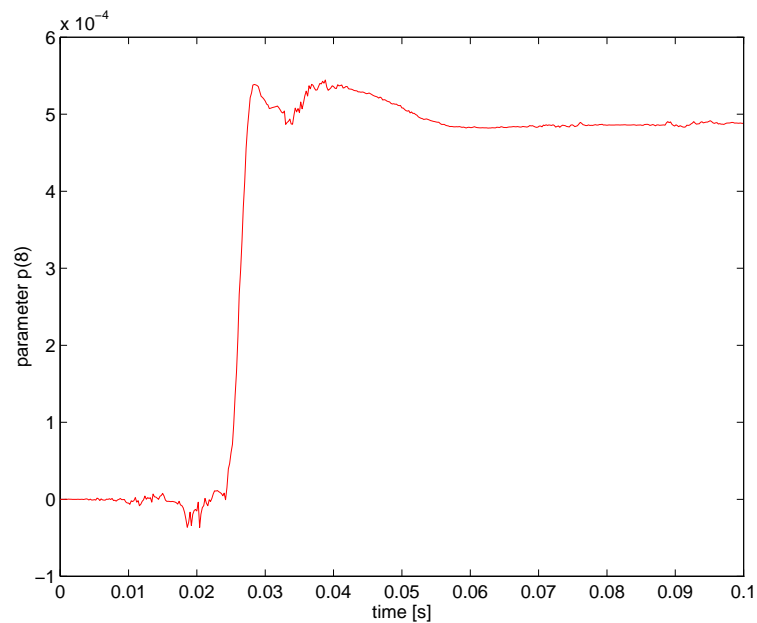


Figure 4.26: Time history of a sample system parameter $\mathbf{p}(8)$ after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (P -structured controller).

PI-structure

In this case, the closed loop system is controlled by a state space controller with additional integrative feedback. A parameter change at $t = 0.01$ s is then applied to the system by a sudden appearance of cross-coupling forces with a non-conservative stiffness factor of $k_n = 6 \cdot 10^6$ N/m.

The behaviour of the entire system under adaptive control is very similar to a system controlled by a simple state feedback controller. The parameter change is triggered a little later, because the stability region is larger for a system with integrative feedback as shown in Fig. 4.12. The history of the forgetting control factor and the plot of the forgetting parameter can be seen in Fig. 4.27. Additionally, a white noise signal

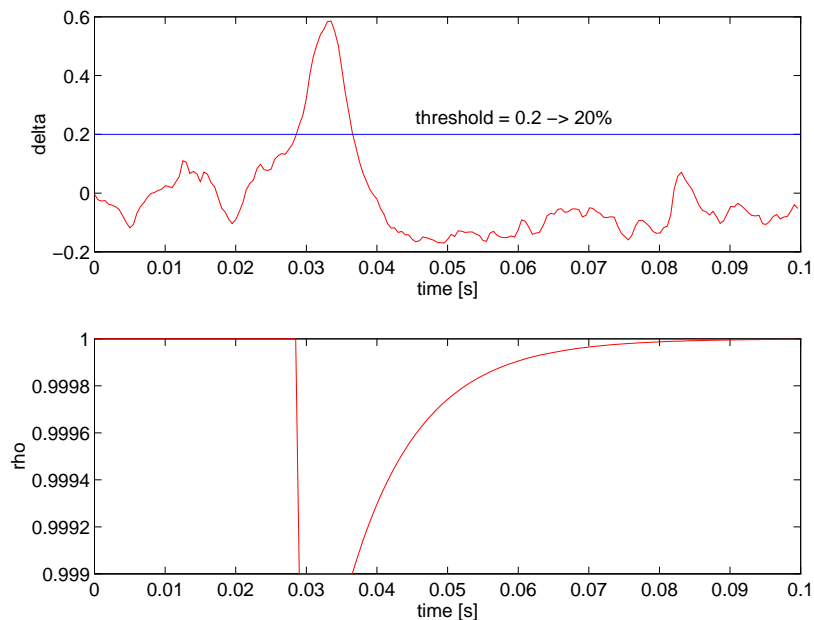


Figure 4.27: Time history of the forgetting control variable δ and the forgetting factor ρ . after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (PI-structured controller).

with a maximum deflection of $100 \mu\text{m}$ is added to the set point. This generates only a small ripple on the control variable, because the set point is not directly fed through the feed forward matrix \mathbf{K}_w , but passes the integrative loop, which is a low pass. Nevertheless, a control variable is generated which is not correlated to the estimation error generated by the measurement noise. The adaptation is performed similarly to the case with a simple state feedback controller and the rotor bearing system recovers from the parameter change. The time history of the rotor displacements and the control currents can be seen in Fig. 4.28 and Fig. 4.29, respectively. The time history of the strongly changing parameter $\mathbf{p}(8)$ can be seen in Fig. 4.30, the time history of the parameter $\mathbf{p}(1)$ with very little change can be seen in Fig. 4.31.

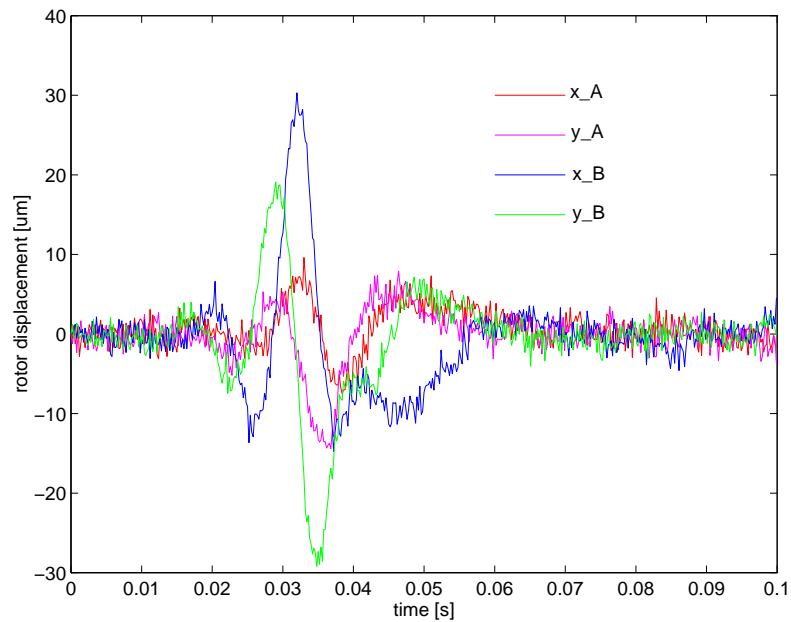


Figure 4.28: Time history of the rotor displacements after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (PI-structured controller).

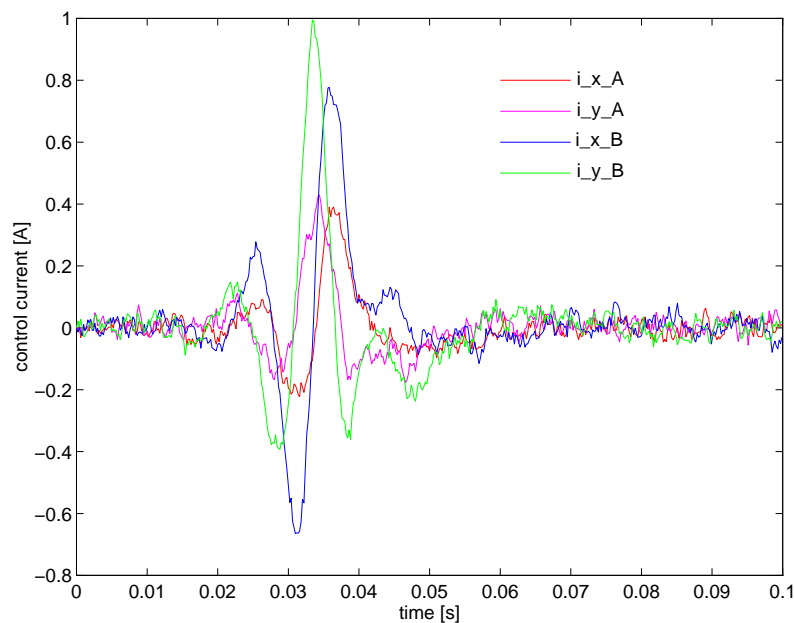


Figure 4.29: Time history of the control currents after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (PI-structured controller).

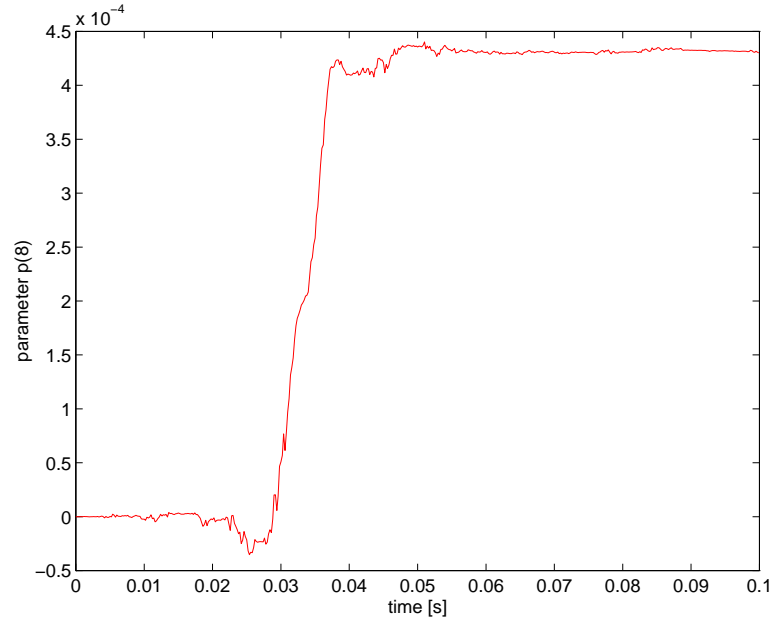


Figure 4.30: Time history of a sample system parameters $\mathbf{p}(8)$ after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (PI-structured controller).

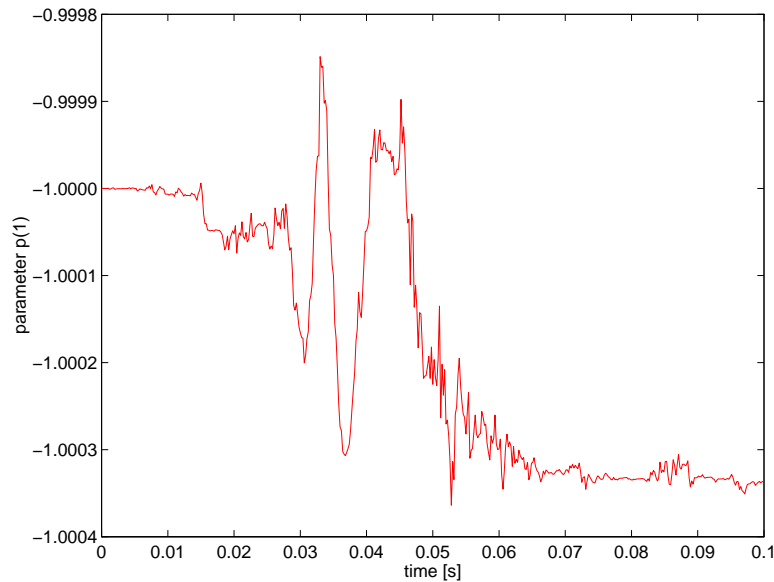


Figure 4.31: Time history of a sample system parameter $\mathbf{p}(1)$ after the appearance of a non-conservative cross-coupling stiffness of $k_n = 6 \cdot 10^6$ N/m at time $t = 0.01$ s (PI-structured controller).

4.5.4 Discussion of the numerical results

The problem involved with state space adaptive control for a rotor suspended in active magnetic bearings is the fast dynamic behaviour of the open loop plant, which is unstable in addition. Usually, control is applied to compensate disturbances or make the closed loop faster than the open loop system. For the rotor bearing system presented in this work, the closed loop eigenvalues have the same absolute values as the open loop ones. If the open loop system is slower than the closed loop system, there is more time to adapt the system. If the open loop system has a fast and unstable behaviour, the time for parameter adaptation is limited. In a pure linear case, this time restriction would infer no problems. If the system is bounded by saturation, however, the dynamical range underlies certain constraints. These are mainly the constraints for displacement and the control current. Although the adaptive control concept is limited in the range of parameter changes, two regions of positive operation remain.

Parameter change does not destabilise the closed loop system

If a parameter change does not destabilise the system, all closed loop system poles lie inside the unit circle. In this first case, the adaptation process lasts very long, which is not a problem as long as the system is stable. This is due to the fact that the output signal contains only little information on the process, because the entire system is only excited by the measurement noise corrupting the output. Moreover, the control variable is correlated with the output variable. An uncorrelated part of the control variable is needed in order to estimate the parameters of the system under investigation quickly and adapt to parameter changes. This requirement could be satisfied by set point changes. Since the main objective of the control for a rotor bearing system is the stabilisation and compensation of disturbances, set point changes are undesirable. Another possibility to excite the system would be by a white noise signal added to the zero set point with a larger amplitude than the measurement noise. This procedure, however, is not applicable, because unnecessary vibrations are generated.

Parameter change destabilises the closed loop system

The second case is that a parameter change destabilises the system. Here, the closed loop poles are moved outside the unit-circle in the z -plane, or into the right half complex plane in the continuous time domain. The new poles determine the speed of the destabilisation process, i.e. the time delay until the rotor touches the backup bearing or the amplifiers reaches saturation. This means that the adaptation algorithm has to adapt the system parameters and hence the controller parameters within a certain time and without making the controller run into saturation. Roughly speaking, the adaptation law has to move the closed loop poles back into the stability region, i.e. inside the unit circle.

