Linear Parameter Varying Control of Induction Motors

Ph.D. thesis
by
Klaus Trangbæk

Department of Control Engineering
Aalborg University
Fredrik Bajers Vej 7
DK-9220 Aalborg Ø
Denmark.

ISBN 87-90664-11-6
Doc. no. D-2001-4**47**8
June 2001

This thesis was typeset using LaTeX2e.
Drawings were made in Xfig 3.2.
Graphs were generated in MatLab from Mathworks Inc.

Printing: Centertrykkeriet, Aalborg University

PREFACE

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Control Engineering at Aalborg University. The work documented herein has been carried out at the Department of Control Engineering, Aalborg University, in the period from July 1997 to April 2001 under the supervision of Professor Jakob Stoustrup and Associate Professor Henrik Rasmussen.

I am very grateful to Jakob and Henrik for being great and always available capacities on theoretical as well as practical problems. I am also grateful for their kind insistence that at some point I should stop researching and start writing.

A sincere thanks goes to Professor Anton Stoorvogel at Technische Universiteit Eindhoven in The Netherlands for interesting and inspiring discussions during my six month stay.

I am also grateful to the staff at the Department of Control Engineering for providing a pleasant and inspiring working environment.

I am especially grateful to Assistant Professor Jan Dimon Bendtsen for proof-reading the script and being a great companion both professionally and socially.

Last, but not least, I want to thank my parents for being patient and supportive during the busy years.

June 2001, Aalborg, Denmark

Klaus Trangbæk

ABSTRACT

The subject of this thesis is the development of linear parameter varying (LPV) controllers and observers for control of induction motors. The induction motor is one of the most common machines in industrial applications. Being a highly nonlinear system, it poses challenging control problems for high performance applications. This thesis demonstrates how LPV control theory provides a systematic way to achieve good performance for these problems. The main contributions of this thesis are the application of the LPV control theory to induction motor control as well as various contributions to the field of LPV control theory itself.

Within the last decade the theoretical background for control of LPV systems has been developed. LPV systems constitute a large class of nonlinear systems with a special structure allowing for a systematic approach to controller design. Based on a widely used model of the induction motor and the well-known rotor flux-oriented control scheme, it is demonstrated how LPV methods can be applied to several subproblems in induction motor control.

The current equations of the induction motor have a particular structure, which allows them to be written on a complex form. It is shown that for an LPV system with this structure, the optimal controller will also possess this structure. This knowledge can be employed to improve the numerics of the controller synthesis and to reduce the computational burden in the implementation.

Viewing the rotational speed as an external parameter, the current equations of the induction motor constitute an LPV system. This is used to design an LPV flux observer. The result is an observer with good performance and very little tuning needed.

At the cost of some conservatism the LPV control theory can be applied to an even wider range of systems known as quasi-LPV systems. It is demonstrated how this can be applied to the design of a stator current controller. As in the case of the flux observer design, the resulting controller performs well and requires very little tuning.

In certain cases it is difficult to obtain accurate models using physical principles. We therefore turn our attention to nonlinear black-box modelling with multi-layer perceptrons (MLPs). A novel method for transforming MLP models into quasi-LPV models is presented. An MLP model of an induction motor system is obtained, and the aforementioned model transformation is performed. The resulting quasi-LPV model is then used in the design of a speed controller. This demonstrates how LPV methods can be used in a systematic approach all the way from modelling to controller implementation.

Finally, robustness to uncertainty in the time-varying parameters is considered. More specifically, we consider the case where the parameter variation is represented by a diagonal gain matrix, which is fully known except for some small perturbation. A novel type of sufficient conditions for robustness is presented, and it is illustrated how this can be used in the speed controller design.

All controllers and observers are tested on a laboratory setup.

The key results have been presented at international conferences or have been submitted for publication in international journals.

DANSK SAMMENFATNING

Denne afhandling omhandler udviklingen af lineære parameter-varierende (LPV) regulatorer og observere til regulering af induktionsmotorer. Induktionsmotoren er en af de mest anvendte maskiner i industrien. Dens ikke-lineære dynamik giver anledning til komplicerede reguleringsproblemer i forbindelse med krævende anvendelser. I denne afhandling demonstreres det, hvorledes LPV-reguleringsteori åbner muligheden for en systematisk tilgang til disse problemer. Afhandlingens væsentligste bidrag er anvendelsen af LPV-teori til regulering af induktionsmotorer, samt diverse bidrag til området LPV-reguleringsteori.

Den teoretiske baggrund for regulering af LPV-systemer er udviklet i løbet af det seneste årti. LPV-systemer udgør en omfattende klasse af ikke-lineære systemer med en speciel struktur, der muliggør en systematisk tilgang til regulatordesign. Med udgangspunkt i en ofte anvendt model for induktionsmotoren og det velkendte rotorflux-orienterede reguleringsprincip demonstrerer denne afhandling, hvordan LPV-metoder kan anvendes på flere delproblemer indenfor regulering af induktionsmotorer.

Induktionsmotorens strømligninger har en speciel struktur, der gør det muligt at skrive dem på kompleks form. Det vises, at for et LPV-system med denne struktur, vil den optimale regulator også besidde denne struktur. Denne viden kan benyttes til at forbedre regulatorsyntesens numeriske egenskaber og til at opnå en mindre beregningskrævende implementation.

Ved at betragte omdrejningshastigheden som en udefra kommende parameter kan induktionsmotorens strømligninger betragtes som et LPV-system. Dette udnyttes til at udvikle en flux-observer. Resultatet er en velfungerende observer med meget lille behov for tuning.

På bekostning af konservatisme kan LPV-teori anvendes på en langt større klasse af ikke-lineære systemer kaldet kvasiLPV-systemer. Dette demonstreres ved udviklingen af en statorstrømsregulator, der, ligesom førnævnte observer-design, resulterer i en velfungerende regulator med meget lille behov for tuning.

I visse tilfælde er det vanskeligt at opnå en tilfredsstillende model baseret på fysiske betragtninger. Derfor rettes opmærksomheden herefter mod ikke-lineær black-box-modellering ved hjælp af multi-lags perceptroner (MLPer). En ny metode til transformation af MLP-modeller til kvasiLPV-modeller præsenteres. En MLP-model af et induktionsmotorsystem optrænes, og den nye tranformationsmetode anvendes. Den resulterende kvasiLPV-model anvendes som grundlag for syntese af en hastighedsregulator. På denne måde demonstreres det, hvordan LPV-metoder danner grundlag for en systematisk tilgang til hele proceduren fra modellering til regulatorimplementation.

Til sidst betragtes robusthed overfor usikkerheder i de tidsvarierende parametre. Specifikt betragtes det tilfælde, hvor parametervariationen kan repræsenteres ved en diagonal forstærkningsmatrix, som er fuldstændigt kendt bortset fra en lille afvigelse. En ny type tilstrækkelige betingelser for robusthed præsenteres, og det illustreres, hvorledes disse kan anvendes i udviklingen af hastighedsregulatoren.

Alle regulatorer og observere er afprøvet eksperimentelt på en laboratorieopstilling.

Hovedresultaterne er præsenteret ved internationale konferencer eller er indsendt til publikation i internationale tidsskrifter.

CONTENTS

NOMENCLATURE

Acronyms and abbreviations

ARMAX	AutoRegressive Moving Average with eXogenous input
ARX	AutoRegressive with eXogenous input
LFT	Linear Fractional Transformation
LMI	Linear Matrix Inequality
LPV	Linear Parameter Varying
LTI	Linear Time Invariant
MLP	Multi-Layer Perceptron
NARMAX	Nonlinear AutoRegressive Moving Average with eXogenous input
NARX	Nonlinear AutoRegressive with eXogenous input
PI	Proportional-Integral
PWM	Pulse-Width Modulated
RQP	Robust Quadratic Performance
s.t.	subject to

Symbols

In this list A denotes a matrix, n a positive integer, Q a square matrix, X a Hermitian matrix, c a complex scalar, r a real scalar.

Various

j	imaginary unit, $j^2 = -1$.
α_{sv}	space vector constant ($= e^{j2\pi/3}$).
$\times$	scalar or matrix multiplication, or
$\times$	Cartesian product of subspaces.
$\mathbf{Co}$	convex hull, see page 13.

Scalar operators

c^*	complex conjugate of c.
$\angle c$	phase angle of c.
$\Im\{c\}$	imaginary part of c.
$\Re\{c\}$	real part of c.
$\text{sign}(r)$	sign function. $= \begin{cases} 1, & r > 0 \\ -1, & r < 0 \\ 0, & r = 0 \end{cases}$.

Vectors and matrices

I_n	identity matrix of size $n \times n$.
A^T	A transposed.
A^*	A complex conjugate transposed (Hermitian transposed).
$A^\dagger$	Moore-Penrose pseudo-inverse of A.
$A^\dashv$	left annihilator of A, see page 10.
$A^\vdash$	right annihilator of A, see page 10.
$X > 0$	X is positive definite.
$X \geq 0$	X is positive semidefinite.
$X < 0$	X is negative definite.
$X \leq 0$	X is negative semidefinite.
$\mathrm{diag}_{1 \leq i \leq n}\{r_i\}$	diagonal matrix with the elements r_i on the diagonal.
$\bar{\Delta}$	matrix set of considered perturbations.
$\mathrm{herm}\,(Q)$	Hermitian part, ($\mathrm{herm}\,(Q) = \frac{1}{2}(Q + Q^*)$).
$\mathbb{H}^{n \times n}$	linear space of Hermitian $n \times n$ matrices.
$\mathrm{im}\,A$	image of the matrix A.
$\mathrm{in}(X)$	inertia of X, number of negative, zero, and positive eigenvalues.
$\ker A$	kernel of the matrix A.
on	see page 11.
$\mathcal{F}_l$	lower LFT, see page 17.
$\mathcal{F}_u$	upper LFT, see page 17.
$\star$	Redheffer star product, see page 18.

Induction motor parameters

J	moment of inertia.
L_m	mutual inductance.
L'_m	referred mutual inductance ($= (1-\sigma)L_s$).
L_r	rotor inductance, self inductance of the (virtual) rotor windings.
L_s	stator inductance, self inductance of the stator windings.
L'_s	referred stator inductance ($= \sigma L_s$).
N_r	fictive number of windings of virtual rotor coils.
N_s	number of windings of stator coils.
R_r	rotor resistance, resistance of the (virtual) rotor windings.
R'_r	referred rotor resistance, ($= (L_m/L_r)^2 R_r$).
R_s	stator resistance, resistance of the stator windings.
T_r	rotor time constant, ($= \frac{L_r}{R_r} = \frac{L'_m}{R'_r}$).
σ	leakage constant ($= 1 - L_m^2/(L_r L_s)$).
Z_p	number of pole pairs.

Induction motor signals

i_{mR}	magnetising current in rotor flux coordinates.
$\bar{i}_m$	magnetising current in stator-fixed coordinates.
$\bar{i}_r$	rotor current in rotor-fixed coordinates.
$\bar{i}_{rs}$	rotor current in stator-fixed coordinates.
$\bar{i}_s$	stator current in stator-fixed coordinates.
i_{sd}	direct component of stator current.
i_{sq}	quadrature component of stator current.
$\bar{i}_{sr}$	stator current in rotor flux coordinates.
m_e	electromagnetic torque.
m_L	load torque.
$\bar{\Psi}_r$	rotor flux linkage in rotor-fixed coordinates.
$\bar{\Psi}_{rs}$	rotor flux linkage in stator-fixed coordinates.
$\bar{\Psi}_s$	stator flux linkage.
θ_{mech}	mechanical angle of rotation of the rotor.
θ_r	electrical angle of rotation of the rotor.
$\bar{u}_s$	stator voltage in stator-fixed coordinates.
u_{sd}	direct component of stator voltage.
u_{sq}	quadrature component of stator voltage.
$\bar{u}_{sr}$	stator voltage in rotor flux coordinates.
ω_{mR}	angular velocity of the magnetising current.
ω_r	angular electrical rotor speed.
ω_{slip}	slip frequency ($= \omega_{mR} - \omega_r$).

Rotor flux oriented controller signals

$\hat{i}_{mR}$	estimate of magnetising current in rotor flux coordinates.
$i_{mR,ref}$	reference for magnetising current.
$i_{sd,ref}$	reference for direct component of stator current.
$i_{sq,ref}$	reference for quadrature component of stator current.
$\hat{i}_{sd}$	estimate of direct component of stator current.
$\hat{i}_{sq}$	estimate of quadrature component of stator current.
$\hat{\bar{i}}_m$	estimate of magnetising current in stator-fixed coordinates.
I_{max}	maximum value for stator current magnitude.
$\bar{i}_{sr,ref}$	reference for stator current.
$\bar{u}_{sr,ref}$	reference for stator voltage.
λ_c	speed estimate update gain of Kubota's speed observer.
$m_{e,ref}$	reference for electromagnetic torque.
$m_{e,max}$	maximal producible electromagnetic torque.
u_{sdff}	feed-forward decoupling voltage, direct component.
u_{sqff}	feed-forward decoupling voltage, quadrature component.
u_{sdc}	control voltage for direct component of stator current.
u_{sqc}	control voltage for quadrature component of stator current.
$\omega_{r,ref}$	reference for angular electrical rotor speed.
$\hat{\omega}_r$	estimate of angular electrical rotor speed.

Chapter 1

INTRODUCTION

This Ph.D. thesis considers the application of linear parameter varying (LPV) control theory to induction motor control. The aim is to develop novel controllers and observers for field-oriented control of induction motors and to provide theoretical contributions in the field of LPV control theory.

1.1 Background

The induction motor has a wide range of applications in industry converting electrical power into mechanical power, for instance in pumps and ventilators. Indeed, in the industrialised countries approximately 60 % of the entire electrical power available is consumed by AC motors, whereof most are induction motors [Kaźmierkowski and Tunia, 1994]. Recent advances in computer technology allows employing control techniques yielding a performance for the induction motor similar to that of the more expensive and less reliable DC motor. This does, however, pose complicated control problems.

Several approaches to induction motor control exist, see for instance [Vas, 1998] or [Kaźmierkowski and Tunia, 1994]. This thesis will focus on one particular approach, the rotor-flux oriented control.

The induction motor is a highly nonlinear system calling for advanced control techniques. Within the last decade the theoretical background for control of linear parameter varying

(LPV) systems has been developed. LPV systems constitute a large class of systems with a special structure allowing for a systematic approach to controller design. At the cost of conservatism the approach can be applied to an even wider range of systems known as quasi-LPV systems.

An LPV system is essentially a linear time-varying system which can be written on the form

$$\begin{aligned} \dot{x} &= A(\theta(t))x + B(\theta(t))u \\ y &= C(\theta(t))x + D(\theta(t))u, \end{aligned}$$

where θ is a time varying parameter vector. As such it has a structure which is similar to a linear time-invariant state space system, and control design methods with some similarity to linear state space methods can indeed be used.

One of the main reasons for LPV control theory being the object of increasing interest is that performance analysis and controller synthesis for these systems can be formulated as linear matrix inequalities (LMIs). LMIs pose convex problems and can be efficiently solved by numerical software such as the MatLab LMI toolbox [Gahinet et al., 1995]. In this sense, once a problem has been cast as an LMI, it can be considered as solved.

An early suggestion that a system on the LPV form could be controlled by a controller of the same form was given in [Becker et al., 1993]. The method given here did however not lead to a convex problem, but the result was later extended in several steps. In [Apkarian et al., 1995] a non-conservative LMI solution was given under the assumption of affine parameter dependence. In [Scherer, 2001] the result is extended to the more general rational parameter dependence. This result seemingly has not been applied to any real-life systems yet. In this thesis it will be applied to the control of an induction motor.

In [Shamma and Athans, 1992] it was suggested that an even wider range of nonlinear systems could be treated as LPV system at the cost of some conservatism. In this quasi-LPV approach the parameters are allowed to depend on the states. By disregarding the explicit dependence and treating these systems as LPV systems, theoretical guarantees of stability and performance can be obtained.

In this thesis these ideas will be used and further developed, aiming for control of induction motors with a rigorous theoretical basis.

1.2 Objectives

The aim of this thesis is to develop novel controllers and observers for field-oriented control of induction motors and to provide theoretical contributions in the field of LPV control theory.

In order to limit the scope we will focus on a particular type of controller, more specifically the rotor-flux oriented controller as discussed in for instance [Rasmussen, 1995], [Vas, 1998], or [Leonhard, 1990]. This controller consists of subblocks as shown in Figure 1.1.

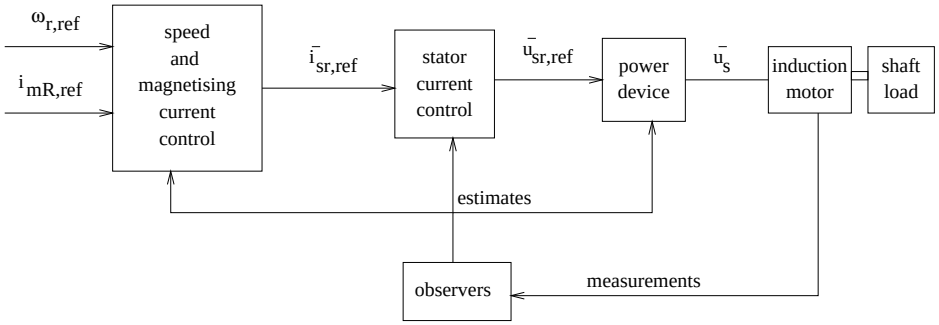


Figure 1.1: *Sketch of the rotor flux oriented speed control scheme.*

The controller objective is to track references for the magnetising current, i_{mR}, and the speed, ω_r. The observers provide estimates of the stator and magnetising currents and of the speed. The speed and magnetising current controllers provide a reference signal, $\bar{i}_{sr,ref}$, for the stator current. The stator current controller tracks this reference by providing the power device with a stator voltage command, $\bar{u}_{sr,ref}$.

The main aim of this thesis is to provide novel methods for designing these sub-blocks using LPV control theory. The theoretical parts will focus on the method described in [Scherer, 2001], which is very general and non-conservative.

1.3 Contributions

The contributions of this thesis are in both the induction motor application and the LPV control theory areas.

The main contributions in the theoretical area are:

- A discrete time version of the continuous time LPV control method presented in [Scherer, 2001] is given.
- The current equations of the induction motor have a particular structure, which allows them to be written on a complex form. It is shown that for an LPV system with this structure, the optimal controller will also possess this structure.
- A systematic approach for transforming neural network state space models into

quasi-LPV models suitable for control design is given. This was presented in the conference paper [Bendtsen and Trangbæk, 2000b].

- A novel approach to robust LPV control for a special class of LPV systems is presented. This has partially been presented in the submitted journal paper [Bendtsen and Trangbæk, 2000a].

The main contributions in the induction motor application area are:

- It is pointed out that the proof of convergence for the speed observer presented in [Kubota et al., 1993] is incorrect and that divergence is possible.
- The discrete time version of the LPV theory is applied to the design of a flux observer. This has partially been documented in the conference paper [Trangbæk, 2000].
- The discrete time version of the LPV theory is applied to the design of a stator current controller using the quasi-LPV approach. This has been documented in the submitted journal paper [Bendtsen and Trangbæk, 2001b].
- A neural network is applied for black-box identification of the motor system.
- The obtained neural network model is transformed into a quasi-LPV model and a speed controller is designed.
- The robust LPV control method is applied to the same model.

1.4 Outline of the thesis

The thesis is organised as follows:

Nomenclature

Provides a list of symbols and abbreviations used throughout the thesis.

Chapter 1 - Introduction

This introduction.

Chapter 2 - Preliminaries

Introduces some basic concepts needed later in the thesis. The theoretical parts of Chapters 5-7 rely heavily on matrix inequalities and in particular linear matrix inequalities (LMIs). These concepts will be briefly discussed.

The model structure used in the same chapters is linear fractional transformations (LFTs). This chapter also discusses interconnections of such systems including the Redheffer star product.

Finally, the use of multi-layer perceptrons (MLPs) as state space models is discussed.

Chapter 3 - Induction Motor System

Describes the dynamic model of a symmetrical three-phase induction motor with a squirrel cage rotor. The part of the model describing the currents is written as a complex second order state space model with the shaft speed as a time-varying parameter. The concept of rotating reference frames is then discussed.

In order to control the speed of the motor it is necessary to use a power device. A commonly used type of power device, the voltage sourced inverter, is described.

Finally, the laboratory setup is discussed. Experiments will be performed on a laboratory system with a $1.5kW$ induction motor.

Chapter 4 - Rotor Flux Oriented Control

The rotor flux oriented control scheme for the induction motor is described. The purpose of the controller is to track a reference speed and a reference magnetising current while rejecting disturbances from the load torque.

First it is discussed how the dynamical equations of the motor are simplified by writing them in a reference system following the angle of the rotor flux. Then the rotor flux oriented control method is described. The method is observer-based and requires an estimate of the rotor flux. A short discussion of flux and speed observers is also given.

Chapter 5 - Linear Parameter Varying Flux Observer

This chapter reviews the LPV synthesis method in [Scherer, 2001] and applies it to the design of a flux observer for the induction motor.

First the historical background of LPV control is reviewed. Then robust quadratic performance analysis of LPV systems is discussed and the so-called full block S-procedure controller synthesis is described. A discrete-time version of these results is then given.

Considering the speed as a time-varying parameter allows writing the induction motor model as either a real fourth order LPV model or as a complex second order LPV model. This is due to a special symmetry in the transfer function. Theoretical justification for the fact that controllers and observers for this type of system can be assumed to have the same type of symmetry without loss of performance is presented.

Finally a discrete-time flux observer is designed using the above theory. The observer is tested on the laboratory setup.

Chapter 6 - Quasi-LPV Current and Speed Controllers

The quasi-LPV approach allows the use of LPV theory for a very general class of nonlinear systems. First a discussion of the quasi-LPV structure is given. The approach is then applied to the design of a stator current controller.

It is then discussed how to transform a neural network state space model into a quasi-LPV model suitable for control design. This method is then applied to the design of a speed controller.

Chapter 7 - Robust LPV Speed Controller

In this chapter a novel approach is given for robust LPV design for systems where the time-varying parameters are uncertain. The method is applied to the design of a speed controller.

Chapter 8 - Conclusions

This chapter contains conclusions and recommendations for further work.

Appendix A - Experimental Setup

In this appendix the experimental setup is discussed in detail.

Appendix B - Lemmas and Proofs

This appendix contains lemmas with no appropriate place in the main thesis as well as proofs which were deemed too long and tedious for the main thesis.

Bibliography

Index

Chapter 2

PRELIMINARIES

This chapter introduces some basic concepts needed later in the thesis. The theoretical parts of Chapters 5-7 rely heavily on matrix inequalities and in particular linear matrix inequalities (LMIs). These concepts are described in Sections 2.2 and 2.3.

The model structure used in the same chapters is linear fractional transformations (LFTs). Section 2.4 discusses interconnections of such systems including the Redheffer star product.

Chapter 6 describes the use of multi-layer perceptrons (MLPs) as a basis for obtaining quasi-LPV models. Multi-layer perceptrons are described in Section 2.5, whereas the discussion of the quasi-LPV structure will be left for Chapter 6.

2.1 Notation

Most of the notation in this thesis is standard with the following exceptions.

Definition 2.1 (Hermitian part, $\mathrm{herm}\,(\cdot)$*)*

Let X *be a square matrix. Then*

$$\mathrm{herm}\,(X) \triangleq \frac{1}{2}(X + X^*).$$

The Hermitian part is a natural extension of the real part of scalars, since the eigenvalues of a Hermitian matrix are real.

The symbol $\perp$ is used as a right annihilator in e.g. [Scherer, 2001] and as a left annihilator in e.g. [Helmersson, 1995]. To avoid confusion when comparing with other literature we shall use the symbols $\vdash$ and $\dashv$ respectively.

Definition 2.2 (Left annihilator, $\dashv$)

$A^{\dashv}$ denotes any full row rank matrix such that $\ker A^{\dashv} = \operatorname{im} A$.

$A^{\dashv}$ only exists if A has linearly dependent rows and then $A^{\dashv}A = 0$.

Definition 2.3 (Right annihilator, $\vdash$)

$A^{\vdash}$ denotes any full column rank matrix such that $\operatorname{im} A^{\vdash} = \ker A$.

Note that $\operatorname{im} A^{*\vdash} = \operatorname{im} A^{\dashv *}$.

Finally, in order to reduce the width of certain matrix equations we will need the following notation. Given a matrix X and a Hermitian matrix P the following expressions are equivalent

$$\begin{bmatrix} X \end{bmatrix}^* \begin{bmatrix} P \end{bmatrix} \begin{bmatrix} X \end{bmatrix} \equiv \begin{bmatrix} * \end{bmatrix}^* \begin{bmatrix} P \end{bmatrix} \begin{bmatrix} X \end{bmatrix}.$$

2.2 Matrix inequalities

A square matrix X is *Hermitian* if $X = X^*$. Let $\mathbb{H}^{n\times n}$ denote the linear space of Hermitian $n \times n$ matrices and let $F \in \mathbb{H}^{n\times n}$. Let $h \in \mathbb{C}^n$ and define the *quadratic form*

$$|h|_F^2 \triangleq h^* F h. \tag{2.1}$$

We say that F is *positive semidefinite* if

$$|h|_F^2 \geq 0 \quad \forall h \tag{2.2}$$

and that F is *positive definite* if

$$\exists \epsilon > 0 : |h|_F^2 \geq \epsilon |h|^2 \quad \forall h. \tag{2.3}$$

We simply denote this $F \geq 0$ and $F > 0$, respectively. $F \leq 0$ (*negative semidefinite*) and $F < 0$ (negative definite) are defined similarly. $\mathcal{S}$ is a *positive subspace* of F if

$$\exists \epsilon > 0 : |h|_F^2 \geq \epsilon |h|^2 \quad \forall h \in \mathcal{S} \tag{2.4}$$

and we write this as

$$F > 0 \text{ on } \mathcal{S}.$$

A *negative subspace* is defined similarly.

A Hermitian matrix has real eigenvalues and the maximal dimension of its positive subspace is equal to the number of positive eigenvalues. Consequently a Hermitian matrix is positive definite if and only if all of its eigenvalues are positive.

Definition 2.4 (Inertia, in*)*

The inertia of a Hermitian matrix H is defined as

$$\text{in}(H) \triangleq (\text{in}_-(H), \text{in}_0(H), \text{in}_+(H)) \tag{2.5}$$

where $\text{in}_-(H), \text{in}_0(H), \text{in}_+(H)$ denote the number of negative, zero and positive eigenvalues of H, respectively.

Definition 2.5 (Inertia on subspace, $\text{in}(\cdot|_\cdot)$*)*

For any subspace $\mathcal{S} \subset \mathbb{C}^n$,

$$\text{in}(H|_{\mathcal{S}}) \triangleq \text{in}(S^*HS) \tag{2.6}$$

for any basis matrix S of $\mathcal{S}$.

The following lemma is a generalisation of the well-known Schur complement lemma.

Lemma 2.6 (Schur complement) [Scherer, 2001]

If A is non-singular and A and B are Hermitian then

$$\text{in}\left(\begin{bmatrix} A & C \\ C^* & B \end{bmatrix}\right) = \text{in}(A) + \text{in}(B - C^*A^{-1}C) \tag{2.7}$$

This lemma is very useful in many situations. An example will be given in Section 2.3.2. The following lemma will be used in the proof of Lemma 2.9.

Lemma 2.7 (Dualisation Lemma) [Scherer, 2001]

Let P be any non-singular Hermitian matrix, and let $\mathcal{S}$ be any subspace such that $\text{in}_0(P|_{\mathcal{S}}) = 0$. Then

$$\text{in}(P) = \text{in}(P|_{\mathcal{S}}) + \text{in}(P^{-1}|_{\mathcal{S}^\perp}) \tag{2.8}$$

Now let $F \in \mathbb{H}^{n\times n}$ be a matrix function of a vector of *decision variables* x. We will call $F(x) > 0$ a *matrix inequality* in x. The *feasibility set* of a matrix inequality is defined as

$$X_{feas} \triangleq \{x : F(x) > 0\}, \tag{2.9}$$

and we say that the matrix inequality is *feasible* if X_{feas} is non-empty. Of course $>$ can be replaced by $\geq$, $\leq$, or $<$ in the above.

Remark 2.8 The decision variables are usually in the form of one or more matrices, but the problem can always be reformulated into the vector form.

Matrix inequalities arise in many control analysis and synthesis problems. There is no general way to solve them, except when F depends affinely on x. In this case, the affine matrix inequality can be solved with convex methods. This will be discussed further in Section 2.3. If the matrix is a quadratic function of x there is generally no way to solve the inequality except in special cases such as the following.

Lemma 2.9 (Elimination Lemma for quadratic matrix inequalities) [Scherer, 1999] [Helmersson, 1999]

Assume that C has dimension $n \times m$ and that

$$\text{in}(P) = (m, 0, n) \tag{2.10}$$

The quadratic matrix inequality

$$\begin{bmatrix} I \\ A^*XB + C \end{bmatrix}^* P \begin{bmatrix} I \\ A^*XB + C \end{bmatrix} < 0 \tag{2.11}$$

in the unstructured unknown X has a solution if and only if

$$B^{\vdash *} \begin{bmatrix} I \\ C \end{bmatrix}^* P \begin{bmatrix} I \\ C \end{bmatrix} B^{\vdash} < 0 \tag{2.12}$$

and

$$A^{\vdash *} \begin{bmatrix} -C^* \\ I \end{bmatrix}^* P^{-1} \begin{bmatrix} -C^* \\ I \end{bmatrix} A^{\vdash} > 0 \tag{2.13}$$

Proof: A constructive proof is given in Appendix B on page 173. $\lhd$

Remark 2.10 X being *unstructured* means that there are no constraints on the structure, e.g. that it must be Hermitian or block diagonal.

This lemma is the basis for the synthesis method described in Section 5.3.

2.3 Linear matrix inequalities

As mentioned in the previous section there is no general way to solve matrix inequalities. However if the matrix depends affinely on the decision variables it is called a *Linear Matrix Inequality* (LMI) and fast and efficient numerical solvers are available. If a problem can be cast as an (finite-dimensional) LMI it can therefore be considered practically solved. This section contains a formal definition of an LMI, a short description of some of the available software packages and the types of problems they can solve, and finally a few examples of control problems that can be cast as LMIs. A more thorough discussion can be found in [Boyd et al., 1994].

Definition 2.11 (Linear matrix inequality, LMI)

A linear matrix inequality is an inequality

$$F(x) > 0$$

where F is an affine (i.e. a constant plus a linear) function mapping a finite dimensional vector space to a Hermitian matrix set $\mathbb{H}^{n\times n}$. The elements of the vector x are called the decision variables*.*

Remark 2.12 In the usual definition the mapping is to a set of real symmetric matrices, and x belongs to a real vector space. Indeed, most available solvers work only with this formulation. The above definition can however be reformulated as an equivalent problem with real matrices of double dimensions. This will be discussed further in Section 5.5.

LMIs have several nice features. For instance the feasibility set is convex. This means that given a set of solutions of an LMI, any *convex combination* of these is also a solution.

Definition 2.13 (Convex combination)

Let $\mathcal{M} = \{x_1, x_2, ..., x_n\}$ be a subset of a linear vector space, and let $\{a_1, a_2, ..., a_n\}$ be a set on non-negative real numbers such that $\sum_{i=1}^{n} a_i = 1$. Then

$$x \triangleq \sum_{i=1}^{n} a_i x_i$$

is called a convex combination *of $\mathcal{M}$.*

Definition 2.14 (Convex hull,$\mathbf{Co}$)

The convex hull, $\mathbf{Co}(\mathcal{M})$ *of a set $\mathcal{M}$ is the intersection of all convex sets containing $\mathcal{M}$.*

If $\mathcal{M}$ is a subset of a linear vector space, then $\mathbf{Co}(\mathcal{M})$ is the set of all convex combinations of elements in $\mathcal{M}$. The convexity of the feasibility set can be used to convert some infinite-dimensional problems into finite-dimensional ones.

Example 2.15 Consider the problem of finding an X such that

$$AX + X^*A^* > 0 \tag{2.14}$$

for all $A \in \bar{A}$, where $\bar{A}$ is convex. If $\bar{A}$ has infinitely many elements, then this is an infinite-dimensional LMI, since (2.14) puts infinitely many constraints on X. For a fixed X we can also view (2.14) as an LMI in A. This means that if X fulfills (2.14) for a number of As, then it is also a solution for a convex combination of these.

Now if $\bar{A}$ is the convex hull of a finite number of *vertices* (or *generator*s) $A_1, A_2, .., A_n$, we only need to solve (2.14) for these vertices in order to obtain a solution for all $A \in \bar{A}$. This is a finite-dimensional problem.

LMIs rarely have an explicit solution but must be solved by numerical iteration. One exception is the LMI in X given by

$$F(X) = A^*XB + B^*X^*A + D > 0. \tag{2.15}$$

This inequality typically needs to be solved as a last step in a controller synthesis, where A, B, and D have been found as solutions to other LMIs. By multiplying from both sides by $A^\vdash$ or $B^\vdash$ it is immediately obvious that

$$A^{\vdash *}DA^\vdash > 0 \text{ and } B^{\vdash *}DB^\vdash > 0 \tag{2.16}$$

are necessary conditions for the feasibility of (2.15). In fact they are also sufficient. This was shown in [Gahinet and Apkarian, 1994], where a constructive proof can also be found. Here we will give an alternative proof based on (2.15) being a special case of the quadratic matrix inequality (2.11).

Lemma 2.16 [Gahinet and Apkarian, 1994]

Let $D \in \mathbb{H}^{n\times n}$. The LMI (2.15) in X is feasible if and only if (2.16) is satisfied.

Proof:

(2.15) is a special case of (2.11) with

$$C + C^* = D,\ P = \begin{bmatrix} 0 & -I_n \\ -I_n & 0 \end{bmatrix}.$$

The inertia condition (2.10) is then immediately fulfilled and the conditions (2.12) and (2.13) can easily be seen to be exactly (2.16). $\lhd$

Remark 2.17 Since the proof of Lemma 2.9 is constructive, this proof can also be considered constructive.

Remark 2.18 In Lemmas 2.9 and 2.16 $<$ can of course be replaced by $>$ and vice versa, but the sufficiency parts do not hold for non-strict inequalities ($\geq$ and $\leq$). Consider for instance the non-strict LMI

$$F(y) = \begin{bmatrix} y & c+y \\ c^*+y & c+c^*+y \end{bmatrix} \geq 0$$

in the real scalar y. This can be written as (2.15) with

$$X + X^* = y, \quad A = B = \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & c \\ c^* & c+c^* \end{bmatrix}.$$

$F(y)$ has eigenvalues $\frac{1}{2}(2y + c + c^* \pm \sqrt{(2y + c + c^*)^2 + 4cc^*})$ which are clearly of different signs no matter how y is chosen. In other words the non-strict LMI $F(y) \geq 0$ has no finite solution, but the non-strict versions of the conditions (2.16) both read

$$\begin{bmatrix} 1 & 1 \end{bmatrix}^{\vdash *} \begin{bmatrix} 0 & c \\ c^* & c+c^* \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix}^{\vdash} = \begin{bmatrix} z \\ -z \end{bmatrix}^* \begin{bmatrix} 0 & c \\ c^* & c+c^* \end{bmatrix} \begin{bmatrix} z \\ -z \end{bmatrix} = 0 \geq 0,$$

where z is an arbitrary non-zero number.

2.3.1 LMI solvers

Several software packages are available for solving LMIs. The most widely used is probably the LMI toolbox for MatLab [Gahinet et al., 1995]. Free alternatives are LMITOOL [Nikoukhah et al., 1995] and sdpsol [Wu and Boyd, 1996]. The MatLab toolbox can solve three different types of problems in addition to an explicit solution of (2.15). The first is the *feasibility problem*:

- find, if it exists, a solution x to the LMI $F(x) > 0$.

The second problem is the *linear objective minimisation problem*:

- $\min_x \{cx : F(x) > 0\}$.

This is a generalisation of the *eigenvalue problem*: $\min_{\lambda,x}\{\lambda : \lambda I - F(x) > 0, G(x) > 0\}$.

The third problem that can be solved by the MatLab toolbox is the *generalised eigenvalue problem*:

- $\min_{\lambda,x}\{\lambda : \lambda G(x) - F(x) > 0, G(x) > 0, H(x) > 0\}$.

The first two problems are convex and are solved by interior point methods. The third is only quasi-convex but can be solved by similar techniques [Gahinet et al., 1995].

It is beyond the scope of this thesis to discuss how these methods work. For an introduction to interior point algorithms in convex programming see [Nesterov and Nemirovski, 1994] or [Nemirovski and Gahinet, 1994] describing the algorithm used in the toolbox.

2.3.2 Examples of LMIs in control

Two examples of LMIs arising in connection with control problems will be given here. For more examples see [Boyd et al., 1994] and [Packard et al., 1991].

Lyapunov stability

Consider the autonomous system

$$\dot{x} = Ax,$$

where A is a constant square matrix. This system is Lyapunov stable if and only if there exists a positive definite *Lyapunov matrix* X such that

$$A^*X + XA < 0.$$

A test for stability can therefore be cast as a feasibility problem in X:

$$\begin{bmatrix} -A^*X - XA & 0 \\ 0 & X \end{bmatrix} > 0.$$

Riccati inequality

$\mathcal{H}_\infty$ control problems often lead to *Riccati inequalities* such as

$$A^*X + XA + X(BB^* - DD^*)X + C^*C < 0, \quad X > 0.$$

Because of the quadratic term this is not an LMI in X. However, if we assume $BB^* > DD^*$ (meaning $BB^* - DD^* > 0$) then we can use Schur complement (Lemma 2.6) to arrive at the equivalent

$$\begin{bmatrix} A^*X + XA + C^*C & X \\ X & -(BB^* - DD^*)^{-1} \end{bmatrix} < 0, \quad X > 0 \tag{2.17}$$

which is indeed an LMI in X. If on the other hand $BB^* - DD^*$ is indefinite then we cannot use this trick. But if C^*C is non-singular (and consequently positive definite) then we multiply from both sides by $Y = X^{-1}$ to arrive at

$$YA^* + AY + (BB^* - DD^*) + YC^*CY < 0, \quad Y > 0.$$

This can then be converted into an LMI in Y. A problem with this last approach arises if (2.17) is just one of several constraints on X even if the other constraints are LMIs. Then we will have LMI constraints on X and Y but also the constraint $X^{-1} = Y$ which is not convex. This problem arises in most robust synthesis problems.