The speed of the adaptation algorithm is determined by the value of the forgetting factor and the covariance of the parameters to be estimated. Of course, a large covariance covers many parameter changes for systems with slow dynamics. As for the

presented system, a large covariance destabilises the system, because unstable predictors are generated permanently, which violate the stability criterion for the adaptation procedure. Therefore, the covariance has to be kept rather small, which could be proved by means of simulation as well. Otherwise the entire system becomes unstable even if there is no parameter change.

The forgetting factor is the second parameter determining the adaptation speed in two ways. On the one hand, a low forgetting factor means that the parameters are considered to be false and the newly estimated are “more true”. Then the parameters can change quickly and adapt to the new situation, but generate unstable predictors as well, which destabilise the adaptive control loop in the end. On the other hand the covariance is increased fast if the forgetting factor is too low. This leads to the aforementioned problem of unstable predictors. What remains is a dilemma. Either the system is destabilised by parameter changes with the adaptation algorithm being too slow due to a small covariance, or by unstable predictors with the covariance being too large.

Whether the adaptation algorithm can follow a parameter change finally depends on the covariance and the forgetting factor within certain constraints:

- If the initial covariance is too large from the beginning ($\mathbf{P} > 10^{-6} \cdot \mathbf{I}$), unstable predictors result.
- If the covariance is small enough ($\mathbf{P} < 10^{-6} \cdot \mathbf{I}$), and if
 - the forgetting factor is too small ($\rho(k) \ll 0.999$) a large variance of the system parameters is caused which finally destabilises the system.
 - the forgetting factor is chosen properly ($\rho(k) \approx 0.999$), the adaptation algorithm can cope with the parameter change.
 - the forgetting factor is too large ($\rho(k) \approx 1$), the system runs into saturation before the algorithm adapts to the true parameters.

As a conclusion it can be pointed out that the algorithm for adaptive control presented in this work performs excellently for a certain range of parameter changes. If these are too large instability cannot be prevented.

Chapter 5

Conclusion

This laster chapter gives a short summary on what has been done in the present work followed by an outlook which focuses on the future work possible in the field of adaptive control for a rotor suspended by active magnetic bearings.

5.1 Summary

The main topic of the present work is the development of state space adaptive control for a rigid rotor suspended by active magnetic bearings. Therefore, the specific objectives were the modelling of a rotor bearing system, the derivation of a linear discrete time model for a chosen point of operation, the search for a proper algorithm which can cope with parameter changes of the controlled system, and the proof of the performance by means of simulation.

To start with, an experimental setup and the rotor model was presented with a rigid rotor suspended by two active magnetic bearings with four axes of control. Non-conservative cross-coupling forces, simulating a system change, were applied to the rotor in a given plane along the rotor axis. For simulation purposes a comprehensive nonlinear model including a rigid rotor, position and current sensors, analogue to digital converters, digital signal processor, digital to analogue converters, switching power amplifiers with pulse width modulators and magnetic actuators was established. Then, a linear continuous time model has been derived from the nonlinear model for a point of operation. For the sake of simplification, an internal current control loop was designed for the point of operation in order to reduce the order of the resulting linear system. Under these assumptions the magnetic actuator can be treated as a negative spring regarding the rotor position, and as a gain regarding the control current. From there a continuous time state space model for the rotor bearing system was derived and transformed into a discrete time state space model with its matrices in controller canonical form.

The estimation algorithm and the controller design were based on the state space model in controller canonical form. The prediction error algorithm was applied to an innovations model derived from the state space model in order to identify the model

parameters under on-line conditions. In this algorithm a state space model and all states have to be calculated after each sampling time interval. To provide numerical stability, a special implementation of this algorithm was used. Additionally, an effective algorithm was proposed to detect system parameter changes in order to start the adaptation process.

In order to justify the application of a linear state space controller for the position control loop of the rotor bearing system, a control current loop was tested by means of simulation with respect to its bandwidth. Using a PI-controller for this task the position stiffness and the current gain derived from the nonlinear model could be confirmed. Additionally, the bandwidth of the current control loop covers the frequency band for the position control loop. This loop was investigated by both step responses and responses to additional disturbances.

Adaptive control was performed via the design of a pole placement controller with and without integrative feedback based on the identified state space model. Simulations proofed the successful operation of the proposed algorithms. The sudden appearance of destabilising non-conservative cross-coupling forces was assumed in order to change the parameters of the system. The simulation results showed that the proposed identification algorithm can cope with parameter changes and the entire system can be stabilised even for high values of non-conservative stiffness coefficients.

5.2 Outlook

Since the adaptive control concept has been extensively investigated by means of simulation, an implementation within a real system is desirable. However, some numerical problems will be involved with that implementation for two reasons.

Firstly, the adaptive control concept includes a large number of matrix operations. The advantage is, that the computation time can be expected to be constant, because no iterations are involved with it. This was a primary constraint for the choice of the state space adaptation algorithm. Still, it will be a challenge for a digital controller to carry out all computation within a sample period of $100 \mu\text{s}$.

Secondly, within a simulation there are no limitations on the computation accuracy. In a digital signal processor one is limited to a system accuracy, which can deteriorate the performance.

Once the control concept is implemented for a real application, all design parameters of the adaptive control algorithm can be tuned, e.g. the initial covariance or the forgetting factor. If this is done, a stable state space control loop is the result which guarantees a certain closed loop performance over a wide range of plant parameters. Moreover, the control concept can be applied to other open loop unstable MIMO-systems.

Bibliography

- [Aling, 1990] Aling, H. (1990). *Identification of closed loop systems: identifiability, recursive algorithms and application to a power plant*. PhD thesis, Delft University of Technology, Delft.
- [Allaire et al., 1983] Allaire, P., Lewis, D., and Knight, J. (1983). Active vibration control of a single mass rotor on flexible supports. Department of Mechanical and Aerospace Engineering, University of Virginia.
- [Åström and Wittenmark, 1988] Åström, K. and Wittenmark, B. (1988). *Adaptive Control*. Addison-Wesley Publishing Company.
- [Basseville, 1983] Basseville, M. and Beneviste, A. (1983). Design and comparative study of some sequential jump detection algorithms for digital signals. *IEEE Transactions on Acoustic Speech and Signal Processing*, ASSP-31(3):521–535.
- [Basseville and Beneviste, 1983] Basseville, M. and Beneviste, A. (1983). Sequential detection of abrupt changes in spectral characteristics of digital signals. *IEEE Transactions on Information Theory*, IT-29(5):709–724.
- [Basseville and Beneviste, 1984] Basseville, M. and Beneviste, A. (1984). *Detection of Abrupt Changes in Signals and Dynamical Systems*, volume 77 of *Lecture Notes in Control and Information Sciences*, edited by M. Thoma and A. Wyner. Springer Verlag Berlin Heidelberg New York.
- [Bierman, 1988] Bierman, G. (1988). *Factorisation Methods for Discrete Sequential Estimation*. Academic Press, New York.
- [Borison, 1978] Borison, U. (1978). Self-tuning regulators for a class of multivariable systems. In *Identification and System Parameter Estimation*, pages 1857–1866. North Holland Publishing Company.
- [Brammer and Siffing, 1975] Brammer, K. and Siffing, G. (1975). *Stochastische Grundlagen des Kalman-Bucy-Filters*. Methoden der Regelungstechnik, herausgegeben von O. Föllinger und H. Sartorius. Oldenburg Verlag, München Wien.
- [Brammer and Siffing, 1985] Brammer, K. and Siffing, G. (1985). *Kalman-Bucy-Filter*. Methoden der Regelungstechnik, herausgegeben von O. Föllinger und H. Sartorius. Oldenburg Verlag, München Wien.

- [Bronstein, 1989] Bronstein, I. (1989). *Taschenbuch der Mathematik*. Teubner Verlagsgesellschaft, Leipzig.
- [Cellier, 1991] Cellier, F. (1991). *Continuous System Modeling*. Springer-Verlag New York Inc., New York Berlin Heidelberg.
- [Chu, 1987] Chu, Q. P. (1987). *Maximum likelihood parameter identification of flexible spacecraft*. PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering.
- [Föllinger, 1990] Föllinger, O. (1990). *Lineare Abtastsysteme*. Oldenburg Verlag, München Wien, 4 edition.
- [Gähler and Herzog, 1994] Gähler, C. and Herzog, R. (1994). Identification of magnetic bearing systems. In Schweitzer, G., Siegart, R., and Herzog, R., editors, *Proceedings to the Fourth International Symposium on Magnetic Bearings*, pages 293–298.
- [Gähler et al., 1996] Gähler, C., Mohler, M., and Herzog, R. (1996). Multivariable identification of active magnetic bearing system. In Matsumura, F., Okada, Y., Fujita, M., and Namerikawa, T., editors, *Proceedings to the Fifth International Symposium on Magnetic Bearings*, pages 7–12.
- [Goodwin and Sin, 1984] Goodwin, C. and Sin, K. (1984). *Adaptive Filtering Prediction and Control*. Information and System Science Series, edited by Kailath, T. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.
- [Haferl, 1991] Haferl, A. (1991). *Magnetisierungseffekte in aktiven Magnetlagern*. Diplomarbeit, Institut für Maschinendynamik und Meßtechnik, Technische Universität Wien.
- [Hanss and Estler, 1994] Hanss, M. and Estler, M. (1994). Anmerkungen zur Online-Parameterschätzung für die adaptive Regelung von Mehrgrößensystemen der Zustandsraumdarstellung. *at-Automatisierungstechnik*, 42:189–205.
- [Hensel, 1990] Hensel, H. (1990). *Methoden des rechnergestützten Entwurfs und Echtzeitsatzes zeitdiskreter Mehrgrößenregelungen und ihre Realisierung in einem CAD-System*, volume 4 of *Reihe 20: Rechnerunterstützte Verfahren*. VDI Verlag.
- [Herzog and Gähler, 1994] Herzog, R. and Gähler, C. (1994). Modelling and identification of flexible rotors in magnetic bearings. In *IMACS/MATHMOD Conference, Vienna*.
- [Hopkins, 1988] Hopkins, M. A. (1988). *Pseudo-Linear Identification: Optimal joint parameter and state estimation of linear stochastic MIMO systems*. PhD thesis, Virginia Polytechnic Institute and State University.
- [Hou, 1982] Hou, D. (1982). *State Space Model Identification of Unstable Multivariable Linear Systems*. PhD thesis, Washington State University.
- [Isermann, 1987a] Isermann, R. (1987a). *Digitale Regelsysteme, Band I*. Springer-Verlag, Berlin Heidelberg New York.

- [Isermann, 1987b] Isermann, R. (1987b). *Digitale Regelsysteme, Band II*. Springer-Verlag, Berlin Heidelberg New York.
- [Isermann, 1992a] Isermann, R. (1992a). *Identifikation dynamischer Systeme, Band I*. Springer-Verlag, Berlin Heidelberg New York.
- [Isermann, 1992b] Isermann, R. (1992b). *Identifikation dynamischer Systeme, Band II*. Springer-Verlag, Berlin Heidelberg New York.
- [Isermann et al., 1992] Isermann, R., Lachmann, K.-H., and Matko, D. (1992). *Adaptive Control Systems*. Systems and Control Engineering, edited by M.J. Grimble. Prentice Hall. ISBN 0-13-005414-3.
- [Jiang and Doraiswami, 1987] Jiang, J. and Doraiswami, R. (1987). Convergence analysis of least-squares identification algorithm for unstable systems. *IEEE Proceedings*, 134(5):301–308.
- [Jörgl, 1993] Jörgl, H. (1993). *Repetitorium Regelungstechnik, Band 1*. Oldenburg Verlag.
- [Jörgl, 1994] Jörgl, H. (1994). *Repetitorium Regelungstechnik, Band 2*. Oldenburg Verlag.
- [Keith et al., 1989] Keith, F., Williams, R., and Allaire, P. (1989). Digital control system design for active magnetic bearings. Department of Mechanical and Aerospace Engineering, University of Virginia.
- [Keuchel, 1988] Keuchel, U. (1988). *Methoden zur rechnergestützten Analyse und Synthese von Mehrgrößensystemen in Polynommatrizzendarstellung*. Dissertation, Ruhr Universität Bochum.
- [Keuchel, 1995] Keuchel, U. (1995). Ein neues Verfahren zur Parameterschätzung und Strukturselektion für lineare Mehrgrößensysteme der Zustandsraumdarstellung. *at-Automatisierungstechnik*, 43:458–466.
- [Kevicky and Hetthéssy, 1977] Kevicky, L. and Hetthéssy (1977). Self-tuning minimum variance control of MIMO discrete time systems. *Automatic Control Theory and Applications*, 5(1):11–17.
- [Knospe et al., 1995] Knospe, C., Hope, W., Fedigan, S., and Williams, R. (1995). Experiments in the control of unbalance response using magnetic bearings. *Mechatronics*, 5(4):385–400. Elsevier Science Ltd.
- [Koivo, 1980] Koivo, H. (1980). A multivariable self-tuning controller. *Automatica*, 16:351–366.
- [Kouvraritakis and Rossiter, 1992] Kouvraritakis, B. and Rossiter, J. (1992). Use of bicausal weighting sequences in least squares identification of open-loop unstable dynamic systems. *IEEE Proceedings*, 139(3):328–336.