2.4 Linear system interconnection

This section is a short introduction to the interconnection of linear systems. The concepts of linear fractional transformations and Redheffer star product will be described. For further details see [Doyle et al., 1991], [Zhou et al., 1996], and [Helmersson, 1995].

Let $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ be a complex matrix. Then the *lower linear fractional transformation* (LFT) with respect to Δ_l is defined as [Doyle et al., 1991]

$$\mathcal{F}_l(M, \Delta_l) \triangleq M_{11} + M_{12}\Delta_l(I - M_{22}\Delta_l)^{-1}M_{21}.$$

The *upper linear fractional transformation* with respect to Δ is defined similarly as

$$\mathcal{F}_u(M, \Delta_u) \triangleq M_{22} + M_{21}\Delta_u(I - M_{11}\Delta_u)^{-1}M_{12}.$$

The transformations are only defined if the inverses exist.

Definition 2.19 (well-posed, well defined)

The lower LFT $\mathcal{F}_l(M, \Delta_l)$ is said to be well-posed *(or* well defined*) if $I - M_{22}\Delta_l$ is non-singular. The upper LFT $\mathcal{F}_u(M, \Delta_u)$ is said to be well-posed if $I - M_{11}\Delta_u$ is non-singular.*

The LFTs are another formulation of the interconnections in Figure 2.1. Often M describes a known and time-invariant system and Δ contains unknown or time-varying gains.

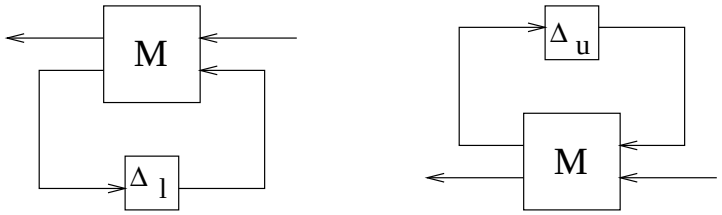


Figure 2.1: *Lower and upper linear transformations.*

The *Redheffer star product*, $\star$, represents the interconnection in Figure 2.2, i.e.

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = (A \star B) \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

Note that $A \star B$ depends on a partitioning of A and B. This partitioning will always be clear from the context.

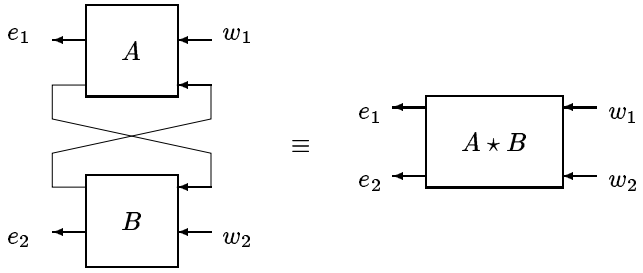


Figure 2.2: Redheffer star product.

Let the partitioning be given by $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and $B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}$. Then we have [Doyle et al., 1991]

$$A \star B = \begin{bmatrix} \mathcal{F}_l(A, B_{11}) & A_{12}(I - B_{11}A_{22})^{-1}B_{12} \\ B_{21}(I - A_{22}B_{11})^{-1}A_{21} & \mathcal{F}_u(B, A_{22}) \end{bmatrix}.$$

Again we need the inverses to be defined.

Definition 2.20 (well-posed,well defined)

We say that the interconnection $A \star B$ is well-posed if $I - B_{11}A_{22}$ and $I - A_{22}B_{11}$ are non-singular.

Remark 2.21 The star product is defined even if for instance e_1 and w_1 are of zero size, i.e. $A = A_{22}$. Then $A \star B = \mathcal{F}_u(B, A_{22})$.

The star product is associative [Helmersson, 1995] i.e.

$$(A \star B) \star C = A \star (B \star C).$$

Furthermore $\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$ is the *unit element* of the star product or *star product identity*, i.e.

$$A \star \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} = A = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \star A.$$

These two facts can be used for transforming problems into special forms. The following example demonstrates how to do this if a given design method requires a system to be strictly proper. The example system has more inputs and outputs than needed to demonstrate the idea, but it has been chosen this way, so that it can be directly applied to the theory described in Section 5.3.

Example 2.22 The synthesis method described in Section 5.3 requires the system

$$\left[\begin{array}{c} \dot{x} \\ \hline z_u \\ z_p \\ y \end{array}\right] = \left[\begin{array}{c|ccc} A & B_u & B_p & B \\ \hline C_u & D_{uu} & D_{up} & E_u \\ C_p & D_{pu} & D_{pp} & E_p \\ C & F_u & F_p & F_3 \end{array}\right] \left[\begin{array}{c} x \\ \hline w_u \\ w_p \\ u \end{array}\right] \tag{2.18}$$

to be strictly proper in the channel from the control signals, u, to the measurements, y, i.e. F_3 must be 0. Then (under some assumptions) a suboptimal controller on the form

$$\left[\begin{array}{c} \dot{x}_c \\ \hline u \\ z_c \end{array}\right] = \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2} \\ \hline C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & D_{c22} \end{array}\right] \left[\begin{array}{c} x_c \\ \hline y \\ w_c \end{array}\right] \tag{2.19}$$

can be constructed. The meaning of the signals w_u, w_p, w_c, z_u, z_p, and z_c is unimportant for now, and will be discussed in Chapter 5.

In practice it may happen that the system we wish to design a controller for is not strictly proper. This problem can be overcome by finding a controller $\tilde{K}$ for the corresponding system with $F_3 = 0$ and then transforming the controller into another controller K yielding the same closed loop system for the actual system. Denote the system matrix in (2.18) by M_s and define

$$\Gamma = \begin{bmatrix} 0 & I \\ I & -F_3 \end{bmatrix}, \quad \Gamma^{-\star} = \begin{bmatrix} 0 & I \\ I & F_3 \end{bmatrix}$$

and observe that $\Gamma \star \Gamma^{-\star} = \Gamma^{-\star} \star \Gamma = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}$ is the Redheffer star product identity, and that

$$M_p \triangleq M_s \star \Gamma = \left[\begin{array}{c|ccc} A & B_u & B_p & B \\ \hline C_u & D_{uu} & D_{up} & E_u \\ C_p & D_{pu} & D_{pp} & E_p \\ C & F_u & F_p & 0 \end{array}\right].$$

Now assume that a controller, $\tilde{K}$, has been obtained for the system defined by M_p, and write this controller as

$$\begin{bmatrix} u \\ \dot{x}_c \\ z_c \end{bmatrix} = \tilde{K}_c \begin{bmatrix} y \\ x_c \\ w_c \end{bmatrix}, \qquad \tilde{K}_c = \begin{bmatrix} D_{c1} & C_{c1} & D_{c12} \\ B_{c1} & A_c & B_{c2} \\ D_{c21} & C_{c2} & D_{c2} \end{bmatrix}.$$

Then, assuming that $I + F_3\tilde{K}_c$ is non-singular, the closed loop is given by

$$\begin{bmatrix} \dot{x} \\ z_u \\ z_p \\ \dot{x}_c \\ z_c \end{bmatrix} = M_c \begin{bmatrix} x \\ w_u \\ w_p \\ x_c \\ w_c \end{bmatrix}$$

in which

$$\begin{aligned} M_c &\triangleq M_p \star \tilde{K}_c = M_p \star (\Gamma^{-\star} \star \Gamma) \star \tilde{K}_c \\ &= (M_p \star \Gamma^{-\star}) \star (\Gamma \star \tilde{K}_c) = M_s \star (\Gamma \star \tilde{K}_c). \end{aligned}$$

A controller for the system defined by M_s is thus given by

$$\begin{bmatrix} u \\ \dot{x}_c \\ z_c \end{bmatrix} = K_c \begin{bmatrix} y \\ x_c \\ w_c \end{bmatrix}, \quad K_c = (\Gamma \star \tilde{K}_c). \tag{2.20}$$

2.5 Multi-layer perceptron

This section provides a short introduction to a particular type of artificial neural network, the multi-layer perceptron (MLP), which can be used to model nonlinear functions. It is beyond the scope of this thesis to go into detail with this subject. For a more thorough introduction see for instance [Suykens et al., 1996] or [Bendtsen, 1999] and the references therein. First the MLP structure will be discussed.

It is then discussed specifically how to obtain nonlinear state space models from input and output measurements using the MLP structure.

The *multi-layer perceptron* (MLP) is composed of layers of perceptrons coupled in parallel. A *perceptron* consists of a memoryless scalar function, the *neuron function*, acting on a weighted sum of input signals, as shown in Figure 2.3.

The neuron function (or simply *neuron*) can be either linear or nonlinear. Some of the traditional neuron functions in MLPs are the unit gain

$$\phi_{lin}(x) = x$$

and the *hyperbolic tangent*

$$\phi_{tanh}(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

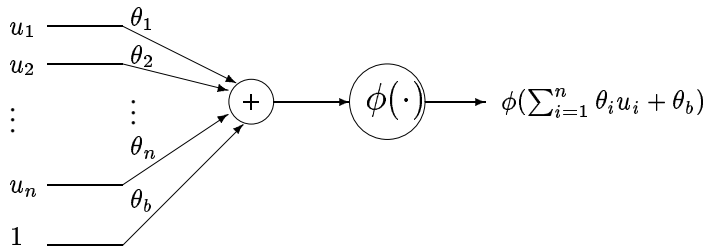


Figure 2.3: A single perceptron.

In this thesis only these two will be used.

A block diagram of an MLP with one hidden layer is shown in Figure 2.4. The input vector z_{in} is multiplied by the *input weight matrix* Θ_1. The *bias vector* Θ_b is then added and the sum is input to ϕ containing a number of neurons in parallel. The resulting output is multiplied by the *output weight matrix* Θ_2 producing the final output z_{out}. Notice that the weights $\theta_1, ..., \theta_n$ in Figure 2.3 form a row in the matrix Θ_1 and that the weight θ_b is an element in Θ_b. The output weight matrix Θ_2 can be seen as stemming from a layer with linear neurons and no biases. The resulting function is

$$z_{out} = \Theta_2 \phi(\Theta_1 z_{in} + \Theta_b) \triangleq M_m(z_{in}, \Theta_1, \Theta_2, \Theta_b). \tag{2.21}$$

By choosing Θ_1, Θ_2, and Θ_b appropriately the MLP can be used to approximate a given static nonlinear function. It has been shown [Hornik et al., 1989] that with a sufficient number of neurons and under certain continuity conditions, the MLP with one hidden layer can act as a universal approximator.

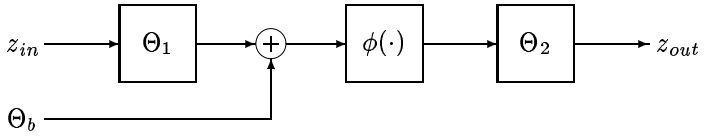


Figure 2.4: Block diagram of an MLP with one hidden layer.

2.5.1 Training

Adjusting the weights of a neural network is known as *training*. For an MLP this is typically done by trying to approximate a set of *output targets* in the following manner. Given a set of target output vectors $z_{t,1}, ...z_{t,m}$ and a corresponding set of input vectors $z_{in,1}, ..., z_{in,m}$ define the approximation error

$$\epsilon(m) \triangleq z_{t,m} - z_{out,m} = z_{t,m} - M_m(z_{in,m}, \Theta_1, \Theta_2, \Theta_b)$$

and the quadratic performance functional

$$J = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{2}\epsilon(i)^T\epsilon(i).$$

The aim is now to minimise the performance functional over the weights and biases, i.e.

$$\min_{\Theta_1,\Theta_2,\Theta_b} J.$$

Remark 2.23 Minimising the performance functional for an MLP with nonlinear neuron functions is a non-convex problem and problems with local minima can arise.

The simplest way to attempt to minimise J is by the *Back Propagation Error Algorithm* (BPEA). This is an iterative method, where the weights and biases are updated by the following rule

$$\theta_{i+1} = \theta_i - \eta\frac{dJ}{d\theta_i},$$

where $\eta > 0$ is the step size and Θ_i is the collection of all elements of Θ_1, Θ_2, or Θ_b at the i'th iteration. This corresponds to going in the opposite direction of the gradient, i.e. to move downhill until the bottom is reached. This method is very stable but also very slow. Including the second order derivative as in the Gauss-Newton learning rule

$$\theta_{i+1} = \theta_i - \left(\frac{d^2J}{d\theta_i d\theta_i^T}\right)^{-1}\frac{dJ}{d\theta_i}$$

increases the convergence rate drastically. This corresponds to approximation the performance function by a paraboloid and jumping directly to the bottom. For linear neuron functions this will immediately yield the global minimum solution. The Levenberg-Marquardt algorithm combines these two to obtain the fast convergence of the Gauss-Newton algorithm with the stability of the BPEA. See [Suykens et al., 1996] and the references therein for a more thorough description of training algorithms.

2.5.2 MLPs as state space models

Assume that we wish to obtain a model of the discrete time nonlinear state space system

$$y_{k+1} = f(y_k, ..., y_{k-n_y+1}, u_k, ..., u_{k-n_u+1}), \tag{2.22}$$

where y is a measured output and u is a known input. This is known as a nonlinear autoregressive model with exogenous inputs (NARX).

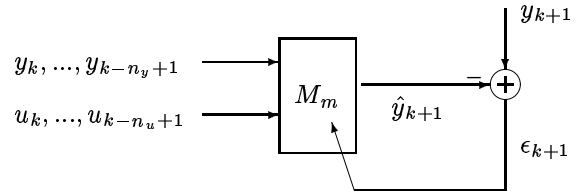


Figure 2.5: MLP as state space model.

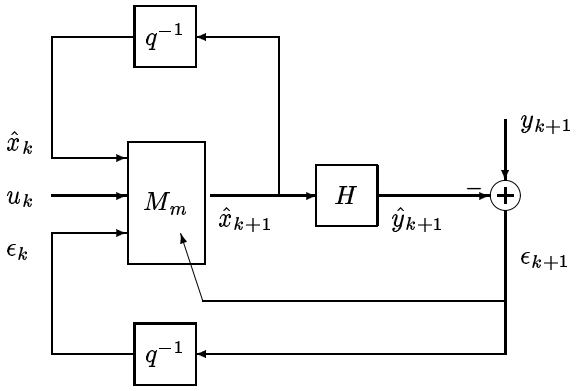


Figure 2.6: Recurrent MLP as state space model.

A way to obtain a model for this system is shown in Figure 2.5. Old measurements of outputs and inputs are fed to the MLP, which provides a prediction of $y(k+1)$. The prediction error $\epsilon(k+1)$ can then be used to adjust the parameters in the MLP. Using a large number of measurements to train the MLP, the effects of white measurement noise can be removed, yielding a model with a prediction error with the same variance as the noise assuming that the MLP has enough neurons to model the system, and that local minima are avoided. A more thorough discussion of this model structure can be found in e.g. [Lightbody and Irwin, 1996].

Once a model has been obtained it could for instance be used as a predictor as already seen. By feeding the predicted outputs back into the MLP, it can also be used as an open-loop simulator of the system. Thirdly, it can be used as a system model for a control design.

The model type (2.22) is somewhat limited due to the fact that the noise must be white.

A more general system structure is

$$x_{k+1} = f(x_k) + g(x_k)u_k + g_d(x_k)d_k, \qquad y_k = Hx_k, \tag{2.23}$$

where x is the state, d_k is white noise, and H is a known matrix. This is a specialised version of the nonlinear autoregressive moving average model with exogenous inputs (NARMAX). An approach to training of this type of model is illustrated by Figure 2.6, where q^{-1} is the delay operator. The MLP provides a state estimate, which is delayed and fed back into the MLP. This is known as a *recurrent MLP*. The output prediction error is again used to adjust the MLP parameters but are also used as inputs to the MLP. In this way it is possible to include more general noise types. The state estimates, $\hat{x}$, do not necessarily correspond directly to the actual states, x. Notice that some of the state estimates could for instance be delayed version of the prediction error.

If there is no noise, it has been shown [Siegelmann et al., 1997] that there is no loss of generality in the mappings that can be achieved by assuming that the state estimates $\hat{x}$ are delayed versions of the output y. But when the noise is not white, then the more general training structure in Figure 2.6 must be used. It is beyond the scope of this thesis to go into details with how the parameters are adjusted. For a discussion of this model and its training see e.g. [Korbicz and Janczak, 1996] or [Bendtsen, 1999]. It should just be noted that from input and output measurements it is possible to obtain a state space model of very general nonlinear systems. Unfortunately there are no training rules for the NARMAX model guaranteeing convergence.

The MLP will be used in Chapter 6 to obtain a model of the induction motor to be used for speed control.

2.6 Summary

This chapter has given an introduction to some of the basic concepts used in this thesis. A matrix inequality is an expression $M(x) > 0$, where M is a Hermitian matrix function of the decision variables x. $M > 0$ means that all the eigenvalues of M are positive. There is in general no way to solve matrix inequalities, but if M is a quadratic function of x with a certain structure and certain inertia properties are fulfilled, then it is possible to construct a solution.

Another case is if M depends affinely on x. Then $M(x) > 0$ is a linear matrix inequality (LMI) and fast and efficient software solvers exist. If a problem can be formulated as an LMI it can therefore be considered solved. Examples of LMIs arising in control problems were given.

The star product, $\star$, is used to denote interconnection of systems. It was demonstrated how the associativity of the star product could be used to transform certain problems into ones with a simpler form.

Finally, the multi-layer perceptron and its application as nonlinear state space model was discussed.

Chapter 3

INDUCTION MOTOR SYSTEM

This chapter describes the system considered in this thesis. First, in Section 3.1, a model for an induction motor with squirrel cage rotor and three star connected stator windings is derived. A number of assumptions are made in order to obtain a simple model. The electro-magnetic model is developed using complex space vector representation, where three real signals are combined into one complex signal. The result is a complex third order nonlinear model. In Section 3.2 the model is written in a stator-fixed reference frame.

In Section 3.3 the part of the model describing the currents is written as a complex second order state space model with the shaft speed as a time-varying parameter. The concept of rotating reference frames is then discussed. This will be used in Chapter 4.

In Section 3.4 the uncertainty on some of the motor parameter values is discussed. The resistances can change due to temperature variations, and the shaft speed can also be considered a time varying parameter.

In order to control the speed of the motor it is necessary to use a power device. In Section 3.5 a commonly used type of power device, the voltage sourced inverter, is described.

Experiments will be performed on a laboratory system with a $1.5kW$ induction motor. This system is discussed in Section 3.7. The laboratory system includes a DC-motor connected to the shaft in order to allow simulation of load torques.

3.1 Induction motor model

This section describes the dynamic model of a symmetrical three-phase induction motor with a squirrel cage rotor. The description is mainly based on [Leonhard, 1990], [Rasmussen, 1995], and [Kaźmierkowski and Tunia, 1994].

The induction motor mainly consists of two parts, the *stator* and the *rotor*. The rotor is rotating inside the stator separated by an *air gap*, as shown by the cross section in Figure 3.1. The rotor is in principle built from parallel conductors short-circuited by a ring at each end, as illustrated in Figure 3.2.

The windings of the three stator coils (A, B and C) are parallel to the rotor bars and distributed sinusoidally around the cylinder displaced by 120 degrees, so that the total number of windings at each angle is approximately constant. Figure 3.1 illustrates the distribution of winding A by the width of the gray area. The stator shown is of the one pole pair type, meaning that the coils will produce one magnetic north and one magnetic south pole. Often a motor will be constructed with several pole pairs by connecting coils in parallel and displacing the coils by $120/Z_p$ degrees, where Z_p is the number of pole pairs. This works as a gearing giving a larger torque and a slower mechanical rotational speed. The derivation of the electrical equations will be for a one pole pair motor. Adapting to multiple pole pairs is simply a question of modifying the mechanical model as in Section 3.1.4.

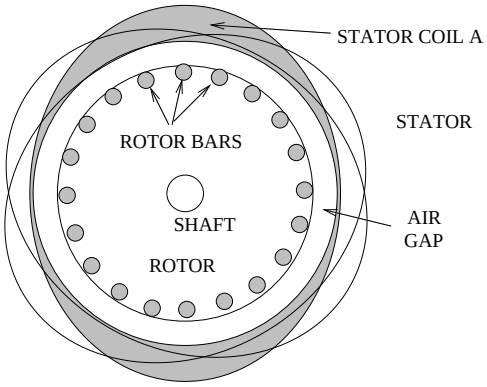


Figure 3.1: *Cross section of induction motor*

The stator windings are fed sinusoidal voltages to create a rotating magnetic field. When the rotor and the magnetic field of the stator rotate at different speeds, currents will be induced in the rotor rods. These currents result in a magneto-motive force perpendicular to the current and to the magnetic field resulting in a torque on the rotor.

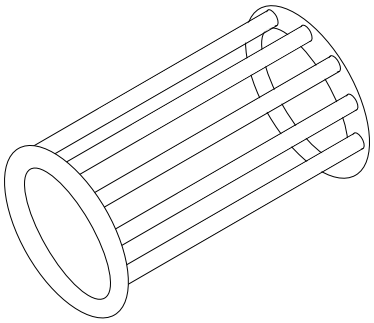


Figure 3.2: *Squirrel cage rotor*

3.1.1 Modelling assumptions

A number of assumptions are made in order to permit a simple model to be obtained.

1. The motor is symmetrical.
2. The rotor is concentric and the air gap has a constant width, h.
3. Only the basic harmonics of the spatial field distribution and of the magneto-motive force in the air gap are considered.
4. The stator windings are star connected (see Figure 3.3) and the neutral is isolated.
5. The ends of the rotor bars are short circuited.
6. The permeability of the iron parts is infinite.
7. The flux density is radial in the air gap.
8. Slotting effects, iron losses and end-effects are negligible.
9. The effects of anisotropy, magnetic saturation and eddy currents are negligible.
10. The coil resistances and reactances are constant or slowly varying.

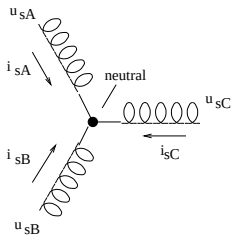


Figure 3.3: *Stator coils in star connection.*

One simplification resulting from these assumptions is that the rotor can be considered as consisting of 3 short-circuited windings distributed in the same way as the stator coils [Leonhard, 1990, page 152]. In the following these virtual coils will be referred to as the *rotor coils* or *rotor windings*. The currents in these coils are referred to as i_{ra}, i_{rb} and i_{rc}. Since the rotor currents cannot be measured, the number of virtual rotor windings can be chosen arbitrarily.

3.1.2 Electro-magnetic model

The electro-magnetic model describes the torque on the rotor as a function of the stator currents i_{sA}, i_{sB} and i_{sC}. Due to the isolated neutral

$$i_{sA}(t) + i_{sB}(t) + i_{sC}(t) = 0 \tag{3.1}$$

is valid at any instant.

The distribution of the stator coils results in the magneto-motive force wave excited by the stator currents at an angle θ being (see Figure 3.4)

$$f_s(\theta, t) = N_s \left(i_{sA}(t)\cos(\theta) + i_{sB}(t)\cos(\theta - 2\pi/3) + i_{sC}(t)\cos(\theta - 4\pi/3) \right).$$

Note that mathematically it may be more correct to write the above equation as

$$f_s(i_{sA}, i_{sB}, i_{sC}, \theta) = N_s \left(i_{sA}\cos(\theta) + i_{sB}\cos(\theta - 2\pi/3) + i_{sC}\cos(\theta - 4\pi/3) \right),$$

since it is not a time-varying function, but the former notation is the one used in most of the literature on the subject and this style will be adopted throughout this chapter.

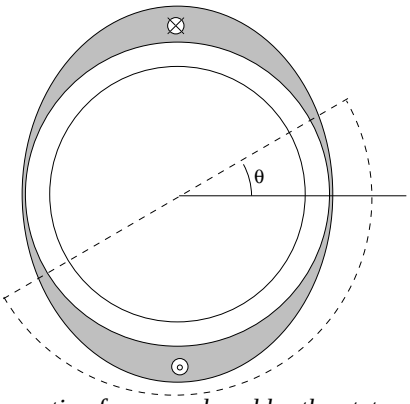


Figure 3.4: *The magneto-motive force produced by the stator currents at the angle θ is the sum of all currents inside the semi-circle.*

N_s is the number of windings on each coil. Likewise the magneto-motive force wave excited by the rotor currents, at the angle θ is

$$f_r(\theta,t) = N_r(i_{ra}(t)cos(\theta-\theta_r(t)) + i_{rb}(t)cos(\theta-\theta_r(t)-2\pi/3) \\ +i_{rc}(t)cos(\theta-\theta_r(t)-4\pi/3)).$$

θ_r is the electrical angle of rotation of the rotor. For a one pole pair machine this angle is the same as the mechanical angle θ_{mech} ($\theta_r = Z_p\theta_{mech}$). N_r is a fictive number of windings on the rotor coils.

As the permeability of the iron is assumed infinite the magneto-motive force is effective only at the air gap, giving the flux density on the stator side:

$$B_s(\theta,t) = \frac{\mu_0}{2h}\left(f_s(\theta,t) + \kappa f_r(\theta,t)\right), \tag{3.2}$$

where μ_0 is the vacuum permeability constant and κ is a coupling factor compensating for magnetic leakage. On the rotor side the flux density is given by

$$B_r(\theta,t) = \frac{\mu_0}{2h}\left(\kappa f_s(\theta,t) + f_r(\theta,t)\right). \tag{3.3}$$

The part of the flux density on the rotor surface due to the stator currents is

$$B_{rs}(\theta,t) = \frac{\kappa\mu_0}{2h} f_s(\theta,t). \tag{3.4}$$

The current distribution along the surface of the rotor, a_r, is the derivative of the rotor magneto-motive force:

$$a_r(\theta,t) = \frac{1}{2r}\frac{\partial f_r(\theta,t)}{\partial \theta}, \tag{3.5}$$

where r is the radius of the rotor. The tangential force df acting on an axial strip of width $rd\beta$ is the (vector) product of the flux density and the current distribution:

$$df = -B_{rs}(\theta,t)a_r(\theta,t)lrd\beta, \tag{3.6}$$

where l is the length of the rotor. Integrating this gives the electro-magnetic torque in the direction of rotation

$$m_e(t) = r \int\limits_{\text{Surface}} df = -lr^2 \int\limits_0^{2\pi} B_{rs}(\theta,t)a_r(\theta,t)d\theta. \tag{3.7}$$

The flux linkage in stator coil A, Ψ_{sA}, is the integrated effect of the stator flux through all loops of coil A. The loop formed by the conductors at angles $\lambda - \pi/2$ and $\lambda + \pi/2$ is penetrated by the field lines passing through the stator between these angles. Assuming a continuous distribution of turns with the incremental density $\frac{1}{2}N_s \cos\lambda$ (for coil A) at this angle, the flux linkage can be obtained as

$$\Psi_{sA}(t) = \frac{1}{2}N_s \int\limits_{\lambda=-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos\lambda\{\int\limits_{\theta=\lambda-\frac{\pi}{2}}^{\lambda+\frac{\pi}{2}} lrB_s(\theta,t)d\theta\}d\lambda. \tag{3.8}$$

The flux linkages in the rotor coils are found in the same manner:

$$\Psi_{ra}(t) = \frac{1}{2}N_r \int\limits_{\lambda=-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos\lambda\{\int\limits_{\theta=\lambda-\frac{\pi}{2}+\theta_r}^{\lambda+\frac{\pi}{2}+\theta_r} lrB_r(\theta,t)d\theta\}d\lambda. \tag{3.9}$$

The corresponding equations for the other coils are:

$$\Psi_{sB}(t) = \frac{1}{2}N_s \int\limits_{\lambda=-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\lambda - 2\pi/3)\{\int\limits_{\theta=\lambda-\frac{\pi}{2}}^{\lambda+\frac{\pi}{2}} lrB_s(\theta,t)d\theta\}d\lambda, \tag{3.10}$$

$$\Psi_{sC}(t) = \frac{1}{2}N_s \int\limits_{\lambda=-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\lambda - 4\pi/3)\{\int\limits_{\theta=\lambda-\frac{\pi}{2}}^{\lambda+\frac{\pi}{2}} lrB_s(\theta,t)d\theta\}d\lambda, \tag{3.11}$$

$$\Psi_{rb}(t) = \frac{1}{2} N_r \int_{\lambda=-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\lambda - 2\pi/3) \{ \int_{\theta=\lambda-\frac{\pi}{2}+\theta_r}^{\lambda+\frac{\pi}{2}+\theta_r} lrB_r(\theta,t)d\theta \} d\lambda, \tag{3.12}$$

$$\Psi_{rc}(t) = \frac{1}{2} N_r \int_{\lambda=-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos(\lambda - 4\pi/3) \{ \int_{\theta=\lambda-\frac{\pi}{2}+\theta_r}^{\lambda+\frac{\pi}{2}+\theta_r} lrB_r(\theta,t)d\theta \} d\lambda. \tag{3.13}$$

The input to neutral voltage in the stator coils are assumed to be given by:

$$u_{sA}(t) = R_s i_{sA}(t) + \frac{d}{dt}\Psi_{sA}(t), \tag{3.14}$$

$$u_{sB}(t) = R_s i_{sB}(t) + \frac{d}{dt}\Psi_{sB}(t), \tag{3.15}$$

$$u_{sC}(t) = R_s i_{sC}(t) + \frac{d}{dt}\Psi_{sC}(t), \tag{3.16}$$

where the stator resistance, R_s, is the resistance of the stator windings, and the equivalent equations for rotor coils are:

$$\begin{aligned} 0 &= R_r i_{ra}(t) + \frac{d}{dt}\Psi_{ra}(t), \\ 0 &= R_r i_{rb}(t) + \frac{d}{dt}\Psi_{rb}(t), \\ 0 &= R_r i_{rc}(t) + \frac{d}{dt}\Psi_{rc}(t), \end{aligned}$$

where the rotor resistance, R_r, is the resistance of the (virtual) rotor windings.

3.1.3 Complex space vector notation

To simplify the equations a complex notation based on the value $\alpha_{sv} \triangleq e^{j2\pi/3}$ is introduced. The complex space vector of the stator current is defined as

$$\bar{i}_s(t) \triangleq \frac{2}{3}(i_{sA}(t) + \alpha_{sv} i_{sB}(t) + \alpha_{sv}^2 i_{sC}(t)). \tag{3.17}$$

For the rotor currents a space vector is defined as well:

$$\bar{i}_r(t) \triangleq \frac{2}{3}(i_{ra}(t) + \alpha_{sv} i_{rb}(t) + \alpha_{sv}^2 i_{rc}(t)). \tag{3.18}$$

Notice that no information is lost in this transformation from three to two degrees of freedom. Using (3.1) the three stator currents can be reconstructed from the space vector. Similarly the rotor currents can be reconstructed due to $i_{ra} + i_{rb} + i_{rc} = 0$.

The space vector for the stator voltage is

$$\bar{u}_s(t) \triangleq \frac{2}{3}(u_{sA}(t) + \alpha_{sv} u_{sB}(t) + \alpha_{sv}^2 u_{sC}(t)). \tag{3.19}$$

The three stator voltages cannot be reconstructed from this value without adding an additional demand, for instance $u_{sA}(t) + u_{sB}(t) + u_{sC}(t) = c$, where c is some chosen constant, but due to the isolated neutral the value of this constant is of no importance.

We also define space vectors for the stator and rotor flux linkages:

$$\bar{\Psi}_s(t) \triangleq \frac{2}{3}(\Psi_{sA}(t) + \alpha_{sv} \Psi_{sB}(t) + \alpha_{sv}^2 \Psi_{sC}(t)), \tag{3.20}$$

$$\bar{\Psi}_r(t) \triangleq \frac{2}{3}(\Psi_{ra}(t) + \alpha_{sv} \Psi_{rb}(t) + \alpha_{sv}^2 \Psi_{rc}(t)). \tag{3.21}$$

Inserting equations (3.8), (3.10), and (3.11) in (3.20) after some calculations gives:

$$\bar{\Psi}_s(t) = L_s \bar{i}_s(t) + L_m \bar{i}_r(t) e^{j\theta_r(t)}, \tag{3.22}$$

where the stator inductance, L_s, and the mutual inductance, L_m are

$$L_s \triangleq 3\pi\mu_o \frac{lrN_s^2}{8h}, \quad L_m \triangleq 3\pi\mu_o \frac{lrN_s N_r}{8h}\kappa. \tag{3.23}$$

A similar equation can be obtained for the rotor:

$$\bar{\Psi}_r(t) = L_r \bar{i}_r(t) + L_m \bar{i}_s(t) e^{-j\theta_r(t)}, \tag{3.24}$$

where the rotor inductance, L_r, is

$$L_r \triangleq 3\pi\mu_o \frac{lrN_r^2}{8h}. \tag{3.25}$$

Combining (3.14)-(3.16) with the space vector definitions (3.19)-(3.17) results in the simple equation:

$$\bar{u}_s(t) = R_s \bar{i}_s(t) + \frac{d\bar{\Psi}_s(t)}{dt}. \tag{3.26}$$

A similar result can be obtained for the rotor:

$$0 = R_r \bar{i}_r(t) + \frac{d\bar{\Psi}_r(t)}{dt}. \tag{3.27}$$

By inserting space vector expressions in the flux density equation (3.4) and inserting this in the torque equation (3.7) the integral can be calculated to get the following expression for the electro-magnetic torque tangential to the rotation:

$$m_e(t) = \frac{3}{2} Z_p L_m \Im\{\bar{i}_s(t)(\bar{i}_r(t)e^{j\theta_r(t)})^*\}. \tag{3.28}$$

Inserting (3.22) and (3.24) in (3.26) and (3.27) gives:

$$\bar{u}_s(t) = R_s \bar{i}_s(t) + L_s \frac{d}{dt}\bar{i}_s(t) + L_m \frac{d}{dt}(\bar{i}_r(t)e^{j\theta_r(t)}) \tag{3.29}$$

$$0 = R_r \bar{i}_r(t) + L_r \frac{d}{dt}\bar{i}_r(t) + L_m \frac{d}{dt}(\bar{i}_s(t)e^{-j\theta_r(t)}). \tag{3.30}$$

3.1.4 Mechanical system

The mechanical rotational speed ω_{mech} is affected by the electro-magnetic torque m_e and the load torque m_L:

$$\dot{\omega}_{mech}(t) = \frac{1}{J}(m_e(t) - m_L(t)), \tag{3.31}$$

where J is the collective moment of inertia of the rotor and the load, assuming the shaft to be rigid. m_L contains the actual load along with the speed-dependent viscous and coulomb friction. The electrical rotational speed is defined as

$$\omega_r(t) = \dot{\theta}_r(t) = Z_p \omega_{mech}(t). \tag{3.32}$$

Equations (3.28)-(3.32) form the model to be used below.

3.2 Stator-fixed coordinates

The model of the induction motor can be expressed in various coordinate systems. Expressing the model in a coordinate system which rotates with the rotor or stator flux gives a model in which the states are constant in steady state operation (constant ω_r and m_L). This model is often desirable for control purposes but in order to perform the non-linear change of coordinates it is necessary to know the rotor flux angle. For flux estimation purposes it is therefore often desirable to work with a model in stator-fixed coordinates.

In (3.29) and (3.30) $\bar{u}_s$ and $\bar{i}_s$ are already in stator-fixed coordinates, while $\bar{i}_r$ is in rotor-fixed coordinates. By defining the stator-fixed rotor current

$$\bar{i}_{rs}(t) \triangleq \bar{i}_r(t)e^{j\theta_r(t)} \tag{3.33}$$

the following stator-fixed model is found:

$$(R_s + L_s\frac{d}{dt})\bar{i}_s(t) + L_m\frac{d}{dt}\bar{i}_{rs}(t) = \bar{u}_s(t), \tag{3.34}$$

$$L_m(\frac{d}{dt} - j\omega_r(t))\bar{i}_s(t) + (R_r + L_r(\frac{d}{dt} - j\omega_r(t)))\bar{i}_{rs}(t) = 0, \tag{3.35}$$

$$m_e(t) = \frac{3}{2}Z_pL_m\Im\{\bar{i}_s(t)\bar{i}_{rs}(t)^*\}, \tag{3.36}$$

$$\dot{\omega}_r(t) = Z_p\dot{\omega}_{mech}(t) = \frac{Z_p}{J}(m_e(t) - m_L(t)). \tag{3.37}$$

To summarise, $\bar{i}_s$ and $\bar{i}_{rs}$ are the stator and rotor currents. $\bar{u}_s$ is the stator voltage, which is often controlled through a voltage sourced inverter as described in Section 3.5. ω_r is the rotational speed of the rotor. m_e is the electro-magnetic torque. m_L is the load torque acting as a disturbance. Z_p, R_s, R_r, L_s, L_r, and L_m are the parameters of the motor, and J is the collective lumped moment of inertia of rotor and load.

3.3 State space model

For several analysis and synthesis methods a state space model is desired. This model will be formulated here. For convenience the time dependency will be dropped in the notation. From (3.34) and (3.35) a complex state space model for the current equations can be obtained:

$$\dot{x}_{src} = A_{src}x_{src} + B_{src}\bar{u}_s,$$
$$x_c = \begin{bmatrix} \bar{i}_s \\ \bar{i}_{rs} \end{bmatrix},$$

$$A_{src} = \begin{bmatrix} \frac{L_rR_s + jL_m^2\omega_r}{L_m^2 - L_rL_s} & \frac{L_m(-R_rL_m + L_rj\omega_r)}{L_m^2 - L_rL_s} \\ \frac{-R_sL_m - jL_sL_m\omega_r}{L_m^2 - L_rL_s} & \frac{L_sR_r - jL_sL_r\omega_r}{L_m^2 - L_rL_s} \end{bmatrix},$$
$$B_{src} = \begin{bmatrix} \frac{L_r}{L_rL_s - L_m^2} \\ \frac{-L_m}{L_rL_s - L_m^2} \end{bmatrix}.$$

The speed ω_r could be included in the state vector as a third state, but the above form has the advantage that the state space part can be seen as a linear parameter varying system with ω_r as the varying parameter, see Chapter 5.

3.3.1 State transformations

For control purposes it is often desirable to work with another state vector than $\begin{bmatrix} \bar{i}_s \\ \bar{i}_{rs} \end{bmatrix}$. A state transformation is obtained by multiplication of a state transformation matrix, T,

$$x_{new} = Tx_s, \qquad A_{new} = TA_{src}T^{-1}, \qquad B_{new} = TB_{src} \tag{3.38}$$

An important choice of states is

$$x_{sc} \triangleq \begin{bmatrix} \bar{i}_s \\ \bar{i}_m \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & \frac{L_r}{L_m} \end{bmatrix} \begin{bmatrix} \bar{i}_s \\ \bar{i}_{rs} \end{bmatrix}. \tag{3.39}$$

The magnetising current, $\bar{i}_m$, has the same angle as the rotor flux

$$\bar{\Psi}_{rs} \triangleq \bar{\Psi}_r e^{j\theta_r} = L_m \bar{i}_m = L_m \bar{i}_s + L_r \bar{i}_{rs}. \tag{3.40}$$

The resulting state space system is

$$\begin{aligned} \dot{x}_{sc} &= A_{sc}x_{sc} + B_{sc}\bar{u}_s, \\ x_{sc} &= \begin{bmatrix} \bar{i}_s \\ \bar{i}_m \end{bmatrix}, \\ A_{sc} &= \begin{bmatrix} \frac{L_m^2 R_r + R_s L_r^2}{L_r(L_m^2 - L_s L_r)} & \frac{L_m^2(jL_r\omega_r - R_r)}{L_r(L_m^2 - L_s L_r)} \\ \frac{R_r}{L_r} & j\omega_r - \frac{R_r}{L_r} \end{bmatrix}, \\ B_{sc} &= \begin{bmatrix} \frac{L_r}{L_s L_r - L_m^2} \\ 0 \end{bmatrix}, \\ m_e &= \frac{3Z_p L_m^2}{2L_r}\Im\{\bar{i}_s(\bar{i}_m - \bar{i}_s)^*\} = \frac{3Z_p L_m^2}{2L_r}\Im\{\bar{i}_s \bar{i}_m^*\}, \\ \dot{\omega}_r &= Z_p\dot{\omega}_{mech} = \frac{Z_p}{J}(m_e - m_L). \end{aligned} \tag{3.41}$$

This particular choice of states has some nice properties for control purposes which will be discussed in Chapter 4.

3.3.2 Real state space model

To obtain a *real* state space model the space vectors are first split into real and imaginary parts:

$$\bar{u}_s = u_{sD} + ju_{sQ}, \tag{3.42}$$

$$\bar{i}_s = i_{sD} + ji_{sQ}, \tag{3.43}$$

$$\bar{i}_m = i_{mD} + j i_{mQ}, \tag{3.44}$$

where u_{sD}, u_{sQ}, i_{sD}, i_{sQ}, i_{mD} and i_{mQ} are real signals. A real state space model can now be obtained by substituting the complex system and input matrices by real matrices of double size

$$A_r + jA_i \rightarrow \begin{bmatrix} A_r & -A_i \\ A_i & A_r \end{bmatrix} \tag{3.45}$$

and by substituting the signal vectors

$$x_r + jx_i \rightarrow \begin{bmatrix} x_r \\ x_i \end{bmatrix}. \tag{3.46}$$

This gives the following model:

$$\dot{x}_{sr} = A_{sr}x_{sr} + B_{sr}u_{sr}$$

$$x_{sr} = \begin{bmatrix} i_{sD} \\ i_{mD} \\ i_{sQ} \\ i_{mQ} \end{bmatrix}, u_{sr} = \begin{bmatrix} u_{sD} \\ u_{sQ} \end{bmatrix}$$

$$A_{sr} = \begin{bmatrix} \frac{L_m^2 R_r + R_s L_r^2}{L_r(L_m^2 - L_s L_r)} & \frac{-L_m^2 R_r}{L_r(L_m^2 - L_s L_r)} & 0 & -\frac{L_m^2 \omega_r}{L_m^2 - L_s L_r} \\ \frac{R_r}{L_r} & -\frac{R_r}{L_r} & 0 & -\omega_r \\ 0 & \frac{L_m^2 \omega_r}{L_m^2 - L_s L_r} & \frac{L_m^2 R_r + R_s L_r^2}{L_r(L_m^2 - L_s L_r)} & \frac{-L_m^2 R_r}{L_r(L_m^2 - L_s L_r)} \\ 0 & \omega_r & \frac{R_r}{L_r} & -\frac{R_r}{L_r} \end{bmatrix},$$

$$B_{sr} = \begin{bmatrix} \frac{L_r}{L_s L_r - L_m^2} & 0 \\ 0 & 0 \\ 0 & \frac{L_r}{L_s L_r - L_m^2} \\ 0 & 0 \end{bmatrix},$$

$$m_e = \frac{3 Z_p L_m^2}{2 L_r}(i_{sQ} i_{mD} - i_{sD} i_{mQ}), \tag{3.47}$$

$$\dot{\omega}_r = Z_p \dot{\omega}_{mech} = \frac{Z_p}{J}(m_e - m_L). \tag{3.48}$$

3.3.3 Rotating reference frames

In normal operation the inputs and states will be rotating, i.e. following circular trajectories in the complex plane. For control and simulation purposes it is often desirable to work with the system in a rotating coordinate system. Defining the signals

$$x_r \triangleq x e^{-j\rho} \qquad u_r \triangleq u e^{-j\rho} \tag{3.49}$$

the system $\dot{x} = Ax + Bu$ can be written as

$$\dot{x}_r = (A - j\omega I)x_r + Bu_r \tag{3.50}$$

where $\omega \triangleq \frac{d}{dt}\rho$.

3.3.4 Steady state

In normal operation the space vectors will rotate around the origin in the complex plane.

Definition 3.1 (Steady state)

The complex state space system

$$\dot{x} = Ax + Bu \tag{3.51}$$

is in steady state if there exists a reference frame rotating at angular velocity ω such that

$$\dot{x}_r = (A - j\omega I)x_r + Bu_r = 0, \text{ and } \dot{u}_r = 0, \tag{3.52}$$

where x_r and u_r are defined as in (3.49).

Let the system (3.41) be in steady state and define $\omega_{mR} = \frac{d}{dt}\angle\bar{i}_s$ such that

$$(A_{sc} - j\omega_{mR})x_{sc} + B_{sc}\bar{u}_s = 0. \tag{3.53}$$

Then $\bar{u}_{ss} \triangleq \bar{u}_s e^{-j\omega_{mR}t}$, $\begin{bmatrix}\bar{i}_{ss}\\ \bar{i}_{ms}\end{bmatrix} \triangleq \begin{bmatrix}\bar{i}_s\\ \bar{i}_m\end{bmatrix} e^{-j\omega_{mR}t}$ are all constant and the states are given from

$$\begin{bmatrix}\bar{i}_{ss}\\ \bar{i}_{ms}\end{bmatrix} = -(A_{sc} - j\omega_{mR})^{-1}B_{sc}\bar{u}_{ss}$$

as

$$\bar{i}_{ss} = \frac{(R_r + j(\omega_{mR} - \omega_r)L_r)\bar{u}_{ss}}{R_sR_r + j(\omega_{mR} - \omega_r)R_sL_r + j\omega_{mR}L_sR_r + \omega_{mR}(\omega_{mR} - \omega_r)(L_m^2 - L_rL_s)} \tag{3.54}$$

$$\bar{i}_{ms} = \frac{R_r \bar{u}_{ss}}{R_s R_r + j(\omega_{mR} - \omega_r) R_s L_r + j\omega_{mR} L_s R_r + \omega_{mR}(\omega_{mR} - \omega_r)(L_m^2 - L_r L_s)}. \tag{3.55}$$

The quantity $\omega_{slip} \triangleq \omega_{mR} - \omega_r$ is known as the *slip frequency* and is related to the torque by

$$m_e = \frac{3 Z_p L_m^2}{2 R_r} |\bar{i}_m|^2 \omega_{slip}. \tag{3.56}$$

3.4 Uncertain and time-varying parameters

The dynamical behaviour of the induction motor is affected by time variations in the parameters.

The rotor resistance R_r can change as much as 50 % due to heating. Furthermore it can be difficult to obtain an accurate estimate of its value especially during steady state operation.

The stator resistance R_s can also change, but the stator windings are usually better ventilated than the rotor windings, so the variations will not be quite as large. In addition obtaining an accurate estimate is easier. Since both of these variations are caused by temperature changes, both R_r and R_s will be *slowly* varying.

The rotational speed ω_r can change due to load disturbances or as a result of a command change to the controller. This variation will typically be fast compared to some of the other dynamics of the motor. Sometimes ω_r (or the position θ_r) is measured, but avoiding the use of a speed (and position) sensor is often desirable, due to the relatively high cost and high sensitivity to the environment of these sensors.

The mutual inductance L_m (and to some extent L_s and L_r) will be affected by magnetic saturation effects when the magnetisation changes. Notice that in the derivation of (3.29) and (3.30) it was assumed that these inductances are constant, so modelling this uncertainty is more complicated than simply assuming the inductances to be time-varying in the above model.

3.5 Power device

In a control context the control objective is usually controlling the angular velocity or position of the rotor shaft. The control signals are usually the stator voltages or currents depending on the type of *power device*. The description here will be limited to a (three-phase bridge) *voltage sourced inverter* (VSI). For a thorough discussion of power devices including current sourced inverters see [Kaźmierkowski and Tunia, 1994].

Figure 3.5 is a sketch of the power device in connection with the induction motor. The DC voltage source converts a three-phase AC supply to a DC voltage which is then supplied to the inverter. The signal conditioning supplies the inverter with a modulation signal to generate the reference stator voltages.

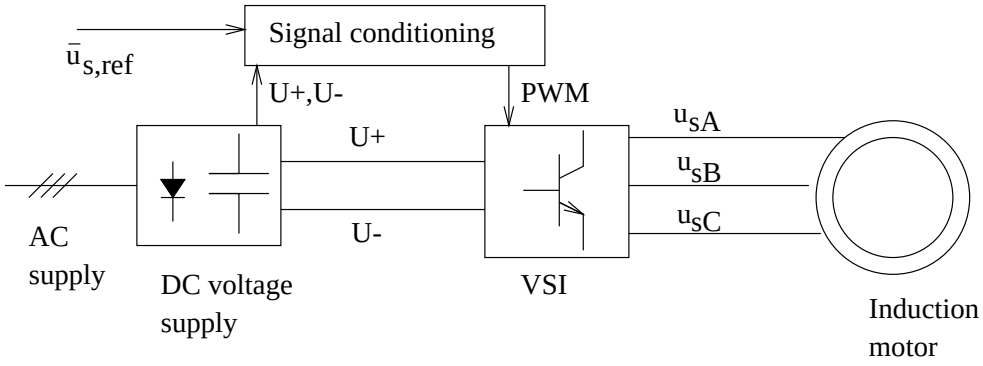


Figure 3.5: *Voltage sourced inverter connected to an induction motor.*

The inverter is sketched in Figure 3.6. By applying pulse width modulation signals to the input terminals $u_{sA,pwm}$, $u_{sB,pwm}$, and $u_{sC,pwm}$ the output voltages u_{sA}, u_{sB}, and u_{sC} can be switched between U^+ and U^-. Since the stator currents only depend on the lower frequency part of the stator voltage this is equivalent to applying a low pass filtered version of the switched voltages. The signal conditioning computes the modulation signals to accomplish $\frac{2}{3}(u_{sA} + \alpha_{sv} u_{sB} + \alpha_{sv}^2 u_{sC}) \approx \bar{u}_{s,ref}$, where the approximation is only considered for the low frequency part.

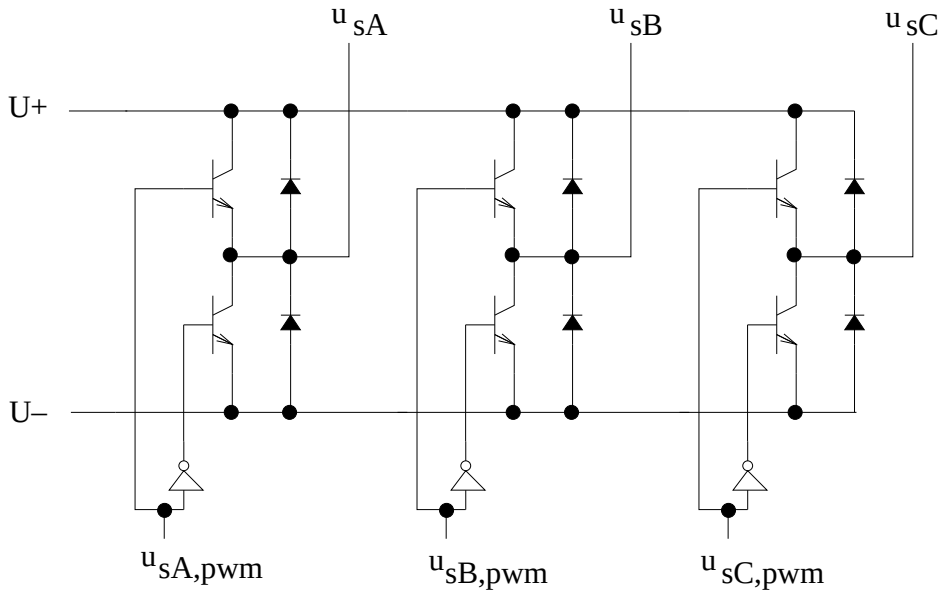


Figure 3.6: *Three-phase bridge voltage sourced inverter.*

3.6 Parameter identification

For control purposes it is necessary to know the motor parameters R_s, R_r, L_m, L_s, and L_r. These can be identified from stator current and voltage measurements. There is however an infinite number of motor parameters all yielding the same behaviour [Gorter, 1997]. It is therefore necessary to make some assumption on the parameters, for instance that $L_r = L_s$. The parameters can then be identified for instance by auto-commissioning at standstill as described in e.g. [Rasmussen, 1995] and [Rasmussen et al., 1995]. The voltage reference is chosen in such a way that the generated torque is not large enough to pull the shaft out of standstill. If voltage measurements are not available the reference voltages for the power device must be used.

3.7 Experimental setup

In this thesis several experiments will be performed on a laboratory induction motor system illustrated in Figure 3.7.

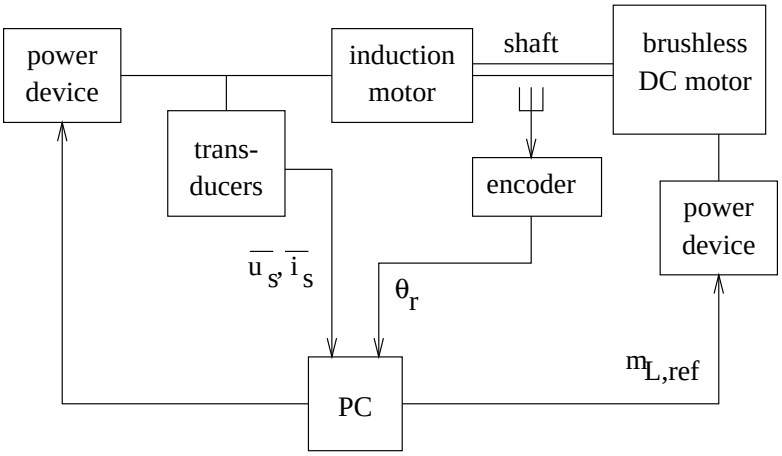


Figure 3.7: *Laboratory induction motor system.*

The induction motor has two pole pairs ($Z_p = 2$), a squirrel-cage rotor, and three star connected stator windings. Its nominal data are

Nominal power	1.5 kW
Nominal speed	1420 rpm
Nominal torque	10 Nm
Nominal current at 380 V	3.6 A

The nominal speed of $1420rpm = 148.7rad/s$ is equivalent to the electrical rotational speed $\omega_r = 297.4rad/s$. The electrical rotational speed in rad/s will be the representation used in the following chapters.

In [Rasmussen, 1995] the parameters of the induction motor were identified at standstill at $20°C$ under the assumption $L_s = L_r$ as

$$L_s = L_r = 0.352H, \quad L_m = 0.341H, \quad R_s = 5.0\Omega, \quad R_r = 3.3\Omega. \tag{3.57}$$

The PC runs the control program to be tested providing a reference voltage for the induction motor power device and a torque reference to the DC motor power device. The brush-less DC motor can be used to simulate a load torque on the shaft. Since the PC receives measurements of the rotor angular position from the encoder this can for instance be a position or speed dependent load.

In addition to the encoder data the PC receives measurements of the stator current and voltage.

The equipment is described in further detail in Appendix A.

3.8 Summary

This chapter described the induction motor system. The following complex state space model of the induction motor in the stator-fixed reference frame was derived:

$$\begin{aligned}
\dot{x}_{sc} &= A_{sc}x_{sc} + B_{sc}\bar{u}_s, \\
x_{sc} &= \begin{bmatrix} \bar{i}_s \\ \bar{i}_m \end{bmatrix}, \\
A_{sc} &= \begin{bmatrix} \frac{L_m^2 R_r + R_s L_r^2}{L_r(L_m^2 - L_s L_r)} & \frac{L_m^2(jL_r\omega_r - R_r)}{L_r(L_m^2 - L_s L_r)} \\ \frac{R_r}{L_r} & j\omega_r - \frac{R_r}{L_r} \end{bmatrix}, \\
B_{sc} &= \begin{bmatrix} \frac{L_r}{L_s L_r - L_m^2} \\ 0 \end{bmatrix}, \\
m_e &= \frac{3Z_p L_m^2}{2L_r}\Im\{\bar{i}_s \bar{i}_m^*\}, \\
\dot{\omega}_r &= Z_p\dot{\omega}_{mech} = \frac{Z_p}{J}(m_e - m_L).
\end{aligned}$$

$\bar{i}_s$ and $\bar{u}_s$ are the stator current and voltage, respectively, and $\bar{i}_m$ is the magnetising current. ω_r is the rotational speed of the shaft. m_e is the torque produced by the induction motor. m_L is the load torque on the shaft acting as a disturbance. L_m, L_s, L_r, R_r, R_s, Z_p, and J are constant or slowly varying real parameters.

The stator voltage, $\bar{u}_s$, is the control signal and is supplied to the motor by a power device, the voltage sourced inverter. $\bar{i}_s$ and $\bar{u}_s$, and in some configurations ω_r, can be measured.

This constitutes the model to be used in the following chapters. Experiments will be performed on a laboratory system described at the end of this chapter and in Appendix A.

Chapter 4

ROTOR FLUX ORIENTED CONTROL

In this chapter the rotor flux oriented controller (or rotor *field* oriented controller) scheme for the induction motor is described. The purpose of the controller is to track a reference speed, $\omega_{r,ref}$, and a reference magnetising current, $i_{mR,ref}$, while rejecting disturbances from the load torque.

The main purpose of this chapter is to present an existing induction motor control method. This controller is a cascade coupling of several sub-blocks. The aim of the following chapters is to develop replacements by new methods for some of these sub-blocks. It has therefore been chosen to focus on field oriented control and in particular the rotor flux orientation rather than give a full review of all the many control methods for induction motors.

Section 4.1 shows the simplification in the dynamical equations of the motor achieved by writing them in a reference system following the angle of the rotor flux. Section 4.2 then describes the rotor flux oriented control method. The method is observer-based and requires an estimate of the rotor flux. A short discussion of flux estimation is given in Section 4.3. If a speed or position measurement is not available it is furthermore

necessary to estimate the speed. A brief introduction to speed observers is given in Section 4.4.

The controllers and observers described in this chapter have all been described before. The only new contribution is the discovery of the potential instability of the speed observer described in [Kubota et al., 1993]. An example of this instability is given in Section 4.4.2.

4.1 Model in rotor flux coordinates

By expressing the model in a coordinate system fixed to the rotor flux angle

$$\rho \triangleq \angle \bar{\Psi}_{rs} = \angle \bar{i}_m, \tag{4.1}$$

where $\bar{i}_m$ is as given in Section 3.3.1, i.e. $\bar{i}_m = \bar{i}_s + \frac{L_r}{L_m}\bar{i}_{rs}$, a partial linearisation of the torque and magnetising current equations can be achieved. Equations (3.34) and (3.35) in rotor flux coordinates are:

$$(R_s + (\frac{d}{dt} + j\omega_{mR})L'_s)\bar{i}_{sr} + (\frac{d}{dt} + j\omega_{mR})L'_m i_{mR} = \bar{u}_{sr}, \tag{4.2}$$

$$R'_r(i_{mR} - \bar{i}_{sr}) + (\frac{d}{dt} + j(\omega_{mR} - \omega_r))L'_m i_{mR} = 0, \tag{4.3}$$

where the following definitions have been used

$$\begin{array}{llll} \bar{i}_{sr} & \triangleq & \bar{i}_s e^{-j\rho}, & \bar{u}_{sr} \triangleq \bar{u}_s e^{-j\rho}, \\ i_{mR} & \triangleq & \bar{i}_m e^{-j\rho} = |i_m|, & \omega_{mR} \triangleq \frac{d}{dt}\rho, \\ \sigma & \triangleq & 1 - L_m^2/(L_r L_s), & L'_s \triangleq \sigma L_s, \\ L'_m & \triangleq & (1-\sigma)L_s = L_m^2/L_r, & R'_r \triangleq (L_m/L_r)^2 R_r. \end{array}$$

The stator current in rotor flux coordinates is split into two real values, the direct and quadrature components:

$$\begin{array}{lll} i_{sd} & \triangleq & \Re\{\bar{i}_s e^{-j\rho}\} = \Re\{\bar{i}_{sr}\}, \\ i_{sq} & \triangleq & \Im\{\bar{i}_s e^{-j\rho}\} = \Im\{\bar{i}_{sr}\}. \end{array}$$

By taking the real part of (4.3), a dynamic equation for the magnetising current, i_{mR}, can be found as

$$\frac{L_r}{R_r}\frac{d}{dt}i_{mR} + i_{mR} = i_{sd}. \tag{4.4}$$

The quantity $T_r \triangleq \frac{L_r}{R_r}$ is called the *rotor time constant*.

The imaginary part gives the slip frequency

$$\omega_{slip} = \frac{1}{T_r}\frac{i_{sq}}{i_{mR}}. \tag{4.5}$$

The produced torque can be found from (3.28) as

$$m_e = \frac{3}{2} Z_p L'_m i_{mR} i_{sq}. \tag{4.6}$$

As seen a decoupling is achieved so that i_{sd} is used for controlling i_{mR} and i_{sq} is used for controlling the torque m_e. Even though i_{mR} does affect m_e, the changes are slow, making m_e an almost linear function of i_{sq}.

4.2 Rotor flux oriented control

Several speed or torque control schemes for induction motors exist, see for instance [Kaźmierkowski and Tunia, 1994] or [Vas, 1998]. Also worth mentioning is the passivity-based approach described in e.g. [Ortega et al., 1996]. In this thesis we will however focus on one particular type of induction motor control, namely direct rotor flux oriented control which will be described in this section. The basic principle is shown in Figure 4.1 and is based on the partial decoupling of the torque and the magnetising current achieved in the rotor flux oriented reference frame.

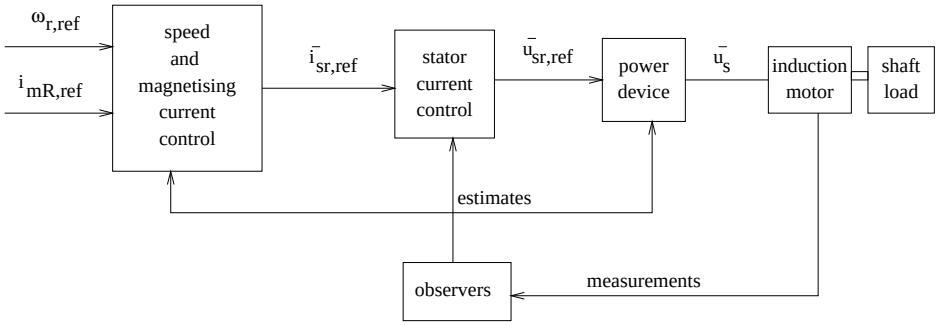


Figure 4.1: *Sketch of the rotor flux oriented speed control scheme.*

The controller objective is to track references for the magnetising current, i_{mR}, and the speed, ω_r (or torque or position). The observers provide estimates of the stator and magnetising currents, $\bar{i}_s$ and $\bar{i}_m$, and of the speed. These estimates are based on measurement of some of these three signals. In some cases the stator voltage, $\bar{u}_s$, is also measured. Otherwise the voltage command, $\bar{u}_{sr,ref}$ can be used. The speed and magnetising current controllers operate in the rotor flux oriented reference frame and provide a reference

signal, $\bar{i}_{sr,ref}$, for the stator current. The stator current controller tracks this reference by providing the power device with a stator voltage command, $\bar{u}_{s,ref}$, in the stator-fixed reference frame.

Examples of how to implement the different blocks will be given in the following. The examples in Sections 4.2.1-4.2.3 are from [Rasmussen, 1995].

4.2.1 Speed control

The speed can be controlled through the torque, m_e, given by equation (4.6). Since the torque usually is not measured, nothing will be gained by estimating and controlling it in a feedback loop [Rasmussen, 1995]. Instead, the reference value for i_{sq} can be found from (4.6) as

$$i_{sq,ref} = \frac{2}{3Z_p L'_m \hat{i}_{mR}} m_{e,ref}. \tag{4.7}$$

The speed can then be controlled for instance by a PI-controller tuned by a relay feedback experiment. Since the speed controller is in a cascade coupling with the stator current controller, the tuning must be performed with the intended stator current controller in operation. The tuning will also be affected by the bandwidth of the speed sensor or estimation.

The torque is limited by the maximum stator current allowed. The limit is found as

$$m_{e,max} = \frac{3Z_p L'_m \hat{i}_{mR}}{2} \sqrt{I^2_{max} - i^2_{sd,ref}}. \tag{4.8}$$

Anti-windup must therefore be implemented if the speed controller contains an integration.

The scheme is illustrated in Figure 4.2. The PI-controller acts on the control error $\omega_{r,ref} - \hat{\omega}_r$. The PI-controller output $m_{e,ref}$ is limited by $m_{e,max}$ computed from (4.8). The torque reference $m_{e,ref}$ is converted into a current reference $i_{sq,ref}$ using (4.7).

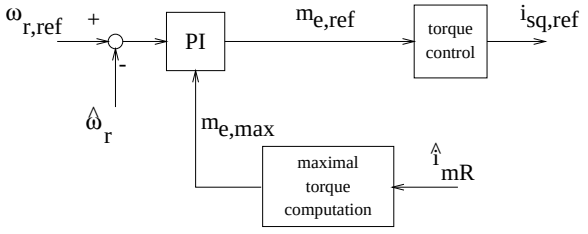


Figure 4.2: *Example of a simple speed control scheme.*

4.2.2 Magnetising current control

The magnetising current i_{mR} is governed by (4.4). The aim of the controller is to keep i_{mR} constant at some predetermined value. At speeds above rated speed it may be necessary to reduce this value in order to avoid saturation of the voltage supply. This situation will not be considered here.

Since the steady state transfer function is 1 independent of physical parameters a proportional controller with gain K_p will have zero steady state error if the reference value is premultiplied by $\frac{K_p+1}{K_p}$ assuming that the stator current controller has zero steady state error. A proportional controller tuned by a relay-feedback experiment can therefore be used.

4.2.3 Stator current control

As described in Section 4.2 a decoupling is achieved in the rotor flux oriented coordinate system so that i_{sd} controls the magnetising current and i_{sq} controls the torque. The stator currents i_{sd} and i_{sq} are controlled through the direct and quadrature components of the stator voltage, u_{sd} and u_{sq}, defined as

$$\begin{aligned} u_{sd} &\triangleq \Re(\bar{u}_{sr}), \\ u_{sq} &\triangleq \Im(\bar{u}_{sr}). \end{aligned}$$

Unfortunately the equations for the currents are cross-coupled so a decoupling is necessary.

Splitting (4.2) into the real and imaginary parts gives

$$u_{sd} = (R_s + L'_s\frac{d}{dt})i_{sd} - \omega_{mR}L'_s i_{sq} + L'_m\frac{d}{dt}i_{mR}, \tag{4.9}$$

$$u_{sq} = (R_s + L'_s\frac{d}{dt})i_{sq} + \omega_{mR}L'_s i_{sd} + L'_m\omega_{mR}i_{mR}. \tag{4.10}$$

Inserting (4.4) in (4.9) a decoupling scheme can be identified from

$$(R_s + L'_s\frac{d}{dt})i_{sd} = u_{sd} + \omega_{mR}L'_s i_{sq} - R'_r(i_{sd} - i_{mR}), \tag{4.11}$$

$$(R_s + L'_s\frac{d}{dt})i_{sq} = u_{sq} - \omega_{mR}L'_s i_{sd} - L'_m\omega_{mR}i_{mR}. \tag{4.12}$$

Decoupling can be achieved by adding the feed-forward voltages

$$u_{sdff} \triangleq -\omega_{mR}L'_s i_{sq} + R'_r(i_{sd} - i_{mR}), \tag{4.13}$$

$$u_{sqff} \triangleq \omega_{mR} L'_s i_{sd} + L'_m \omega_{mR} i_{mR} \tag{4.14}$$

to the outputs of the current controllers u_{sdc} and u_{sqc}. We then achieve

$$\begin{aligned} (R_s + L'_s \frac{d}{dt}) i_{sd} &= u_{sdc}, \\ (R_s + L'_s \frac{d}{dt}) i_{sq} &= u_{sqc}. \end{aligned}$$

The stator current can therefore be controlled by for instance PI-controllers tuned by relay-feedback experiments. This current control system is illustrated in Figure 4.3. In practice the values of i_{sd}, i_{sq}, i_{mR}, and ω_{mR} all have to be replaced by their respective estimates. Anti-windup again has to be implemented due to limits in the voltage supply.

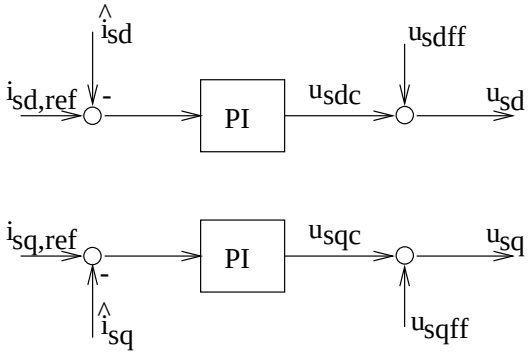


Figure 4.3: *Example of a simple stator current controller.*

4.3 Rotor flux estimation

In the stator current, magnetising current, and torque controllers estimates of the magnitude and angle of the magnetising current (or equivalently the rotor flux $\bar{\Psi}_{rs} = L_m \bar{i}_m$) are needed. This section will give an example of how to estimate $\bar{i}_m$ based on measurements of $\bar{i}_s$, $\bar{u}_s$, and possibly ω_r. A flux observer can be based on respectively the *voltage model* (3.34) and the *current model* (3.35) or a combination of these. The observer based on the current model (3.35) is:

$$\dot{\hat{\bar{i}}}_m = (-\frac{R_r}{L_r} + j\omega_r) \hat{\bar{i}}_m + \frac{R_r}{L_r} \bar{i}_s \tag{4.15}$$

Note that this model requires accurate estimates of R_r and ω_r. It is essentially an open loop simulation of the motor regarding the stator current as input.

Another observer is based on integrating the voltage model

$$\bar{u}_s = R_s \bar{i}_s + \frac{d\bar{\Psi}_s}{dt} \tag{4.16}$$

to obtain an estimate for $\bar{\Psi}_s$ and then finding $\bar{i}_m$ from

$$\bar{i}_m = \frac{L_r}{L_m^2}\bar{\Psi}_s + (1 - \frac{L_r L_s}{L_m^2})\bar{i}_s. \tag{4.17}$$

The voltage model method does not need values of R_r and ω_r, but it can be very noise sensitive especially for low frequencies due to the pure integration.

4.3.1 Closed-loop observer of Jansen and Lorenz

The rotor flux observer described in [Jansen and Lorenz, 1992], here called the *JL-observer*, is an example of a combination of the current and the voltage model. This observer combines the good qualities of the current model in low speed operation with the good qualities of the voltage model in high speed operation. The JL-observer is shown in Figure 4.4.

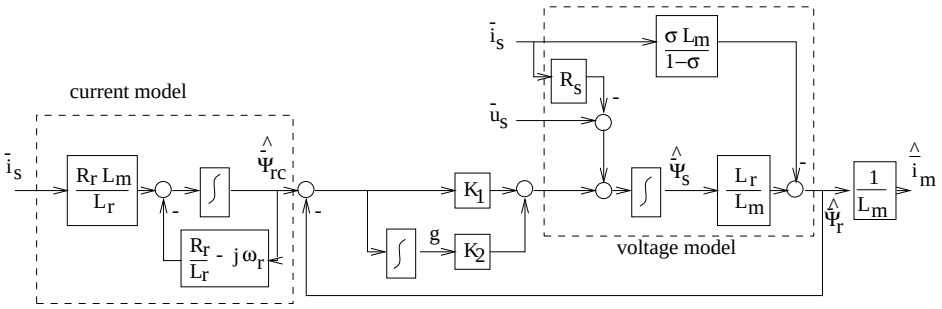


Figure 4.4: *Flux observer of Jansen and Lorenz (JL-observer).*

The K_1- and K_2-blocks constitute a frequency dependent weighting between the two observer types. For frequencies, s, where $K_1 + K_2 s^{-1}$ is large (low frequencies), the main emphasis will be on the current model. For $K_1 + s^{-1}K_2 = 0$ the JL-observer is exactly the voltage model observer. The idea is that at high frequencies we can exploit the robustness of the voltage model to uncertainties in R_r and ω_r and at low frequencies we can use the current model which has lower gains and therefore is less noise sensitive.

The JL-observer can be written as the state space system:

$$\begin{aligned}
\dot{x}_{jl} &= A_{jl}x_{jl} + B_{jl}y_m, \\
\hat{\bar{i}}_m &= C_{jl}x_{jl} + D_{jl}y_m, \\
x_{jl} &= \begin{bmatrix} \hat{\bar{\Psi}}_s \\ g \\ \hat{\bar{\Psi}}_{rc} \end{bmatrix}, \\
y_m &= \begin{bmatrix} \bar{i}_s \\ \bar{u}_s \end{bmatrix},
\end{aligned}$$

$$\begin{aligned}
A_{jl} &= \begin{bmatrix} -\frac{K_1 L_r}{L_m} & K_2 & K_1 \\ -\frac{L_r}{L_m} & 0 & 1 \\ 0 & 0 & -\frac{R_r}{L_r} + j\omega_r \end{bmatrix}, \\
B_{jl} &= \begin{bmatrix} \frac{K_1 L_r L_s}{L_m} - K_1 L_m - R_s & 1 \\ \frac{L_r L_s}{L_m} - L_m & 0 \\ \frac{L_m R_r}{L_r} & 0 \end{bmatrix}, \\
C_{jl} &= \begin{bmatrix} \frac{L_r}{L_m^2} & 0 & 0 \end{bmatrix}, \quad D_{jl} = \begin{bmatrix} 1 - \frac{L_r L_s}{L_m^2} & 0 \end{bmatrix}.
\end{aligned}$$

4.4 Speed estimation

For systems without speed or position sensors the speed ω_r must be estimated. This section presents two ways of doing this. Section 4.4.1 presents a simple method based on equation (3.35). The method presented in Section 4.4.2 is based on adaptive control theory. Section 4.4.3 presents simulation results.

4.4.1 Speed estimation from rotor equation

One way to estimate the speed is to isolate it from the rotor equation (3.35). In the stator-fixed reference frame this gives

$$j\bar{i}_m\omega_r = \frac{R_r}{L_r}(\bar{i}_m - \bar{i}_s) + \frac{d}{dt}\bar{i}_m. \tag{4.18}$$

In the rotor flux oriented reference frame the equivalent equation is

$$\omega_r = \omega_{mR} - \frac{R_r}{L_r}\frac{i_{sq}}{i_{mR}}. \tag{4.19}$$

Note that the last term is an expression for the slip frequency and that the accuracy of the estimate of R_r is essential, especially at high slip frequencies. Both versions require estimates of the flux. The bandwidth and noise sensitivity of the speed observer therefore depends on the choice of flux observer. A flux estimate based on the rotor equation (JL-observer with $K_1 \to \infty, K_2 \to \infty$) is useless since it will always return the same speed estimate as the flux estimate is based on due to the open-loop nature of this flux observer. The main problems with a flux observer based on the voltage model (JL-observer with $K_1 = K_2 = 0$) is that it depends on the stator resistance R_s, and that it contains a pure integration. To obtain a reasonable speed estimate a compromise between the two must be found.