- [Kumpati and Annaswamy, 1989] Kumpati, S. and Annaswamy, A. (1989). *Stable Adaptive Systems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.
- [Kuo, 1980] Kuo, B. (1980). *Digital Control Systems*. Series in Electrical and Computer Engineering, edited by M.E. Van Valkenburg. Holt, Rinehart and Winston, Inc., New York Chicago San Francisco.
- [Lang, 1997] Lang, O. (1997). *Vibration Control of a Self-Excited Rotor by Active Magnetic Bearings*. Dissertation, Institut für Maschinendynamik und Meßtechnik, Technische Universität Wien.
- [Lang et al., 1995a] Lang, O., Wassermann, J., and Springer, H. (1995a). Adaptive Regelung von zirkulatorischen Kräften in rotierenden Maschinen mit Magnetlagern. In Prof.Dr.-Ing.habil. R.Hampel, D.-I. F., editor, *Proceeding of the 2nd Zittau Workshop on Magnetic Bearing*, Theodor-Körner-Allee 16, D-02763 Zittau. Rektor der Hochschule für Technik, Wirtschaft und Sozialwesen Zittau/Görlitz.
- [Lang et al., 1995b] Lang, O., Wassermann, J., and Springer, H. (1995b). Adaptive vibration control of a rigid rotor supported by active magnetic bearings. In *International Gas Turbine and Aeroengine Congress and Exposition*, 345 E 47th St., New York, N.Y. 10017. The American Society of Mechanical Engineers.
- [Larsonneur, 1990] Larsonneur, R. (1990). *Design and control of active magnetic bearing systems for high speed rotation*. Dissertation, Thesis ETH No. 9140, Swiss Federal Institute of Technologie, Zürich, Switzerland.
- [Ljung and Söderström, 1983] Ljung, L. and Söderström, T. (1983). *Theory and Practice of Recursive Identification*. MIT Press, Cambridge Massachusetts.
- [Lottin and Saïdi, 1994] Lottin, J. and Saïdi, L. (1994). Identification of physical parameters in active magnetic bearing suspension. In Schweitzer, G., Siegwart, R., and Herzog, R., editors, *Proceedings of the Fourth International Symposium on Magnetic Bearings*, pages 299–304.
- [Lum et al., 1996] Lum, K., Coppola, V., and Bernstein, D. (1996). Adaptive auto-centering control for an active magnetic bearing supporting a rotor with unknown mass imbalance. *IFAC Proceedings*, 3b(13):415–420.
- [MATLAB, 1992] MATLAB (1992). *MATLAB - Reference Guide*. The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Massachusetts 01760.
- [Mo and Bayoumi, 1993] Mo, L. and Bayoumi, M. (1993). Adaptive pole placement control of MIMO systems. *IEEE Transactions on Automatic Control*, 38(6):967–970.
- [Nazaruddin, 1994] Nazaruddin, Y. (1994). *Adaptive Regelung von Ein- und Mehrgrößensystemen auf der Basis der Zustandsraumdarstellung*. Dissertation, Ruhr Universität Bochum.
- [Ng et al., 1977] Ng, T., Goodwin, G., and Anderson, B. (1977). Identifiability of MIMO linear dynamic systems operating in closed loop. *Automatica*, 13:477–485.

- [Nonami et al., 1994] Nonami, K., He, W., and Nishimura, H. (1994). Robust control of magnetic levitation systems by means of H_∞ control μ -Synthesis. *JSME International Journal, Ser. C.*, 37(3):513–520.
- [Nonami and Ito, 1994] Nonami, K. and Ito, T. (1994). μ -Synthesis of flexible rotor magnetic bearing system. In Schweitzer, G., Siegwart, R., and Herzog, R., editors, *Proceedings to the Fourth International Symposium on Magnetic Bearings*, pages 73–78.
- [Nour Eldin, 1976] Nour Eldin, H. (1976). Minimalrealisierung der Matrixübertragungsfunktion. *Regelungstechnik*, 1(3):82–87.
- [Nour Eldin and Heisters, 1980] Nour Eldin, H. and Heisters, M. (1980). Zwei neue Zustandsdarstellungsformen zur Gewinnung von Kroneckerindizes, Entkopplungsindizes und eines Prim-Matrix-Produktes, Teil 1. *Regelungstechnik*, 1(12):420–425.
- [Nour Eldin and Heisters, 1981] Nour Eldin, H. and Heisters, M. (1981). Zwei neue Zustandsdarstellungsformen zur Gewinnung von Kroneckerindizes, Entkopplungsindizes und eines Prim-Matrix-Produktes, Teil 2. *Regelungstechnik*, 1(1):26–30.
- [Ogata, 1987] Ogata, K. (1987). *Discrete Time Control Systems*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632.
- [Robbins and Monroe, 1951] Robbins, H. and Monroe, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407.
- [Salminen, 1992] Salminen, R. (1992). *Robust pole placement control of a trolley crane system*. Reports of the Automation Technology Laboratory: Series A, Research reports, Helsinki, Espoo.
- [Schlitt, 1992] Schlitt, H. (1992). *Systemtheorie für stochastische Prozesse*. Springer-Verlag, Berlin Heidelberg New York.
- [Schumann, 1982] Schumann, R. (1982). *Digitale Parameteradaptive Mehrgrößenregelung, Ein Beitrag zu Entwurf und Analyse*. Dissertation, Technische Fachhochschule Darmstadt.
- [Schweitzer et al., 1993] Schweitzer, G., Traxler, A., and Bleuler, H. (1993). *Magnetlager, Grundlagen, Eigenschaften und Anwendungen berührungsfreier, elektromagnetischer Lager*. Springer Verlag Berlin. ISBN 3-540-55868-3.
- [Siegwart et al., 1991] Siegwart, R., Vischer, D., Larssonneur, R., Herzog, R., Traxler, A., Bleuler, H., and Schweitzer, G. (1991). Control concepts for active magnetic bearings. In *Int.Symp. on Magnetic Susp.Techn.*, pages 401–422, Hampton, Va. NASA-Conf.Publ.
- [SIMULINK, 1992] SIMULINK (1992). *SIMULINK - Dynamic System Simulation Software*. The MathWorks, Inc., Cochituate Place, 24 Prime Park Way, Natick, Massachusetts 01760.

- [Sivrioglu et al., 1996] Sivrioglu, S., Nonami, K., and Ueyama, H. (1996). Advanced mixed H_2/H_∞ control design for active magnetic bearing system. In Matsumura, F., Okada, Y., Fujita, M., and Namerikawa, T., editors, *Proceedings to the Fifth International Symposium on Magnetic Bearings*, pages 159–164.
- [Springer, 1989] Springer, H. (1989). Nonlinearities and hysteresis effects in ferromagnetic circuits of electromagnetic bearings. In *ROMAC-Report UVA / 643092 / MAE89 / 405*. Department of Mechanical and Aerospace Engineering, University of Virginia.
- [Stephens and Knospe, 1996] Stephens, L. and Knospe, C. (1996). μ -Synthesis based, robust controller design for AMB machining spindle. In Matsumura, F., Okada, Y., Fujita, M., and Namerikawa, T., editors, *Proceedings to the Fifth International Symposium on Magnetic Bearings*, pages 153–158.
- [Tolle, 1983] Tolle, H. (1983). *Mehrgrößen-Regelkreissynthese, Grundlagen und Frequenzbereichsverfahren*. Oldenburg Verlag GmbH, München.
- [Tolle, 1985] Tolle, H. (1985). *Mehrgrößen-Regelkreissynthese, Entwurf im Zustandsraum*. Oldenburg Verlag GmbH, München.
- [Tsui and Chen, 1983] Tsui, C. and Chen, C. (1983). An algorithm for companion form realization. *International Journal of Control*, 38(4):769–777.
- [Ulbrich, 1993] Ulbrich, H. (1993). *Haben aktive Lagerungen Zukunft ? Schwingungen in Rotierenden Maschinen (SIRM'93)*, herausgegeben von H. Irretier, R. Nordmann und H. Springer. Vieweg Verlag.
- [Unbehauen and Nazaruddin, 1995] Unbehauen, H. and Nazaruddin, Y. (1995). Adaptive Zustandsregler für Mehrgrößensysteme und ihre praktische Anwendung der Zustandsraumdarstellung. *at-Automatisierungstechnik*, 43:236–241.
- [Wassermann, 1995] Wassermann, J. (1995). Konzept eines leistungsfähigen Multiprozessorsystems mit DSP's für die Magnetlagerforschung. In Prof. Dr.-Ing. habil. R. Hampel, D.-I. F. W., editor, *Proceeding of 2nd Zittau Workshop Magnetic Bearing*, Theodor-Körner-Allee 16, D-02763 Zittau. Rektor der Hochschule für Technik, Wirtschaft und Sozialwesen Zittau/Görlitz.
- [Wassermann and Springer, 1994] Wassermann, J. and Springer, H. (1994). A linear power amplifier with current injection (LACI) for magnetic bearings. In Schweitzer, G., Siegwart, R., and Herzog, R., editors, *Proceedings of the Fourth International Symposium on Magnetic Bearings*, pages 371–376.
- [Wurmsdobler, 1992] Wurmsdobler, P. (1992). Simulation eines nichtlinearen Modells für einen magnetgelagerten Mehrscheibenrotor. Diplomarbeit, Institut für Maschinendynamik und Meßtechnik, Technische Universität Wien.
- [Wurmsdobler, 1997] Wurmsdobler, P. (1997). *State Space Adaptive Control for a Rigid Rotor Suspended in Active Magnetic Bearings*. Dissertation, Institut für Maschinen- und Prozeßautomatisierung, Technische Universität Wien.

- [Wurmsdobler et al., 1996a] Wurmsdobler, P., Jörgl, H., and Springer, H. (1996a). State space adaptive control for a lumped mass rotor excited by nonconservative cross-coupling forces. In Childs, D., editor, *Proceedings of The Eight Workshop on Rotordynamic Instability Problems in High-Performance Turbomachinery*. Texas A&M University.
- [Wurmsdobler et al., 1996b] Wurmsdobler, P., Jörgl, H., and Springer, H. (1996b). State space adaptive control for a rigid rotor suspended in active magnetic bearings. In Matsumura, F., Okada, Y., Fujita, M., and Namerikawa, T., editors, *Proceedings to the Fifth International Symposium on Magnetic Bearings*, pages 185–190.
- [Zhu and Backx, 1993] Zhu, Y. and Backx, T. (1993). *Identification of Multivariable industrial Processes*. Advances in Industrial Control, edited by M.J. Grimble and M.A. Johnson. Springer-Verlag, Berlin Heidelberg New York.

Appendix A

Controller canonical form

Given the discrete time state space model determined by the system matrix Φ , its state vector \mathbf{x}_b , the control matrix Γ and the measurement matrix \mathbf{C} as presented in Section 2.3.5, the basis transformation Eqn.(2.88) (see [Tolle, 1985])

$$\mathbf{x} = \mathbf{T} \mathbf{x}_b \quad \text{and} \quad \mathbf{x}_b = \mathbf{T}^{-1} \mathbf{x} \quad (\text{A.1})$$

can be carried out using the transformation matrix

$$\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_N]^T, \quad (\text{A.2})$$

with $N = 8$ determining the order of the system with 4 subsystems of order $\nu_i = 2$ each. The new system matrices can be calculated via the transformation matrix \mathbf{T} as

$$\mathbf{A} = \mathbf{T} \Phi \mathbf{T}^{-1}, \quad (\text{A.3})$$

$$\mathbf{B} = \mathbf{T} \Gamma, \quad (\text{A.4})$$

$$\mathbf{C} = \mathbf{C} \mathbf{T}^{-1}. \quad (\text{A.5})$$

The desired structure of the system matrix in controller canonical form is according to Eqn.(2.94) and Eqn.(2.95)

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & \dots & 0 & 0 \\ -a_2^{(11)} & -a_1^{(11)} & \dots & -a_2^{(14)} & -a_1^{(14)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_2^{(41)} & -a_1^{(41)} & \dots & -a_2^{(44)} & -a_1^{(44)} \end{bmatrix}, \quad (\text{A.6})$$

and the structure of the control matrix in controller canonical form according to Eqn.(2.96) and Eqn.(2.97)

$$\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_4] = \begin{bmatrix} 0 & \dots & 0 \\ 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix}. \quad (\text{A.7})$$

Using the transformation matrix defined by Eqn.(A.2), equation $\mathbf{A} \mathbf{T} = \mathbf{T} \Phi$ can be rewritten for 4 subsystems as

$$\mathbf{t}_2^T = \mathbf{t}_1^T \Phi \quad (\text{A.8})$$

$$-a_2^{(11)} \mathbf{t}_1^T - a_1^{(11)} \mathbf{t}_2^T \dots - a_2^{(14)} \mathbf{t}_7^T - a_1^{(14)} \mathbf{t}_8^T = \mathbf{t}_2^T \Phi \quad (\text{A.9})$$

$$\vdots = \vdots$$

$$\mathbf{t}_8^T = \mathbf{t}_7^T \Phi \quad (\text{A.10})$$

$$-a_2^{(41)} \mathbf{t}_1^T - a_1^{(41)} \mathbf{t}_2^T \dots - a_2^{(44)} \mathbf{t}_7^T - a_1^{(44)} \mathbf{t}_8^T = \mathbf{t}_8^T \Phi \quad (\text{A.11})$$

or for all 4 independent vectors \mathbf{t}_1 , \mathbf{t}_3 , \mathbf{t}_5 and \mathbf{t}_7 in matrix form

$$\begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_7 \end{bmatrix} \cdot \begin{bmatrix} a_2^{(11)} + a_1^{(11)} \Phi + \Phi^2 & \dots & a_2^{(14)} + a_1^{(14)} \Phi \\ \vdots & \ddots & \vdots \\ a_2^{(41)} + a_1^{(41)} \Phi & \dots & a_2^{(44)} + a_1^{(44)} \Phi + \Phi^2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \quad (\text{A.12})$$

Using the Hamilton-Caley theorem [Tolle, 1985] leaves the determinant of the upper matrix zero and herewith the vectors \mathbf{t}_1 , \mathbf{t}_3 , \mathbf{t}_5 and \mathbf{t}_7 arbitrary. Therefore, the transformation matrix results in

$$\mathbf{T} = [\mathbf{t}_1, \Phi^T \mathbf{t}_1, \dots, \mathbf{t}_7, \Phi^T \mathbf{t}_7]^T. \quad (\text{A.13})$$

With $\mathbf{B} = \mathbf{T} \Gamma$ and the decomposition of the control matrix

$$\Gamma = [\gamma_1, \dots, \gamma_4] \quad (\text{A.14})$$

for all subsystems the following conditions hold

$$\mathbf{b}_1 = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_1^T \Phi \\ \vdots \\ \mathbf{t}_7^T \\ \mathbf{t}_7^T \Phi \end{bmatrix} \gamma_1, \quad \dots, \quad \mathbf{b}_4 = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_1^T \Phi \\ \vdots \\ \mathbf{t}_7^T \\ \mathbf{t}_7^T \Phi \end{bmatrix} \gamma_4 \quad (\text{A.15})$$

or rearranged

$$\mathbf{b}_1 = \mathbf{t}_1^T \underbrace{\begin{bmatrix} \gamma_1 \\ \Phi \gamma_1 \\ \vdots \\ \gamma_4 \\ \Phi \gamma_4 \end{bmatrix}}_{\mathbf{T}_c}, \quad \dots, \quad \mathbf{b}_4 = \mathbf{t}_4^T \begin{bmatrix} \gamma_1 \\ \Phi \gamma_1 \\ \vdots \\ \gamma_4 \\ \Phi \gamma_4 \end{bmatrix}. \quad (\text{A.16})$$

If all states are controllable and thus the matrix \mathbf{T}_c is regular, the vectors for the transformation can be calculated by

$$\mathbf{t}_1 = [\mathbf{T}_c^T]^{-1} \mathbf{b}_1, \quad (\text{A.17})$$

$$\vdots = \vdots$$

$$\mathbf{t}_4 = [\mathbf{T}_c^T]^{-1} \mathbf{b}_4. \quad (\text{A.18})$$

Appendix B

Derivation of matrices \mathbf{M}_k and \mathbf{V}_k

In Subsection 3.1.3 the matrices \mathbf{M}_k and \mathbf{V}_k have been defined in order to compute the gradient matrix Ψ recursively. A derivation of those matrices is given in the following.