4.4.2 Speed estimation method by Kubota et al.

An alternative approach based on adaptive control theory is suggested by Kubota et al. in [Kubota et al., 1993]. The scheme is illustrated in Figure 4.5.

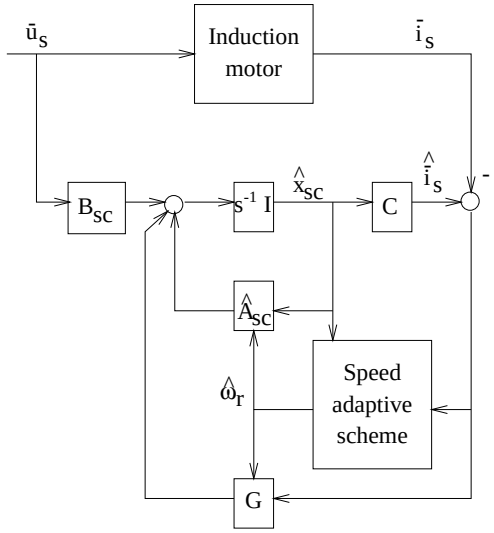


Figure 4.5: *Adaptive flux and speed estimation scheme of Kubota et al.*

$\hat{A}_{sc}$ is the system matrix of the model based on the current estimate of the speed. B_{sc} is the input matrix of the model. $C = [1 \quad 0]$. G is the observer gain matrix and is chosen so that the observer poles are proportional to the motor model poles. The state estimate is updated by the equation

$$\frac{d}{dt}\hat{x} = \hat{A}_{sc}\hat{x}_{sc} + B_{sc}\bar{u}_s + G(\hat{\bar{i}}_s - \bar{i}_s), \tag{4.20}$$

where the first two terms provide an open-loop simulation and the last term provides a correction based on the stator current estimation error.

In [Kubota et al., 1993] it is shown through Lyapunov theory that the speed adaption scheme

$$\frac{d}{dt}\hat{\omega}_r = \lambda_c \Im\{e_{is}^* \hat{\bar{i}}_m\}, \tag{4.21}$$

where $e_{is} = \bar{i}_s - \hat{\bar{i}}_s$ and the *speed estimate update gain* λ_c is any positive constant, will make the estimation error converge to zero. Unfortunately the proof is incorrect. The proof is based on the Lyapunov function

$$V = e^* e + (\Delta_{\omega_r})^2/\lambda \tag{4.22}$$

where $e = x_{sc} - \hat{x}_{sc}$, $\Delta_{\omega_r} = \hat{\omega}_r - \omega_r$, and λ is a positive constant. The time derivative of V is

$$\begin{gathered}
\tfrac{d}{dt}V = \dot{e}^* e + e^* \dot{e} + 2\Delta_{\omega_r}\dot{\Delta}_{\omega_r}/\lambda = \\
2e^* herm(A_{sc} + GC)e - 2herm(e^*(\hat{A}_{sc} - A_{sc})\hat{x}_{sc}) + 2\Delta_{\omega_r}\dot{\hat{\omega}}_r/\lambda = \\
2e^* herm(A_{sc} + GC)e + \\
2\Delta_{\omega_r}\Im\{\tfrac{-L_m^2}{L_s L_r - L_m^2} e_{is}^* \hat{\bar{i}}_m + \bar{i}_m^* \hat{\bar{i}}_m - \hat{\bar{i}}_m^* \hat{\bar{i}}_m\} + 2\Delta_{\omega_r}\dot{\hat{\omega}}_r/\lambda = \\
2e^* herm(A_{sc} + GC)e + 2\Delta_{\omega_r}\tfrac{-L_m^2}{L_s L_r - L_m^2}\Im\{e_{is}^* \hat{\bar{i}}_m\} + \\
2\Delta_{\omega_r}\Im\{\bar{i}_m^* \hat{\bar{i}}_m\} + 2\Delta_{\omega_r}\dot{\hat{\omega}}_r/\lambda
\end{gathered} \tag{4.23}$$

where it has been assumed that $\dot{\omega}_r = 0$. However, in [Kubota et al., 1993] the third term, $2\Delta_{\omega_r}\Im\{\bar{i}_m^* \hat{\bar{i}}_m\}$, has been forgotten. It is then stated that $\frac{d}{dt}V$ can be made negative by choosing G so that $herm(A_{sc} + GC) < 0$, and by setting

$$\frac{d}{dt}\hat{\omega}_r = \lambda \frac{L_m^2}{L_s L_r - L_m^2}\Im\{e_{is}^* \hat{\bar{i}}_m\}. \tag{4.24}$$

Since $\Im\{u^* y\} = |u||y| \sin(\angle y - \angle u)$, where $\angle(\cdot)$ is the angle of a complex number, the forgotten term will be positive when $\sin(\angle\hat{\bar{i}}_m - \angle\bar{i}_m)$ has the same sign as Δ_{ω_r}, which is always true in steady state. This can be seen from (3.55) in the following way: Write (3.55) as

$$\bar{i}_{ms}(\omega_r) = \frac{c_1}{c_2 + c_3\omega_r}, \tag{4.25}$$

where c_1, c_2, c_3 are complex constants (depending on ω_{mR} and $\bar{u}_{ss}$, but these are constant in steady state). Then, observing that the denominator is non-zero due to $L_s L_r > L_m^2$, we have

$$\begin{aligned}
\angle\hat{\bar{i}}_m - \angle\bar{i}_m = \angle\frac{c_1}{c_2 + c_3\hat{\omega}_r} - \angle\frac{c_1}{c_2 + c_3\omega_r} = \\
\angle(c_2 + c_3\omega_r)(c_2 + c_3\hat{\omega}_r)^*.
\end{aligned} \tag{4.26}$$

Tedious computations will show that

$$\Im\{(c_2 + c_3\omega_r)(c_2 + c_3\hat{\omega}_r)^*\} = \Delta_{\omega_r} R_r (R_s^2 L_r + \omega_{mR}^2 L_s (L_s L_r - L_m^2)) \qquad (4.27)$$

which has the same sign as Δ_{ω_r}. In other words the forgotten term will be positive in steady state and probably in most other cases as well, so there is no guarantee that the Lyapunov function will converge to zero.

The reason that the method usually works anyway, is that the speed estimate in most cases will converge to the true value, and subsequently the state estimates will converge too.

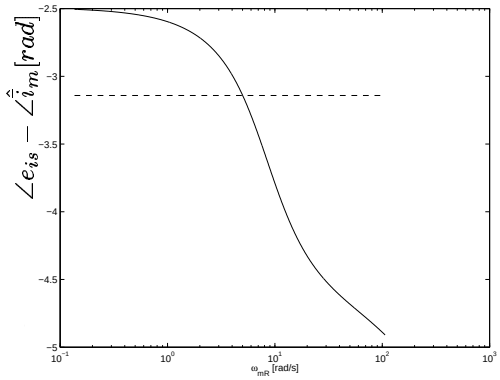


Figure 4.6: *The angle $\angle e_{is} - \hat{\angle i}_m$ in steady state for the example motor.*

When $\sin(\angle e_{is} - \hat{\angle i}_m)$ has the same sign as Δ_{ω_r}, the speed estimation error will increase. Figure 4.6 shows the angle

$$\begin{aligned}\angle e_{is} - \hat{\angle i}_m = \\ \angle \begin{bmatrix}1 & 0\end{bmatrix} \left((-A_{sc} + j\omega_{mR})^{-1} - (-\hat{A}_{sc} + j\omega_{mR})^{-1}\right) B_{sc} - \\ \angle \begin{bmatrix}0 & 1\end{bmatrix} (-\hat{A}_{sc} + j\omega_{mR})^{-1} B_{sc}\end{aligned} \qquad (4.28)$$

in steady state for $\omega_r = 10 rad/s$ and $\hat{\omega}_r = 15 rad/s$, as a function of the angular velocity of the flux, ω_{mR} for an example motor with the parameters in (3.57). As seen the speed estimation error will increase when $\omega_{mR} < 5 rad/s$ (regenerative mode). This is also demonstrated by the simulation shown in Figure 4.7. The system is in steady state at $\omega_r = 10 rad/s$ and $\omega_{mR} = 4 rad/s$. The estimator constants are chosen as $G = 0$ (guarantees $\mathrm{herm}\,(A_{sc} + GC) < 0$) and $\lambda_c = 1000$.

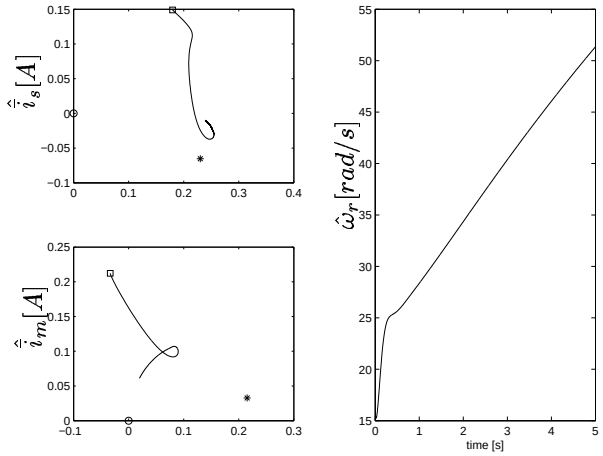


Figure 4.7: *Simulation of the speed estimator with the system in steady state at* $\omega_r = 10rad/s$ *and* $\omega_{mR} = 4rad/s$*. The two figures to the left show the estimates of* $\bar{i}_s$ *and* $\bar{i}_m$*, respectively, in a coordinate system rotating at the same angular velocity as the flux. The actual currents are constant in this coordinate system and are marked by* $*$*. The initial estimates are marked by* $\square$*. The bottom figure shows the speed estimate as a function of time. The speed estimate is seen to diverge from the actual speed.*

As seen the speed estimate diverges from the true speed. Initially the current estimates converge towards the true values, but the increasing speed estimation error causes them to diverge eventually as well.

4.4.3 Speed estimation simulation results

Figure 4.8 shows a test of the two speed estimation schemes. The actual speed changes from $10rad/s$ to $12rad/s$ and back again after 0.5 seconds. This is done for both a low ($2Nm$) and a high ($9Nm$) load situation. The figure shows this test for three different versions of the rotor equation scheme (Figures A, B, and C). The JL-observer has been used to provide the flux estimates and the difference between the three figures is the choice of frequency weighting constant K_1 and K_2. All three have been low-pass filtered to reduce measurement noise. In Figure A the weighting is mainly on the current model showing the expected reluctance to change away from the current speed estimate. In Figure B the weighting has been chosen to give a reasonable result. In Figure C the weighting is mainly on the voltage model. The fluctuations in the high slip case have the frequency ω_{mR} and are due to the almost pure integration which causes an almost constant offset error. This causes the fluctuations when the nonlinear transformation from $\bar{i}_m$ to i_{mR} is performed.

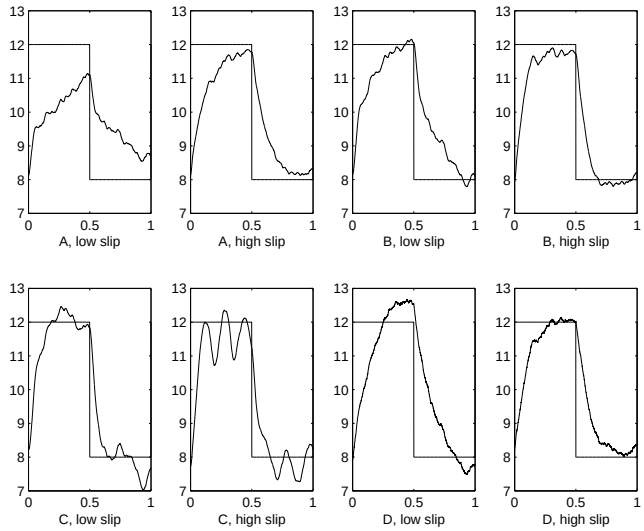


Figure 4.8: *Speed estimation at low and high load. In Figure A the weighting is mainly on the current model. In Figure B the weighting has been chosen to give a reasonable result. In Figure C the weighting is mainly on the voltage model. Figure D shows the same test for the method by Kubota et al.*

Figure D shows the same test for the method by Kubota et al. with the observer poles at 1.2 times the system poles. The performance is similar to the one obtained in Figure B. It is found that performances similar to Figures A and C can also be obtained by changing the ratio between observer and model poles.

Instantaneous changes in the speed are not realistic. Figure 4.9 shows a more realistic simulation where the speed is controlled by a rotor flux oriented controller with speed measurements available along where the speed reference is changed in steps. Also plotted is the speed estimate from Kubota's method. Figure 4.10 shows the speed estimation error.

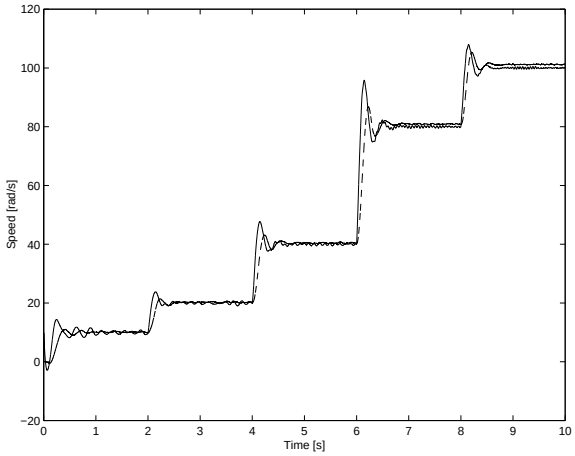


Figure 4.9: *Speed changes performed by rotor flux oriented controller with speed measurements, and the speed estimate of Kubota's method (dashed)*

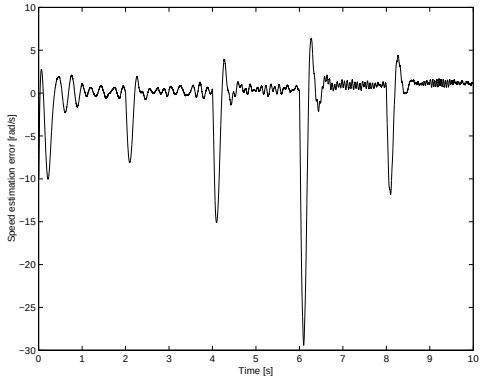


Figure 4.10: *Speed estimation error*

Notice that the speed estimation error resulting from the steps is practically the same for low and high speed. This indicates that the main problems for the flux observer caused by speed uncertainty will be greatest at low speeds since a change of for example $10rad/s$ causes a bigger difference in the model for low speeds than for high.

Figure 4.11 shows a similar simulation, where the speed reference sweeps slowly from $-15rad/s$ to $15rad/s$ and back again. Every 2 seconds the load changes in a step from $3Nm$ to $7Nm$ or back. The figure also shows the speed estimation. Figure 4.12 shows the speed estimation error.

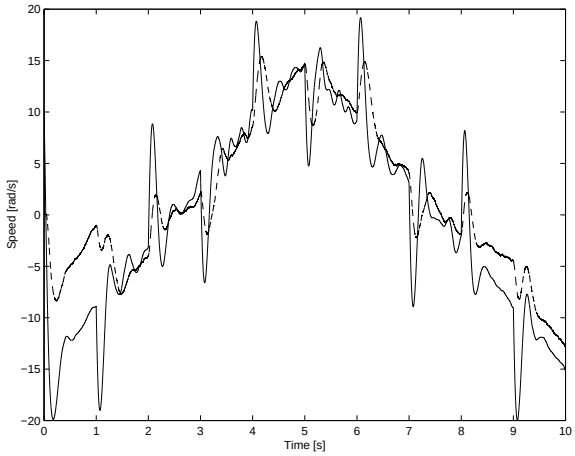


Figure 4.11: *Speed sweep with step changes in load and speed estimate of Kubota's method (dashed).*

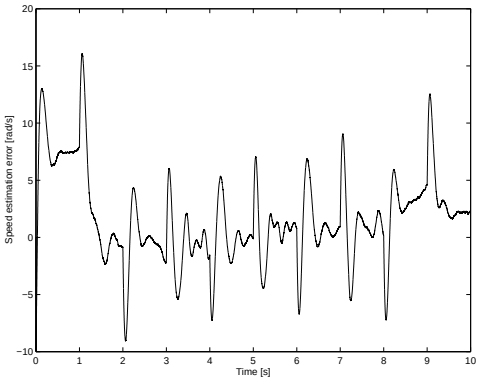


Figure 4.12: *Speed estimation error*

4.5 Summary

In this chapter the rotor flux oriented control method has been described. The controller consists of a cascade coupling with a stator current controller in the inner loop. The outer loop controls the shaft speed and the magnitude of the rotor flux. Examples of how to construct these controllers were given.

The control method is observer-based and requires an estimate of the rotor flux. An example of a flux observer, the JL-observer presented in [Jansen and Lorenz, 1992], was given.

If a speed or position measurement is not available it is furthermore necessary to estimate the speed. An example of a speed observer, the one presented in [Kubota et al., 1993], was given. It was shown that under certain conditions the observer will diverge. However, in normal operation the estimate will usually converge to the correct value assuming that the correct motor parameters are used.

Chapter 5

LINEAR PARAMETER VARYING FLUX OBSERVER

In recent years efficient ways to design controllers for a particular type of non-linear systems, *linear parameter varying* (LPV) systems, have been developed.

This chapter will review the LPV synthesis method in [Scherer, 2001] and apply it to the design of a flux observer for the induction motor. In Section 5.1 the historical background of LPV control is reviewed. In Section 5.2 robust quadratic performance analysis of LPV systems is discussed. In Section 5.3 the so-called full block S-procedure controller synthesis is discussed. Section 5.4 discusses how to obtain a discrete-time version of the results in Sections 5.2 and 5.3. This turns out to be surprisingly simple.

Considering the speed ω_r as a time-varying parameter allows writing the induction motor model obtained in Section 3.3.1 as either a real fourth order LPV model or as a complex second order LPV model. This is due to a special symmetry in the transfer function.

Section 5.5 provides theoretical justification for the fact that controllers and observers for this type of system can be assumed to have the same type of symmetry without loss of performance.

In Section 5.6 a discrete-time flux observer for a wide range of speeds is designed using the theory described in the previous sections. The observer is tested on the laboratory setup. Even though the performance is not significantly better than the JL-observer described in Section 4.3.1 it is worth noting that practically no tuning had to be done.

5.1 LPV background

In the famous DGKF paper [Doyle et al., 1989] state space solutions for the standard suboptimal $\mathcal{H}_\infty$ problem were given, that is, for an LTI state space system and a given $\gamma > 0$, find all controllers such that the $\mathcal{H}_\infty$-norm of the closed-loop system is less than γ. The solution was found by solving two coupled Riccati equations. In two independent papers [Gahinet and Apkarian, 1994] (continuous and discrete time) and [Iwasaki and Skelton, 1994] (continuous time only) the problem was reformulated into three coupled Riccati *inequalities* yielding an LMI problem. There were many advantages to this approach. The DGKF solution required several assumptions on the system that were not inherent to the problem, but rather to the solution method. Most of these assumptions could be removed with the inequality formulation. Furthermore it became more obvious how to extend the result to *linear parameter varying* (LPV) systems.

Definition 5.1 (Continuous time linear parameter varying system, LPV system)

A continuous time linear parameter varying system is a system which can be written on the form

$$\begin{aligned} \dot{x} &= A(\theta(t))x + B(\theta(t))u \\ y &= C(\theta(t))x + D(\theta(t))u, \end{aligned} \tag{5.1}$$

where θ is a bounded time-varying parameter vector which can be measured online.

Definition 5.2 (Discrete time linear parameter varying system, LPV system)

A discrete time linear parameter varying system is a system which can be written on the form

$$\begin{aligned} x_{k+1} &= A(\theta_k)x_k + B(\theta_k)u_k \\ y_k &= C(\theta_k)x_k + D(\theta_k)u_k, \end{aligned}$$

where θ is a bounded time-varying parameter vector which can be measured online.

There are two important points to make about the parameter vector. Firstly, it is not necessarily known a priori but can be measured in real-time, so traditional time-varying

control methods based on future as well as past values cannot be used. Secondly, the variations can be fast, so traditional gain scheduling between a finite grid of linearised operation points will not necessarily work.

These facts were pointed out in [Shamma and Athans, 1992], where it was also suggested that the problem might be solved by designing a controller with the same type of structure, i.e. an *LPV controller*. An early suggestion of a controller design of this type is given in [Becker et al., 1993]. The synthesis method is convex, but unless the parameter dependence is affine, it requires an infinite number of constraints.

If the parameter vector enters the state space matrices in a rational manner an LPV system can be written as an LFT as illustrated in Figure 5.1, where the *uncertainty block* (or *residual gains*) Δ is a matrix function of $\theta(t)$ and M is an LTI system. (Despite the term "uncertainty block", Δ is assumed to be fully known in real time, but it is uncertain in the sense that it is not known a priori.)

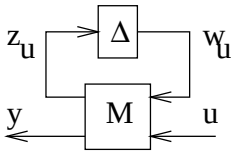


Figure 5.1: *LPV system on LFT form.*

In two independent papers, [Apkarian and Gahinet, 1995] (continuous and discrete time) and [Packard, 1994] (discrete time), controller synthesis for LPV systems on the LFT form were given in terms of LMIs. The idea was to provide the controller with a copy of the uncertainty block as illustrated in Figure 5.2, $\Delta_c(t) = \Delta(t)$.

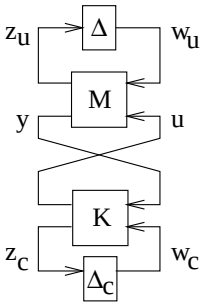


Figure 5.2: *LPV system with LPV controller.*

If z_u can be reconstructed from the measurements y, then by setting $z_c = z_u$ we will have $w_c = w_u$. w_c can then be used to compensate for the effects of w_u on the system. But even if z_u is not fully known, some of the effects can still be compensated by an

observer-based estimate.

The methods in [Packard, 1994, Apkarian and Gahinet, 1995] are based on the scaled Small Gain Theorem. The scaling is done by *multipliers*, J, which have to commute with Δ, i.e. $J\Delta = \Delta J$ for all possible Δ. This condition greatly limits the available multipliers thereby introducing conservatism. In addition these multipliers allow for Δ to be complex even if it is actually known to be real introducing further conservatism. Furthermore the Small Gain Theorem is more conservative than *robust quadratic performance* [Zhou et al., 1992] (Robust quadratic performance is discussed in Section 5.2).

Under the assumption that the system is *polytopic* these problems are all dealt with in [Apkarian et al., 1995]. The system being polytopic means that the matrix

$$\begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix} \tag{5.2}$$

belongs to the convex hull of a finite number of matrices. The controller resulting from the synthesis is also on polytopic form, i.e. the controller matrices are found as a convex combination of *vertex matrices*.

Staying in the LFT setting [Helmersson, 1995] takes realness of Δ into account using a structured singular value approach, but due to the frequency domain nature of the result the improvement is only useful for time-invariant parameters.

In [Scorletti and Ghaoui, 1995] the analysis results in [Rantzer and Megretski, 1994] (for journal versions see [Scorletti and Ghaoui, 1998] and [Megretski and Rantzer, 1997]) are extended to synthesis. By using skew-symmetric multipliers the realness of the time-varying parameters are taken into account. However, the multipliers are still required to be block diagonal, maintaining a lot of conservatism.

In [Scherer, 2001] the controller synthesis problem for systems with rational parameter dependence is solved with full block multipliers. In other words, the conservatism due to the block diagonal multipliers is removed. The result has the least possible conservatism if we allow for arbitrarily fast parameter variations. The only downside to the method is that the *scheduling function*, Δ_c, now has to be a nonlinear function of Δ.

This chapter will review the synthesis method in [Scherer, 2001] and apply it to the design of a flux observer for the induction motor. In Section 5.2 robust quadratic performance analysis of LPV systems is discussed. Assuming that the state space matrices depend on θ in a rational manner a full block S-procedure is used to transform between the LPV on the form in (5.1) and an equivalent system on LFT form. In Section 5.3 full block controller synthesis is discussed. These two sections contain no new contributions except for Lemma 5.12, which provides a partial solution to some of the numerical problems associated with Lemma 2.9. Furthermore Scherer's results are extended to complex systems. Careful inspection reveals that this is mainly a question of substituting transposed $(\cdot^T)$ with complex conjugated transposed $(\cdot^*)$.

5.2 LPV analysis

This section reviews the robust quadratic performance (RQP) analysis method that is the basis for the controller synthesis described in [Scherer, 2001]. First the concept of RQP is introduced. Then the equivalence between LPV systems on the form (5.1) and a generalised LFT representation is discussed. The latter provides a separation into an LTI system and time-varying parameters which is essential to the following section on controller synthesis.

5.2.1 Robust quadratic performance

Consider the LPV system

$$\begin{aligned} \dot{\xi} &= \mathcal{A}(\Delta)\xi + \mathcal{B}(\Delta)w_p \\ z_p &= \mathcal{C}(\Delta)\xi + \mathcal{D}(\Delta)w_p, \end{aligned} \tag{5.3}$$

where the state space matrices depend on Δ in a linear fractional manner, which we will discuss later. Δ is a time-varying matrix belonging to a compact and path-connected set $\bar{\Delta}$.

Definition 5.3 (Robust quadratic performance,RQP)

We say that the system (5.3) achieves robust quadratic performance *with* performance index

$$P_p = P_p^* \triangleq \begin{bmatrix} Q_p & S_p \\ S_p^* & R_p \end{bmatrix}, \quad R_p \geq 0 \tag{5.4}$$

if

- *Positive constants K and α exist such that*

  $\|\xi(t_1)\| \leq \|\xi(t_0)\| K e^{-\alpha(t_1-t_0)}$ *for* $t_1 \geq t_0$ *and all* $\Delta \in \bar{\Delta}$ *if* $w_p(t) = 0$.

- *The quadratic performance specification*

  $$\exists \varepsilon > 0 : \int_{t_0}^{t_1} \begin{bmatrix} w_p(t)^* & z_p(t)^* \end{bmatrix} P_p \begin{bmatrix} w_p(t) \\ z_p(t) \end{bmatrix} dt \leq -\varepsilon \int_{t_0}^{t_1} w_p(t)^* w_p(t) dt,$$

  holds for all $t_1 \geq t_0$ *and all* $\Delta(t) \in \bar{\Delta}$ *if* $\xi(t_0) = 0$.

The first condition guarantees exponential stability. The second condition provides a performance specification depending on the choice of P_p. For instance the choice

$$P_p = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix}$$

provides the $\mathcal{L}_2$-induced norm bound

$$\sup_{w \neq 0} \frac{||z||_2}{||w||_2} < \gamma.$$

Other possibilities are passivity and positive real constraints. RQP has been shown to be less conservative than small gain conditions [Zhou et al., 1992].

RQP analysis can be formulated as a (possibly infinite-dimensional) LMI:

Theorem 5.4 [Scherer, 1999](Robust quadratic performance analysis)

Suppose there exist an $\mathcal{X} > 0$ *satisfying the LMI*

$$\left[\begin{array}{cc|c} I & 0 & \\ \mathcal{A}(\Delta) & \mathcal{B}(\Delta) & \\ \hline 0 & I & \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) & \end{array}\right]^* \left[\begin{array}{cc|cc} 0 & \mathcal{X} & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 \\ \hline 0 & 0 & Q_p & S_p \\ 0 & 0 & S_p^* & R_p \end{array}\right] \left[\begin{array}{cc|c} I & 0 & \\ \mathcal{A}(\Delta) & \mathcal{B}(\Delta) & \\ \hline 0 & I & \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) & \end{array}\right] < 0 \tag{5.5}$$

for all $\Delta \in \bar{\Delta}$ *and with* P_p *satisfying (5.4). Then the system (5.3) achieves robust quadratic performance with performance index* P_p*.*

Proof: A proof by standard dissipativity arguments is given in [Scherer, 1999]. A proof for the discrete-time version is given in Section 5.4.2. ◁

5.2.2 Linear fractional dependency

Now consider the following LPV system on a generalised LFT form:

$$\left[\begin{array}{c} \dot{\xi} \\ \hline z_u \\ z_c \\ z_p \end{array}\right] = \left[\begin{array}{c|ccc} \mathcal{A} & \mathcal{B}_u & \mathcal{B}_c & \mathcal{B}_p \\ \hline \mathcal{C}_u & \mathcal{D}_{uu} & \mathcal{D}_{uc} & \mathcal{D}_{up} \\ \mathcal{C}_c & \mathcal{D}_{cu} & \mathcal{D}_{cc} & \mathcal{D}_{cp} \\ \mathcal{C}_p & \mathcal{D}_{pu} & \mathcal{D}_{pc} & \mathcal{D}_{pp} \end{array}\right] \left[\begin{array}{c} \xi \\ \hline w_u \\ w_c \\ w_p \end{array}\right] \tag{5.6}$$

with a parameter dependency given by

$$\left[\begin{array}{c} w_u \\ w_c \\ \hline z_u \\ z_c \end{array}\right] \in \mathcal{S}_a(\Delta) = \operatorname{im} S_a(\Delta) = \operatorname{im} \left[\begin{array}{c} S_{a1}(\Delta) \\ \hline S_{a2}(\Delta) \end{array}\right] = \operatorname{im} \left[\begin{array}{c} S_{1u}(\Delta) \\ S_{1c}(\Delta) \\ \hline S_{2u}(\Delta) \\ S_{2c}(\Delta) \end{array}\right], \tag{5.7}$$

where $\xi \in \mathbb{C}^{n_s+n_c}, z_u \in \mathbb{C}^{n_{zu}}, z_c \in \mathbb{C}^{k_c}, z_p \in \mathbb{C}^{n_{zp}}, w_u \in \mathbb{C}^{n_{wu}}, w_c \in \mathbb{C}^{m_c}$, and $w_p \in \mathbb{C}^{n_{wp}}$. This type of parameter dependency is more general than what is usually

seen in LPV literature. If S_{a2} is nonsingular we have a standard feedback loop

$$\begin{bmatrix} z_u \\ z_c \end{bmatrix} = S_{a1}(\Delta) S_{a2}(\Delta)^{-1} \begin{bmatrix} w_u \\ w_c \end{bmatrix}.$$

If we have $S_{a1}(\Delta) = \Delta$ and $S_{a2}(\Delta) = I$ we recover the standard LFT representation. This more general representation allows for instance for affine parameter dependence. It turns out in the synthesis problem, that even if the plant has a standard LFT dependency, it may still be necessary to let the controller have the more general form in order to avoid conservatism [Scherer, 2001].

Notice that the interconnection of the system in (5.6) and the parameter dependency in (5.7) is only well-posed if [Scherer, 2001]

$$\mathcal{S}_a(\Delta) \oplus \text{im} \begin{bmatrix} I & 0 \\ 0 & I \\ \mathcal{D}_{uu} & \mathcal{D}_{uc} \\ \mathcal{D}_{cu} & \mathcal{D}_{pp} \end{bmatrix} = \mathbb{C}^{n_{wu}+m_c+n_{zu}+k_c}, \forall \Delta \in \bar{\Delta}.$$

In the standard LFT case this simply amounts to $I - \begin{bmatrix} \mathcal{D}_{uu} & \mathcal{D}_{uc} \\ \mathcal{D}_{cu} & \mathcal{D}_{pp} \end{bmatrix} \Delta$ being non-singular for all Δ.

If we would like to examine RQP for the system (5.6)-(5.7), we need to put it on the form in (5.3). The following lemma from [Scherer, 2001] provides a way to do this.

Lemma 5.5 Assume that $S_a(\Delta)$ is a continuous function. If the interconnection of (5.6) and (5.7) is well-posed, then

$$S_{a2} - \begin{bmatrix} \mathcal{D}_{uu} & \mathcal{D}_{uc} \\ \mathcal{D}_{cu} & \mathcal{D}_{pp} \end{bmatrix} S_{a1}$$

has full row rank for all $\Delta \in \bar{\Delta}$. Therefore

$$S_{a1} \left[S_{a2} - \begin{bmatrix} \mathcal{D}_{uu} & \mathcal{D}_{uc} \\ \mathcal{D}_{cu} & \mathcal{D}_{pp} \end{bmatrix} S_{a1} \right]^{\dagger}$$

is continuous in Δ. Furthermore the system (5.3) with

$$\begin{bmatrix} \mathcal{A}(\Delta) & \mathcal{B}(\Delta) \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) \end{bmatrix} = \begin{bmatrix} \mathcal{A} & \mathcal{B}_p \\ \mathcal{C}_p & \mathcal{D}_{pp} \end{bmatrix} + \begin{bmatrix} \mathcal{B}_u & \mathcal{B}_c \\ \mathcal{D}_{pu} & \mathcal{D}_{pc} \end{bmatrix} S_{a1} \left[S_{a2} - \begin{bmatrix} \mathcal{D}_{uu} & \mathcal{D}_{uc} \\ \mathcal{D}_{cu} & \mathcal{D}_{pp} \end{bmatrix} S_{a1} \right]^{\dagger} \begin{bmatrix} \mathcal{C}_u & \mathcal{D}_{up} \\ \mathcal{C}_c & \mathcal{D}_{cp} \end{bmatrix} \quad (5.8)$$

has the same trajectories for $\xi(t), w_p(t)$, and $z_p(t)$ as (5.6)-(5.7).

In other words the system (5.6)-(5.7) is equivalent to (5.3) with (5.8).

The following theorem provides a way to check for RQP for the LFT system.

In order to align the notation with the one in the following section define the following subspace by swapping coordinates of $\mathcal{S}_a$:

$$\mathcal{S}_e(\Delta) \triangleq \operatorname{im} \begin{bmatrix} S_{1u}(\Delta) \\ S_{2u}(\Delta) \\ S_{1c}(\Delta) \\ S_{2c}(\Delta) \end{bmatrix}.$$

Theorem 5.6 (Full block S-procedure for LPV system) [Scherer, 2001]

Robust quadratic performance for the system (5.6)-(5.7) is achieved if the following two equivalent conditions are fulfilled:

1. *The interconnection (5.6)-(5.7) is well-posed and there exists an $\mathcal{X} > 0$ such that (5.8) satisfies (5.5) for all $\Delta \in \bar{\Delta}$.*
2. *$\dim(S_a(\Delta)) \geq n_{zu} + k_c$ and there exist $\mathcal{X} > 0$ and a Hermitian multiplier*

$$P_e \triangleq \left[\begin{array}{cc|cc} Q & S & Q_{12} & S_{12} \\ S^* & R & S_{21}^* & R_{12} \\ \hline Q_{12}^* & S_{21} & Q_2 & S_2 \\ S_{12}^* & R_{12}^* & S_2^* & R_2 \end{array}\right] > 0 \text{ on } \mathcal{S}_e(\Delta), \quad \forall \Delta \in \bar{\Delta}, \tag{5.9}$$

satisfying

$$\left[\begin{array}{c} * \\ * \\ \hline * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{array}\right]^* \left[\begin{array}{cc|cc|cc|cc} 0 & \mathcal{X} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathcal{X} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S & Q_{12} & S_{12} & 0 & 0 \\ 0 & 0 & S^* & R & S_{21}^* & R_{12} & 0 & 0 \\ \hline 0 & 0 & Q_{12}^* & S_{21} & Q_2 & S_2 & 0 & 0 \\ 0 & 0 & S_{12}^* & R_{12}^* & S_2^* & R_2 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & 0 & 0 & 0 & S_p^* & R_p \end{array}\right] \left[\begin{array}{cccc} I & 0 & 0 & 0 \\ \mathcal{A} & \mathcal{B}_u & \mathcal{B}_c & \mathcal{B}_p \\ \hline 0 & I & 0 & 0 \\ \mathcal{C}_u & \mathcal{D}_{uu} & \mathcal{D}_{uc} & \mathcal{D}_{up} \\ \hline 0 & 0 & I & 0 \\ \mathcal{C}_c & \mathcal{D}_{cu} & \mathcal{D}_{cc} & \mathcal{D}_{cp} \\ \hline 0 & 0 & 0 & I \\ \mathcal{C}_p & \mathcal{D}_{pu} & \mathcal{D}_{pc} & \mathcal{D}_{pp} \end{array}\right] < 0. \tag{5.10}$$

Proof:

We have already seen that 1. guarantees RQP. The equivalence of 1. and 2. follows from the so called full block S-procedure found in [Scherer, 2001], where it is given for real systems. Careful inspection reveals that extending it to complex systems is merely a question of substituting $.^T \to .^*$ and $\mathbb{R} \to \mathbb{C}$. $\lhd$

The main advantage of using 2. instead of 1. is that the parameter dependency only affects the choice of multipliers. This turns out to be very helpful in the synthesis to follow in Section 5.3.

5.3 LPV controller synthesis

This section reviews the LPV controller synthesis described in [Scherer, 2001]. First the interconnection of an LPV system and an LPV controller on the generalised LFT form described in the previous section is described. This leads to a closed-loop system on the same form. The Elimination Lemma for quadratic matrix inequalities (Lemma 2.9) is then applied in order to turn the analysis equation for the closed-loop into LMI synthesis equations. A partial proof is then given providing a construction for the controller. The synthesis LMIs are infinite-dimensional. In Section 5.3.3 an example is given of how to make them finite-dimensional. Finally Section 5.3.4 discusses how to overcome some of the numerical problems associated with solving the quadratic matrix inequality.