The predictor based on the innovations model is introduced by Eqn.(3.3) and Eqn.(3.4) in Chapter 3 as

$$\hat{\mathbf{x}}(k+1, \mathbf{p}) = \mathbf{A}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}) + \mathbf{B} \mathbf{u}(k) + \mathbf{K}(\mathbf{p}) \varepsilon(k), \quad (\text{B.1})$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}(\mathbf{p}) \hat{\mathbf{x}}(k, \mathbf{p}), \quad (\text{B.2})$$

with its system matrices in controller canonical form as described in Chapter 2 and Appendix A as

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & \dots & 0 & 0 \\ -a_2^{(11)} & -a_1^{(11)} & \dots & -a_2^{(14)} & -a_1^{(14)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_2^{(41)} & -a_1^{(41)} & \dots & -a_2^{(44)} & -a_1^{(44)} \end{bmatrix}, \quad (\text{B.3})$$

$$\mathbf{B} = \begin{bmatrix} 0 & \dots & 0 \\ 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \\ 0 & \dots & 1 \end{bmatrix}, \quad (\text{B.4})$$

$$\mathbf{C} = \begin{bmatrix} c_2^{(11)} + c_1^{(11)} & \dots & c_2^{(14)} + c_1^{(14)} \\ \vdots & \ddots & \vdots \\ c_2^{(41)} + c_1^{(41)} & \dots & c_2^{(44)} + c_1^{(44)} \end{bmatrix}, \quad (\text{B.5})$$

and the Kalman matrix

$$\mathbf{K}(z) = \begin{bmatrix} k_2^{(11)} & \dots & k_2^{(14)} \\ k_1^{(11)} & \dots & k_1^{(14)} \\ \vdots & \ddots & \vdots \\ k_2^{(41)} & \dots & k_2^{(44)} \\ k_1^{(41)} & \dots & k_1^{(44)} \end{bmatrix}. \quad (\text{B.6})$$

This model yields 96 parameters to be estimated which are gathered within the parameter vector

$$\mathbf{p} = \begin{bmatrix} -a_2^{(11)}, -a_1^{(11)}, \dots, -a_2^{(14)}, -a_1^{(14)}, \dots, \\ -a_2^{(41)}, -a_1^{(41)}, \dots, -a_2^{(44)}, -a_1^{(44)}, \\ c_2^{(11)}, c_1^{(11)}, \dots, c_2^{(14)}, c_1^{(14)}, \dots, \\ c_2^{(41)}, c_1^{(41)}, \dots, c_2^{(44)}, c_1^{(44)}, \\ k_2^{(11)}, k_1^{(11)}, \dots, k_2^{(14)}, k_1^{(14)}, \dots, \\ k_2^{(41)}, k_1^{(41)}, \dots, k_2^{(44)}, k_1^{(44)} \end{bmatrix}. \quad (\text{B.7})$$

The parameter vector simply contains each second row of the system matrix \mathbf{A} i.e. only $\frac{n_s^2}{2} = 32$ parameters, and all rows of the measurement matrix \mathbf{C} and the Kalman matrix \mathbf{K} , both having no particular structure.

B.1 Derivation of the matrix \mathbf{M}_k

With the definition of the parameter vector the partial derivative \mathbf{M}_k defined in Eqn.(3.38) is

$$\mathbf{M}_k(\hat{\mathbf{p}}, \mathbf{x}, \boldsymbol{\varepsilon}) = \left. \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{x}}(k+1) \right|_{\mathbf{p}=\hat{\mathbf{p}}} = \left. \frac{\partial}{\partial \mathbf{p}} [\mathbf{A}(\mathbf{p}) \mathbf{x} + \mathbf{B} \mathbf{u} + \mathbf{K}(\mathbf{p}) \boldsymbol{\varepsilon}] \right|_{\mathbf{p}=\hat{\mathbf{p}}}. \quad (\text{B.8})$$

Using the innovations model yields

$$\begin{aligned} \hat{x}_1(k+1) &= \hat{x}_2(k) + k_2^{(11)} \varepsilon_1(k) + \dots + k_2^{(14)} \varepsilon_4(k), \\ \hat{x}_2(k+1) &= -a_2^{(11)} \hat{x}_1(k) - a_1^{(11)} \hat{x}_2(k) \dots \\ &\quad -a_2^{(14)} \hat{x}_7(k) - a_1^{(14)} \hat{x}_8(k) \\ &\quad + k_1^{(11)} \varepsilon_1(k) + \dots + k_1^{(14)} \varepsilon_4(k), \\ &\quad \vdots \\ \hat{x}_7(k+1) &= \hat{x}_8(k) + k_2^{(41)} \varepsilon_1(k) + \dots + k_2^{(44)} \varepsilon_4(k), \\ \hat{x}_8(k+1) &= -a_2^{(41)} \hat{x}_1(k) - a_1^{(41)} \hat{x}_2(k) \dots \\ &\quad -a_2^{(44)} \hat{x}_7(k) - a_1^{(44)} \hat{x}_8(k) \\ &\quad + k_1^{(41)} \varepsilon_1(k) \dots + k_1^{(44)} \varepsilon_4(k). \end{aligned} \quad (\text{B.9})$$

Differentiating the above equations with respect to the parameter vector \mathbf{p} results in the matrix

$$\mathbf{M}_k = \left. \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{x}}(k+1) \right|_{\mathbf{p}=\hat{\mathbf{p}}} = [\mathbf{M}_x, \mathbf{0}^{8 \times 32}, \mathbf{M}_\varepsilon] \quad (\text{B.10})$$

with the 8×32 -dimensional matrices

$$\mathbf{M}_x = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{0} \\ \hat{\mathbf{x}}^T(k) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \dots & \hat{\mathbf{x}}^T(k) \end{bmatrix} \quad \text{and} \quad (\text{B.11})$$

$$\mathbf{M}_\varepsilon = \begin{bmatrix} \varepsilon(k) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \varepsilon(k) & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \varepsilon(k) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \varepsilon(k) \end{bmatrix}. \quad (\text{B.12})$$

Note that the matrix \mathbf{M}_x does not depend on $\mathbf{u}(k)$, because the model is used in controller canonical form.

B.2 Derivation of the matrix \mathbf{V}_k

The partial derivative of the expected output, \mathbf{V}_k , defined in Eqn.(3.31) is

$$\mathbf{V}_k(\hat{\mathbf{p}}, \mathbf{x}) = \left. \frac{\partial}{\partial \mathbf{p}} \hat{\mathbf{y}}(k) \right|_{\mathbf{p}=\hat{\mathbf{p}}} = \left. \frac{\partial}{\partial \mathbf{p}} [\mathbf{C}(\mathbf{p}) \mathbf{x}] \right|_{\mathbf{p}=\hat{\mathbf{p}}}. \quad (\text{B.13})$$

Using the innovations model yields

$$\begin{aligned} \hat{y}_1(k) &= c_2^{(11)} \hat{x}_1(k) + c_1^{(11)} \hat{x}_2(k) + \dots \\ &\quad c_2^{(14)} \hat{x}_7(k) + c_1^{(14)} \hat{x}_8(k), \\ &\quad \vdots \\ \hat{y}_4(k) &= c_2^{(41)} \hat{x}_1(k) + c_1^{(41)} \hat{x}_2(k) + \dots \\ &\quad c_2^{(44)} \hat{x}_7(k) + c_1^{(44)} \hat{x}_8(k). \end{aligned} \quad (\text{B.14})$$

Differentiating the above equations with respect to the parameter vector \mathbf{p} results in the matrix

$$\mathbf{V}_k = [\mathbf{0}^{4 \times 32}, \mathbf{V}_x, \mathbf{0}^{4 \times 32}], \quad (\text{B.15})$$

with

$$\mathbf{V}_x = \begin{bmatrix} \hat{\mathbf{x}}^T(k) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \hat{\mathbf{x}}^T(k) \end{bmatrix} \quad (\text{B.16})$$

The matrices \mathbf{M}_k and \mathbf{V}_k are static in their structure, but have to be updated with the estimated states $\hat{\mathbf{x}}^T(k)$ and the prediction error $\varepsilon(k)$ each sample interval.

Appendix C

Derivation of PI-controller

In Section 3.2 several derivations are implied which are presented in the following sections.

C.1 Closed loop transfer function

The output of the system controlled by a state feedback controller is according to Eqn.(3.83)

$$\mathbf{Y}(z) = \mathbf{G}_{cl}(z) \mathbf{U}_I(z) \quad (\text{C.1})$$

with the transfer function matrix for the closed loop with simple state feedback

$$\mathbf{G}_{cl}(z) = \mathbf{C} \left(z \mathbf{I} - \mathbf{A} + \mathbf{B} \mathbf{K}_x \right)^{-1} \mathbf{B}. \quad (\text{C.2})$$

Since the model is used in controller canonical form, the matrix \mathbf{B} consists mainly of zeros and ones (see Appendix A). Thus, the the closed loop transfer function \mathbf{G}_{cl} can be written with the definitions for the state space controller defined by Eqn.(3.70) and Eqn.(3.71) as

$$\mathbf{G}_{cl} = \begin{bmatrix} \frac{c_2^{(11)} + c_1^{(11)} z}{z^2 + p_1^{(11)} z + p_2^{(11)}} & \cdots & \frac{c_2^{(14)} + c_1^{(14)} z}{z^2 + p_1^{(11)} z + p_2^{(11)}} \\ \vdots & \ddots & \vdots \\ \frac{c_2^{(41)} + c_1^{(41)} z}{z^2 + p_1^{(44)} z + p_2^{(44)}} & \cdots & \frac{c_2^{(44)} + c_1^{(44)} z}{z^2 + p_1^{(44)} z + p_2^{(44)}} \end{bmatrix}. \quad (\text{C.3})$$

This transfer function matrix can be put as a product of two transfer function matrices such that

$$\mathbf{G}_{cl} = \mathbf{N}(z) \mathbf{D}^{-1}(z) \quad (\text{C.4})$$

with matrix $\mathbf{D}(z)$ as a diagonal matrix

$$\mathbf{D}(z) = \begin{bmatrix} z^2 + p_1^{(11)} z + p_2^{(11)} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & z^2 + p_1^{(44)} z + p_2^{(44)} \end{bmatrix}, \quad (\text{C.5})$$

and $\mathbf{N}(z)$ depending on the entries of matrix \mathbf{C} as

$$\mathbf{N}(z) = \begin{bmatrix} c_2^{(11)} + c_1^{(11)} z & \dots & c_2^{(14)} + c_1^{(14)} z \\ \vdots & \ddots & \vdots \\ c_2^{(41)} + c_1^{(41)} z & \dots & c_2^{(44)} + c_1^{(44)} z \end{bmatrix}. \quad (\text{C.6})$$

Using this convention, the output of the system controlled by a state feedback controller is

$$\mathbf{Y}(z) = \mathbf{N}(z) \mathbf{D}^{-1}(z) \mathbf{U}_I(z) \quad (\text{C.7})$$

and with

$$\mathbf{U}_I(z) = \frac{1}{z-1} \mathbf{K}_I \mathbf{E}(z), \quad (\text{C.8})$$

$$\mathbf{E}(z) = \mathbf{W}(z) - \mathbf{Y}(z), \quad (\text{C.9})$$

the output finally depends only on the set-point error with

$$\mathbf{Y}(z) = \mathbf{N}(z) \mathbf{D}^{-1}(z) \frac{1}{z-1} \mathbf{K}_I (\mathbf{W}(z) - \mathbf{Y}(z)). \quad (\text{C.10})$$

Solving Eqn.(C.10) for $\mathbf{Y}(z)$ yields

$$\mathbf{Y}(z) = \mathbf{G}_{cl_I}(z) \mathbf{W}(z) \quad (\text{C.11})$$

with the transfer function matrix

$$\mathbf{G}_{cl_I}(z) = \left(\mathbf{I} + \mathbf{N} \mathbf{D}^{-1} \frac{1}{z-1} \mathbf{K}_I \right)^{-1} \mathbf{N} \mathbf{D}^{-1} \frac{1}{z-1} \mathbf{K}_I \quad (\text{C.12})$$

The detailed derivation necessary in order to gain Eqn.(3.89) is presented in the following section.

C.2 Simplification of closed loop transfer function

Equation Eqn.(C.12) or Eqn.(3.89) can be simplified using the matrix inversion lemma with general matrices \mathbf{X} and \mathbf{Y}

$$\left(\mathbf{X} \mathbf{Y} \right)^{-1} = \mathbf{Y}^{-1} \mathbf{X}^{-1} \quad (\text{C.13})$$

and vice versa as

$$\mathbf{G}_{cl_I}(z) = \left(\mathbf{I} + \mathbf{N} \mathbf{D}^{-1} \frac{1}{z-1} \mathbf{K}_I \right)^{-1} \mathbf{N} \mathbf{D}^{-1} \frac{1}{z-1} \mathbf{K}_I, \quad (\text{C.14})$$

$$= \left(\mathbf{I}(z-1) + \mathbf{N} \mathbf{D}^{-1} \mathbf{K}_I \right)^{-1} \mathbf{N} \mathbf{D}^{-1} \mathbf{K}_I, \quad (\text{C.15})$$

$$= \left(\mathbf{N} \left(\mathbf{N}^{-1}(z-1) + \mathbf{D}^{-1} \mathbf{K}_I \right) \right)^{-1} \mathbf{N} \mathbf{D}^{-1} \mathbf{K}_I, \quad (\text{C.16})$$

$$= \left(\mathbf{N}^{-1}(z-1) + \mathbf{D}^{-1} \mathbf{K}_I \right)^{-1} \mathbf{N}^{-1} \mathbf{N} \mathbf{D}^{-1} \mathbf{K}_I, \quad (\text{C.17})$$

$$= \left(\mathbf{N}^{-1}(z-1) + \mathbf{D}^{-1} \mathbf{K}_I \right)^{-1} \mathbf{D}^{-1} \mathbf{K}_I, \quad (\text{C.18})$$

$$= \left(\mathbf{D} \left(\mathbf{N}^{-1}(z-1) + \mathbf{D}^{-1} \mathbf{K}_I \right) \right)^{-1} \mathbf{K}_I, \quad (\text{C.19})$$

$$= \left(\mathbf{D} \mathbf{N}^{-1}(z-1) + \mathbf{D} \mathbf{D}^{-1} \mathbf{K}_I \right)^{-1} \mathbf{K}_I, \quad (\text{C.20})$$

$$= \left(\mathbf{D} \mathbf{N}^{-1}(z-1) + \mathbf{K}_I \right)^{-1} \mathbf{K}_I, \quad (\text{C.21})$$

$$= \left(\left(\mathbf{D}(z-1) + \mathbf{K}_I \mathbf{N} \right) \mathbf{N}^{-1} \right)^{-1} \mathbf{K}_I, \quad (\text{C.22})$$

$$= \mathbf{N} \left(\mathbf{D}(z-1) + \mathbf{K}_I \mathbf{N} \right)^{-1} \mathbf{K}_I \quad (\text{C.23})$$

$$= \mathbf{N} \mathbf{D}_I^{-1} \mathbf{K}_I \quad (\text{C.24})$$

with

$$\mathbf{D}_I = \mathbf{D}(z-1) + \mathbf{K}_I \mathbf{N}. \quad (\text{C.25})$$

C.3 Controller computation for PI-structure

Similarly to [Nazaruddin, 1994] the controller computation for PI-structure is derived in the following for the rotor bearing system.

The desired polynomial matrix for the closed loop system with integrative feedback is

$$\mathbf{P}_I = \begin{bmatrix} z^3 + p_{I_1}^{(11)} z^2 + p_{I_2}^{(11)} z + p_{I_3}^{(11)} & & & \mathbf{0} \\ & \ddots & & \\ & & z^3 + p_{I_1}^{(44)} z^2 + p_{I_2}^{(44)} z + p_{I_3}^{(44)} & \\ \mathbf{0} & & & \end{bmatrix}. \quad (\text{C.26})$$

After the matrix gain has been found as

$$\mathbf{K}_I = \mathbf{P}_I(z) \mathbf{N}^{-1}(z) \Big|_{z=1} \quad (\text{C.27})$$

with the controller gain matrix for the integrative feedback

$$\mathbf{K}_I = \begin{bmatrix} k_I^{(11)} & \dots & k_I^{(14)} \\ \vdots & \ddots & \vdots \\ k_I^{(41)} & \dots & k_I^{(44)} \end{bmatrix}. \quad (\text{C.28})$$

The equation for the polynomial matrix \mathbf{D} has to be solved as

$$\mathbf{P}_I \stackrel{\dagger}{=} \mathbf{D}_I = (z - 1) \mathbf{D} + \mathbf{K}_I \mathbf{N}. \quad (\text{C.29})$$

Using the previous definitions for all matrices, the diagonal entries of the closed loop polynomial matrix collected in positive powers of z are

$$\begin{aligned} \mathbf{D}_I^{(ii)} &= z^3 + \left(p_1^{(ii)} - 1 \right) z^2 \\ &\quad + \left(\sum_{j=1}^4 c_2^{ji} k_w^{(ij)} - p_1^{(ii)} + p_2^{(ii)} \right) z \\ &\quad + \left(\sum_{j=1}^4 c_2^{ji} k_w^{(ij)} - p_2^{(ii)} \right) \\ &= z^3 + p_{I_1}^{(ii)} z^2 + p_{I_2}^{(ii)} z + p_{I_3}^{(ii)} \end{aligned} \quad (\text{C.30})$$

Comparing these diagonal polynomials to the desired ones in \mathbf{P}_I , yields

$$p_1^{ii} = p_{I_1}^{ii} + 1, \quad (\text{C.31})$$

$$p_2^{ii} = \sum_{j=1}^4 c_2^{ji} k_w^{(ij)} - p_{I_3}^{ii}. \quad (\text{C.32})$$

Note that the solution is unique, although only two parameters were calculated from three equations. If the solution is derived from the equation with z^2 and z^0 the expression with z^1 turns out to be redundant. The non-diagonal elements in \mathbf{D}_I are zero according to Eqn.(C.27).

Appendix D

Simulation tools

The entire simulation was carried out with MATLAB/SIMULINK, a comfortable engineering tool for numerical calculations and simulations. The plant model was built using SIMULINK standard blocks from its library in a hierarchical manner as it is usual for SIMULINK. Separately, the control algorithm was coded in MATLAB for two reasons. Firstly, it seems to be more realistic to simulate a DSP by a function with the same inputs and outputs as in a real time application. Secondly, a sophisticated algorithm can be coded in MATLAB very easily, and can later on be translated into C-code with MATLAB functions. Therefore, the first section shows all SIMULINK block diagrams used, the second section presents the data file and all other functions which are part of the algorithm.

D.1 Simulink block diagrams

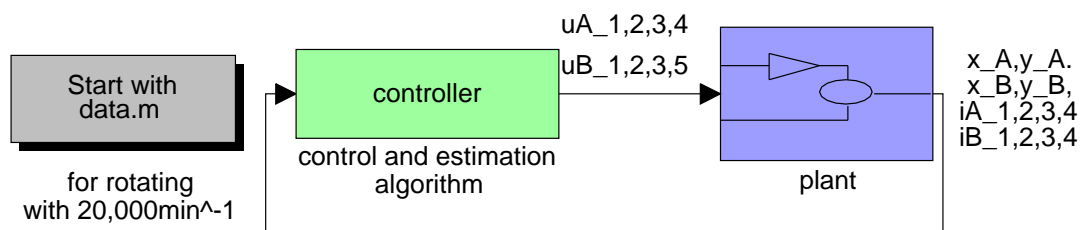


Figure D.1: Block diagram of the entire simulation loop with plant and the SIMULINK s-function “control and estimation algorithm”.

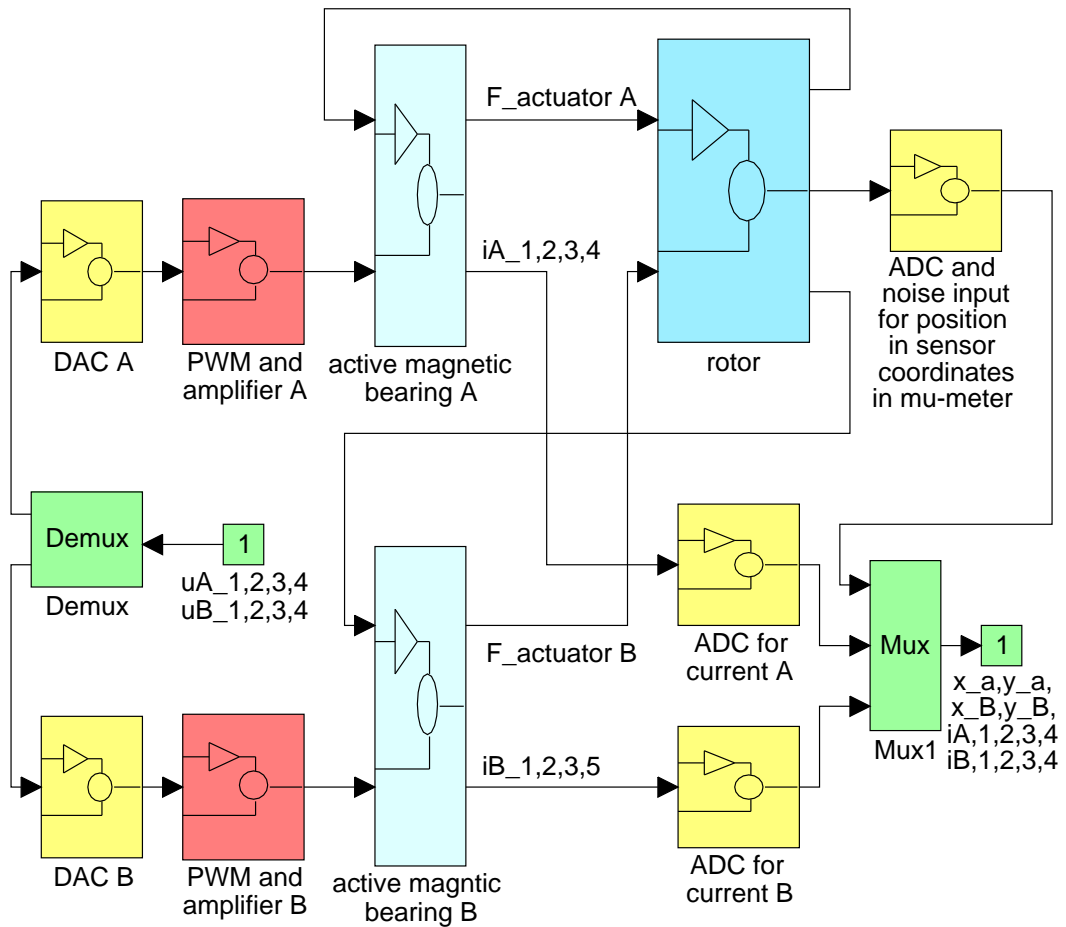


Figure D.2: SIMULINK block diagram of the plant with the elements controller, DAC, PWM with amplifier, actuators, rotor, and ADCs.

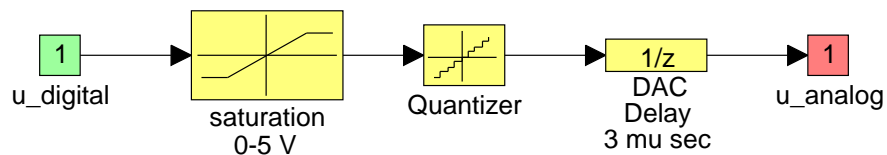


Figure D.3: One DAC for the output voltage as an example for all 8 channels.

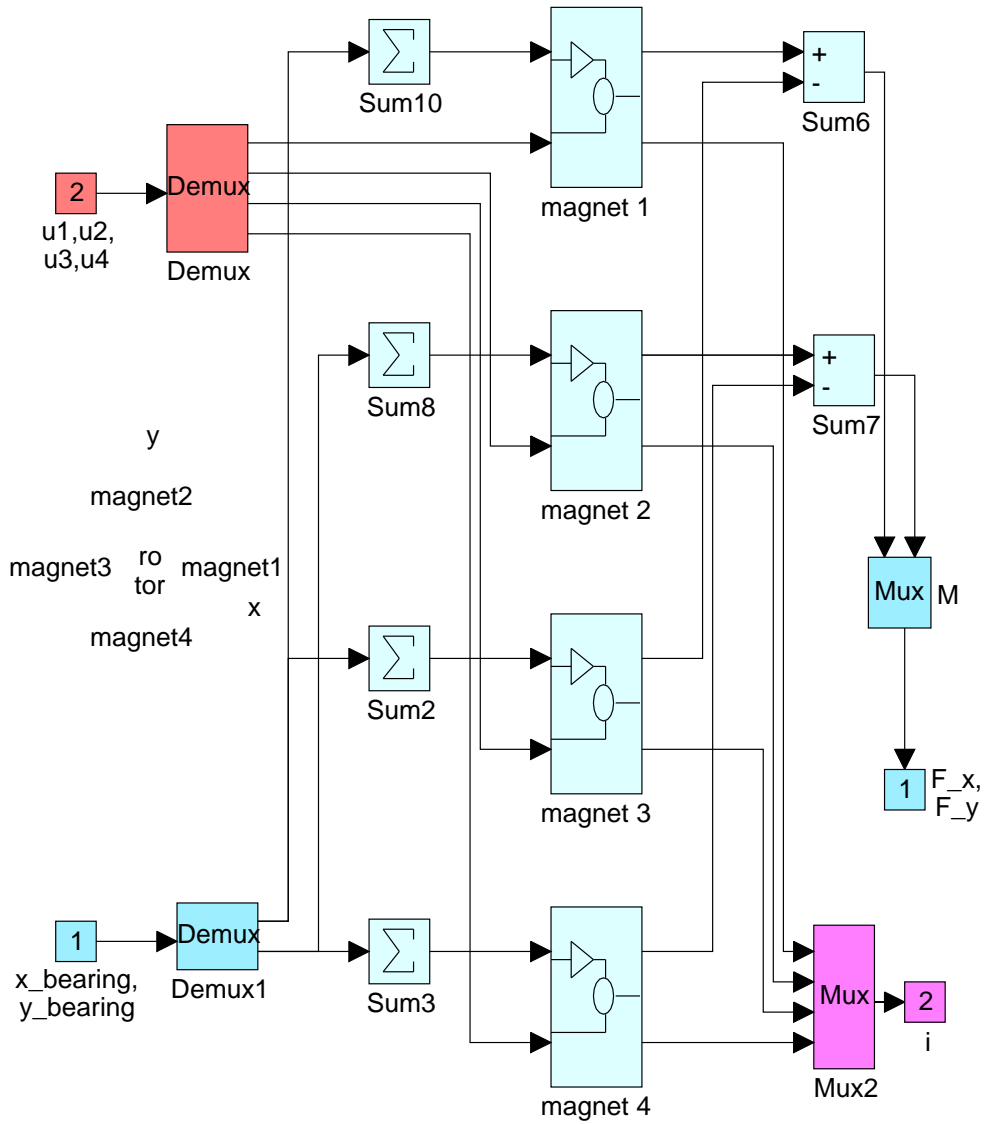


Figure D.4: An active magnetic bearing with four magnets.