5.3.1 Closed-loop system

Consider the LPV system

$$\left[\begin{array}{c} \dot{x} \\ \hline z_u \\ z_p \\ y \end{array}\right] = \left[\begin{array}{c|ccc} A & B_u & B_p & B \\ \hline C_u & D_{uu} & D_{up} & E_u \\ C_p & D_{pu} & D_{pp} & E_p \\ C & F_u & F_p & 0 \end{array}\right] \left[\begin{array}{c} x \\ \hline w_u \\ w_p \\ u \end{array}\right] \tag{5.11}$$

with $x \in \mathbb{C}^{n_s}, u \in \mathbb{C}^m$ and $y \in \mathbb{C}^p$ representing states, inputs and outputs. $w_p \in \mathbb{C}^{n_{wp}}$ contains disturbance, noise, and command signals. $z_p \in \mathbb{C}^{n_{zp}}$ is the performance output, i.e. the signals to be controlled. $w_u \in \mathbb{C}^{n_{wu}}, z_u \in \mathbb{C}^{n_{zu}}$ are the channels connecting the time-varying parameters in Δ with the nominal system described by (5.11). Let the time-variations be given by

$$\left[\begin{array}{c} w_u \\ \hline z_u \end{array}\right] \in \mathcal{S}(\Delta) = \operatorname{im} S(\Delta) = \operatorname{im} \left[\begin{array}{c} S_1(\Delta) \\ \hline S_2(\Delta) \end{array}\right]. \tag{5.12}$$

This type of parameter dependency was discussed in Section 5.2.2. Assume that the time-varying parameters are known to be bounded by $\Delta \in \bar{\Delta}$.

Remark 5.7 As seen the system is required to be strictly proper in the channel from controller input u to the measurements y. In Section 2.4 it was discussed what to do when presented with a non-strictly proper system.

As discussed in the introduction to this chapter, the main idea behind LPV control is to let the controller have online access to the time-varying parameters. If both the system and the controller have standard LFT parameter dependency this can be represented as in Figure 5.3.

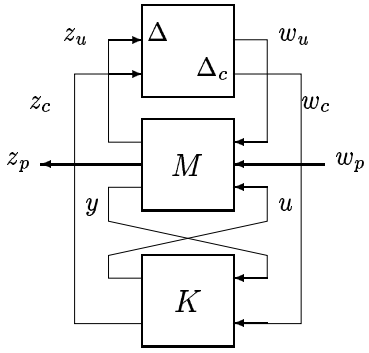


Figure 5.3: *Expanding the time-varying block, Δ, by a scheduling function, Δ_c.*

Remark 5.8 This could be viewed as a robust synthesis problem: Design the controller K such that the closed loop system is stable and fulfills some performance specification for any $\Delta \in \bar{\Delta}$. This would normally be a non-convex problem except for special cases such as state-feedback. Luckily, due to the particular structure of the problem, copying Δ into Δ_c will allow a wider set of multipliers, making the problem convex with some conservatism. Letting Δ_c be a nonlinear function of Δ will even remove this conservatism.

Now let the controller be given by

$$\left[\begin{array}{c} \dot{x}_c \\ \hline u \\ z_c \end{array}\right] = \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2} \\ \hline C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & D_{c22} \end{array}\right] \left[\begin{array}{c} x_c \\ \hline y \\ w_c \end{array}\right], \tag{5.13}$$

where $x_c \in \mathbb{C}^{n_c}, z_c \in \mathbb{C}^{k_c}$, and $w_c \in \mathbb{C}^{m_c}$. Let the *controller scheduling subspace* be given by

$$\left[\begin{array}{c} w_c \\ \hline z_c \end{array}\right] \in \mathcal{S}_c(\Delta) = \operatorname{im} S_c(\Delta) = \operatorname{im} \left[\begin{array}{c} S_{c1}(\Delta) \\ \hline S_{c2}(\Delta) \end{array}\right]. \tag{5.14}$$

We then have a closed-loop system on the form (5.6) with

$$\left[\begin{array}{c|c|c|c} \mathcal{A} & \mathcal{B}_u & \mathcal{B}_c & \mathcal{B}_p \\ \hline \mathcal{C}_u & \mathcal{D}_{uu} & \mathcal{D}_{uc} & \mathcal{D}_{up} \\ \mathcal{C}_c & \mathcal{D}_{cu} & \mathcal{D}_{cc} & \mathcal{D}_{cp} \\ \hline \mathcal{C}_p & \mathcal{D}_{pu} & \mathcal{D}_{pc} & \mathcal{D}_{pp} \end{array}\right] = \left[\begin{array}{cc|cc|c} A & 0 & B_u & 0 & B_p \\ 0 & 0 & 0 & 0 & 0 \\ \hline C_u & 0 & D_u & 0 & D_{up} \\ 0 & 0 & 0 & 0 & 0 \\ \hline C_p & 0 & D_{pu} & 0 & D_p \end{array}\right] +$$

$$\left[\begin{array}{ccc} 0 & B & 0 \\ I_{n_c} & 0 & 0 \\ \hline 0 & E_u & 0 \\ 0 & 0 & I_{k_c} \\ \hline 0 & E_p & 0 \end{array}\right] \left[\begin{array}{ccc} A_c & B_{c1} & B_{c2} \\ C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & D_{c22} \end{array}\right] \left[\begin{array}{cc|cc|c} 0 & I_{n_c} & 0 & 0 & 0 \\ C & 0 & F_u & 0 & F_p \\ 0 & 0 & 0 & I_{m_c} & 0 \end{array}\right], \quad (5.15)$$

and a parameter dependency given by (5.7), with

$$\left[\begin{array}{c} S_{1u}(\Delta) \\ S_{1c}(\Delta) \\ \hline S_{2u}(\Delta) \\ S_{2c}(\Delta) \end{array}\right] = \left[\begin{array}{cc} S_1(\Delta) & 0 \\ 0 & S_{c1}(\Delta) \\ \hline S_2(\Delta) & 0 \\ 0 & S_{c2}(\Delta) \end{array}\right].$$

We can now analyse the stability and performance of the closed-loop system by directly inserting in Theorem 5.6. Notice that

$$\mathcal{S}_e(\Delta) = \mathcal{S}(\Delta) \times \mathcal{S}_c(\Delta).$$

5.3.2 Controller synthesis

Assuming that (5.9) defines a convex set of possible multipliers, analysing the stability and performance of the closed-loop system by testing the feasibility of (5.10) is a convex problem since the only unknowns, $\mathcal{X}$ and P_e, enter linearly. In connection with synthesis it becomes a non-convex problem due to the presence of products of the multipliers and the closed-loop matrices. However, due to the specific structure of the problem, by applying the Elimination Lemma for quadratic matrix inequalities (Lemma 2.9), the synthesis problem can be cast as an LMI. This is due to the following lemma, which has been instrumental in most LPV results in the literature:

Lemma 5.9 (Hermitian multiplier extension) [Packard, 1994]

Let $X > 0, Y > 0 \in \mathbb{H}^{n\times n}$, and let r be a positive integer. Then there exist matrices $X_2 \in \mathbb{C}^{n\times r}, X_3 \in \mathbb{H}^{r\times r}$ such that

$$\begin{bmatrix} X & X_2 \\ X_2^* & X_3 \end{bmatrix} > 0 \text{ and } \begin{bmatrix} X & X_2 \\ X_2^* & X_3 \end{bmatrix}^{-1} = \begin{bmatrix} Y & * \\ * & * \end{bmatrix} \quad (5.16)$$

if and only if

$$\begin{bmatrix} X & I_n \\ I_n & Y \end{bmatrix} \geq 0 \text{ and } \operatorname{rank} \begin{bmatrix} X & I_n \\ I_n & Y \end{bmatrix} \leq n + r \tag{5.17}$$

Proof: See [Packard, 1994] or [Helmersson, 1995]. $\lhd$

Lemma B.3 gives a construction for the full rank Hermitian case.

Before giving the theorem for controller synthesis we need the following assumption on the performance index:

$$\operatorname{in}(P_p) = (\dim(Q_p), 0, \dim(R_p)),$$

which is needed in order to fulfill the inertia condition in the Elimination Lemma. This condition is non-restrictive, since most sensible choices of quadratic performance objectives do indeed fulfill this.

Partition the inverse of the performance matrix P_p as

$$P_p^{-1} = \begin{bmatrix} \tilde{Q}_p & \tilde{S}_p \\ \tilde{S}_p^* & \tilde{R}_p \end{bmatrix} \tag{5.18}$$

and define the multipliers

$$P \triangleq \begin{bmatrix} Q & S \\ S^* & R \end{bmatrix}, \qquad \tilde{P} \triangleq \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^* & \tilde{R} \end{bmatrix}, \tag{5.19}$$

partitioned to conform with $S(\Delta)$.

Theorem 5.10 (LPV controller synthesis) [Scherer, 2001]

The following two statements are equivalent

1. *A controller on the form (5.13) exists such that the closed loop system (5.15) admits an $\mathcal{X} > 0$ and a Hermitian multiplier P_e satisfying (5.9) and (5.10).*
2. *There exist Hermitian X,Y and Hermitian multipliers*

$$P > 0 \text{ on } \mathcal{S}(\Delta) \text{ and } \tilde{P} < 0 \text{ on } \mathcal{S}(\Delta)^{\perp}, \quad \forall \Delta \in \bar{\Delta} \tag{5.20}$$

satisfying the linear matrix inequalities

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} \geq 0 \tag{5.21}$$

$$\Psi^* \left[\begin{array}{c} * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{array}\right]^* \left[\begin{array}{cc|cc|cc} 0 & X & 0 & 0 & 0 & 0 \\ X & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S & 0 & 0 \\ 0 & 0 & S^* & R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & 0 & S_p^* & R_p \end{array}\right] \left[\begin{array}{ccc} I & 0 & 0 \\ A & B_u & B_p \\ \hline 0 & I & 0 \\ C_u & D_{uu} & D_{up} \\ \hline 0 & 0 & I \\ C_p & D_{pu} & D_{pp} \end{array}\right] \Psi \quad < \quad 0 \tag{5.22}$$

$$\Phi^* \left[\begin{array}{c} * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{array}\right]^* \left[\begin{array}{cc|cc|cc} 0 & Y & 0 & 0 & 0 & 0 \\ Y & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q} & \tilde{S} & 0 & 0 \\ 0 & 0 & \tilde{S}^* & \tilde{R} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \tilde{Q}_p & \tilde{S}_p \\ 0 & 0 & 0 & 0 & \tilde{S}_p^* & \tilde{R}_p \end{array}\right] \left[\begin{array}{ccc} -A^* & -C_u^* & -C_p^* \\ I & 0 & 0 \\ \hline -B_u^* & -D_{uu}^* & -D_{pu}^* \\ 0 & I & 0 \\ \hline -B_p^* & -D_{up}^* & -D_{pp}^* \\ 0 & 0 & I \end{array}\right] \Phi > 0 \tag{5.23}$$

where

$$\Phi = \begin{bmatrix} B^* & E_u^* & E_p^* \end{bmatrix}^\vdash, \quad \Psi = \begin{bmatrix} C & F_u & F_p \end{bmatrix}^\vdash. \tag{5.24}$$

Proof: For the full proof see [Scherer, 2001]. Here only a constructive proof of 2. $\Rightarrow$ 1. will be given with the following assumptions.

We assume that the system is on the standard LFT-form, i.e.

$$w_u(t) = \Delta(t) z_u(t) \tag{5.25}$$

or equivalently

$$S(\Delta) = \begin{bmatrix} \Delta \\ I \end{bmatrix},$$

and that Δ is square so that

$$n_{zu} = n_{wu} \triangleq n_u.$$

Then we have

$$\mathcal{S}(\Delta)^\perp = \operatorname{im} \tilde{S}(\Delta), \qquad \tilde{S}(\Delta) \triangleq \begin{bmatrix} I \\ -\Delta^* \end{bmatrix}.$$

We furthermore assume that

$$\operatorname{in}_-(\tilde{P}) = n_u \tag{5.26}$$

which simplifies the proof greatly. With this condition it is possible to let the controller be on the standard LFT form as well. The scheduling function will however still have to be a nonlinear function of Δ.

The proof will be in the form of a controller construction algorithm comprised of the following steps. First the extended multiplier P_e is constructed from P and $\tilde{P}$ and $\mathcal{X}$ is constructed from X and Y. Then a controller scheduling function is constructed to assure (5.9). The LTI part of the controller can then be found using Lemma 2.9. Assume now that $P, \tilde{P}, X$ and Y satisfying (5.20)-(5.23) have been found.

Extension of multipliers

By compactness of $\bar{\Delta}$ and by the strictness of (5.20) we can, if necessary, perturb $\tilde{P}$ to render it nonsingular. Define

$$N \triangleq P - \tilde{P}^{-1} \tag{5.27}$$

and let U be an orthonormal basis for the image of N such that

$$U^* N U = \begin{bmatrix} N_- & 0 \\ 0 & N_+ \end{bmatrix}, \quad N_- < 0, \quad N_+ > 0. \tag{5.28}$$

Choose the scheduling function dimensions as $k_c = \dim(N_+)$ and $m_c = \dim(N_-)$ and define

$$\begin{bmatrix} V_-(\Delta) & V_+(\Delta) \end{bmatrix} = S(\Delta)^* U \tag{5.29}$$

with V_-, V_+ having m_c, k_c columns respectively. By defining the extended multiplier as

$$P_e = \begin{bmatrix} P & U \\ U^* & (U^* N U)^{-1} \end{bmatrix} \tag{5.30}$$

we find by Lemma B.3 and Schur complement (Lemma 2.6) that P_e is nonsingular and

$$P_e^{-1} = \begin{bmatrix} \tilde{P} & * \\ * & * \end{bmatrix}, \quad \text{in}(P_e) = \text{in}(U^* N U) + \text{in}(\tilde{P}) = (m_c + n_u, 0, k_c + n_u). \tag{5.31}$$

To construct $\mathcal{X}$ we pick Z_x as an orthonormal basis of the image of $X - Y^{-1}$ and set

$$\mathcal{X} = \begin{bmatrix} X & Z_x \\ Z_x^* & \left(Z_x^*(X - Y^{-1})Z_x\right)^{-1} \end{bmatrix}. \tag{5.32}$$

By Lemma B.3 and Schur complement (Lemma 2.6) $\mathcal{X}$ will be positive definite and $\mathcal{X}^{-1} = \begin{bmatrix} Y & * \\ * & * \end{bmatrix}$.

Scheduling function

Defining $S_c(\Delta) = \begin{bmatrix} \Delta_c(\Delta) \\ I \end{bmatrix}$ we need to find a Δ_c such that

$$P_e > 0 \text{ on } \operatorname{im} S(\Delta) \times \operatorname{im} S_c(\Delta), \quad \forall \Delta \in \bar{\Delta}. \tag{5.33}$$

With the above definitions this can be written as

$$\begin{bmatrix} S(\Delta)^* P S(\Delta) & V_-(\Delta)\Delta_c(\Delta) + V_+(\Delta) \\ \Delta_c(\Delta)^* V_-(\Delta)^* + V_+(\Delta)^* & \Delta_c(\Delta)^* N_-^{-1} \Delta_c(\Delta) + N_+^{-1} \end{bmatrix} > 0. \tag{5.34}$$

Using Schur complement and a congruence transformation this is equivalent to

$$\begin{bmatrix} N_+^{-1} & \Delta_c(\Delta)^* \\ \Delta_c(\Delta) & -N_- \end{bmatrix} - \\ \begin{bmatrix} * & * \end{bmatrix}^* \left[S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* \right]^{-1} \begin{bmatrix} V_+(\Delta) & V_-(\Delta) N_-^* \end{bmatrix} > 0. \tag{5.35}$$

By choosing

$$\Delta_c(\Delta) = N_- V_-(\Delta)^* \left(S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* \right)^{-1} V_+(\Delta) \tag{5.36}$$

to make the off-diagonal blocks equal to zero, inequality (5.35) is equivalent to

$$N_+^{-1} - V_+(\Delta)^* \left[S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* \right]^{-1} V_+(\Delta) > 0$$

and

$$-N_- - N_- V_-(\Delta)^* \left[S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* \right]^{-1} V_-(\Delta) N_-^* > 0.$$

These are equivalent to

$$\begin{bmatrix} N_+^{-1} & V_+(\Delta)^* \\ V_+(\Delta) & S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* \end{bmatrix} > 0$$

and

$$\begin{bmatrix} -N_- & N_- V_-(\Delta)^* \\ V_-(\Delta) N_- & S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* \end{bmatrix} > 0$$

by Schur complement arguments. By Schur complement the latter is equivalent to $S(\Delta)^* P S(\Delta) > 0$, which is always true due to (5.20). The first inequality is by Schur complement equivalent to

$$S(\Delta)^* P S(\Delta) - V_-(\Delta) N_- V_-(\Delta)^* - V_+(\Delta) N_+ V_+(\Delta)^* = S(\Delta)^* \tilde{P}^{-1} S(\Delta) > 0,$$

where the equality follows from the definitions of N, V_-, and V_+. From Lemma 2.7, (5.20), and (5.26) we have

$$\text{in}(\tilde{P}^{-1}|_{\mathcal{S}(\Delta)}) = \text{in}(\tilde{P}) - \text{in}(\tilde{P}|_{\mathcal{S}(\Delta)^\perp}) = (n_u, 0, n_u) - (n_u, 0, 0) = (0, 0, n_u),$$

i.e. the first inequality is also always true.

Controller construction

Once the multipliers have been constructed and the fulfilment of (5.9) has been assured through (5.36), the controller matrices can be found as a solution to (5.10). This is a quadratic matrix inequality in the form given in Lemma 2.9, which can be seen after some rearrangements of rows and columns and by observing that

$$\left[\begin{array}{cc|cc|c} 0 & I_{n_c} & 0 & 0 & 0 \\ B^* & 0 & E_u^* & 0 & E_p^* \\ 0 & 0 & 0 & I_{k_c} & 0 \end{array}\right]^\vdash = \begin{bmatrix} \Phi_1 \\ 0 \\ \Phi_2 \\ 0 \\ \Phi_3 \end{bmatrix}, \tag{5.37}$$

and

$$\left[\begin{array}{cc|cc|c} 0 & I_{n_c} & 0 & 0 & 0 \\ C & 0 & F_u & 0 & F_p \\ 0 & 0 & 0 & I_{m_c} & 0 \end{array}\right]^\vdash = \begin{bmatrix} \Psi_1 \\ 0 \\ \Psi_2 \\ 0 \\ \Psi_3 \end{bmatrix}, \tag{5.38}$$

where

$$\begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix} = \Phi \quad \text{and} \quad \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_3 \end{bmatrix} = \Psi. \tag{5.39}$$

The inertia condition (2.10) is satisfied if the inner factor in (5.10) has negative inertia $(n_s + n_c + n_u + m_c + n_{wp})$ and positive inertia $(n_s + n_c + n_u + k_c + n_{zp})$. This is clearly fulfilled since

$$\text{in}\begin{pmatrix} 0 & \mathcal{X} \\ \mathcal{X} & 0 \end{pmatrix} = (n_s + n_c, 0, n_s + n_c), \tag{5.40}$$

$$\text{in}(P_e) = (n_u + m_c, 0, n_u + k_c), \tag{5.41}$$

$$\text{in}(P_p) = (n_{wp}, 0, n_{zp}). \tag{5.42}$$

$\triangleleft$

Remark 5.11 The proof given here is less general than in [Scherer, 2001]. For instance, the inertia condition (5.26) on $\tilde{P}$ is not necessary, but the constructive proof is a lot more complicated, and the scheduling function is given in a less explicit manner. The condition is non-convex, so it is not easy to enforce. However, practical experience shows that it often is fulfilled anyway, at least if enforcing the convex inertia restrictions discussed below in Section 5.3.3.

To summarise, assuming that the parameter dependency is given by (5.25), the algorithm for finding an LPV controller is as follows

- Solve the LMIs (5.21)-(5.23) with the additional conditions (5.20) and (5.26) to obtain X, Y, P, and $\tilde{P}$.
- Construct P_e as in (5.30) and $\mathcal{X}$ as in (5.32).
- Solve the quadratic matrix inequality (5.10) to obtain the controller matrices.
- The controller is now given by (5.13) with $w_c = \Delta_c z_c$, where Δ_c is given by (5.36).

5.3.3 Finite dimensional global solution

Since $\bar{\Delta}$ usually has infinitely many elements, (5.20) poses an infinite number of constraints on P and $\tilde{P}$. Let us again consider the LFT case. Then (5.20) is equivalent to

$$\begin{bmatrix}\Delta\\ I\end{bmatrix}^* \begin{bmatrix}Q & S\\ S^* & R\end{bmatrix}\begin{bmatrix}\Delta\\ I\end{bmatrix} > 0 \text{ and } \begin{bmatrix}I\\ -\Delta^*\end{bmatrix}^* \begin{bmatrix}\tilde{Q} & \tilde{S}\\ \tilde{S}^* & \tilde{R}\end{bmatrix}\begin{bmatrix}I\\ -\Delta^*\end{bmatrix} < 0. \tag{5.43}$$

Let us assume that $\bar{\Delta}$ can be described by

$$\bar{\Delta} = \mathbf{Co}(\bar{\Delta}_g), \qquad \bar{\Delta}_g = \{\sum_{i=1}^{n} L_i \Delta_i K_i^* : \Delta_i \in \bar{\Delta}_i\},$$

where $\bar{\Delta}_i$ are finite sets and L_i, K_i are matrices of full column rank. Since (5.43) is not linear in Δ it is necessary to introduce constraints on P and $\tilde{P}$ in order to have (5.43) on $\bar{\Delta}$ implied by (5.43) on the finite generator set $\bar{\Delta}_g$. Enforcing $Q < 0$ and $\tilde{R} > 0$ is sufficient to have this implication [Scherer, 1999]. Notice that if the parameter dependency is affine, i.e. $\mathcal{D}_{uu} = 0$, then the $(2,2)$-block of (5.22) is $Q + \mathcal{D}_{pu}^* R_p \mathcal{D}_{pu} < 0$. Due to $R_p \geq 0$ this implies $Q < 0$. Similarly (5.23) implies $\tilde{R} > 0$, so in the affine parameter dependency case there is no need for conservatism.

In the general rational case the least conservative constraints would appear to be

$$L_i^* Q L_i < 0, \quad K_i^* \tilde{R} K_i > 0, \qquad \forall i \in 1, ..., n.$$

With these constraints, (5.43) on $\bar{\Delta}_g$ implies (5.43) on $\bar{\Delta}$ [Scherer, 2001]. By adding these constraints, the synthesis inequalities (5.21)-(5.23) are therefore a finite-dimensional LMI, which can be implemented in a standard LMI solver as described in Section 2.3.1.

5.3.4 Solving the quadratic matrix inequality

The proof of Lemma 2.9 in Appendix B on page 173 provides a way to construct a solution to the quadratic matrix inequality (2.11). Numerical problems may arise, especially if P is ill-conditioned. The only inversions in the proof which need to be performed are the inversions of $S^*\Pi S$ in equation (B.19) and of Z_1. The following lemma provides a choice of L that will make $S^*\Pi S \approx -I$, making it easy to invert. Refer to the proof in the Appendix for definitions of the matrices.

Lemma 5.12 Let $U_B \begin{bmatrix}\Sigma_B & 0\end{bmatrix} V_B^* = B$ *be a singular value decomposition of* B*. With the definitions in the proof of Lemma 2.9 and with*

$$J = \begin{bmatrix} J_{11} & J_{12} \\ J_{12}^* & J_{22} \end{bmatrix} = V_B^* \begin{bmatrix} I \\ C \end{bmatrix}^* P \begin{bmatrix} I \\ C \end{bmatrix} V_B, \tag{5.44}$$

partitioned so $J_{22} \in \mathbb{H}^{(m-c_z)\times(m-c_z)}$*, choosing*

$$L = V_B \begin{bmatrix} I & 0 \\ 0 & Q \end{bmatrix}, \quad QQ^* = -J_{22}^{-1} \tag{5.45}$$

will provide

$$S^*\Pi S = -I. \tag{5.46}$$

Proof: Inserting $\begin{bmatrix} D_{12} \\ D_{22} \end{bmatrix} = K^* C V_B \begin{bmatrix} 0 \\ Q \end{bmatrix}$ and

$$\Pi \triangleq \begin{bmatrix} L & 0 \\ 0 & K^{-*} \end{bmatrix}^* P \begin{bmatrix} L & 0 \\ 0 & K^{-*} \end{bmatrix}$$

we have

$$S^*\Pi S = \begin{bmatrix} 0 \\ I \\ K^*CV_B \begin{bmatrix} 0 \\ Q \end{bmatrix} \end{bmatrix}^* \Pi \begin{bmatrix} 0 \\ I \\ K^*CV_B \begin{bmatrix} 0 \\ Q \end{bmatrix} \end{bmatrix} =$$

$$\begin{bmatrix} V_B \begin{bmatrix} 0 \\ Q \end{bmatrix} \\ CV_B \begin{bmatrix} 0 \\ Q \end{bmatrix} \end{bmatrix}^* P \begin{bmatrix} V_B \begin{bmatrix} 0 \\ Q \end{bmatrix} \\ CV_B \begin{bmatrix} 0 \\ Q \end{bmatrix} \end{bmatrix} =$$

$$\begin{bmatrix} 0 \\ Q \end{bmatrix}^* J \begin{bmatrix} 0 \\ Q \end{bmatrix} = Q^* J_{22} Q = -I. \quad (5.47)$$

The existence of a Q fulfilling $QQ^* = -J_{22}^{-1}$ follows from $S^*\Pi S < 0 \Rightarrow J_{22} < 0$. $\lhd$

Of course finding a suitable Q is no more numerically tractable than inverting J_{22}, but the point is that it does not have to be exact, an approximate solution will still make inversion of $S^*\Pi S$ well-conditioned.

5.4 Discrete time controller synthesis

When a model has been obtained based on physical considerations, it will be usually in continuous time. On the other hand, the controller usually has to be implemented in discrete time. There are two ways to do this. Either the controller is designed in continuous time and then discretised, or the model is discretised first, and a controller is then designed in discrete time.

Some of the advantages and disadvantages of the first method (controller discretisation) compared to the second method (model discretisation) are:

- Since the real plant operates in continuous time, stability and performance requirements are more naturally expressed in continuous time.
- The optimal sampling rate may depend more on the controller dynamics than on the plant dynamics. If the sampling frequency can be chosen freely, then it might be better to determine this after the controller synthesis.
- On the other hand, if the sampling frequency is given beforehand, then controller synthesis in continuous time might result in a controller with very high gains or unstable open-loop dynamics, which could result in the closed-loop behaviour not being preserved when implemented in discrete time. Limitations on the sampling frequency will be taken into account with model discretisation.

- Time delays can be modelled easily in discrete time, whereas in continuous time approximations such as Padé approximants (see e.g. [Franklin et al., 1994]) must be used.
- Depending on the type of parametrisation, it might be necessary to perform the discretisation of the controller at each sample. This may be computationally demanding.
- On the other hand, discretisation of a continuous time model with a simple parametrisation may result in a discrete time model with a more complicated parametrisation.

No matter which of the two ways is chosen, a discretisation has to be performed. Discretisation of LPV systems is less trivial than discretising LTI systems, since the system changes from sample to sample. Section 5.4.1 discusses discretisation of LPV systems on the LFT form. If the model is discretised, a discrete time version of the theory in Sections 5.2-5.3 is needed. Section 5.4.2 gives the discrete time version of robust quadratic performance. Section 5.4.3 describes LPV synthesis for discrete time synthesis. Finally, Section 5.4.4 discusses a few issues of discrete time controller design and implementation.

5.4.1 Discretisation

In [Apkarian, 1997] some suggestions on discretisation of LPV controllers are given. One of the main problems is that the controller has to be discretised at each sample, which is not necessarily feasible in real-time. If discretising the model instead, this will not be a problem. On the other hand, it is unclear how to contain the infinitely many models obtained at different parameter values in a simple LPV representation.

Fortunately, the LFT representation allows for a simpler alternative due to associativity of the star product. Define

$$D_{bl} = \begin{bmatrix} I & \sqrt{T_s}I \\ \sqrt{T_s}I & \frac{T_s}{2}I \end{bmatrix},$$

where T_s is the sampling period. A trapezoidal approximation (or bilinear transformation) of the continuous time system

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

is then given by

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} = D_{bl} \star \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}.$$

Now consider the LPV system (5.11) with $w_u(t) = \Delta(t) z_u(t)$ and write this as

$$\left[\begin{array}{c} \dot{x} \\ \hline z_p \\ y \end{array}\right] = \left(\left[\begin{array}{c|ccc} A & B_p & B & B_u \\ \hline C_p & D_{pp} & E_p & D_{pu} \\ C & F_p & 0 & F_u \\ C_u & D_{up} & E_u & D_{uu} \end{array}\right] \star \Delta \right) \left[\begin{array}{c} x \\ \hline w_p \\ u \end{array}\right]. \tag{5.48}$$

From the associativity of the star product we have

$$D_{bl} \star \left(\left[\begin{array}{c|ccc} A & B_p & B & B_u \\ \hline C_p & D_{pp} & E_p & D_{pu} \\ C & F_p & 0 & F_u \\ C_u & D_{up} & E_u & D_{uu} \end{array}\right] \star \Delta \right) = \left(D_{bl} \star \left[\begin{array}{c|ccc} A & B_p & B & B_u \\ \hline C_p & D_{pp} & E_p & D_{pu} \\ C & F_p & 0 & F_u \\ C_u & D_{up} & E_u & D_{uu} \end{array}\right] \right) \star \Delta.$$

Since the inside of the bracket on the right hand side is constant, it needs only be computed once, and the LFT structure is maintained.

$\Delta(t)$ is replaced by $\Delta_k \triangleq \Delta(kT_s)$. Notice that an underlying assumption is

$$\Delta(t) \approx \Delta(kT_s), \quad \text{for } kT_s \leq t \leq (k+1)T_s.$$

5.4.2 Discrete time analysis

Consider the discrete time version of the LPV system (5.3):

$$\begin{array}{rcl} \xi_{k+1} & = & \mathcal{A}(\Delta_k)\xi_k + \mathcal{B}(\Delta_k) w_{p,k} \\ z_{p,k} & = & \mathcal{C}(\Delta_k)\xi_k + \mathcal{D}(\Delta_k) w_{p,k}. \end{array} \tag{5.49}$$

Notice that it is an underlying assumption that the parameters are constant during the entire sampling period from time kT_s to $(k+1)T_s$, where T_s is the sampling period.

We will also define robust quadratic performance for discrete time systems:

Definition 5.13 (Robust quadratic performance,RQP)

We say that the system (5.49) achieves robust quadratic performance *with* performance index

$$P_p = P_p^* \triangleq \begin{bmatrix} Q_p & S_p \\ S_p^* & R_p \end{bmatrix}, \quad R_p \geq 0 \tag{5.50}$$

if

- *Positive constants K and α exist such that*

$$\|\xi_{k_1}\| \leq \|\xi_{k_0}\| K e^{-\alpha(k_1-k_0)} \text{ for } k_1 \geq k_0 \text{ and all } \Delta \in \bar{\Delta} \text{ if } w_{p,k} = 0.$$

- *The quadratic performance specification*

$$\exists \varepsilon > 0 : \sum_{k=k_0}^{k_1} \begin{bmatrix} w_{p,k}^* & z_{p,k}^* \end{bmatrix} P_p \begin{bmatrix} w_{p,k} \\ z_{p,k} \end{bmatrix} \leq -\varepsilon \sum_{k=k_0}^{k_1} w_{p,k}^* w_{p,k}, \tag{5.51}$$

holds for all $k_1 \geq k_0$ and all $\Delta_k \in \bar{\Delta}$ if $\xi_{k_0} = 0$.

We can now formulate the discrete time equivalent to Theorem 5.4. A similar result for the l_2-induced norm was given in [Doyle et al., 1991].

Theorem 5.14 (Robust quadratic performance analysis)

Suppose there exists an $\mathcal{X} > 0$ satisfying the LMI

$$\left[\begin{array}{cc} I & 0 \\ \mathcal{A}(\Delta) & \mathcal{B}(\Delta) \\ \hline 0 & I \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) \end{array}\right]^* \left[\begin{array}{cc|cc} -\mathcal{X} & 0 & 0 & 0 \\ 0 & \mathcal{X} & 0 & 0 \\ \hline 0 & 0 & Q_p & S_p \\ 0 & 0 & S_p^* & R_p \end{array}\right] \left[\begin{array}{cc} I & 0 \\ \mathcal{A}(\Delta) & \mathcal{B}(\Delta) \\ \hline 0 & I \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) \end{array}\right] < 0 \tag{5.52}$$

for all $\Delta \in \bar{\Delta}$ and with P_p satisfying (5.4). Then the system (5.49) achieves robust quadratic performance with performance index P_p.

Proof: Let $w_{p,k} = 0$ in (5.49) and choose

$$\mathcal{V}_k = \xi_k^* \mathcal{X} \xi_k$$

as a Lyapunov candidate for the unforced system. The difference from sample to sample is

$$\mathcal{V}_{k+1} - \mathcal{V}_k = \xi_k^* \mathcal{A}(\Delta)^* \mathcal{X} \mathcal{A}(\Delta) \xi_k - \xi_k^* \mathcal{X} \xi_k,$$

which implies that the system is uniformly exponentially stable if $\mathcal{A}(\Delta)^* \mathcal{X} \mathcal{A}(\Delta) < \mathcal{X}$. But this is immediately deduced from the upper left block in (5.52), which can be written as

$$\mathcal{A}(\Delta)^* \mathcal{X} \mathcal{A}(\Delta) - \mathcal{X} + \mathcal{D}(\Delta)^* R_p \mathcal{D}(\Delta) < 0.$$

Since $R_p \geq 0$ it is seen that if $\mathcal{X}$ renders (5.52) satisfied, the unforced system is uniformly exponentially stable.

Furthermore, due to the strictness of (5.52), we can add a small perturbation $G = \left[\begin{smallmatrix} 0 & 0 \\ 0 & \varepsilon I \end{smallmatrix}\right]$ to the left-hand side of the inequality without rendering it unsatisfied. Multiplying from the left and right by $\begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix}$ then gives

$$\begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix}^* \begin{bmatrix} \mathcal{A}(\Delta)^* \mathcal{X} \mathcal{A}(\Delta) - \mathcal{X} & \mathcal{A}(\Delta)^* \mathcal{X} \mathcal{B}(\Delta) \\ \mathcal{B}(\Delta)^* \mathcal{X} \mathcal{A}(\Delta) & \mathcal{B}(\Delta)^* \mathcal{X} \mathcal{B}(\Delta) \end{bmatrix} \begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix} +$$
$$\begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix}^* \begin{bmatrix} 0 & I \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) \end{bmatrix}^* P_p \begin{bmatrix} 0 & I \\ \mathcal{C}(\Delta) & \mathcal{D}(\Delta) \end{bmatrix} \begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix} + \begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix}^* G \begin{bmatrix} \xi_k \\ w_{p,k} \end{bmatrix} \leq 0$$

which reduces to

$$(\xi_{k+1} - \xi_k)^* \mathcal{X} (\xi_{k+1} - \xi_k) + \begin{bmatrix} w_{p,k} \\ z_{p,k} \end{bmatrix}^* P_p \begin{bmatrix} w_{p,k} \\ z_{p,k} \end{bmatrix} + \varepsilon w_{p,k}^* w_{p,k} \leq 0.$$

Summing from $k = k_0$ to $k = k_1$ with $\xi_{k_0} = 0$, $\mathcal{X} > 0$ then implies (5.51). $\lhd$

Notice that the only difference between Theorems 5.4 and 5.14 is the block containing $\mathcal{X}$. It turns out that this change can be carried through all the way to the synthesis LMIs.

Consider the discrete time version of the closed-loop LPV system (5.6) on LFT form

$$\left[\begin{array}{c} \xi_k \\ \hline z_{u,k} \\ z_{c,k} \\ z_{p,k} \end{array}\right] = \left[\begin{array}{c|ccc} \mathcal{A} & \mathcal{B}_u & \mathcal{B}_c & \mathcal{B}_p \\ \hline \mathcal{C}_u & \mathcal{D}_{uu} & \mathcal{D}_{uc} & \mathcal{D}_{up} \\ \mathcal{C}_c & \mathcal{D}_{cu} & \mathcal{D}_{cc} & \mathcal{D}_{cp} \\ \mathcal{C}_p & \mathcal{D}_{pu} & \mathcal{D}_{pc} & \mathcal{D}_{pp} \end{array}\right] \left[\begin{array}{c} \xi_k \\ \hline w_{u,k} \\ w_{c,k} \\ w_{p,k} \end{array}\right] \tag{5.53}$$

with a parameter dependency given by

$$\left[\begin{array}{c} w_{u,k} \\ w_{c,k} \\ \hline z_{u,k} \\ z_{c,k} \end{array}\right] \in \mathcal{S}_a(\Delta_k) = \operatorname{im} S_a(\Delta_k) = \operatorname{im} \left[\begin{array}{c} S_{a1}(\Delta_k) \\ \hline S_{a2}(\Delta_k) \end{array}\right] = \operatorname{im} \left[\begin{array}{c} S_{1u}(\Delta_k) \\ S_{1c}(\Delta_k) \\ \hline S_{2u}(\Delta_k) \\ S_{2c}(\Delta_k) \end{array}\right], \tag{5.54}$$

where $\xi \in \mathbb{C}^{n_s + n_c}$, $z_u \in \mathbb{C}^{n_{zu}}$, $z_c \in \mathbb{C}^{k_c}$, $z_p \in \mathbb{C}^{n_{zp}}$, $w_u \in \mathbb{C}^{n_{wu}}$, $w_c \in \mathbb{C}^{m_c}$, and $w_p \in \mathbb{C}^{n_{wp}}$.

The trajectories of (5.53) are then the same as for (5.49) with (5.8). Furthermore the full block S-procedure providing an RQP test for the LFT form is exactly as in Theorem 5.6 except for the change of the block containing $\mathcal{X}$.