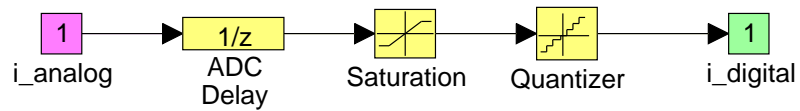


Figure D.5: One ADC for the current sampling as an example for all 8 channels.

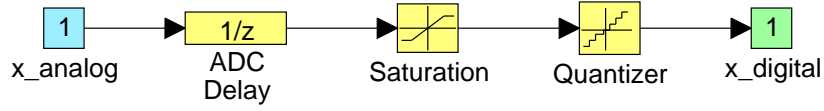


Figure D.8: One ADC for the position sampling as example for all 8 channels.

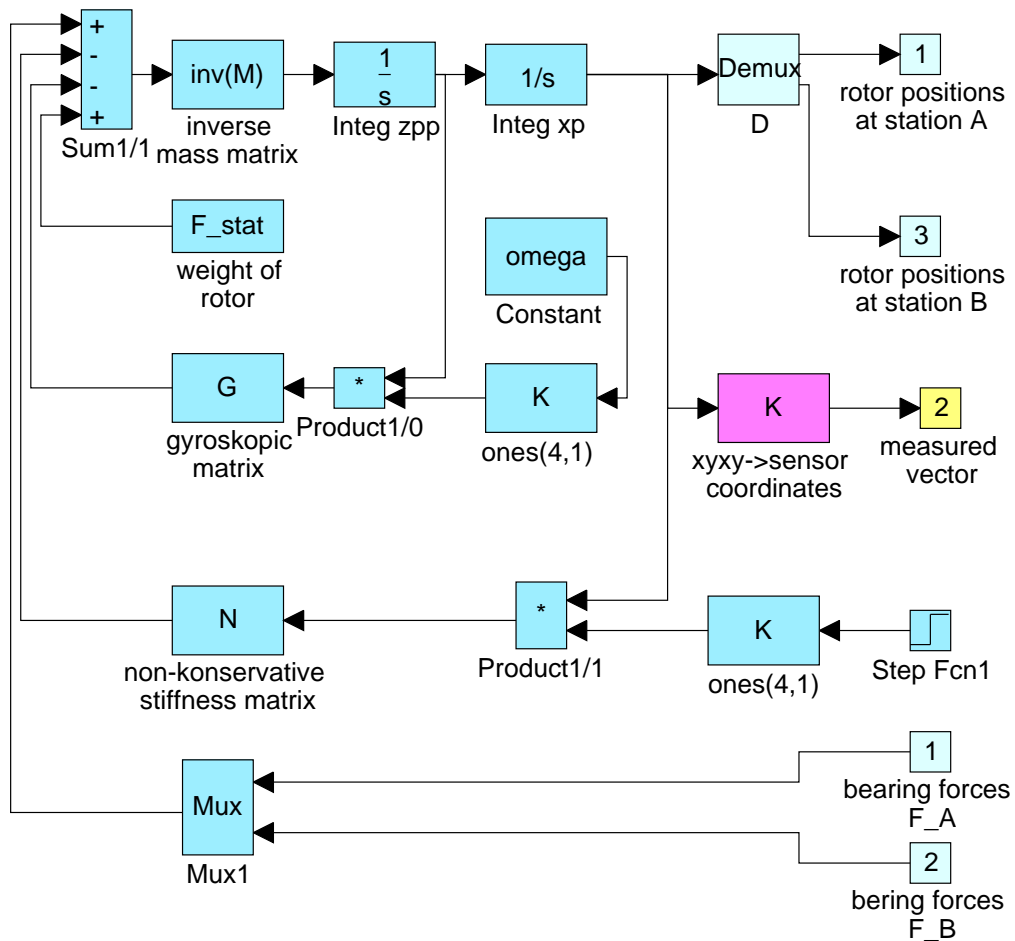


Figure D.9: SIMULINK model of the rigid rotor.

D.2 MATLAB code

All computations necessary for the model identification adaptive control algorithm were coded in the MATLAB programming language for the reasons mentioned above. Of course, the resulting code is not very efficient with respect to computation time and DSP resources, because the readability is more important during the developing stage and computation time is no issue within a simulation. For a real implementation in C or assembler code, the presented code has to be optimised.

D.2.1 Data file

The following data file contains all information necessary for a simulation run. Within this file the current controller is defined, the coefficients for the active magnetic bearing are calculated and the linear state space model is derived in controller canonical form.

```
%
%      data.m
%
%      date file used for rotor simulation

%
%      general
%
fprintf('\n\n Loading data.m');
%
IDACflag  = 1;    % flag for adaptive control
integflag = 0;    % flag for integrative feedback
excitflag = 1;    % flag for excitation
loadflag  = 0;    % flag for additional load
weightflag = 1;  % flag for rotor weight
rotflag   = 1;    % flag for rotation
setflag   = 0;    % flag for set point

t0        = 0;    % [s] simulation start time
tterm     = 1E-2; % [s] simulation termination time
tstep     = 1E-1; % [s] time of step

%
% Integration
%
Tint      = 1E-6; % [s] integration time
Tcomp     = 4E-6; % [s] computation delay
Tsamp     = 1E-4; % [s] sample time

%
% globals
%
global invTl2s R_max n_in n_out order index k_m % model
global Ahat Bhat Chat Ac0 Bc0 Cc0 % estimated model
global Kt Kw Ki Khat % controller
global U Y Xhat Yhat Ui u_act e_act epsilon % controller states
```

```

global par d_par Dp0 Dp Pp Up Dpn Upn Psi Wk      % estimation algorithm
global k_rho rho rho_0 rho_inf maxnoise         % forgetting factor control
global sigmaN_1 sigmaN_2 delta N_1 N_2         % forgetting factor control
global threshold minvar maxvar                 % forgetting factor control
global trigger Dp0
global i0A i0B u0A u0B nn_i U_DAC_i n_mag      % current controller
global I_max U_max u_act e_act                 % current controller
global IDACflag      integflag setflag        % general
global U_s X_s Y_s                                       % save variables
global sigma_s delta_s rho_s                                       % save variables
global par_s det_s Kt_s Ki_s                                       % save variables
global u_s i_s t_s t0 tterm tstep                                       % save variables

%
%      system modelling
%
% electro mechanical part
%
mu_0      = 4*pi*1E-7;      % [Vs/Am] permeability iv vacuum
mu_r_s    = 2000;          % [-] relative stator permeability
mu_r_r    = 1000;          % [-] relative rotor permeability
l_mag_s   = 84*1E-3;       % [m] length of magnetic path in stator
l_mag_r   = 20*1E-3;       % [m] length of magnetic path in rotor
A_mag_s   = 700*1E-6;      % [m^2] cross section area of length of
% magnetic path in stator
A_mag_r   = 420*1E-6;      % [m^2] cross section area of length of
% magnetic path in rotor
B_max_s   = 1.3;           % [T] maximal induction in stator
B_max_r   = 1.95;          % [T] maximal induction in rotor
A_mag_l   = A_mag_s;       % [m^2] cross section area of length of
% magnetic path in air
alpha     = pi/4;          % [rad] angle between pole shoes
R_0hm     = 0.8;           % [Ohm] ohmic resistance
N_mag     = 2*65;          % [-] number of windings = 2 poles
n_mag     = 4;             % [-] number of magnets
l0        = 0.5*1E-3;      % [m] nominal air gap
i00       = 4;             % [A] bias current
u0        = i00*R_0hm;     % [V] bias voltage
I_max     = 8;             % [A] maximum current

%
% constants for active magnetic bearing
% according to chapter 2
%
R_mag0    = ( l_mag_s/(mu_r_s*A_mag_s) ...
+ l_mag_r/(mu_r_r*A_mag_r) ...
+ 2*l0/A_mag_s)/mu_0;      % magnetic reluctance
L_mag0    = N_mag^2/R_mag0; % inductivity
Phi00     = N_mag*i00/R_mag0; % magnetic flux for bias current
Khiphiphi = R_0hm*R_mag0/(N_mag^2);
Khipl     = -2*R_0hm*Phi00/((N_mag^2)*mu_0*A_mag_l);
Khipu     = 1/N_mag;
KFphi     = 2*Phi00*cos(alpha/2)/(mu_0*A_mag_l);
Kiphi     = R_mag0/N_mag;
Kil       = 2*Phi00/(N_mag*mu_0*A_mag_l);

```

```

KiFc      = 2*cos(alpha/2)*Phi00*N_mag/(A_mag_l*mu_0*R_mag0);
KlFc      = -4*cos(alpha/2)/R_mag0*(Phi00/(A_mag_l*mu_0))^2;

K_i = 2*KiFc;          % current stiffness
K_s = 2*KlFc;          % position stiffness

%
% switching amplifier and current controller
%
TsADC_i   = 3E-6;      % [s] conversion delay for ADC for current
TsDAC_i   = 3E-6;      % [s] conversion delay for DAC for voltage
Tswitch   = 1/60000;   % [s] switching time
Kswitch   = 1;         % [-] amplification
ADCbit_i  = 9;         % [-] bit-resolution for ADC for current
DACbit_i  = 11;        % [-] bit-resolution for DAC for voltage
U_ADC_i   = 8;         % [A] input range for current
U_DAC_i   = 5;         % [V] output range for voltage
voltage   = 75;        % [V] switching voltage
K_a       = 2*voltage/U_DAC_i;
n_m       = Kphi*Khiphu;
d_m       = [1 Khipphi];
K_c       = 3.5;
n_i       = K_c*[1 Khipphi]; % numerator for PI-controller
d_i       = [1 0];          % denominator for PI-controller
[nn_i,dd_i] = ...
c2dm(n_i,d_i,Tsamp,'tustin'); % discrete time PI controller
U_max     = 75;            % maximum voltage for anti windup

n_s = K_a*R_mag0/N_mag^2;
d_s = [1 0];
n_h0 = 1;
d_h0 = [Tsamp/2 1];
n_o = n_s*n_h0;
d_o = conv(d_s,d_h0);
T_tot = Tsamp;

%
% measurement in mu-meter
%
k_m      = 1E6;          % [-] measurement gain
R_max    = 250E-6*k_m;  % [mum] Maximum displacement in mu-meter
ADCbit_x = 10;         % [-] bit-resolution for ADC for position
TsADC_x  = 3E-6;      % [s] conversion delay for ADC for position
maxnoise = 1;          % [mum] measurement noise in mu-meter

%
% mechanical part
%
m_r      = 28.768;      % [kg] rotor mass
Iar      = 0.8632;      % [kgm2] axial moment of inertia for rotor
Ipr      = 0.02188;     % [kgm2] polar moment of inertia for rotor
a_s      = 0.23877;     % [m] distance to bearing A from centre of gravity
b_s      = -0.24123;    % [m] distance to bearing B from centre of gravity
c_s      = 0.18977;     % [m] distance to sensor A from centre of gravity
d_s      = -0.19233;    % [m] distance to sensor B from centre of gravity

```

```

if rotflag == 1,
    omega = 20000*2*pi/60;          % [rad/s] rotor speed
else
    omega = 0;                      % [rad/s]
end

%
% rotor weight
%
g = 9.81;
mg0 = m_r*g;                        % [kg] rotor weight
if weightflag == 1,
    F_stat = -mg0/(a_s-b_s)*[0 -b_s 0 a_s]; % [N] static force
else
    F_stat = [0 0 0 0];             % [N] static force
end
%
if loadflag == 1,
    F_load = [100 0 100 0];        % [N] disturbance force
else
    F_load = [0 0 0 0];           % [N] disturbance force zero
end

%
% step of non conservative stiffness parameter
%
if excitflag == 1,
    excitation = 1E7;              % [N/m] non conservative stiffness parameter
else
    excitation = 0;                % [N/m] non conservative stiffness parameter = zero
end
n = -0.1;                          % [m] distance from centre of gravity
n_0 = 0;                            % [N/m] quantity for design

%
% system matrices
%
Ks0 = diag([K_s,K_s,K_s,K_s]); % position stiffness
Ki0 = diag([K_i,K_i,K_i,K_i]); % current stiffness
M0 = diag([Iar,m_r,Iar,m_r]); % mass matrix in inertia coordinates
G0 = [ 0 0 Ipr 0;                 % gyroscopic matrix in inertia coordinates
      0 0 0 0;
      -Ipr 0 0 0;
      0 0 0 0];

Nn = [ 0 -1 ;                     % matrix of non conservative stiffness
      1 0 ];                       % at plane N'

%
% transformation matrices
%
Tl = [a_s 1 0 0;                  % x(B) = Tl x(0)T

```

```

        0 0 a_s 1;          %
        b_s 1 0 0;         %
        0 0 b_s 1];
%
Ts = [c_s 1 0 0;         % x(S) = T1 x(0)T
      0 0 c_s 1;         %
      d_s 1 0 0;         %
      0 0 d_s 1];
%
Tn0n = [n 1 0 0;         %
        0 0 n 1];       % x(N) = Tn0n x(0)
% f(0) = Tnn0 f(N) = Tn0n' f(N)
Tn0nT = Tn0n';

%
invT1 = inv(T1);
invTs = inv(Ts);
invT1T = inv(T1)';
T12s = Ts*invT1;        % x(S) = T12s x(B)
%
invT12s = inv(T12s);
%
M0 = invT1'*M0*invT1;   % mass matrix in bearing coordinates
invM0 = inv(M0);        % inverse mass matrix in bearing coordinates

G0 = invT1'*G0*invT1;   % gyroscopic matrix in bearing coordinates

N1 = Tn0n'*Nn*Tn0n;     % matrix of non conservative stiffness
N0 = invT1'*N1*invT1;   % in bearing coordinates

Omeгадаesign=0;

A0 = [          zeros(4)          eye(4);
      -invM0*(Ks0+n_0*N0)  -invM0*G0*Omeгадаesign];

B0 = [          zeros(4)          ;
      invM0*Ki0   ];

C0 = k_m*[eye(4,4) zeros(4,4)]; % output in mu-meter !!

D0 = zeros(4,4);

index = [2 2 2 2];
dof = max(size(M0));
order = max(size(A0));
[n_out,n_in] = size(D0);
[A0d,B0d,C0d,D0d] = c2dm(A0,B0,C0,D0,Tsamp,'zoh'); % Discretise
[Ac0,Bc0,Cc0,Tc0] = ctrlform(A0d,B0d,C0d,index); % Transformation

%
% initial values for simulation
%
X0 = zeros(order,1);    % initial state
i0A = [i00-F_stat(1)/K_i; ...

```

```

        i00-F_stat(2)/K_i; ...
        i00+F_stat(1)/K_i; ...
        i00+F_stat(2)/K_i];          % static current bearing A
i0B = [i00-F_stat(3)/K_i; ...
        i00-F_stat(4)/K_i; ...
        i00+F_stat(3)/K_i; ...
        i00+F_stat(4)/K_i];          % static current bearing B
%
u0A = i0A*R_0hm/K_a;                % static voltage bearing A
u0B = i0B*R_0hm/K_a;                % static voltage bearing B
Phi0A = N_mag*i0A/R_mag0;           % static flux bearing A
Phi0B = N_mag*i0B/R_mag0;           % static flux bearing B
%
fprintf('\n\n Done. \n\n');
%
```

D.2.2 Estimation algorithm

The state space estimation algorithm and all control algorithms are contained in one function. This function simulates all tasks of a DSP with the measured values for the rotor position and all coil currents as input from the ADCs, and the control voltage for all DACs as output, respectively. This function is called at each sample interval T_s .

```

function [sys, x0] = controller(t,x,u,flag,Tsamp)
%
%      function [sys, x0] = controller(t,x,u,flag,Tsamp)
%
%      model identification state space adaptive control
%
%
% global definitions
%
global invTl2s R_max n_in n_out order index k_m % model
global Ahat Bhat Chat Ac0 Bc0 Cc0             % estimated model
global Kt Kw Ki Khat                          % controller
global U Y Xhat Yhat Ui u_act e_act epsilon  % controller states
global par d_par Dp Dp0 Up Dpn Upn Psi Wk     % estimation algorithm
global k_rho rho rho_0 rho_inf maxnoise       % forgetting factor control
global sigmaN_1 sigmaN_2 delta N_1 N_2       % forgetting factor control
global threshold minvar maxvar               % forgetting factor control
global i0A i0B u0A u0B nn_i U_DAC_i n_mag     % current controller
global I_max U_max u_act e_act              % current controller
global trigger Dp0
global IDACflag      integflag setflag       % general
global U_s X_s Y_s   % save variables
global sigma_s delta_s rho_s                 % save variables
global par_s det_s Kt_s Ki_s                 % save variables
global u_s i_s t_s t0 tterm tstep           % save variables
%
% flag = 0 --> Return sizes of parameters and initial conditions
%
if flag == 0,
```

```

%
% initialisation of state space controller
%
Ahat = Ac0; % initial system matrix
Bhat = Bc0; % initial controller matrix
Chat = Cc0; % initial measurement matrix
Khat = obscalc(Ahat,Bhat,Chat,[index],'n'); % initial Kalman matrix
if integflag == 1
    [Kt,Ki] = ...
    ctrlcalci(Ahat,Bhat,Chat,index,'y'); % initial controller
    Ui = zeros(n_out,1); % initial output
else
    [Kt,Kw] = ...
    ctrlcalc(Ahat,Bhat,Chat,index,'y'); % initial controller
end
U = zeros(n_in,1); % initial input
Y = zeros(n_out,1); % initial output
Xhat = zeros(order,1); % initial estimated state
Yhat = Chat*Xhat; % initial estimated output
epsilon = Y - Yhat; % initial estimation error
u_act = [u0A; u0B]; % initial control voltage
e_act = zeros(n_mag*2,1); % initial current error

%
% forgetting factor control
%
N_1 = 16; % short memory
N_2 = 16^2; % long memory
sigmaN_1 = sqrt(2)*maxnoise^2; % initial variance with short memory
sigmaN_2 = sqrt(2)*maxnoise^2;
threshold = 0.2; % trigger value
rho_0 = 0.999; % rho_0
rho_inf = 1; % rho infinity
rho = rho_inf; % initial rho
delta = 0; % initial delta
k_rho = 0.99;

%
% parameter estimation
%
par = [
    Ahat(2,:)';
    Ahat(4,:)';
    Ahat(6,:)';
    Ahat(8,:)';
    Chat(:);
    Khat(:);
    ]; % initial parameter vector
d_par = max(size(par)); % number of parameters
Dpmin = [
    ones(32,1)*1E-8; % initial covariance
    ones(32,1)*1E-8; % diagonal matrix replaced
    ones(32,1)*1E-8; % by vector
    ];

```

```

Dp = diag(Dpmin);
Dp0 = Dp;
Up = eye(d_par,d_par);
Dpn = eye(d_par,d_par);
Upn = eye(d_par,d_par);
minvar = 1E-12;           % minimal covariance
maxvar = 1E-6;           % maximal covariance
Psi = zeros(n_out,d_par)'; % initial gradient
Wk = zeros(order,d_par); % initial Wk

%
% save variables initialisation
%
n_s = ceil((tterm-t0)/Tsamp);
U_s = zeros(n_in,n_s);
X_s = zeros(order,n_s);
Y_s = zeros(n_out,n_s);
if IDACflag == 1,
    sigma_s = zeros(2,n_s);
    delta_s = zeros(1,n_s);
    rho_s = zeros(1,n_s);
    par_s = zeros(d_par,n_s);
    det_s = zeros(d_par,n_s);
    Kt_s = zeros(n_in*order,n_s);
    if integflag == 1
        Ki_s = zeros(n_out*n_in,n_s);
    end
end
end
i_s = zeros(n_mag*2,n_s);
u_s = zeros(n_mag*2,n_s);
t_s = zeros(1,n_s);

%
% verbose output
%
if IDACflag == 1,
    fprintf('\n Starting estimation with %2.0f parameters \n',d_par);
end
if integflag == 1
    fprintf('\n Starting controller with integrative feedback \n')
else
    fprintf('\n Starting state space controller \n')
end

%
% simulation
%
set_param('rotor','Stop time','tterm'); % stop simulation
sys = [0,1,2*n_mag,n_out+2*n_mag,0,0];
x0 = [0];

%
%-----
% flag = 2 --> real time hit
%
elseif flag == 2,

```

```

        sample_hit = (abs(rem(1E6*t,1E6*Tsamp)) < Tsamp/1E6);
        if sample_hit
%           fprintf('Flag 2 %e\n',t)
            counter = x(1) + 1;

%
%           input of measured data and transformation
%
%
Ys = u(1:n_out);           % read rotor position in sensor
                           % coordinates in mu-meter
Y = invTl2s*Ys;           % transformation to bearing
                           % coordinates
if max(abs(Y))>R_max,      % check whether rotor is out of
                           % range
    set_param('rotor','Stop time',t); % stop simulation
    fprintf('t = %g, ...
            Rotor touched bearing, ...
            emergency stop\n',t)
    return;
end
I_list = u(n_out+1:n_out+2*n_mag); % read currents for all magnets

%
%           Set point error
%
%
if setflag == 1
    if t > tstep,
        W = 1E-6*k_m*[100 100 100 100]';
    else
        W = [0 0 0 0]';
    end
else
    W = [0 0 0 0]';
end

%
%           Prediction error
%
%
epsilon = Y - Yhat;

%
%           Start estimation algorithm
%
%
if IDACflag == 1,
%
%           Two estimates for the sum of the variances
%
%
sigmaN_1 = (N_1-2)*sigmaN_1/(N_1-1) + ...
            epsilon'*epsilon/n_out/N_1;   % short memory
sigmaN_2 = (N_2-2)*sigmaN_2/(N_2-1) + ...
            epsilon'*epsilon/n_out/N_2;   % long memory

%

```

```

%      Calculation of trigger value deltahat(k)
%
deltan = (sigmaN_1-sigmaN_2)/sigmaN_2;
delta = (N_1-2)*delta/(N_1-1) + (deltan-delta)/N_1;

%
%      Compute forgetting factor rho(k)
%
triggern = sign(delta-threshold);
if triggern>0,
    if sign(triggern - trigger)>0
        fprintf('t = %1.6f, Parameter change triggered ... \n',t)
%        fprintf('t = %g, Add Dp0 to Dp \n',t)
%        Dp = Dp + Dp0;
    end
    rhon = rho_0;
    fprintf('t = %g, Reset rho to %f\n',t,rhon)
    burst = noise(4,1,100);
else
    burst = zeros(4,1);
    rhon = k_rho*rho + rho_inf*(1-k_rho);
%    fprintf('t = %1.6f, rho = %f \n',t,rhon)
end

%
%      covariance update using Bierman's factorisation
%
for k = 1:n_out,
    f = Up'*Psi(:,k);
    g = Dp*f;
    beta(1) = rhon;
    for j = 1:d_par,
        beta(j+1) = beta(j) + f(j)*g(j);
        Dpn(j,j) = Dp(j,j)*beta(j)/beta(j+1)/rhon;
        Dpn(j) = max([minvar min([ Dpn(j) maxvar] )]);
        mu(j) = -f(j)/beta(j);
        nu(j) = g(j);
        for i = 1:j-1,
            Upn(i,j) = Up(i,j) + nu(i)*mu(j);
            nu(i) = nu(i) + Up(i,j)*nu(j);
        end
    end
    L(:,k) = (nu/beta(d_par+1))';
    Dp = Dpn;
    Up = Upn;
end

%
%      Model update
%
parn = par + L*epsilon;
Ahat(2,:) = parn(1:8)';
Ahat(4,:) = parn(9:16)';
Ahat(6,:) = parn(17:24)';
Ahat(8,:) = parn(25:32)';

```

```

Chat(:) = parn(33:64)';
Khat(:) = parn(65:96)';

%
%      Stability check of predictor
%
eigenvalues = abs(eig(Ahat -Khat*Chat));
evil = find(eigenvalues>=1);
if evil == [],          % no eigenvalue outside the unit circle
    par = parn;        % parameter update
%      Dp = Dpn;
else
    fprintf('t = %g, Predictor unstable, no update\n',t)
    Ahat(2,:) = par(1:8)';
    Ahat(4,:) = par(9:16)';
    Ahat(6,:) = par(17:24)';
    Ahat(8,:) = par(25:32)';
    Chat(:) = par(33:64)';
    Khat(:) = par(65:96)';
end

%
%      New controller
%
if integflag == 1
    [Kt,Ki] = ctrllcalci(Ahat,Bhat,Chat,index,'y');
else
    [Kt,Kw] = ctrllcalc(Ahat,Bhat,Chat,index,'y');
end

%
%      State space controller with or without integrative feedback
%
if integflag == 1
    E = W + burst - Y;
    U = -Kt*Xhat + Ui;
    Uin = Ui + Ki*E;
else
    U = -Kt*Xhat + Kw*(W + burst);
end

%
%      New state estimation
%
Xhatn = Ahat*Xhat + Bhat*U + Khat*epsilon;
Yhatn = Chat*Xhatn;

%
%      New Gradient Matrix
%
[Mk,Vk] = MkVk_ACK(Xhat,epsilon);          % Mk(x(k)) and Vk(x(k))
Wkn = (Ahat-Khat*Chat)*Wk + Mk + Khat*Vk; % Wk(k+1)
[Mkn,Vkn] = MkVk_ACK(Xhatn,epsilon);      % Mk(x(k+1)) and Vk(x(k+1))
Psin = Wkn'*Chat' + Vkn';                 % Psi(k+1)