5.4.3 Discrete time synthesis

As discussed in the previous section, the only difference between the continuous and discrete time versions of RQP analysis of LPV systems is the block which is $\begin{bmatrix} 0 & \mathcal{X} \\ \mathcal{X} & 0 \end{bmatrix}$

in continuous time and $\begin{bmatrix} -\mathcal{X} & 0 \\ 0 & \mathcal{X} \end{bmatrix}$ in discrete time. Since these two blocks have the same inertia, we can still apply the Elimination Lemma for quadratic matrix inequalities (Lemma 2.9). Furthermore, the annihilators still cancel the same rows and columns as in the continuous time case. This means that it is again the upper left blocks of $\mathcal{X}$ and $\mathcal{X}^{-1}$ which are preserved.

Consider the discrete time system

$$\left[\begin{array}{c} x_{k+1} \\ \hline z_{u,k} \\ z_{p,k} \\ y_k \end{array}\right] = \left[\begin{array}{c|ccc} A & B_u & B_p & B \\ \hline C_u & D_{uu} & D_{up} & E_u \\ C_p & D_{pu} & D_{pp} & E_p \\ C & F_u & F_p & 0 \end{array}\right] \left[\begin{array}{c} x_k \\ \hline w_{u,k} \\ w_{p,k} \\ u_k \end{array}\right] \tag{5.55}$$

with parameter dependence given by

$$\left[\begin{array}{c} w_{u,k} \\ \hline z_{u,k} \end{array}\right] \in \mathcal{S}(\Delta_k) = \operatorname{im} S(\Delta_k) = \operatorname{im} \left[\begin{array}{c} S_1(\Delta_k) \\ \hline S_2(\Delta_k) \end{array}\right]. \tag{5.56}$$

The controller will be on the form

$$\left[\begin{array}{c} x_{c,k+1} \\ \hline u_k \\ z_{c,k} \end{array}\right] = \left[\begin{array}{c|cc} A_c & B_{c1} & B_{c2} \\ \hline C_{c1} & D_{c11} & D_{c12} \\ C_{c2} & D_{c21} & D_{c22} \end{array}\right] \left[\begin{array}{c} x_{c,k} \\ \hline y_k \\ w_{c,k} \end{array}\right], \tag{5.57}$$

with the *controller scheduling subspace*

$$\left[\begin{array}{c} w_{c,k} \\ \hline z_{c,k} \end{array}\right] \in \mathcal{S}_c(\Delta_k) = \operatorname{im} S_c(\Delta_k) = \operatorname{im} \left[\begin{array}{c} S_{c1}(\Delta_k) \\ \hline S_{c2}(\Delta_k) \end{array}\right]. \tag{5.58}$$

Theorem 5.15 (Discrete time LPV controller synthesis)

The following two statements are equivalent

1. *A controller on the form (5.57)-(5.58) exists for the LPV system (5.55)-(5.54) such that the closed loop system achieves robust quadratic performance with performance index P_p satisfying (5.50).*
2. *There exist Hermitian X,Y and Hermitian multipliers*

$$P > 0 \text{ on } \mathcal{S}(\Delta) \text{ and } \tilde{P} < 0 \text{ on } \mathcal{S}(\Delta)^{\perp}, \quad \forall \Delta \in \bar{\Delta} \tag{5.59}$$

satisfying the linear matrix inequalities

$$\begin{bmatrix} X & I \\ I & Y \end{bmatrix} \geq 0 \tag{5.60}$$

$$\Psi^* \left[\begin{array}{c} * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{array}\right]^* \left[\begin{array}{cc|cc|cc} -X & 0 & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & Q & S & 0 & 0 \\ 0 & 0 & S^* & R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & Q_p & S_p \\ 0 & 0 & 0 & 0 & S_p^* & R_p \end{array}\right] \left[\begin{array}{ccc} I & 0 & 0 \\ A & B_u & B_p \\ \hline 0 & I & 0 \\ C_u & D_{uu} & D_{up} \\ \hline 0 & 0 & I \\ C_p & D_{pu} & D_{pp} \end{array}\right] \Psi \; < \; 0 \quad (5.61)$$

$$\Phi^* \left[\begin{array}{c} * \\ * \\ \hline * \\ * \\ \hline * \\ * \end{array}\right]^* \left[\begin{array}{cc|cc|cc} -Y & 0 & 0 & 0 & 0 & 0 \\ 0 & Y & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & \tilde{Q} & \tilde{S} & 0 & 0 \\ 0 & 0 & \tilde{S}^* & \tilde{R} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \tilde{Q}_p & \tilde{S}_p \\ 0 & 0 & 0 & 0 & \tilde{S}_p^* & \tilde{R}_p \end{array}\right] \left[\begin{array}{ccc} -A^* & -C_u^* & -C_p^* \\ I & 0 & 0 \\ \hline -B_u^* & -D_{uu}^* & -D_{pu}^* \\ 0 & I & 0 \\ \hline -B_p^* & -D_{up}^* & -D_{pp}^* \\ 0 & 0 & I \end{array}\right] \Phi \; > \; 0 \quad (5.62)$$

where

$$\Phi = \begin{bmatrix} B^* & E_u^* & E_p^* \end{bmatrix}^\vdash \quad \Psi = \begin{bmatrix} C & F_u & F_p \end{bmatrix}^\vdash$$

Proof: The proof follows the same lines as the proof of Theorem 5.10. $\mathcal{X}$ is constructed in exactly the same way from X and Y. $\lhd$

As indicated the controller construction follows the exact same lines as in the proof of Theorem 5.10.

5.4.4 Discrete time controller design and implementation

This section discusses a few issues on designing and implementing discrete time LPV controllers.

Restrictions on pole placement

The LMI approach to control allows for simple ways to restrict the closed-loop poles to certain areas of the s- or z-plane, see for instance [Chilali et al., 1999]. Note that it does not really make sense to take about poles for a time-varying system, but restricting the eigenvalues of $\mathcal{A}(\Delta)$ has a similar meaning. For example, it is very easy to ensure a certain decay rate in discrete time by restricting the eigenvalues to be within a margin β of the unit circle, i.e. $\rho(\mathcal{A}(\Delta)) < 1 - \beta, \quad \forall \Delta \in \bar{\Delta}$, where ρ denotes the spectral radius. This can be enforced by replacing the $\begin{bmatrix} -\mathcal{X} & 0 \\ 0 & \mathcal{X} \end{bmatrix}$-block in Theorem 5.52 by

$\begin{bmatrix} -\mathcal{X} & 0 \\ 0 & (1-\beta)^{-2}\mathcal{X} \end{bmatrix}$. In the synthesis in Theorem 5.15 this corresponds to changing $\begin{bmatrix} -X & 0 \\ 0 & X \end{bmatrix}$ to $\begin{bmatrix} -X & 0 \\ 0 & (1-\beta)^{-2}X \end{bmatrix}$ in (5.61) and $\begin{bmatrix} -Y & 0 \\ 0 & Y \end{bmatrix}$ to $\begin{bmatrix} -Y & 0 \\ 0 & (1-\beta)^{2}Y \end{bmatrix}$ in (5.62).

Implementation

A controller on the form (5.57)-(5.58) can be be implemented through a reformulation along the lines of Lemma 5.5. The controller (5.57)-(5.58) can be written as

$$\begin{array}{rcl} x_{c,k+1} & = & A_c(\Delta_k)x_{c,k} + B_c(\Delta_k)y_k \\ u_k & = & C_c(\Delta_k)x_{c,k} + D_c(\Delta_k)y_k \end{array} \tag{5.63}$$

where

$$\begin{bmatrix} A_c(\Delta_k) & B_c(\Delta_k) \\ C_c(\Delta_k) & D_c(\Delta_k) \end{bmatrix} \triangleq \begin{bmatrix} A_c & B_{c1} \\ C_{c1} & D_{c11} \end{bmatrix} + \begin{bmatrix} B_{c2} \\ D_{c12} \end{bmatrix} S_{c1}(\Delta_k)\left[S_{c2}(\Delta_k) - D_{c22}S_{c1}(\Delta_k)\right]^{\dagger}\begin{bmatrix} C_{c2} & D_{c21} \end{bmatrix}. \tag{5.64}$$

Some computation power may be saved be working with the scheduling vectors w_c and z_c rather than computing the entire matrix in (5.64) along the following lines

$$\begin{array}{rcl} \tilde{z}_{c,k} & = & C_{c2}x_{c,k} + D_{c21}y_k, \\ w_{c,k} & = & S_{c1}(\Delta_k)\left[S_{c2}(\Delta_k) - D_{c22}S_{c1}(\Delta_k)\right]^{\dagger}\tilde{z}_{c,k}, \\ u_k & = & C_{c1}x_{c,k} + D_{c11}y_k + D_{c12}w_{c,k}, \\ x_{c,k+1} & = & A_c x_{c,k} + B_{c1}y_k + B_{c2}w_{c,k}, \end{array}$$

but unless D_{c22} happens to be zero, there is in general no way to avoid the inversion.

5.5 Complex LPV systems

Certain systems with a particular structure can be transformed into an equivalent system with half the number of states, inputs, and outputs, assuming that the output can be decomposed in a meaningful way. We shall refer to this type of system as *complex-formed*, since the transformation will typically be from a real-valued system to a complex-valued one.

The current part of the induction motor model in Section 3.3.2 is complex-formed if we view ω_r as a parameter rather than a state.

One would expect that if a system is complex-formed, then the optimal controller is complex-formed as well. This section establishes theoretical evidence of this. Section 5.5.1 shows that addition, multiplication, and inversion of complex-formed matrices preserves the structure. Section 5.5.2 considers complex-formed linear systems and shows that for a special case, restricting the controller to be complex-formed as well does not limit the achievable performance. Section 5.5.3 considers LMIs with complex-formed matrices and shows that for a simple type, restricting the solution to be complex-formed as well will not affect the feasibility. Finally Section 5.5.4 considers LPV controller synthesis. It is shown that for a complex-formed LPV system the optimal controller is also complex-formed.

5.5.1 Complex-formed matrices

Define the following convex set:

Definition 5.16 (Complex-formed matrix, $\mathcal{C}_M^{2\cdot\times 2\cdot}$)

$$\mathcal{C}_M^{2m\times 2n} \triangleq \{M : M = \begin{bmatrix} M_r & -M_i \\ M_i & M_r \end{bmatrix}, M_r, M_i \in \mathbb{C}^{m\times n}\}.$$

If a matrix is in $\mathcal{C}_M^{2\cdot\times 2\cdot}$ we will call it complex-formed.

It is obvious that if $M_1, M_2 \in \mathcal{C}_M^{2m\times 2n}$, $M_3 \in \mathcal{C}_M^{2n\times 2p}$ we also have

$$M_1 + M_2 \in \mathcal{C}_M^{2m\times 2n} \text{ and } M_1 M_3 \in \mathcal{C}_M^{2m\times 2p}.$$

If $M_4 \in \mathcal{C}_M^{2m\times 2m}$ is non-singular we also have $M_4^{-1} \in \mathcal{C}_M^{2m\times 2m}$:

Lemma 5.17 Let $M \in \mathcal{C}_M^{2m\times 2m}$ be nonsingular. Then $M^{-1} \in \mathcal{C}_M^{2m\times 2m}$ as well.

Proof: A proof is given in the appendix on page 175. ◁

Define the following transformation from $\mathcal{C}_M^{2m\times 2n}$ to $\mathbb{C}^{m\times n}$:

Definition 5.18 ($T_{\mathcal{C}}(\cdot)$-transformation)

$$T_{\mathcal{C}}\left(\begin{bmatrix} M_r & -M_i \\ M_i & M_r \end{bmatrix}\right) \triangleq M_r + jM_i. \tag{5.65}$$

Remark 5.19 This transformation has no unique inverse unless some specific constraints on M_r and M_i are given, for instance that they should be real.

We can now give the following extension to Lemma 5.17:

Lemma 5.20 Let $M \in \mathcal{C}_M^{2m\times 2m}$ be nonsingular. Then

$$T_{\mathcal{C}}\left(M^{-1}\right) = T_{\mathcal{C}}\left(M\right)^{-1}. \tag{5.66}$$

Proof:

A proof is given in the appendix on page 176. $\triangleleft$

5.5.2 Complex-formed linear systems

By Lemma 5.17 it follows that if all matrices of a state space system are of this particular form, then the transfer function is too.

Define the following set of matrix transfer functions:

Definition 5.21 (Complex-formed system, $\mathcal{C}_S$)

$$\mathcal{C}_S^{2m\times 2n} = \{T(s) : T(s) \in \mathcal{C}_M^{2m\times 2n}, \quad \forall s\}.$$

Corollary 5.22 Consider the state space system

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du,$$

and assume $A \in \mathcal{C}_M^{2n\times 2n}, B \in \mathcal{C}_M^{2n\times 2m}, C \in \mathcal{C}_M^{2p\times 2n}$, and $D \in \mathcal{C}_M^{2p\times 2m}$. Then the transfer function $G(s) \triangleq C(sI_n - A)^{-1}B + D$ is in $\mathcal{C}_S$:

$$G(s) \in \mathcal{C}_S^{2p\times 2m}.$$

Furthermore

$$T_{\mathcal{C}}\left(G(s)\right) = T_{\mathcal{C}}\left(C\right)\left(sI_{\frac{n}{2}} - T_{\mathcal{C}}\left(A\right)\right)^{-1} T_{\mathcal{C}}\left(B\right) + T_{\mathcal{C}}\left(D\right).$$

Remark 5.23 The eigenvalues of a square complex-formed real matrix, A, consist of complex conjugated and double real pairs. The eigenvalues of $T_{\mathcal{C}}\left(A\right)$ consist of one from each pair.

Remark 5.24 The transfer function, $G(s)$ of a state space system with real matrices is symmetric around the real axis. This is not necessarily true for $T_{\mathcal{C}}\left(G(s)\right)$. One consequence is that bode plots need to be shown for both positive and negative frequencies.

The following theorem shows that for a special system with complex-formed transfer functions there is no loss in achievable performance by restricting the controller to be complex-formed as well.

Define the matrix

$$Q \triangleq \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \tag{5.67}$$

where the dimension will be apparent from the context.

Lemma 5.25 Let $G, H, F = F^* \in \mathcal{C}_S$ and

$$QG + G^*Q + QHH^*Q + F < 0 \qquad \forall s \tag{5.68}$$

Then

$$F < 0 \qquad \forall s \tag{5.69}$$

Proof: Obviously (5.68) implies $N \triangleq QG + G^*Q + F < 0$. Then also

$$\begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} N \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} + N = 2F < 0 \tag{5.70}$$

$\triangleleft$

Theorem 5.26 Let

$$M \triangleq \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \qquad M_{22} = 0, \tag{5.71}$$

where M_{11}, M_{12}, $M_{21} \in \mathcal{C}_S$. If a controller K exists yielding closed-loop $\mathcal{H}_\infty$ performance $||M \star K||_\infty < \gamma$ then a controller $K_1 \in \mathcal{C}_S$ exists with equal or better closed loop performance, i. e. $||M \star K_1||_\infty < \gamma$.

Proof: The closed loop transfer function is $C_l = M_{11} + M_{12}KM_{21}$. $||C_l||_\infty < \gamma$ if and only if $C_l^* C_l < \gamma^2 I$. Decompose K as $K = K_1 + K_2 Q$, where $K_1, K_2 \in \mathcal{C}_S$. When calculating $C_l^* C_l - \gamma^2 I$ it can be put on the form in (5.68), where G and H are zero when K_2 is zero and where F does not depend on K_2. By Lemma 5.25 the performance will then at least be maintained by setting $K_2 = 0$. $\triangleleft$

Remark 5.27 Note that since the performance holds at all frequencies, Theorem 5.26 holds equally well for $\mathcal{H}_2$ performance.

Remark 5.28 The system in (5.71) is somewhat specialised due to the condition $M_{22} = 0$. Given a non-zero (but stable) M_{22}, we can design a K for M with $M_{22} = 0$ and then implement the controller

$$\begin{bmatrix} 0 & I \\ I & -M_{22} \end{bmatrix} \star K. \tag{5.72}$$

If $M_{22} \in \mathcal{C}_S$ then so is (5.72) *assuming* that $(-M_{22}) \star K$ is well-posed. Otherwise the controller has to be implemented as the two separate blocks in (5.72).

The special system (5.71) with $M_{22} = 0$ can be seen as an observer problem.

5.5.3 Complex-formed LMIs

We will now turn our attention to LMIs where the known matrices are complex-formed. We will show that for a basic type of LMI, there is no loss of feasibility by restricting the solution to be complex-formed as well.

Lemma 5.29 Define

$$J_k \triangleq \begin{bmatrix} 0 & -I_k \\ I_k & 0 \end{bmatrix},$$

and assume $A \in \mathcal{C}_M^{2m\times 2n}$. *Then*

$$J_m A = A J_n.$$

Proof: By trivial calculation. $\lhd$

Notice that $T_{\mathcal{C}}(J_k) = jI_k$.

Theorem 5.30 Let $A \in \mathcal{C}_M^{2n\times 2m}, B \in \mathcal{C}_M^{2l\times 2n}, C = C^* \in \mathcal{C}_M^{2n\times 2n}$. *If the LMI*

$$L(X) = AXB + B^*X^*A^* + C > 0 \tag{5.73}$$

has a solution $X \in \mathbb{C}^{2m\times 2l}$, *then it also has a solution* $X_c \in \mathcal{C}_M^{2m\times 2l}$.

Proof: Define J_k as in Lemma 5.29. Let

$$X = \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix}$$

be a solution to (5.73). Decompose X as

$$X = X_c + \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} X_{\bar{c}}, \quad X_c \in \mathcal{C}_M^{2l\times 2n}, X_{\bar{c}} \in \mathcal{C}_M^{2l\times 2n}.$$

X_c is unique and given by

$$X_c = \frac{1}{2}(X + J_m^* X J_l) = \frac{1}{2}\begin{bmatrix} X_1 + X_4 & -(X_3 - X_2) \\ X_3 - X_2 & X_1 + X_4 \end{bmatrix}. \tag{5.74}$$

$L(X) > 0$ implies

$$J_n^* L(X) J_n > 0, \tag{5.75}$$

and thus

$$\frac{1}{2}\left(L(X) + J_n^* L(X) J_n\right) \quad = \quad \frac{1}{2}\left(L(X + J_m^* X J_l)\right) \quad = \quad L(X_c) \quad > \quad 0.$$

$\triangleleft$

In this theorem, X is assumed to be unstructured, but the statement holds whenever the projection in (5.74) preserves the structure, for instance if X_1, X_2, X_3, and X_4 all have the same block-diagonal structure.

Since J_k is real the above theorem also holds in the real domain:

Corollary 5.31 Let $A \in \mathcal{C}_M^{2n\times 2m} \cap \mathbb{R}^{2n\times 2m}$, $B \in \mathcal{C}_M^{2l\times 2n} \cap \mathbb{R}^{2l\times 2n}$, $C = C^* \in \mathcal{C}_M^{2n\times 2n} \cap \mathbb{R}^{2n\times 2n}$. *If the LMI*

$$AXB + B^T X^T A^T + C > 0 \tag{5.76}$$

has a solution $X \in \mathbb{R}^{2m\times 2l}$, *then it also has a solution* $X_c \in \mathcal{C}_M^{2m\times 2l} \cap \mathbb{R}^{2m\times 2l}$.

5.5.4 Complex-formed LPV synthesis

Consider the system (5.11) and assume all matrices to be complex-formed. Assume furthermore that the parameter dependency is given by (5.12) with $S_1(\Delta)$ and $S_2(\Delta)$ complex-formed for all $\Delta \in \bar{\Delta}$.

Now consider controller synthesis to obtain RQP with a performance index with complex-formed sub-matrices. Since the annihilators can be chosen as complex-formed, all sub-matrices in the LMIs in statement 2. of Theorem 5.10 are complex-formed. By suitable permutations, all the synthesis LMIs can be brought onto the form (5.73). To illustrate this, consider the following simple example.

Example 5.32 Consider the LMI in Q, S, and R

$$M = \begin{bmatrix} D \\ E \end{bmatrix}^* \begin{bmatrix} Q & S \\ S^* & R \end{bmatrix} \begin{bmatrix} D \\ E \end{bmatrix} > 0,$$

where $D \in \mathcal{C}_M^{2m\times 2(n+p)}$, $E \in \mathcal{C}_M^{2l\times 2(n+p)}$, $Q \in \mathbb{H}^{2n\times 2n}$, and $R \in \mathbb{H}^{2p\times 2p}$. Partition the decision variables as

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^* & Q_{22} \end{bmatrix}, \qquad R = \begin{bmatrix} R_{11} & R_{12} \\ R_{12}^* & R_{22} \end{bmatrix}, \qquad S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}.$$

Define the permutation matrix

$$P \triangleq \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & 0 & I_n & 0 \\ 0 & I_p & 0 & 0 \\ 0 & 0 & 0 & I_p \end{bmatrix}.$$

Since $PP^* = I_{2(n+p)}$ we then have

$$M = \underbrace{\begin{bmatrix} D \\ E \end{bmatrix}^*}_{D_p^*} \underbrace{P\,P^* \begin{bmatrix} Q & S \\ S^* & R \end{bmatrix} P\,P^*}_{X_p} \underbrace{\begin{bmatrix} D \\ E \end{bmatrix}}_{D_p} > 0,$$

where $D_p \in \mathcal{C}_M^{2(m+l)\times 2(n+p)}$. The important point now is that performing the projection in (5.74) on X_p and then swapping the rows and columns back preserves all structural requirements on Q, S, and R, for instance the projections of Q and R will still be Hermitian. In conclusion, if we have Q_0, S_0, and R_0 solving $M > 0$, then so will

$$\begin{bmatrix} Q & S \\ S^* & R \end{bmatrix} = \frac{1}{2}\left(PX_0P^* + PJ_{n+p}^* X_0 J_{n+p} P^*\right),$$

where

$$X_0 = P^* \begin{bmatrix} Q_0 & S_0 \\ S_0^* & R_0 \end{bmatrix} P.$$

This method can be applied to all the LMIs in statement 2. of Theorem 5.10. We can therefore always restrict ourselves to a solution where $X, Y, Q, R, S, \tilde{Q}, \tilde{S}$, and $\tilde{R}$ are all complex-formed. The extended multiplier P_e, the scheduling function $S_c(\Delta)$ and the Lyapunov matrix $\mathcal{X}$ will then have complex-formed submatrices. When permuting (5.10) in order to make it correspond to the quadratic matrix inequality in (2.11), it can be done in such a way that A, B, C, and P in (2.11) are all complex-formed. It is then possible to solve the quadratic matrix inequality along the lines of the proof in the appendix in such a way that all controller matrices are also complex-formed.

Corollary 5.33 Consider system (5.11) and assume all matrices to be complex-formed. Assume furthermore that the parameter dependency is given by (5.12) with $S_1(\Delta)$ and $S_2(\Delta)$ complex-formed for all $\Delta \in \bar{\Delta}$. In addition, let the performance index P_p have complex-formed sub-matrices.

If a controller exists achieving RQP with performance index P_p, then there exists a controller with complex-formed matrices and scheduling function also achieving this.

In practical situations we will typically be dealing with an LPV system with *real* complex-formed matrices. The best way to design a controller would then seem to be

- Solve the synthesis LMIs in Theorem 5.10 restricting $X, Y, Q, R, S, \tilde{Q}, \tilde{S}$, and $\tilde{R}$ to be have complex-formed submatrices. This approximately halves the number of decision variables.
- Perform the $T_{\mathcal{C}}(\cdot)$-transformation on all submatrices, thereby obtaining complex matrices of half the original size.

- Extend the multiplier and the Lyapunov matrix.
- Solve the quadratic matrix inequality. Doing these last two steps in the complex domain makes the problems slightly sounder numerically.

How to implement the controller depends on whether complex arithmetic is available on the particular platform for implementation. If so, then convert the measurement

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \to y_1 + jy_2$$

and compute the control signal u. Output the signal $\begin{bmatrix} \Re\{u\} \\ \Im\{u\} \end{bmatrix}$ to the system.

If no complex arithmetic is available, then the controller matrices must be transformed back, for instance

$$A_c \to \begin{bmatrix} \Re\{A_c\} & -\Im\{A_c\} \\ \Im\{A_c\} & \Re\{A_c\} \end{bmatrix}.$$

Care should be taken to exploit the complex-formed structure in order to reduce computational complexity, for instance in computing the inverse in (5.36).

5.6 LPV flux observer

An essential part of the rotor flux oriented control scheme discussed in Chapter 4 is the rotor flux observer. If a good estimate of the rotor flux (or equivalent magnetising current) is not available, the partial decoupling of the stator current into torque and rotor flux controlling parts, as described in Section 4.1, will not be achieved, and the closed-loop performance will be degraded.

As discussed in Section 3.3 the induction motor model can be considered as an LPV system by considering the rotor speed as a time-varying parameter. In this section a discrete time flux observer is designed using the synthesis method described in Sections 5.2-5.4. The aim is to design an observer with theoretically guaranteed convergence, which works well over a wide range of speeds, is simple to implement in real-time, and furthermore requires very little tuning.

In Section 5.6.1, the induction motor model is put on the LFT form needed for the synthesis, and it is discussed how it is discretised. In Section 5.6.2 the actual synthesis of the observer is discussed. Finally, experiments are performed on the laboratory setup. This is done both with and without a speed sensor. The experiments are described in Section 5.6.3.

First, some other flux observers found in the literature are briefly discussed. The flux observer presented in [Jansen and Lorenz, 1992] (the JL-observer) has already been described in Section 4.3.1. Based on physical insight a simple combination of two basic models were combined. The JL-observer can be tuned to have fast convergence and is relatively simple to implement.The main complication is the tuning of the two complex constants to ensure satisfactory behaviour over a wide speed range. The following is a non-exhaustive review of other existing rotor flux observer designs. Common to all of the following (as well as the JL-observer) is that they are based on measurements of the rotor speed, the stator current, and the stator voltage. In some of the following papers the voltage command is used instead of stator voltage measurements, but in general some representation of the stator voltage is needed. In several of the papers adaption of the rotor time constant and other parameters is also discussed. We will, however, not focus on parameter adaption.

In [Nilsen and Kaźmierkowski, 1992] the stator current is considered as the input to the induction motor, and the stator voltage prediction error is used to update the flux estimate. The observer gain is chosen from physical considerations. The observer would appear to work well in practice, but the theoretical justification is somewhat unclear. The paper also suggests how to observe parameter variations in the rotor time constant T_r and the mutual inductance L_m.

In [Manes et al., 1996] it is suggested to use an extended Luenberger observer with the load torque m_L to be slowly varying. Even though the design requires the inversion of a (sparse) 6×6 matrix at each time step, the observer performance is demonstrated on a real-time system and performs well. The main problem is that it requires tuning of six parameters. Even though these six parameters can be directly interpreted as closed-loop eigenvalues, this tuning may be tricky.

In [Martin and Rouchon, 2000] two simple flux observers are discussed. The first is basically an open-loop simulation of the motor. It is shown that if the correct parameters are used, the estimate will converge to the true value. The convergence rate is, however, very slow. The second observer is somewhat similar to the JL-observer although it is less obvious how to interpret it from a physical point of view. It is proved that the estimate converges to the true value. The convergence rate can be made arbitrarily fast. Just as the JL-observer it requires the tuning of two complex parameters, which may be difficult. In addition it is unclear how well the observer works in practice.

In [Benchaib and Edwards, 2000] a sliding mode observer is designed as part of a sliding mode control. The overall control systems displays some chattering, but the flux estimates are satisfactory. It would, however, be tricky to implement the scheme in real-time.

In [Marino et al., 1996] a controller is designed using a backstepping method. The scheme includes an observer which also estimates the rotor time constant and the load torque m_L, assuming the latter to be constant. Simulation results indicate a good performance, but no real-time experiments are presented.

In [Petersen and Pulle, 1998] it is suggested to use a Kalman filter with the gain designed from a deterministic viewpoint. In [Petersen and Pulle, 1997] it is described how to extend the method to be robust to parameter uncertainty. The basic idea of the deterministic viewpoint is not to compute the Kalman gain from an estimate of the noise variance, since it can be difficult to obtain a reasonable model of the noise. Instead, the noise variance is used as a tuning parameter. At each time step an ellipsoid containing all the possible system states that are consistent with the measurements are found. The centre of this ellipsoid is then used as the current estimate. This method is somewhat ad hoc, but is on the other hand easy to tune. The main problem is that it is computationally heavy.

In [Trangbæk, 2000] an LPV observer was designed using the structured singular value approach with block-diagonal multipliers. The performance was good, but the conservatism due to restrictions on the multipliers made it necessary to restrict operation to a small range of speeds.

The following describe a novel approach, the use of full block multipliers, to flux observer design. The performance is theoretically justified in a wide range of speeds. The resulting observer is easy to implement and most importantly requires practically no tuning in order to perform well.

Simultaneously with this work, a similar LPV observer design has been developed in [Darengosse et al., 2000]. As the main difference it is designed in continuous time and a somewhat unorthodox online discretisation has to be performed.

5.6.1 Induction motor model

From Section 3.3.1 we have the following model of the induction motor:

$$\begin{aligned} \dot{x}_{sc} &= A_{sc}x_{sc} + B_{sc}\bar{u}_s, \\ x_{sc} &= \begin{bmatrix} \bar{i}_s \\ \bar{i}_m \end{bmatrix}, \\ A_{sc} &= \begin{bmatrix} \frac{L_m^2 R_r + R_s L_r^2}{L_r(L_m^2 - L_s L_r)} & \frac{L_m^2 (jL_r\omega_r - R_r)}{L_r(L_m^2 - L_s L_r)} \\ \frac{R_r}{L_r} & j\omega_r - \frac{R_r}{L_r} \end{bmatrix}, \\ B_{sc} &= \begin{bmatrix} \frac{L_r}{L_s L_r - L_m^2} \\ 0 \end{bmatrix}, \end{aligned} \tag{5.77}$$

$$\begin{aligned} m_e &= \tfrac{3 Z_p L_m^2}{2 L_r} \Im\{\bar{i}_s \bar{i}_m^*\}, \\ \dot{\omega}_r &= Z_p \dot{\omega}_{mech} = \tfrac{Z_p}{J}(m_e - m_L). \end{aligned} \tag{5.78}$$

We assume that we can measure the stator voltage $\bar{u}_s$, the stator current $\bar{i}_s$, and the rotational speed ω_r. The aim is to estimate the magnetising current $\bar{i}_m$ using these

measurements. We will assume that the rotational speed belongs to a finite interval, $\omega_r(t) \in [-\omega_{r,max}; \omega_{r,max}]$. We shall make no assumptions on the load torque m_L. Consequently, very little information about $\bar{i}_m$ can be recovered by considering the mechanical equations (5.78). We will therefore consider ω_r as a time-varying parameter and base the observer on the current equations (5.77). The stator voltage is considered as a measurable disturbance.

The model can now be written directly on the basic LPV form (5.1), with $\theta(t) = \omega_r$, $A(\theta(t)) = A_{sc}$, $B(\theta(t)) = B_{sc}$, and $u = \bar{u}_s$.

LFT form

The model can easily be put on the LFT form by the standard "pulling out the uncertainties" procedure (see [Zhou et al., 1996]). First write A_{sc} as

$$A_{sc} = A_0 + \omega_r A_{wr},$$

and factor A_{wr} as

$$A_{wr} = A_1 A_2^*,$$

where A_1 and A_2 are of full column rank. The model can now be put on the form in (5.11) with

$$A = A_0, \quad B_u = \sqrt{\omega_{r,max}} A_1, \quad B_p = B_{sc}, \tag{5.79}$$

$$C_u = \sqrt{\omega_{r,max}} A_2^*, \quad D_{uu} = 0, \quad D_{up} = 0, \tag{5.80}$$

and a parameter dependency given by the standard LFT representation

$$\left[\begin{array}{c} w_u \\ \hline z_u \end{array}\right] \in \mathcal{S}(\Delta) = \operatorname{im} S(\Delta), \qquad S(\Delta) = \begin{bmatrix} \Delta \\ I \end{bmatrix}, \qquad \Delta = \frac{\omega_r}{\omega_{r,max}}.$$

Notice that the synthesis method in Section 5.3 does not require Δ to be scaled to $[-1; 1]$. A larger interval would simply limit the range of multipliers allowed by (5.20). The scaling is performed here for numerical reasons.

The rest of the system matrices are designed in order to pose the observer problem as a control problem:

$$\begin{aligned} B &= 0, \\ E_u &= 0, \\ C_p = \begin{bmatrix} 0 & -1 \end{bmatrix}, \qquad D_{pu} &= 0, \qquad D_{pp} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \\ C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \qquad F_u &= 0, \qquad F_p = \begin{bmatrix} 0 & \sigma_i & 0 \\ 1 & 0 & \sigma_u \end{bmatrix}. \end{aligned}$$

This setup is also illustrated in Figure 5.4 as the part within the dashed box.

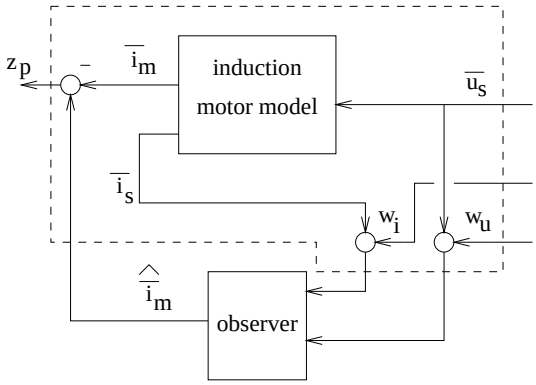


Figure 5.4: *Model setup for the formulation of the flux observer as a generic control problem.*

The performance input $w_p = \begin{bmatrix} \bar{u}_s \\ w_i \\ w_u \end{bmatrix}$ consists of the stator voltage acting as a disturbance, and scaled measurement noise. The measurement vector

$$y = \begin{bmatrix} \bar{i}_s + \sigma_i w_i \\ \bar{u}_s + \sigma_u w_u \end{bmatrix}$$

consists of stator current and voltage corrupted by noise. The performance output is $z_p = u - \bar{i}_m$. The observer will be designed to minimise the l_2-induced norm of the gain from w_p to z_p, and thus the controller output u will be an estimate of $\bar{i}_m$, i.e. $u = \hat{\bar{i}}_m$.

The measurement noise is added to avoid the observer having very high gain at low frequencies. The norm bounds σ_i and σ_u can be seen as tuning parameters.

Discretisation

Since the induction motor model was obtained in continuous time and the observer must be implemented in discrete time, a choice must be made. Either the observer is designed in continuous time and then discretised, or the model is discretised first, and an observer is designed by the discrete time version of the synthesis described in Section 5.4.3.

Based on practical experience it is chosen to discretise the model and then design the observer in discrete time. The discretisation was performed by the bilinear transformation method described in Section 5.4.1. In practice the discretisation was only performed on

the part of the model specified by the matrices in (5.79). After this the rest of the matrices were chosen as above.

5.6.2 Observer synthesis

An observer for the example motor with the parameters in (3.57) and a sampling frequency of $3kHz$ was synthesised using Theorem 5.15. The noise bounds were chosen rather arbitrarily as $\sigma_i = 10^{-4}$ and $\sigma_u = 10^{-4}$.

As seen in Section 5.5, nothing will be gained from transforming the second order complex model into a fourth order real model and allowing for solution that are not complex-formed. By restricting the solution to be complex-formed we have the following advantages

- Fewer decision variables. Since the problem is fairly small, this is not so important here.
- Better numerics for the quadratic matrix inequality solution.
- A complex-formed scheduling function. This is a great advantage, since instead of inverting a 2×2 real matrix, we only have to perform a complex division.

Before solving the LMI, the model was balanced to better the numerics. The balancing was performed on the model including parameter dependence and performance channels.

The performance index P_p was chosen as the l_2-induced norm specification

$$P_p = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix},$$

and a bisectional search was performed to determine the lowest value of γ for which the LMIs were feasible. Figure 5.5 shows the achievable γ, denoted γ_a, as a function of $\omega_{r,max}$. Each observer computation was done in a few seconds using the MatLab LMI toolbox [Gahinet et al., 1995]. The small increase in γ_a for small speed ranges is due to the combined effect of the balancing and bounds on the Frobenius norm of the multipliers, i.e. a numerical effect.

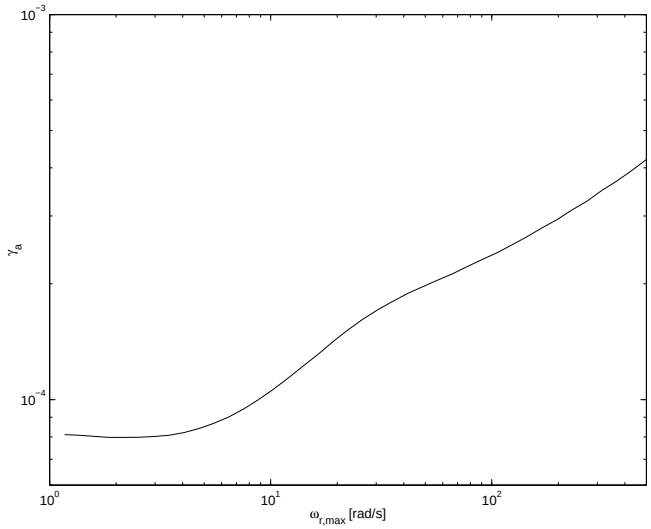


Figure 5.5: *The achievable l_2-induced norm, γ_a, as a function of range of rotational speeds, $\omega_{r,max}$.*

An observer with $\omega_{r,max} = 200\text{rad}/s$ was chosen for further experiments. The observer was computed with a suboptimal $\gamma = 1.1\gamma_a = 0.0003$ in order to improve the numerics.