```

```

%
% end estimation algorithm

else          % no adaptive control

%
%          State space controller with or without integrative feedback
%
%   if integflag == 1
%       E   = W - Y;
%       U   = -Kt*Xhat + Ui;
%       Uin = Ui + Ki*E;
%   else
%       U   = -Kt*Xhat + Kw*W;
%   end

%
%          State estimation for state space controller
%
%   Xhatn = Ahat*Xhat + Bhat*U + Khat*epsilon;
%   Yhatn = Chat*Xhatn;

end          % if IDACflag == 1,

%
% Necessary for step response
%
%if t < 10*Tsamp,
% U = zeros(4,1);
%else
% U = [0.25 0.5 0.75 1]';
%end

%
% Necessary for frequency response
%
%U = [0.25 0.5 0.75 1]*sin(2*pi*1000*t);

%
% setpoints for each current control loop
%
%   I_sit(1,1) = i0A(1)+U(1);
%   I_sit(2,1) = i0A(2)+U(2);
%   I_sit(3,1) = i0A(3)-U(1);
%   I_sit(4,1) = i0A(4)-U(2);
%   I_sit(5,1) = i0B(1)+U(3);
%   I_sit(6,1) = i0B(2)+U(4);
%   I_sit(7,1) = i0B(3)-U(3);
%   I_sit(8,1) = i0B(4)-U(4);

%
% Necessary for step response
%
%if t < 10*Tsamp,

```

```

% I_sit = 4*ones(8,1);
%else
% I_sit = [4.25 4.5 5 5 3 3.5 3.9 4]';
%end

%
% Necessary for frequency response
%
% I_sit = 4*ones(8,1) + [0.25 0.5 1 1 1 0.5 0.9 1]*sin(2*pi*10*t);

%
%      PI-current controller
%
for k = 1:2*n_mag,
    I_sit(k,1) = max([ 0 min([ I_max I_sit(k,1) ]) ]);
    % Saturation i
    e_new(k,1) = I_sit(k,1)-I_ist(k,1);
    u_new(k,1) = u_act(k,1) + nn_i(1)*e_new(k,1) + ...
                nn_i(2)*e_act(k,1); % PI-controller
    u_new(k,1) = max([ -U_max min([ U_max u_new(k,1) ]) ]);
    % Anti windup
end

%
% memory shift
%
Xhat = Xhatn; % estimated rotor states
Yhat = Yhatn; % estimated rotor states
u_act = u_new; % old control voltage
e_act = e_new; % old current error
if integflag == 1
    Ui = Uin;
end
Wk = Wkn;
Psi = Psin;
trigger = triggern;
rho = rhon;

%
% save variables
%
U_s(:,counter) = U;
X_s(:,counter) = Xhat;
Y_s(:,counter) = Y;
if IDACflag == 1,
    sigma_s(:,counter) = [sigmaN_1; sigmaN_2];
    delta_s(:,counter) = delta;
    rho_s(:,counter) = rho;

    par_s(:,counter) = par;
    det_s(:,counter) = diag(Dp);
    Kt_s(:,counter) = Kt(:);
    if integflag == 1
        Ki_s(:,counter) = Ki(:);
    end
end

```

```

    end
    i_s(:,counter) = I_ist;
    u_s(:,counter) = u_new;
    t_s(:,counter) = t;

%
% simulink system
%
    sys = [counter];

%     end of sample_hit loop
    else
        sys = x;
    end

%
%-----
% flag = 3 --> Return state vector
%
elseif flag == 3
%
    sys = [u_act+U_DAC_i/2];           % shift voltage to DAC range

%
%-----
% flag = 4 --> Return next sample hit
%
elseif flag == 4
%
    sys=ceil(t/Tsamp+Tsamp/1e8)*Tsamp;

%
end

```

D.2.3 Controller calculation

The automated controller computation based on the identified model is carried out in a special function for a controller with or without an integrative feedback.

D.2.4 Controller calculation for P-structure

```

function [Kt,Kw] = ctrlcalc(A,B,C,N,flag);
%
%     function [Kt,Kw] = ctrlcalc(A,B,C,N,flag);
%
%     State space controller calculation,
%     A,B,C must be in controller canonical form
%     If flag == 'n', the system is transformed
%     N is the vector of controller index N = [n1 n2 ...]

%
%     Poles at z1/2 = 0.9691,  or
%           at s1/2 = -313.5,  (corresponds to sqrt(K_s/(m_r/2)))

```

```

%
z = 0.9691;
P = [1 -2*z z^2];
mui = 2;          % ni = 2
%
if flag == 'n',
    [A,B,C,T] = ctrlform(A,B,C,N);
end
N_N = max(size(N)); % number of second order sub-systems
%
for k = 1:N_N,
    mui = N(k);
    ii = sum(N(1:k))-N(k)+1;
    iii = sum(N(1:k));
    XX = [ zeros(mui-1,mui-1)   -eye(mui-1,mui-1)   ; ...
           P(mui+1:-1:2)       ] ;
    IABk(ii:iii,ii:iii) = XX + eye(mui,mui);
%
    for l = 1:N_N,
        muj = N(l);
        jj = sum(N(1:l))-muj+1;
        jjj = sum(N(1:l));
        if l == k,
            Kt(k,jj:jjj) = P(muj+1:-1:2) + A(iii,jj:jjj);
        else
            Kt(k,jj:jjj) = A(iii,jj:jjj);
        end
    end
end
%
end
%
Kw = inv(C*inv(IABk)*B);
if flag == 'n',
    Kt = Kt*inv(T);
end

```

D.2.5 Controller calculation for PI-structure

```

function [Kt,Ki] = ctrlcalci(A,B,C,N,flag);
%
%     function [Kt,Ki] = ctrlcalci(A,B,C,N,flag);
%
%     State space controller calculation with additional
%     integrative feedback,
%     A,B,C must be in controller canonical form
%     If flag == 'n', the system is transformed
%     N is the vector of controller index N = [n1 n2 ...]
%
%     Poles at z1/2/3 = 0.9691, or
%           at s1/2/3 = -313.5, (corresponds to sqrt(K_s/(m_r/2)))
%
z = 0.9691;
Ps = [1 -3*z 3*z^2 -z^3];

```

```

ps1 = Ps(2);
ps2 = Ps(3);
ps3 = Ps(4);
mui = 2;           % ni = 2
%
if flag == 'n',
    [A,B,C,T] = ctrlform(A,B,C,N);
end
N_N = max(size(N));           % number of second order sub-systems
%
Sr = zeros(2*N_N,N_N);
for k = 1:N_N,
    Sr(2*k-1:2*k,k) = [1 1]';
end
N01 = C*Sr;                   % N0(z=1)
Ps1 = sum(Ps)*eye(N_N,N_N);   % P_Tilde(z=1)
%
Ki = Ps1*inv(N01);           % Gain Ki
%
for k = 1:N_N,
    mui = N(k);
    p1 = ps1 + 1;
    p2 = Ki(k,:)*C(:,2*k-1) - ps3;
    P = [1 p1 p2];
    ii = sum(N(1:k))-N(k)+1;
    iii = sum(N(1:k));
    %
    for l = 1:N_N,
        muj = N(l);
        jj = sum(N(1:l))-muj+1;
        jjj = sum(N(1:l));
        if l == k,
            Kt(k,jj:jjj) = P(muj+1:-1:2) + A(iii,jj:jjj);
        else
            Kt(k,jj:jjj) = A(iii,jj:jjj);
        end
    end
end
%
end
%
if flag == 'n',
    Kt = Kt*inv(T);
end

```

D.2.6 Helper applications

In order to compute the state space representation in controller canonical form, the transformation derived in Appendix A was implemented in a MATLAB function. The same is true for the observer canonical form.

Transformation to controller canonical form

```

function [Ac, Bc, Cc, Gamma] = ctrlform(A,B,C,N)
%
%     function [Ac, Bc, Cc, Gamma] = ctrlform(A,B,C,N)
%
%     Transformation into controller canonical form
%     system matrices A,B,C and structural indices N = [n1,n2 ..]
%
%     general definitions
%
m = max(size(N));
n = max(size(A));
b = zeros(m,n);
Tc = [];
for k = 1:m,
    b(k,sum(N(1:k))) = 1;
    for l = 1:N(k),
        Tc = [Tc, A^(l-1)*B(:,k)];
    end
end
%
Gamma = [];
for k = 1:m,
    gamma = inv(Tc')*b(k,:);
    for l = 1:N(k),
        Gamma = [Gamma; gamma'*A^(l-1)];
    end
end
%
Ac = Gamma*A*inv(Gamma);
Bc = Gamma*B;
Cc = C*inv(Gamma);

```

Transformation to observer canonical form

```

function [Ao, Bo, Co, Gamma] = obsform(A,B,C,N)
%
%     function [Ao, Bo, Co, Gamma] = obsform(A,B,C,N)
%
%     Transformation into observer canonical form
%     system matrices A,B,C and structural indices N = [n1,n2 ..]
%
%     general definitions
%
m = max(size(N));
n = max(size(A));
c = zeros(m,n);
To = [];
for k = 1:m,
    c(k,sum(N(1:k))) = 1;
    for l = 1:N(k),

```

```

        To = [To; C(k,:)*A^(l-1)];
    end
end
%
Gamma = [];
for k = 1:m,
    gamma = inv(To)*c(k,:)' ;
    for l = 1:N(k),
        Gamma = [Gamma, A^(l-1)*gamma];
    end
end
%
Ao = inv(Gamma)*A*Gamma;
Bo = inv(Gamma)*B;
Co = C*Gamma;

```

Computation of an observer

```

function K = obscalc(A,B,C,N,flag);
%
%     function K = obscalc(A,B,C,N,flag)
%
%     State space observer calculation,
%     A,B,C must be in observer canonical form
%     If flag == 'n', the system is transformed
%     N is the vector of controller N = [n1 n2 ...]
%
%     general definitions
%
z = 0.90;           % all poles equal
P = [1 -2*z z^2];
muj = 2;           % nj = 2
%
if flag == 'n',
    [A,B,C,T] = obsform(A,B,C,N);
end
N_N = max(size(N));
%
for k = 1:N_N,
    muj = N(k);
    jjj = sum(N(1:k))-N(k)+1;
    jjj = sum(N(1:k));
    %
    for l = 1:N_N,
        mui = N(l);
        ii = sum(N(1:l))-mui+1;
        iii = sum(N(1:l));
        if l == k,
            K(ii:iii,k) = P(mui+1:-1:2)' + A(ii:iii,jjj);
        else
            K(ii:iii,k) = A(ii:iii,jjj);
        end
    end
end
end

```

```

%
end
%
if flag == 'n',
    K = T*K;
end

```

Computation of the matrices M_k and V_k

Note that the following implementation is not very efficient, but easy. In a real implementation this problem would be solved in a more sophisticated way.

```

function [Mk, Vk] = MkVk(Xhat,epsilon)
%
%     function MkVk(Xhat,epsilon)
%
%     Compute Mk and Vk depending on Xhat and epsilon
Mk = [ zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,32) ...
      epsilon'      zeros(1,28) ; ...
      Xhat'         zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,32) ...
      zeros(1,4)    epsilon'      zeros(1,24) ; ...
      zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,32) ...
      zeros(1,8)    epsilon'      zeros(1,20) ; ...
      zeros(1,8)    Xhat'         zeros(1,8)    zeros(1,8)    zeros(1,32) ...
      zeros(1,12)   epsilon'      zeros(1,16) ; ...
      zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,32) ...
      zeros(1,16)   epsilon'      zeros(1,12) ; ...
      zeros(1,8)    zeros(1,8)    Xhat'         zeros(1,8)    zeros(1,32) ...
      zeros(1,20)   epsilon'      zeros(1,8)    ; ...
      zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,8)    zeros(1,32) ...
      zeros(1,24)   epsilon'      zeros(1,4)    ; ...
      zeros(1,8)    zeros(1,8)    zeros(1,8)    Xhat'         zeros(1,32) ...
      zeros(1,28)   epsilon'      ];
%
Vk = [ zeros(1,32)    Xhat'         zeros(1,8)    zeros(1,8)    zeros(1,8)    ...
      zeros(1,32) ; ...
      zeros(1,32)    zeros(1,8)    Xhat'         zeros(1,8)    zeros(1,8)    ...
      zeros(1,32) ; ...
      zeros(1,32)    zeros(1,8)    zeros(1,8)    Xhat'zeros(1,8) ...
      zeros(1,32) ; ...
      zeros(1,32)    zeros(1,8)    zeros(1,8)    zeros(1,8)    Xhat'         ...
      zeros(1,32) ];
%

```

Curriculum vitae

Name: Dipl.-Ing. Peter Wurmsdobler
Geburt: 14.8.1968 in Schärding
Staatsbürgerschaft: Österreich

Ausbildung

1974-1978: Volksschule St. Willibald, A-4762 St. Willibald
1978-1986: Humanistisches Gymnasium des Stiftes Wilhering,
A-4073 Wilhering
10.6.1986: Matura am selbigen Gymnasium
1986-1992: Studium des Allgemeinen Maschinenbaus an der
Technischen Universität Wien, Diplomarbeit:
*„Simulation eines nichtlinearen Modells für
einen magnetgelagerten Mehrscheibenrotor“*
17.6.1992: Sponsion zum Diplomingenieur für Maschinenbau
seit 1992: Doktoratsstudium der Technischen Wissenschaften
an der Technischen Universität Wien, Dissertation:
*„State Space Adaptive Control for a Rigid Rotor
Suspended in Active Magnetic Bearings“*

Berufliche Tätigkeit

Juli 1988,1989: Ferialpraxis bei IPH-Industrieanlagen St.Willibald
August 1990: Ferialpraxis bei Voest-Alpine MCE
Juli/August 1991: Ferialpraxis bei UNITECH A/S in Bergen, Norwegen
Juli/August 1992: Projektarbeit bei MECOS-Traxler in Winterthur, Schweiz:
*„Entwurf einer Feed-Forward-Kompensation zur
Schwingungsunterdrückung bei magnetgelagerten
Rotoren“*
seit 1992: Universitätsassistent am Institut für Maschinen- und
Prozeßautomatisierung an der Technischen Universität
Wien, Arbeitsgebiete:
*„Einsatz von Fuzzy-Logic für die intelligente
Steuerung von Jalousien in Bürogebäuden“,
„Adaptive Regelung und Identifikation von
magnetgelagerten Rotoren“*