The observer will be compared to a JL-observer, which has been hand-tuned to yield reasonable behaviour over the speed range. The constants are $K_1 = 32(1 + 0.1j)$ and $K_2 = 2(1 + 0.1j)$. In Figure 5.6 bode plots of the two observers are given for $\omega_r = 10rad/s$. For comparative purposes, a JL-observer with $K_1 = 10000$ and $K_2 = 0$ is also given. This is practically an open-loop simulation of the current model, which is probably the most commonly used observer in induction motor control due to its simplicity. We will refer to the observer designed above as the LPV observer.

The bode plots are shown for positive frequencies from $10^{-1}rad/s$ to $10^4rad/s$. Notice that since the observer is complex, there is no symmetry between positive and negative frequencies. However, the positive frequencies are the most important for a positive ω_r, so only these are shown. The left column shows the gain from stator current to magnetising current estimate. The right column shows the gain from stator voltage to magnetising current estimate.

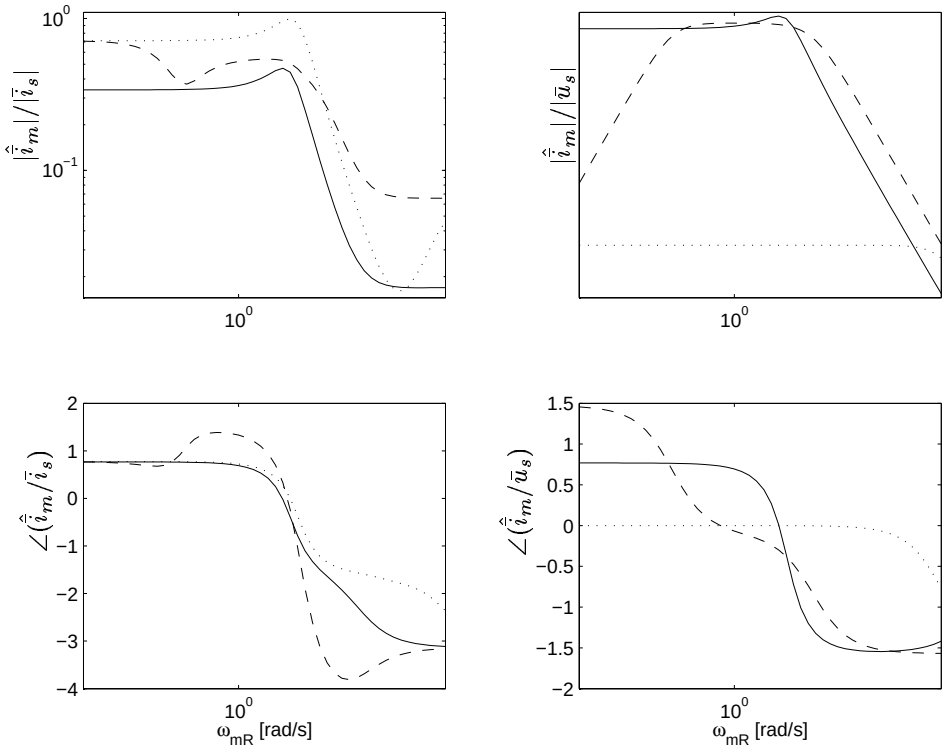


Figure 5.6: *Bode plots of three observers operating at $\omega_r = 10rad/s$, the LPV observer (solid), the JL-observer (dashed), and the current model (dotted).*

As seen, the gains are similar in amplitude for the LPV and the JL-observer in the frequency area around the rotational speed. A major difference is at low frequencies, where the JL-observer will always be based on the current model and thus only current measurements. The LPV observer in contrast uses a combination of both current and voltage measurements. It does, however, not have the very high gains that an observer based on the voltage model has at low frequencies.

5.6.3 Experiments

Experiments to confirm the performance of the LPV observer are performed on the laboratory setup. Since the flux cannot be measured, the only way to test the observers, is to examine the closed-loop behaviour of the rotor flux oriented control, when the observer is inserted. All the other parts of the control scheme are implemented as described in Section 4.2.

Figure 5.7 shows an experiment, where the speed reference is $\omega_{r,ref} = 10rad/s$. The

motor is disturbed by the load torque m_L, which switches between $0Nm$ and $3Nm$ every $1/3$ seconds. The load is plotted by the dashed lines. The plot on the left shows the closed-loop behaviour with the LPV flux observer. The plot on the right shows the same for the JL-observer.

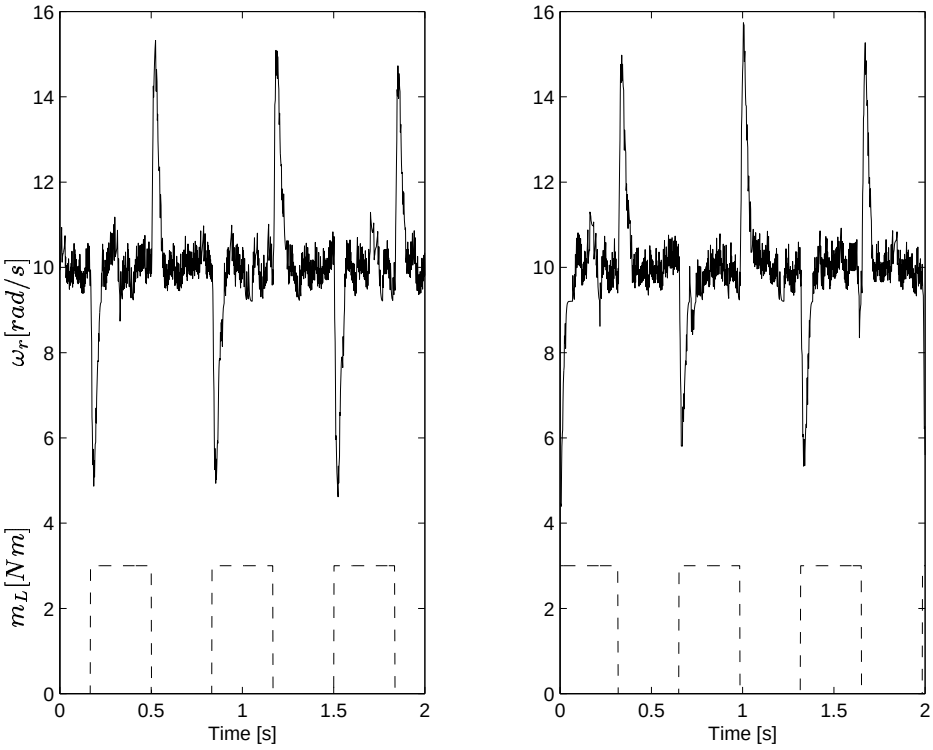


Figure 5.7: *Closed-loop operation at $\omega_{r,ref} = 10rad/s$. The left plot shows the behaviour with the LPV observer. The right plot shows the same for the JL-observer.*

Figure 5.8 shows a similar experiment, but now the speed reference is changed in steps of $15rad/s$ every $0.6s$ from $-150rad/s$ to $150rad/s$ and then back to $-150rad/s$. The load torque changes between $0Nm$ and $3\text{sign}(\omega_r)Nm$ every 0.1 seconds.

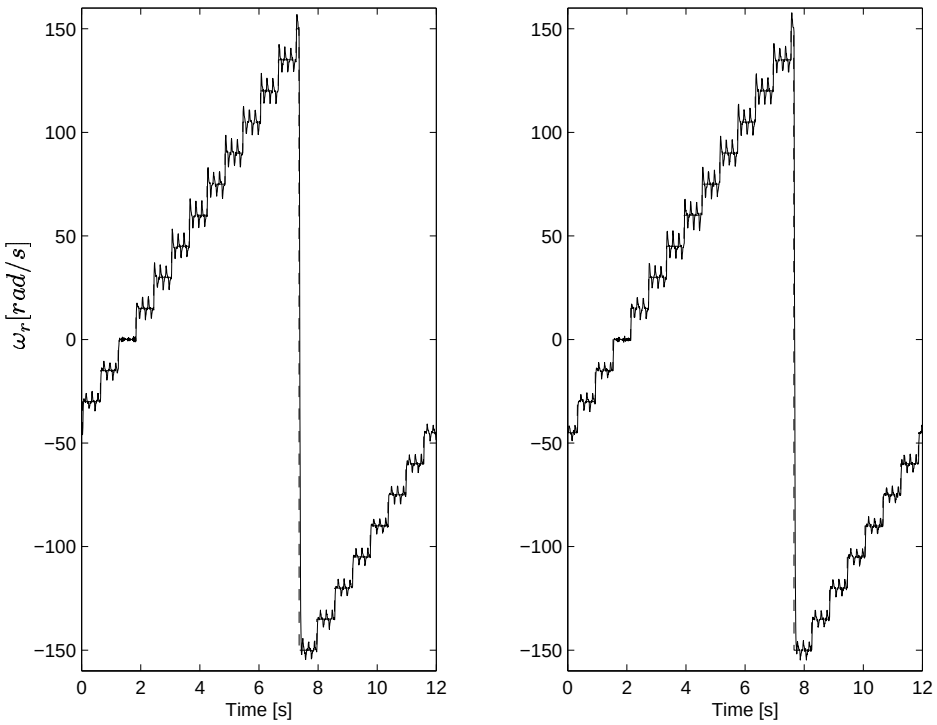


Figure 5.8: *Stepwise sweep through the speed range. The left plot shows the behaviour with the LPV observer. The right plot shows the same for the JL-observer. The dashed lines show the reference speed,* $\omega_{r,ref}$.

The main conclusion from these two tests is that the behaviour of the two observers is very similar.

Speed sensor-less control

Very often it is desirable to avoid the use of a speed or position sensor. The speed (or position) then has to be estimated by a speed observer such as Kubota's speed observer described in Section 4.4.2. The following tests are similar to those above, but the speed measurements were replaced by the speed estimates from Kubota's speed observer with a speed estimate update gain of $\lambda_c = 1000$ and the current observer gain G chosen so that the observer closed-loop eigenvalues (eigenvalues of $\hat{A}_{sc} + GC$) were equal to 1.1 times the eigenvalues of $\hat{A}_{sc}$. Since the speed observer reacts slower than the speed sensor, the speed controller was retuned to preserve stability.

In addition to the LPV observer and the JL-observer, tests are also performed using

the magnetising current estimate provided by Kubota's speed observer as suggested in [Kubota et al., 1993].

It should be noted that both the LPV observer and the JL-observer were designed under the assumption of exact knowledge of ω_r. There is therefore no theoretical guarantee that either will work.

Figures 5.9 and 5.10 show the same experiments as above but now using the observed speed instead of the measured. The left columns show the behaviour with the LPV observer. The middle and the right columns show the same for the JL-observer and the Kubota observer, respectively. The top rows show the estimated speed, whereas the bottom rows show the actual (measured) speed.

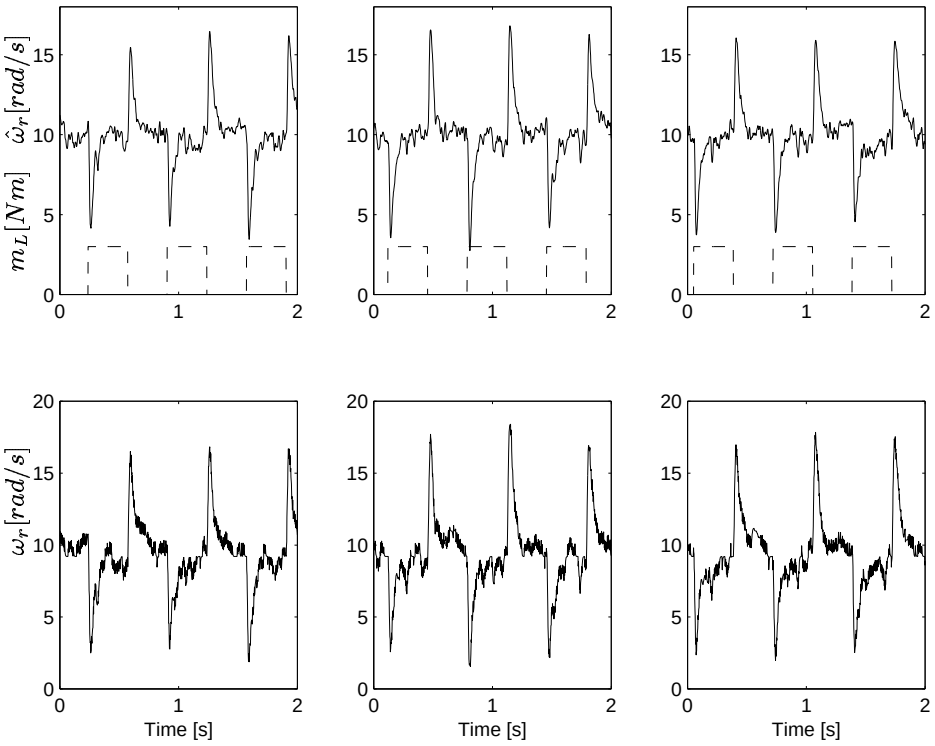


Figure 5.9: *Speed sensor-less operation at* $\omega_{r,ref} = 10rad/s$. *The left column shows the behaviour with the LPV observer. The middle and the right column show the same for the JL-observer and the Kubota observer, respectively. The top row shows the estimated speed. The bottom row shows the measured speed.*

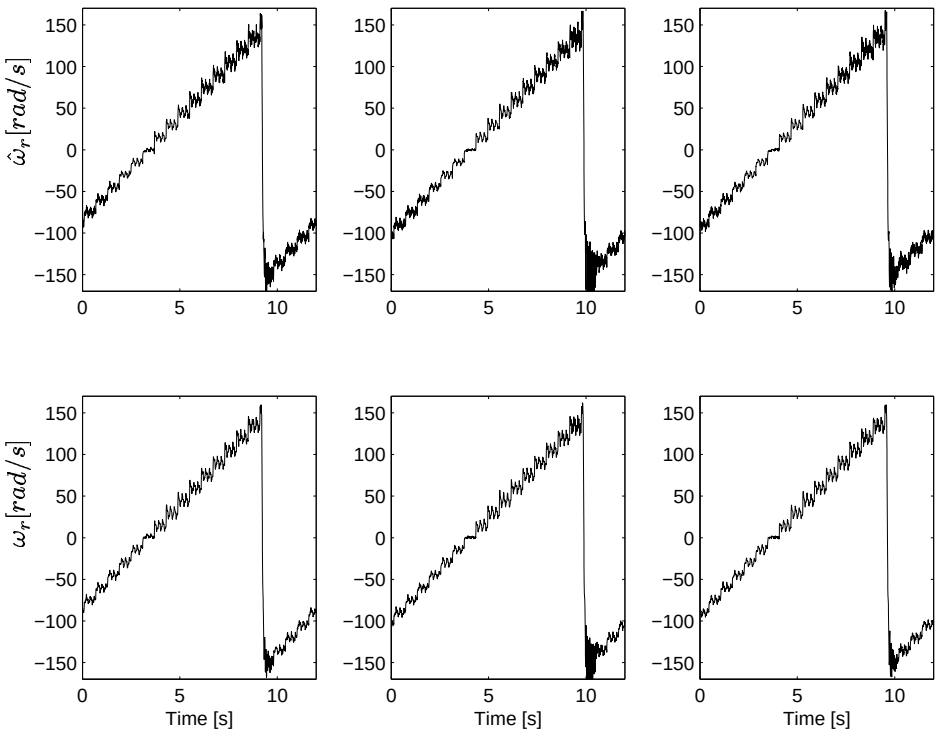


Figure 5.10: *Speed sensor-less stepwise sweep through the speed range. The left column shows the behaviour with the LPV observer. The middle and the right column show the same for the JL-observer and the Kubota observer, respectively. The top row shows the estimated speed. The bottom row shows the measured speed.*

The behaviour of the closed-loop system with the three different observers are quite similar. If anything, the LPV observer is slightly better, especially in the large speed reversal step, but not enough to claim that it is superior to the other observers.

A difficult task in speed sensor-less control is to operate around zero speed. Figure 5.11 shows the results of a test where the speed reference slowly sweeps from $-10rad/s$ to $10rad/s$ and then back again. On the upward slope the load torque is given by $m_L = 6\text{sign}(\omega_r)Nm$. On the downward slope the load torque is zero. The large spikes are caused by the sudden change in load.

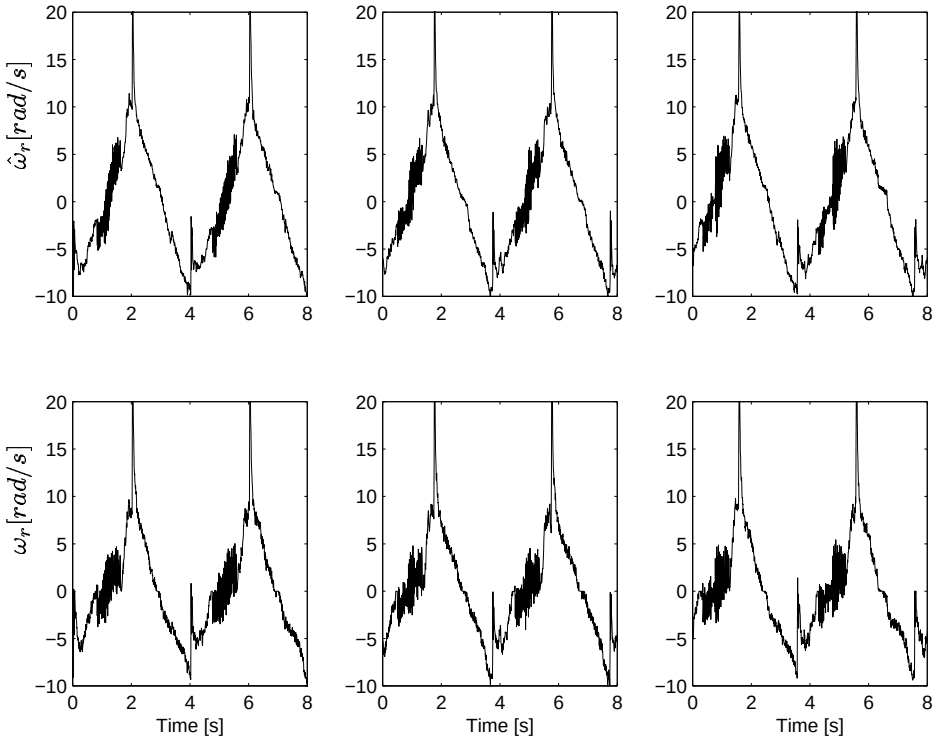


Figure 5.11: *Speed sensor-less slow sweep through zero speed. The left column shows the behaviour with the LPV observer. The middle and the right column show the same for the JL-observer and the Kubota observer, respectively. The top row shows the estimated speed. The bottom row shows the measured speed.*

As expected, there is some degradation of performance close to zero speed. The performance with the three different observers is very similar. Again the LPV observer might be slightly better than the others.

5.7 Summary

In this chapter a controller synthesis method for LPV systems was described and applied to the design of a rotor flux observer.

First, the controller synthesis method of [Scherer, 2001] was described. By using full block scalings the method provides the least conservative way of designing controllers yielding robust quadratic performance for an LPV system with rational parameter dependence. A new result on how to improve the numerics of a part of the synthesis method

was given.

Secondly, it was discussed how to change the theory to the discrete time domain. This can be done by a simple substitution of some of the matrix blocks in the LMIs.

Inspired by the special structure of the current part of the induction motor model, which allows it to be written either as a fourth order real model or as a second order complex model, some theoretical results on this type of structure were then given. The main result was that for an LPV system with this particular type of structure, the controller can be restricted to have the same structure without loss of achievable performance. This have several advantages, especially in implementation of the controller.

Finally the theory was applied to the design of a discrete time LPV flux observer. The observer was then tested on a laboratory setup. It was used as part of a speed control scheme, both with and without speed sensor. The resulting performance was compared to the performance when using the JL-observer or the flux estimates from Kubota's speed observer, both described in Chapter 4. Although the performance was not significantly better, it is worth noting that practically no tuning had to be performed for the LPV observer.

Chapter 6

QUASI-LPV CURRENT AND SPEED CONTROLLERS

The quasi-LPV approach allows the use of LPV theory for a very general class of non-linear systems. In Section 6.1 a discussion of the quasi-LPV structure is given. The approach is then applied to the design of a stator current controller in Section 6.2.

In Section 6.3 a novel method is presented for transforming a multi-layer perceptron state space model into a quasi-LPV model suitable for control design.

This method is then applied to the design of a speed controller in Section 6.4.

6.1 Quasi-LPV systems

The following is a short introduction to the concept of using quasi-LPV models as a basis for control of nonlinear system. For a more thorough discussion see the survey paper [Rugh and Shamma, 2000].

Before the 1990's, theoretical treatments of gain scheduling in nonlinear control systems was very rare [Rugh and Shamma, 2000]. A theoretical discussion of potential problems of traditional *linearisation scheduling* was given in [Shamma and Athans, 1990].

Linearisation scheduling, the traditional approach to gain scheduling as a means of designing controllers for nonlinear systems, is based on Jacobian linearisations of the system model in a finite number of equilibria. For each of these linearised models a controller is designed by linear design methods. As the system moves between these points, the control system then switches between these controllers. This provides a theoretical guarantee of *local* stability and performance around the equilibria, but there is no guarantee in between these points. Furthermore, the method also assumes that the system dynamics change is *slow*.

An early suggestions that a quasi-linear approach could overcome these problems was given in [Shamma and Athans, 1992]. The idea is to view some of the system state variables as both state variables *and time-varying parameters*.

Consider the nonlinear system

$$\begin{array}{rcl} \dot{x} & = & f(x,u,w,v), \quad x = \begin{bmatrix} x_p \\ x_l \end{bmatrix}, \\ z & = & p(x,u,w,v), \\ y & = & h(x,w,v), \end{array} \tag{6.1}$$

where x is the system state, u is the control input, y is the measurement vector, z is the performance output, w are measurable inputs, and v is noise. Assume that these nonlinear functions can be written as

$$\begin{array}{rcccc} f(x,u,w,v) & = & A(\sigma)x & +B(\sigma)u_p(\sigma,u) & +B_p(\sigma)v, \\ p(x,u,w,v) & = & C_p(\sigma)x & +E_p(\sigma)u_p(\sigma,u) & +D_{pp}(\sigma)v, \\ h(x,u,w,v) & = & C(\sigma)x & & +F_p(\sigma)v, \\ \sigma & = & s(x_p,w), & & \end{array} \tag{6.2}$$

where u_p is invertible with respect to u, i.e. there exists a function u_p^{-1} such that

$$u_p^{-1}(u_p(\sigma,u),\sigma) = u.$$

Notice that this representation is valid anywhere and not just in equilibria. We then have the following *quasi-LPV system*:

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A(\sigma) & B(\sigma) & B_p(\sigma) \\ C_p(\sigma) & D_{pp}(\sigma) & E_p(\sigma) \\ C(\sigma) & F_p(\sigma) & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ u_p(\sigma,u) \end{bmatrix}, \qquad \sigma = s(x_p,w). \tag{6.3}$$

Assume that s is known to belong to some bounded set S at all times. The nonlinear system (6.1) is then contained in the *relaxed quasi-LPV model*

$$\begin{bmatrix} \dot{x} \\ z \\ y \end{bmatrix} = \begin{bmatrix} A(\theta) & B(\theta) & B_p(\theta) \\ C_p(\theta) & D_{pp}(\theta) & E_p(\theta) \\ C(\theta) & F_p(\theta) & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ u_p(\theta, u) \end{bmatrix}, \qquad \theta \in S. \tag{6.4}$$

We can now apply LPV control design techniques, such as those discussed in Chapter 5, to this system. Once an LPV controller has been obtained, we simply implement the controller and input $u = u_p^{-1}(u_p(\sigma, u), \sigma)$ to the system. The advantage of this approach over the traditional linearisation scheduling is that stability and performance is conserved in all of S and not just locally at a finite number of equilibria. Furthermore, there is no restriction on how fast the parameters are allowed to vary. A few points must be made about this approach:

- Conservatism is introduced, since the model (6.4) allows for any parameter trajectory within S, whereas the possible trajectories of σ may be far more restricted.
- The representation in (6.2) is not unique. Some representations may yield better results than others.
- The state variables x_p and the inputs w must be known in real-time in order to allow gain scheduling control.
- A necessary condition for writing (6.1) as (6.2) is that (6.1) has an equilibrium at the origin (after a possible transformation of u). Otherwise a coordinate transformation must be performed first, see [Packard and Kantner, 1996].

Example 6.1 Consider the nonlinear system

$$\dot{x}_1 = \sin x_1 + x_2, \qquad \dot{x}_2 = (x_1 + 1)x_2 + u^3. \tag{6.5}$$

This system can be written as the quasi-LPV system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{\sin x_1}{x_1} & 1 \\ 0 & (x_1 + 1) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_p \tag{6.6}$$

or alternatively

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \frac{\sin x_1}{x_1} & 1 \\ x_2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_p, \tag{6.7}$$

where $u = u_p^{\frac{1}{3}}$. If only x_1 is measured, then only the representation (6.6) can be used for LPV control, since the system matrix in (6.7) contains x_2.

6.1.1 LFT representation

Obtaining a relaxed quasi-LPV model in LFT form follows the same lines as above, as illustrated by the following example.

Example 6.2 Consider again the system (6.5). This can be written on the LFT form by "pulling out the nonlinearities" (see [Zhou et al., 1996]):

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} w_u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_p,$$

$$w_u = \Omega(z_u), \qquad z_u = \begin{bmatrix} z_{u1} \\ z_{u2} \end{bmatrix} = x,$$

$$\Omega(z_u) = \begin{bmatrix} sin z_{u1} \\ z_{u1} z_{u2} \end{bmatrix}.$$

A possible quasi-LPV representation is then obtained by rewriting the nonlinearities as

$$w_u = \Delta(z_u) z_u, \qquad \Delta(z_u) = \begin{bmatrix} \frac{sin(z_{u1})}{z_{u1}} & 0 \\ 0 & z_{u1} \end{bmatrix}.$$

A relaxed quasi-LPV model is then obtained by viewing Δ as a time-varying gain rather than a nonlinearity. Notice that in order to apply LPV control design techniques, we again need to have access to x_1 in order to know this gain in real-time.

Definition 6.3 (Residual gains)

We will refer to the function Δ *when used as above as* residual gains.

6.2 Quasi-LPV stator current controller

In this section we will use the quasi-LPV approach to design a novel type of stator current controller. Recall the configuration of the control scheme as shown in Figure 4.1. The speed and magnetising current controller provides a stator current reference for the stator current controller, which controls the stator current by sending a reference for the stator voltage to the power device. A wide variety of schemes for stator current control exists, both in hardware and software. For surveys on current controllers see [Kaźmierkowski and Dzieniakowski, 1994] and [Kaźmierkowski and Malesani, 1998]. Here we will focus on the situation where the hardware is already given as a PWM voltage sourced inverter. We will furthermore assume that the switching frequency of the inverter is fixed and furthermore is so much higher than the sampling frequency of the control system that we can consider the inverter as being able to produce any complex stator voltage within a given limit of magnitude. Some current control schemes, for

instance [Liu et al., 1998], do not consider the rotor dynamics. It is expected that better performance can be achieved by considering these dynamics, which is also indicated by experiments in [Rasmussen, 1995].

Several continuous-time controllers have been presented, for instance using a simple Lyapunov approach [Shyu and Shieh, 1995], minimum-time control [Choi and Sul, 1998], sliding mode control [Shiau and Lin, 2001], as well as a special type of decoupling with special regard to robustness [Jung et al., 1997].

A general problem with these schemes is that it is unclear if they will work well when implemented in discrete time at a sampling frequency which is not considerably faster than the motor dynamics. Here we will design a novel type of controller based on a discrete-time model, thus incorporating the limitations in the sampling frequency. Other examples of discrete-time designs can be found in [Blaabjerg et al., 1996], where RST-controllers are designed, and in [Yang and Lee, 1999] where a simple decoupling is designed under the assumption that the change in speed and rotor flux from sample to sample is negligible.

Section 6.2.1 describes the quasi-LPV model used for the controller design. Section 6.2.2 describes the controller design. The sampling frequency is chosen as 600 Hz, both due to the computational complexity and in order to demonstrate that it is possible to achieve good results at a low sampling frequency. In Section 6.2.3 the closed-loop is simulated in order to verify the stability before implementation. Finally, in Section 6.2.4 the controller is tested on the laboratory setup. The results are satisfactory considering the low sampling frequency.

6.2.1 Quasi-LPV model

In Section 3.3.1 we found the following model of the induction motor:

$$\begin{aligned} \dot{x}_{sc} &= A_{sc}x_{sc} + B_{sc}\bar{u}_s, \\ x_{sc} &= \begin{bmatrix} \bar{i}_s \\ \bar{i}_m \end{bmatrix}, \\ A_{sc} &= \begin{bmatrix} \frac{L_m^2 R_r + R_s L_r^2}{L_r(L_m^2 - L_s L_r)} & \frac{L_m^2 (jL_r\omega_r - R_r)}{L_r(L_m^2 - L_s L_r)} \\ \frac{R_r}{L_r} & j\omega_r - \frac{R_r}{L_r} \end{bmatrix}, \\ B_{sc} &= \begin{bmatrix} \frac{L_r}{L_s L_r - L_m^2} \\ 0 \end{bmatrix}, \end{aligned} \tag{6.8}$$

$$\begin{aligned} m_e &= \tfrac{3 Z_p L_m^2}{2 L_r} \Im\{\bar{i}_s \bar{i}_m^*\}, \\ \dot{\omega}_r &= Z_p \dot{\omega}_{mech} = \tfrac{Z_p}{J}(m_e - m_L). \end{aligned} \tag{6.9}$$

$\bar{i}_s$ and $\bar{u}_s$ are the stator current and voltage, respectively, and $\bar{i}_m$ is the magnetising

current. ω_r is the rotational speed of the shaft. m_e is the torque produced by the induction motor. m_L is the load torque on the shaft acting as a disturbance. L_m, L_s, L_r, R_r, R_s, Z_p, and J are real parameters.

With the definitions of the referred parameters in Section 4.1 and with a transformation to a rotating coordinate system as discussed in Section 3.3.3 we can write the subsystem (6.8) as

$$\frac{d\bar{i}_{s,cc}}{dt} = -\left(\frac{R_s + R'_r}{L'_s} + j\omega_{cc}\right)\bar{i}_{s,cc} + \left(\frac{R'_r}{L'_s} - j\frac{L'_m}{L'_s}\omega_r\right)\bar{i}_{m,cc} + \frac{\bar{u}_{s,cc}}{L'_s}, \tag{6.10}$$

$$\frac{d\bar{i}_{m,cc}}{dt} = \frac{R'_r}{L'_m}\bar{i}_{s,cc} - \left(\frac{R'_r}{L'_m} + j(\omega_{cc} - \omega_r)\right)\bar{i}_{m,cc}, \tag{6.11}$$

where all signals are given in a reference frame with the angle ρ_{cc}, i.e.

$$\begin{array}{lllllll} w_{cc} & \triangleq & \dot{\rho}_{cc}, & & \bar{i}_{s,cc} & \triangleq & \bar{i}_s e^{-j\rho_{cc}}, \\ \bar{i}_{m,cc} & \triangleq & \bar{i}_m e^{-j\rho_{cc}}, & & \bar{u}_{s,cc} & \triangleq & \bar{u}_s e^{-j\rho_{cc}}. \end{array}$$

We wish to design a controller for the complex stator current $\bar{i}_{s,cc}$ using the complex stator voltage $\bar{u}_{s,cc}$ as the control input.

The reference frame is chosen as the same as the one used by the outer control loop, since the reference signal for the stator current will be constant in steady state in this frame. We assume that the flux observer is the simple observer (4.15) based on the current model, and that the reference frame is the angle of the rotor flux estimate as discussed in Chapter 4. In estimated rotor flux coordinates this observer is

$$\dot{\hat{i}}_{mR} = \frac{1}{T_r}(i_{sd} - \hat{i}_{mR}), \tag{6.12}$$

$$\omega_{cc} = \omega_r + \frac{i_{sq}}{T_r \hat{i}_{mR}}, \tag{6.13}$$

with $i_{sd} \triangleq \Re\{\bar{i}_{s,cc}\}$ and $i_{sq} \triangleq \Im\{\bar{i}_{s,cc}\}$.

(6.10)-(6.11) can be viewed as an LPV system with ω_r and ω_{cc} as time-varying parameters, but we wish to exploit the knowledge that the slip frequency $\omega_{slip} = \omega_{cc} - \omega_r$ is usually small compared to the maximal value of ω_r. This can of course be exploited by restricting the parameters ω_r and ω_{cc} to the polygon

$$\begin{bmatrix} \omega_r \\ \omega_{cc} \end{bmatrix} \in \left\{ \begin{bmatrix} \omega_r \\ \omega_{cc} \end{bmatrix} : |\omega_r| \leq \omega_{r,max}, |\omega_{cc} - \omega_r| \leq \omega_{slip,max} \right\}. \tag{6.14}$$

We will instead exploit this knowledge by inserting (6.13) in (6.10)-(6.11) resulting in

$$\frac{d\bar{i}_{s,cc}}{dt} = -\left(\frac{R_s + R'_r}{L'_s} + j\omega_{cc}\right) i_{s,cc} + \left(\frac{R'_r}{L'_s} - j\frac{L'_m}{L'_s}\left(\omega_{cc} - \frac{i_{sq}}{T_r \hat{i}_{mR}}\right)\right)\bar{i}_{m,cc} + \frac{\bar{u}_{s,cc}}{L'_s},$$

$$\frac{d\bar{i}_{m,cc}}{dt} = \frac{R'_r}{L'_m}\bar{i}_{s,cc} - \left(\frac{R'_r}{L'_m} + j\frac{i_{sq}}{T_r\hat{i}_{mR}}\right)\bar{i}_{m,cc}.$$

The subsystem (6.8) can now be written as

$$\dot{x}_{cc} = (A_0 + \delta_1 A_1 + \delta_2 A_2)\, x_{cc} + B_{sc}u_{s,cc},\ x_{cc} \triangleq \begin{bmatrix}\bar{i}_{s,cc}\\ \bar{i}_{m,cc}\end{bmatrix}, \tag{6.15}$$

in which

$$A_0 = \begin{bmatrix} -\frac{R_s+R'_r}{L'_s} & \frac{R'_r}{L'_s} \\ \frac{R'_r}{L'_m} & -\frac{R'_r}{L'_m}\end{bmatrix} \text{ and } B_{sc} = \begin{bmatrix}\frac{1}{L'_s}\\ 0\end{bmatrix}$$

represent the nominal model, which is a linear time invariant system, and

$$A_1 = \begin{bmatrix} -j & -j\frac{L'_m}{L'_s} \\ 0 & 0\end{bmatrix} \text{ and } A_2 = \begin{bmatrix} 0 & j\frac{L'_m}{T_r L'_s} \\ 0 & -j\frac{1}{T_r}\end{bmatrix}$$

represent the nonlinearities entering through the residual gains

$$\delta_1(t) \triangleq \omega_{cc}(t) \text{ and } \delta_2(t) \triangleq \frac{i_{sq}(t)}{\hat{i}_{md}(t)}.$$

To put the system on an LFT form we write (6.15) as

$$\begin{array}{rcl} \dot{x}_{cc} & = & A_0 x_{cc} + B_{u0}w_u + B_{sc}\bar{u}_{s,cc}, \\ w_u = \Delta z_u,\ z_u & = & C_{u0}x_{cc}. \end{array} \tag{6.16}$$

First we observe that A_1 and A_2 are both of rank 1. Consequently we can parametrise these as

$$A_1 = U_1\Sigma_1 V_1^* = \begin{bmatrix} u_{1,1} & u_{1,2}\end{bmatrix}\begin{bmatrix}\sigma_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix} v_{1,1}^* \\ v_{1,2}^*\end{bmatrix}$$

and

$$A_2 = U_2\Sigma_2 V_2^* = \begin{bmatrix} u_{2,1} & u_{2,2}\end{bmatrix}\begin{bmatrix}\sigma_2 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix} v_{2,1}^* \\ v_{2,2}^*\end{bmatrix}$$

and let

$$B_{u0} = \begin{bmatrix} u_{1,1}\sigma_1 & u_{2,1}\sigma_2\end{bmatrix} \text{ and } C_{u0} = \begin{bmatrix} v_{1,1}^* \\ v_{2,1}^*\end{bmatrix}.$$

The parameter variation can then be written as $\delta_1 A_1 x_{cc} + \delta_2 A_2 x_{cc} = B_{u0}\left[\begin{smallmatrix}\delta_1 & 0\\ 0 & \delta_2\end{smallmatrix}\right] C_{u0}x_{cc}$. The parameter variation channel $z_u \to w_u$ is hence defined as

$$w_u(t) = \Delta(t)z_u(t) = \begin{bmatrix}\omega_{cc}(t) & 0\\ 0 & \frac{i_{sq}(t)}{\hat{i}_{md}(t)}\end{bmatrix} z_u(t). \tag{6.17}$$

The main advantage of this representation over the one obtained directly from the polygon (6.14) is the simplicity of Δ. With the methods in most of the references in Section 5.1, using (6.14) directly, we would have to use the parameter variation

$$\Delta = \begin{bmatrix} \omega_r & 0 \\ 0 & \omega_{cc} I_2 \end{bmatrix},$$

since the dependency on ω_{cc} has rank 2. With the full block S-procedure in [Scherer, 2001] (described in Sections 5.2-5.3) we can simply use

$$\Delta = \begin{bmatrix} \omega_{cc} & \frac{L'_m}{L'_s}\omega_r \\ 0 & (\omega_{cc} - \omega_r) \end{bmatrix}$$

with $z_u = -jx_{cc}$. The representation in (6.17) is slightly simpler. On the other hand we do introduce a singularity for small $\hat{i}_{md}$. In normal operation this is not a problem, but during startup the performance may be degraded.

6.2.2 Controller synthesis

A stator current controller was designed for the example motor with the parameters in (3.57). The first step of the controller synthesis was to augment the LPV system defined by (6.16) and (6.17) with performance and control channels to get the system

$$\left[\begin{array}{c} \dot{x}_{cc} \\ \hline z_u \\ z_p \\ y \end{array}\right] = \left[\begin{array}{c|ccc} A_0 & B_{u0} & 0 & B_{sc} \\ \hline C_{u0} & 0 & 0 & 0 \\ C_p & 0 & D_{pp} & E_p \\ C & 0 & F_p & 0 \end{array}\right] \left[\begin{array}{c} x_{cc} \\ \hline w_u \\ w_p \\ \bar{u}_{s,cc} \end{array}\right].$$

The noise signal $w_p = \begin{bmatrix} \bar{i}_{s,cc,ref} \\ n_m \end{bmatrix}$ consists of the reference signal for the stator current and measurement noise. The performance output $z_p = \begin{bmatrix} \bar{i}_{s,cc} - \bar{i}_{s,cc,ref} \\ \sigma_u \bar{u}_{s,cc} \end{bmatrix}$ consists of the stator current error and the stator voltage scaled by the weight σ_u in order to punish large control signals. The measurement $y = \bar{i}_{s,cc} - \bar{i}_{s,cc,ref} + \sigma_n n_m$ is the stator current error corrupted by the measurement noise scaled by the weight σ_n. This was achieved with the matrices $C_p = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $D_{pp} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}$, $E_p = \begin{bmatrix} 0 \\ \sigma_u \end{bmatrix}$, $C = \begin{bmatrix} 1 & 0 \end{bmatrix}$, and $F_p = \begin{bmatrix} -1 & \sigma_n \end{bmatrix}$. The first element in the performance output, the stator current error, was then augmented by a first-order low-pass filter with a pole in $s = p_f$ in order to put more emphasis on the low frequency error. The constants p_f, σ_u, and σ_w can be considered as tuning parameters. p_f is used to obtain a low steady state error. σ_u is mainly included to compensate for the fact that saturation of the stator voltage is not included in the model.

The LPV system was then discretised by the bilinear transformation described in Section 5.4 under the assumption that Δ is approximately constant from sample to sample. The sampling frequency was chosen as 600 Hz. The main reason for this choice is the computational complexity of the LPV controller.

A discrete time stator current controller was synthesised by the LPV method described in Section 5.4.3 with the parameters $p_f = -100$, $\sigma_u = 10^{-6}$, and $\sigma_n = 10^{-8}$. The time-varying parameters were allowed to vary in the intervals $\delta_1 = \omega_{cc} \in [-800\ ;\ 800]$ and $\delta_2 = \frac{i_{sq}}{\hat{i}_{md}} \in [-10\ ;\ 10]$. The performance index P_p was chosen as the l_2-induced norm specification

$$P_p = \begin{bmatrix} -\gamma^2 I & 0 \\ 0 & I \end{bmatrix},$$

and a bisectional search was performed to determine the lowest value of γ for which the LMIs were feasible. With the above parameters a $\gamma = 0.0011$ was achieved. The left hand side of (5.60) became close to singular making it possible to reduce the controller order to 2.

6.2.3 Simulation results

Before implementation, the closed-loop behaviour was simulated in order to verify the stability and performance. In the simulations the reference sequence was chosen as a series of steps of a duration of 250 samples. For each step, the reference for i_{sq} was allowed to take random values in the interval $[-10\ ;\ 10]$, while the reference for i_{sd} was chosen from the interval $[1\ ;\ 3]$. The system was disturbed by a load torque m_L, which was a sequence of uniformly distributed white noise filtered through a first-order filter with a time constant of $1/2$ second. Subject to these external signals, the nonlinear model generated the δ_1 and δ_2 sequences based on which the controller scheduling function was calculated. Motivated by limitations of the hardware of the experimental setup, a saturation on the control voltage $\bar{u}_{s,cc}$ at 600 V was imposed.

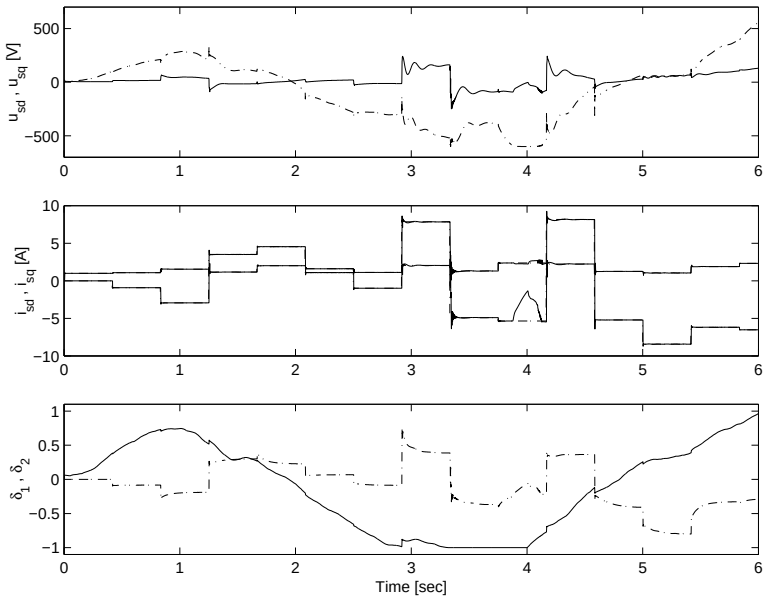


Figure 6.1: *LPV current control, simulation. The top figure shows the real and imaginary components of the control voltage generated by the controller. The middle figures show the real and imaginary components, i_{sd} and i_{sq}, of the controlled currents, plotted with full lines (—) along with their reference signals, plotted with dash-dotted lines ($-\cdot-$). The bottom plot shows δ_1 (—) and δ_2 ($-\cdot-$) scaled to the interval $[-1\ ;\ 1]$. As can be seen, the tracking of the current reference satisfies the performance requirement except when the control voltage saturates (at around 4 sec).*

Figure 6.1 shows a simulation of the closed loop system. It is seen that the control loop achieves good tracking, in accordance with the performance value achieved for all values of the parameter variations, except when the control signal saturates. The parameter variations are shown in the bottom plot in Figure 6.1, scaled to the interval $[-1\ ;\ 1]$. It is noted that the generated stator voltage compensates for the parameter variations throughout the allowed range.

6.2.4 Experimental results

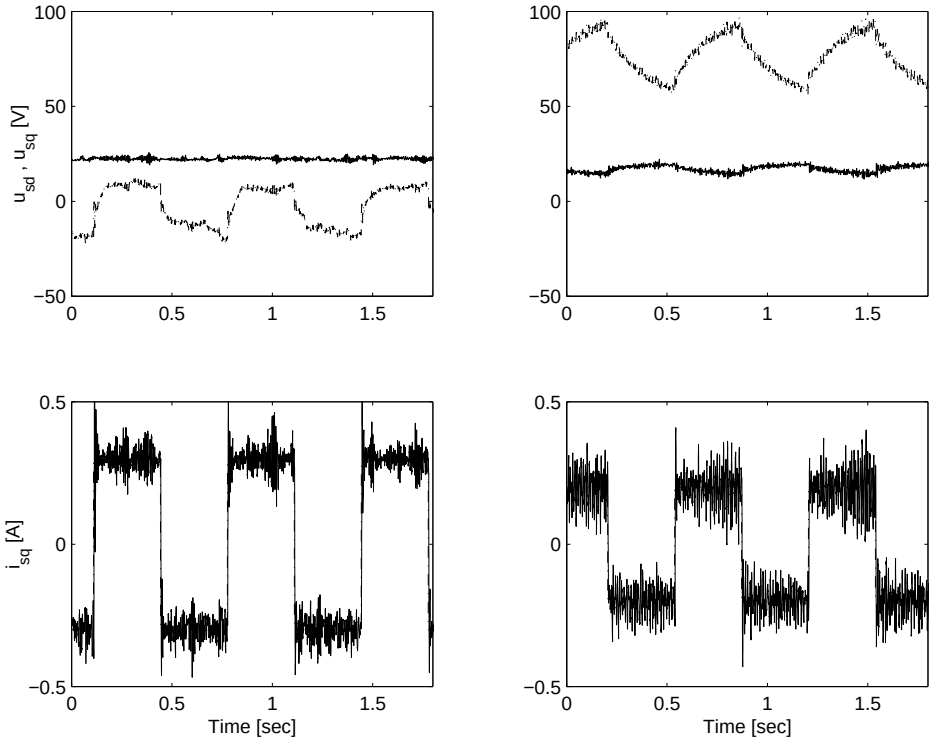


Figure 6.2: *LPV current control, experimental results. The top figures show the real and imaginary components of the control voltage generated by the implemented LPV controller. The lower figures show the controlled current i_{sq}. The reference signals are shown with dash-dotted lines ($-\cdot-$), while the measurements are shown with full lines (—). The left figures are with no load, while the right figures are recorded with a load torque of $m_L = 4Nm$.*

The controller was implemented on the laboratory setup described in Appendix A using the algorithm discussed in Section 5.4.4.

In the first two experiments, the stator current reference was generated in open-loop, i.e. there was no speed controller. The aim was to keep the magnetising current constant and make the imaginary part of the stator current follow a series of steps. The first experiment was conducted without load, while in the second experiment the motor shaft was subjected to a load torque of $4Nm$. The results are shown in Figure 6.2, where it is observed that the current tracks the reference steps adequately well. Looking at the stator voltage, it is noted that the imaginary part of the voltage is significantly different between the two experiments. This is due to the two different disturbance load torques, which

cause the scheduling controller to yield significantly different control signals. Some variation can be noted in the real part of the voltage as well, caused by the cross couplings.

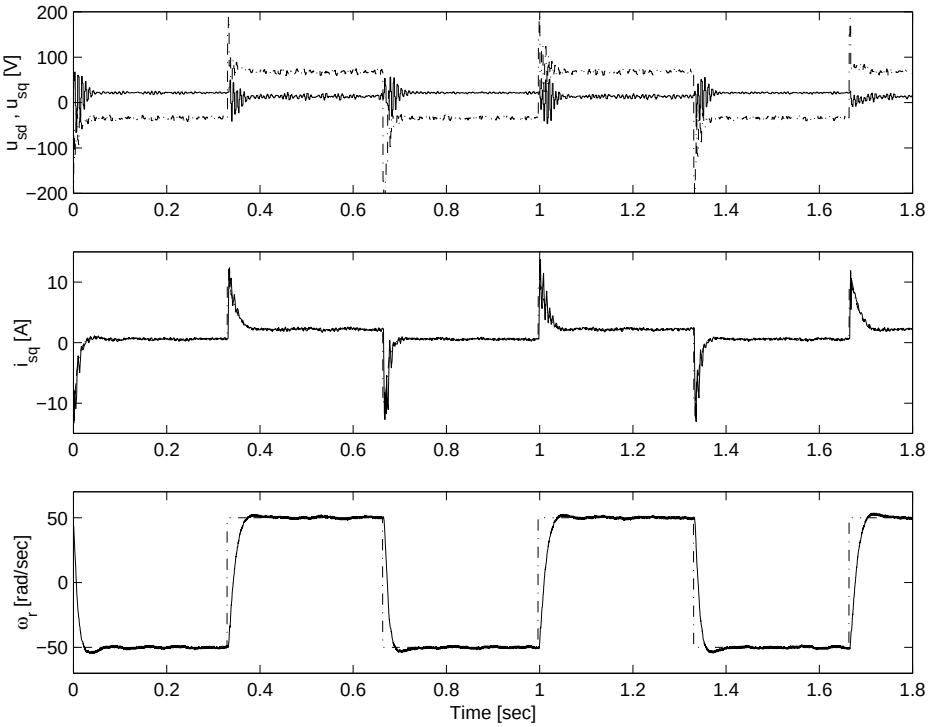


Figure 6.3: *LPV current control in cascade with rotational speed controller, experimental results. The top figure shows the real and imaginary components of the control voltage generated by the implemented LPV controller. The middle and lower figures show the controlled current i_{sq} and the rotational speed, respectively. The reference signals are shown with dash-dotted lines ($-\cdot-$), while the measurements are shown with full lines (—). The current reference signal was generated by an outer loop speed controller.*

In the third experiment the speed loop was closed using an outer PI-controller. In this case the stator current reference signals were thus generated by the PI-controller, and the LPV controller had to track these signals. The results of this experiment is shown in Figure 6.3. As can be seen, the control loop performs satisfactorily.

6.3 Quasi-LPV control based on neural network modelling

So far we have only dealt with control based on grey-box identification of physical models. In some cases it may however be desirable to work with nonlinear black-box models such as the multi-layer perceptron (MLP) discussed in Section 2.5, especially if a physical model is hard to obtain. This may for instance be the case if the load torque is some unknown nonlinear function of the rotational speed. In that case a neural network such as an MLP could be used to obtain a nonlinear model of the system.

The problem is now how to design a controller for a system modelled in this way. The classical approach has been to linearise the system model in some set of operating points and design one or more linear controllers for the system in said points. As discussed in Section 6.1 this approach has several hazards.

Other applications of neural network models in control theory are for instance for feedback linearisation [Chen and Khalil, 1995, He et al., 1998, Levin and Narendra, 1993] and sliding mode control laws [Mears and Polycarpou, 1999]. They have also been proven useful as observers [Kim et al., 1997], in direct adaptive control [French and Rogers, 1998, Su and Annaswamy, 1998], and in other roles. However, not much work has been done on achieving gain scheduling control based on artificial neural networks. In [Lee et al., 1996] a previously tuned gain scheduling controller was approximated by a neural network which then replaced the gain scheduling controller in the loop. Other approaches (e.g. [Chai et al., 1996]) use a neural network to schedule between a finite set of previously designed classical controllers, and have been somewhat ad hoc.

[Suykens et al., 1999] (with the corrections in [Bendtsen and Trangbæk, 2001a]) presents an analysis method of stability and performance of a closed-loop interconnection of two MLPs. In other words, the suggestion is to let the system modelled by an MLP be controlled by a controller also containing an MLP. The method is essentially based on diagonal multipliers. The problem with this analysis method is that it is unclear how to extend it to synthesis, in particular how to choose the MLP part of the controller. The problem is that any coupling between the nonlinearities in the system and in the controller are not exploited.

With the emergence of LPV control theory based on LMIs, as discussed in Chapter 5, a door has been opened for an efficient approach to gain scheduling control based on neural state space models. Such an extension of controller synthesis ideas from linear theory to the nonlinear framework of neural networks is a fundamentally sound idea, of course, but requires a method for reformulating the neural network model as an LPV model suitable for controller synthesis. More specifically, we would like to transform an MLP model into a quasi-LPV model on the LFT form as discussed in Section 6.1.1.

Some work along these lines has already been presented in [Suykens et al., 1995a] and [Suykens et al., 1995b] with robust $\mathcal{H}_\infty$ control in mind. The idea was to split the MLP

model into a linear part and a nonlinear part and then design a robust $\mathcal{H}_\infty$ controller for the linear system treating the nonlinear part as an uncertainty.

However, the fact that the nonlinearities are actually known at the design stage means that the controller can be designed by LPV methods taking advantage of this information as well, achieving a nonlinear and less conservative controller. In addition to this idea the method described in this section extends the results in [Suykens et al., 1995a, Suykens et al., 1995b] by achieving less conservative bounds on the nonlinear part. This improvement was presented in [Bendtsen and Trangbæk, 2000b].

6.3.1 From neural state space model to an LFT framework

We consider a system of the form

$$\dot{\tilde{x}} = f(\tilde{x}, \tilde{u}), \quad \tilde{y} = C\tilde{x} \tag{6.18}$$

where $\tilde{x} \in \mathbb{R}^n$ is the state vector, $\tilde{u} \in \mathbb{R}^m$ is a control signal and $\tilde{y} \in \mathbb{R}^p$ is the output vector for the system. $f(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is an unknown continuous function of the states and inputs describing the system dynamics.

As discussed in Section 2.5 we can approximate this function to a desired accuracy with a single hidden layer MLP with l neurons (assuming l is chosen large enough):

$$f(\tilde{x}, \tilde{u}) = \Theta_o \sigma\left(\Theta_x \tilde{x} + \Theta_u \tilde{u} + \tilde{\Theta}_b\right) + \varepsilon_x$$

where $\Theta_o \in \mathbb{R}^{n\times l}$ and $\Theta_x \in \mathbb{R}^{l\times n}, \Theta_u \in \mathbb{R}^{l\times m}$ contain the output and hidden layer weights, respectively. $\sigma(\cdot) : \mathbb{R}^l \to \mathbb{R}^l$ is a continuous, diagonal, static nonlinearity. $\tilde{\Theta}_b \in \mathbb{R}^l$ contains a set of biases which will allow us to model non-odd functions with odd neuron functions $\sigma(\cdot)$ such as the hyperbolic tangent. We assume it is possible to achieve a smaller modelling error than the measurement noise by choosing the MLP large enough and train it long enough on a sufficiently rich training set.

Consider a system for which a neural state space model has been trained according to the guidelines given above, until ε_x is small enough to be ignored:

$$\dot{\tilde{x}} = \Theta_o \sigma\left(\Theta_x \tilde{x} + \Theta_u \tilde{u} + \tilde{\Theta}_b\right), \quad \tilde{y} = C\tilde{x}. \tag{6.19}$$

We wish to rewrite the neural model (6.19) as the linear fractional transformation

$$\begin{aligned} \dot{x} &= Ax + Bu + B_u\Omega(\xi) \\ \xi &= \Theta_x x + \Theta_u u \\ y &= Cx \end{aligned} \tag{6.20}$$

where the *residual function* $\Omega(\cdot) : \mathbb{R}^l \to \mathbb{R}^l$ is a static diagonal nonlinearity, and where the coordinates (x, u) only differ from $(\tilde{x}, \tilde{u})$ by the possible subtraction of an equilibrium point. The presented method was first discussed in [Bendtsen and Trangbæk, 2000b].

We assume that there exists an equilibrium, $(\tilde{x}, \tilde{u}) = (\tilde{x}^\circ, \tilde{u}^\circ)$, i.e.

$$0 = \Theta_o \sigma(\Theta_x \tilde{x}^\circ + \Theta_u \tilde{u}^\circ + \tilde{\Theta}_b).$$

We can then change the network coordinates in such a way that instead of the arbitrary equilibrium point $(\tilde{x}^\circ, \tilde{u}^\circ)$ we have $0 = \Theta_o \sigma'(0)$ (σ' is a new neuron function mapping which will be defined shortly). Let the new coordinates be given as $x = \tilde{x} - \tilde{x}^\circ, u = \tilde{u} - \tilde{u}^\circ$. Then (6.19) can be written as

$$\dot{x} = \Theta_o \sigma \left(\Theta_x (x + \tilde{x}^\circ) + \Theta_u (u + \tilde{u}^\circ) + \tilde{\Theta}_b \right).$$

Here we will define a new bias vector $\Theta_b = \Theta_x \tilde{x}^\circ + \Theta_u \tilde{u}^\circ + \tilde{\Theta}_b$ and the new neuron function $\sigma'(\xi)$, where ξ is defined as in (6.20):

$$\begin{aligned} \sigma'(\xi) &= \sigma \left(\xi + \Theta_x \tilde{x}^\circ + \Theta_u \tilde{u}^\circ + \tilde{\Theta}_b \right) - \sigma(\Theta_b) \\ &= \sigma \left(\Theta_x (x + \tilde{x}^\circ) + \Theta_u (u + \tilde{u}^\circ) + \tilde{\Theta}_b \right) - \sigma(\Theta_b). \end{aligned}$$

Adding and subtracting $\Theta_o \sigma(\Theta_b)$ in (6.19) then gives

$$\begin{aligned} \dot{x} &= \Theta_o \sigma \left(\Theta_x \tilde{x} + \Theta_u \tilde{u} + \tilde{\Theta}_b \right) + \Theta_o \sigma(\Theta_b) - \Theta_o \sigma(\Theta_b) \\ &= \Theta_o \left(\sigma \left(\Theta_x \tilde{x} + \Theta_u \tilde{u} + \tilde{\Theta}_b \right) - \sigma(\Theta_b) \right) + \Theta_o \sigma(\Theta_b) \\ &= \Theta_o \sigma'(\Theta_x x + \Theta_u u) = \Theta_o \sigma'(\xi). \end{aligned}$$

$\Theta_o \sigma(\Theta_b) = 0$, because this is in fact the equilibrium point.

Remark 6.4 Note that, apart from providing a way to shift the operating point to the origin, the main purpose of the steps given above is to remove the bias from σ instead of having to consider it as a constant disturbance input, as suggested in [Suykens et al., 1995b].

Remark 6.5 It should furthermore be noted that the method given above applies equally well to sampled-data systems $\tilde{x}_{k+1} = f(\tilde{x}_k, \tilde{u}_k)$. In this case the MLP equilibrium point is of the form $\tilde{x}^\circ_{k+1} = f(\tilde{x}^\circ_k, \tilde{u}^\circ_k),\ \forall k$, but the definition of $\sigma'(\cdot)$ turns out to be the same.

Now we can find the effective range of the input arguments to the neuron functions. This is simply done by calculating

$$\begin{aligned} \xi_{j,max} &= \sup_{0 \geq t \geq T} \left\{ \Theta_x^j x(t) + \Theta_u^j u(t) \right\}, \\ \xi_{j,min} &= \inf_{0 \geq t \geq T} \left\{ \Theta_x^j x(t) + \Theta_u^j u(t) \right\} \end{aligned}$$

for $1 \leq j \leq l$, where $t \in [0; T]$ is the time interval in which the training data have been acquired and Θ_x^j, Θ_u^j denote the j'th rows in the hidden layer weight matrices. Then we have the following bounds on the active input range of the j'th neuron:

$$\xi_j = \Theta_x^j x + \Theta_u^j u \in [\xi_{j,min}; \xi_{j,max}].$$

Hence the neuron function response to the active input range belongs to the sector $\sigma_j' \in [k_{j,min}\ ,\ k_{j,max}]$ where

$$k_{j,min} = \inf_{\xi_j \in [\xi_{j,min};\xi_{j,max}]\setminus\{0\}} \left\{ \frac{\sigma_j'(\xi_j)}{\xi_j} \right\} \tag{6.21}$$

and

$$k_{j,max} = \sup_{\xi_j \in [\xi_{j,min};\xi_{j,max}]\setminus\{0\}} \left\{ \frac{\sigma_j'(\xi_j)}{\xi_j} \right\}. \tag{6.22}$$

In other words, the *sector bounds* are determined such that

$$k_{j,min}\xi_j^2 \leq \sigma'(\xi_j)\xi_j \leq k_{j,max}\xi_j^2. \tag{6.23}$$

The actual expressions for these sector bounds must be found for each neuron function individually and will in general depend on the bias, but the bounds obviously exist and are the least conservative easily achievable bounds. A procedure for finding these for $\tanh(\cdot)$ neuron functions is given below in Section 6.3.2.

Once the sector bounds are found, we return to vector notation and define the nonlinear function $\omega(\cdot):\ \mathbb{R}^{n+m} \to \mathbb{R}^n$ as

$$\omega(\xi) = \sigma'(\xi) - \frac{1}{2}\left(K_{min} + K_{max}\right)\xi \tag{6.24}$$

where $K_{min} = \text{diag}\{k_{j,min} - \epsilon\}$ and $K_{max} = \text{diag}\{k_{j,max} + \epsilon\}, 1 \leq j \leq l$. ϵ is a small positive quantity included to make the sector bounds strict. It is observed that $\omega(\cdot)$ belongs to the sector

$$(-\frac{1}{2}(K_{max} - K_{min})\ ,\ \frac{1}{2}(K_{max} - K_{min})).$$

We can now write the equation for $\dot{x}$ as

$$\begin{aligned}
\dot{x} &= \Theta_o\sigma'(\Theta_x x + \Theta_u u) \\
&= \Theta_o\left(\omega(\xi) + \frac{1}{2}\left(K_{min} + K_{max}\right)\xi\right) \\
&= Ax + Bu + B_u\Omega(\xi),
\end{aligned}$$

in which A, B, B_u and Ω are given by

$$A = \frac{1}{2}\Theta_o\left(K_{min} + K_{max}\right)\Theta_x \tag{6.25}$$

$$B = \frac{1}{2}\Theta_o\left(K_{min} + K_{max}\right)\Theta_u \tag{6.26}$$

$$B_u = \frac{1}{2}\Theta_o\left(K_{max} - K_{min}\right) \tag{6.27}$$

$$\Omega(\xi) = 2\left(K_{max} - K_{min}\right)^{-1}\omega(\xi). \tag{6.28}$$

Note that the diagonal scaling by $\frac{1}{2}(K_{max} - K_{min})$ is included in order to make the diagonal static nonlinearity Ω belong to the sector $(-1\ ,\ 1)$.

Remark 6.6 When designing LPV or quasi-LPV controllers, we are interested in the tightest possible bounds $K_{max} - K_{min}$ in order to avoid conservatism. Although the LPV synthesis method described in Section 5.3 is essentially non-conservative, it is usually necessary to use simplified multipliers, for instance by disregarding knowledge on the rate of change of the gains of the residual function, to make the synthesis implementable and to avoid controller switching. A quasi-LPV representation potentially introduces further conservatism due to non-uniqueness of the nonlinear function representation. For the sake of the controller synthesis we are therefore interested in keeping these gains from varying too much.

6.3.2 Sector bounds for tangent hyperbolic neuron functions

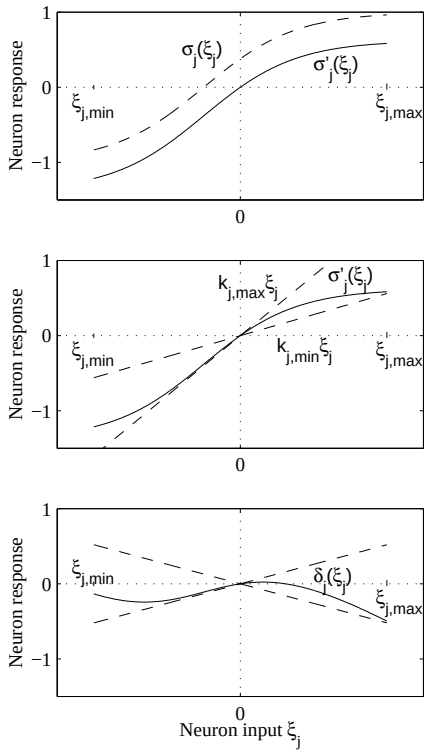


Figure 6.4: *Extraction of linear content from a hyperbolic tangent neuron.*

In order to illustrate the procedure above we will provide an expression for the sector bounds (6.21) and (6.22) for the $\tanh(\cdot)$ neuron function, which is probably the most popular neuron function employed in MLPs. Consider the neuron $\sigma_j(\xi_j) = \tanh(\xi_j + \theta_b)$

where θ_b is the scalar bias on the input ξ_j. Refer to Figure 6.4, where the top plot shows the parallel translation of the original neuron function with bias θ_b to the origin. We will without loss of generality assume that $\theta_b > 0$. Only the section of the neuron function, which corresponds to the input interval $[\xi_{j,min}; \xi_{j,max}]$, is considered.

On the middle plot the straight lines $k_{j,min}\xi_j$ and $k_{j,max}\xi_j$ have been added. Since $\frac{d^2 \tanh(s)}{ds^2} < 0$ for $s > 0$ it is immediately concluded that $k_{j,min}$ is given by

$$k_{j,min} = \frac{\sigma_j'(\xi_{j,max})}{\xi_{j,max}}.$$

$k_{j,max}$, on the other hand, can either be given by $\frac{\sigma_j'(\xi_{j,min})}{\xi_{j,min}}$ if the endpoint of the input range is sufficiently close to zero, or by the slope of the tangent to the neuron function which intersects 0. The relationship between the bias and the argument ξ_b for which said tangent coincides with the neuron function has been found numerically as

$$\xi_b = -0.00379\theta_b^3 + 0.07274\theta_b^2 - 1.5146\theta_b.$$

A closed form most likely does not exist. The polynomial given here provides values of $k_{j,max}$ with errors of the order of magnitude 10^{-5}.

Hence we have

$$k_{j,max} = \begin{cases} \frac{\sigma_j'(\xi_b)}{\xi_b}, & \text{for } \xi_b > \xi_{j,min} \\ \frac{\sigma_j'(\xi_{j,min})}{\xi_{j,min}}, & \text{for } \xi_b \leq \xi_{j,min}. \end{cases}$$

Note that there is no loss of generality in the assumption $\theta_b > 0$ since the fact that the (original) neuron function is odd ensures that the expressions given above hold for negative biases as well, with a few simple sign changes and swapping of minimum and maximum values.

To summarise, this section has presented a systematic method for transforming an MLP state space model of a nonlinear into a quasi-LPV model on the LFT form with a static and diagonal residual gain function. The transformation is performed in a way making the model suitable for LPV controller synthesis.

6.3.3 Uncertainty on the residual gains

Once the sector bounds for the nonlinearity have been determined we also have an explicit, smooth expression for the new set of neuron functions (given by eqns. (6.24) and (6.28)). If there is any uncertainty in the knowledge of x and u, then this will of course result in an uncertainty on the knowledge of $\Omega(\cdot)$. However, if we assume that some bound on the uncertainty of the inputs to the nonlinearity is known, then the above expression

can be exploited to provide a bound on the uncertainty of the gain of the nonlinearity:

$$\frac{\hat{\Omega}_j(\xi_j)}{\xi_j} = \frac{\Omega_j(\xi_j)}{\xi_j} + \varepsilon_{\Omega j}, \quad |\varepsilon_{\Omega j}| < \bar{\varepsilon}_{\Omega j}$$

where Ω_j is the j'th diagonal element of Ω. Such a bound can for instance be found by conducting a numerical search over the range of all permissible values of ξ. The bound on the measurement noise can be used together with Θ_x and Θ_u to estimate the uncertainty on ξ; then this uncertainty can be used to calculate an upper bound on ε_Ω.

6.4 Quasi-LPV speed controller

In Section 6.2 a stator current controller was designed based on a physical model of the induction motor. It is not always possible to construct a good model of a system based on physical considerations. In that case a nonlinear black-box model approach can be used, for instance using an MLP as a model. As discussed in Section 2.5, under certain assumptions it is possible to train an MLP as a nonlinear state space model with the same behaviour as the system using only input and output measurements.

When designing a speed controller the dynamics are heavily affected by the profile of the load torque. Thus a controller designed by linear methods in one operating point may not work in another operating point. This kind of problem can be overcome by obtaining a nonlinear model capturing the behaviour in the entire range of operation, and then use a nonlinear control design method.

In this section we will design a speed controller by the following steps:

- The system to be controlled is modelled by an MLP state space model.
- The obtained model is transformed into a quasi-LPV model on the LFT form by the method presented in Section 6.3.
- A controller is designed by the method described in Section 5.4.3.

6.4.1 Strategy

The overall aim in this section is to design a controller for the rotational speed ω_r and the magnetising current $\bar{i}_m$. The controller should work in the cascade coupling discussed in Section 4.2. This is illustrated in Figure 6.5. The entire block on the right containing the stator current controller, the power device, the induction motor system, as well as the speed and flux observers is considered as the system to be controlled.

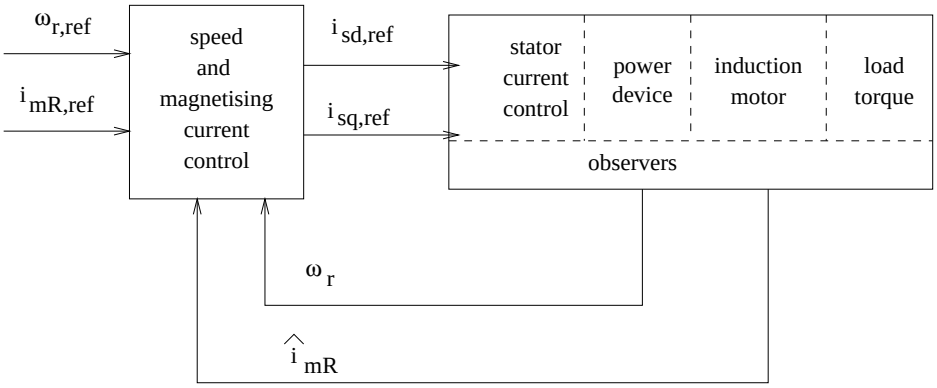


Figure 6.5: *Speed and magnetising current control scheme.*

Ideally, by using a black-box modelling strategy of the entire block, the speed and magnetising current controller should be able to compensate for errors introduced by the model used for the stator current control and to some extent for those introduced by the observers. This of course requires that the system behaviour can be modelled by the black-box model, and that an accurate model is indeed obtained.

The aim is to model the system by an MLP, and transform this into a quasi-LPV model on the LFT form as discussed in Section 6.3. This model will then by used for designing an LPV controller.

In order to simplify the controller, it is decided to assume that only the estimates of the rotor speed ω_r and the magnitude $\bar{i}_{mR}$ of the magnetising current are needed in order to obtain a good model of the system. An estimate of the angular velocity of the flux ω_{mR} could for instance have been used as an input to the model. In the final controller design this would then have entered as a parameter when calculating the residual gains. Similarly, measurements of the stator voltage could have been used.

A main factor in choosing the sampling frequency is the computational complexity of an LPV controller. The stator current controller as well as the observers are implemented at a sampling frequency of $3kHz$, but the complexity of the LPV controller and the limitations of the available hardware makes it necessary to implement the controller at a sampling frequency of just $600Hz$. Since the magnetising current is governed by relatively slow dynamics, this is no problem for this part of the controller. On the other hand it may limit the achievable performance for the speed controller slightly.

Two different types of model structures for MLPs were discussed in Section 2.5. The NARX model structure assumes that the output can be accurately predicted based only on old outputs and inputs. This limits the types of noise than can be modelled. On the other hand, the NARMAX allows for a very general model structure. However due to the possibility of convergence problems with the NARMAX model structure it is chosen to work with a NARX model here. For this problem, this means that the load torque m_L

must be chosen as a function of the speed ω_r only. Providing measurements of the shaft position to the model would allow for a position dependence as well, but it would still not be possible to let the load torque depend dynamically on some unknown disturbance.

It is chosen to let the load torque be a nonlinear function of the speed:

$$m_L = 8(1 - \frac{2}{1 + e^{0.01\omega_r}}),$$

where ω_r is in radians per second and m_L is in Newton meter. Notice that this information is only used to control the DC motor simulating the load torque and is assumed unknown in the modelling.

It is furthermore chosen to let the flux observer be the simple observer (4.15) based on the current model. In rotor flux coordinates the part concerning the magnitude i_{mR} of the magnetising current $\overline{i}_m$ is simply

$$\dot{\hat{i}}_{mR} = \frac{R_r}{L_r}(i_{sd} - \hat{i}_{mR}). \tag{6.29}$$

If we assume that i_{sd} is equal to $i_{sd,ref}$ then this estimate is a known function of the input, and there is no need to attempt to model it with the MLP. This is a reasonable assumption, since the magnetising current dynamics are much slower than the stator current dynamics. Thus the MLP is to predict ω_r based on old measurements of ω_r and estimates of i_{mR}.

6.4.2 MLP model

The first step in the control design procedure is to obtain an MLP model of the system. This consists of the following steps:

- Create data sets for training and validation.
- Choose the model type and parameters, for instance the model order.
- Train the MLP.
- Validate the achieved model.

The reason that we need both a training set and a validation set is the inherent danger in training neural networks of *overtraining*. If the MLP has a large number of neurons and therefore a large number of adjustable parameters, it may happen that the MLP learns the behaviour of the training set including noise rather than that of the actual underlying system. It is therefore necessary to test the behaviour of the MLP on a validation set.

Training set

When black-box modelling a nonlinear system, it is not sufficient for the data to contain a large number of frequencies in the range of interest. Since the dynamic behaviour can vary between operating points and even with the magnitude of the signals, it is necessary for the training set to cover a large number of operating points and signal amplitudes as well. This can be achieved by letting the input signals be pseudo-random signals, for instance steps of varying amplitude and length at various operating points.

It is chosen to create the data in partial closed-loop operation with a loosely tuned PI-controller for the following reasons. Firstly, the signals obtained from closed-loop experiments will mimic an environment which is closer to the one, in which the final controller will operate. Secondly, the range of operation will typically be specified in terms of the outputs rather than the inputs. Closed-loop operation allows us to make sure that data from the entire range are obtained. Finally, it may not be desirable to impose open-loop input signals which could cause the system to leave some allowed region of operation.

There are two problems with the closed-loop approach. Firstly, it is necessary to already have a functioning controller available. On the other hand, this controller does not have to achieve a good performance, so any loosely tuned controller which just stabilises the system in the entire operating range is all that is needed. The second problem is that the measured behaviour will be that of the closed-loop system rather than the open-loop system to be controlled. This problem can be alleviated by adding small pseudo-random signals to the output of the controller [Billings et al., 1992].

A section of the training set is shown in Figure 6.6. The reference for the speed was varied in steps of random length and amplitude in the range from -250 rad/s to 250 rad/s. The speed was controlled by a loosely tuned PI-controller of the type discussed in Section 4.2.1 generating a reference for i_{sq}. The reference for the magnetising current was generated in a similar manner in the range from 0.8 A to 2.8 A. The magnetising current was controlled by a P-controller generating a reference for i_{sd} as discussed in Section 4.2.2. Random signals with an amplitude of 0.8 A were added to the stator current references generated by these controllers.

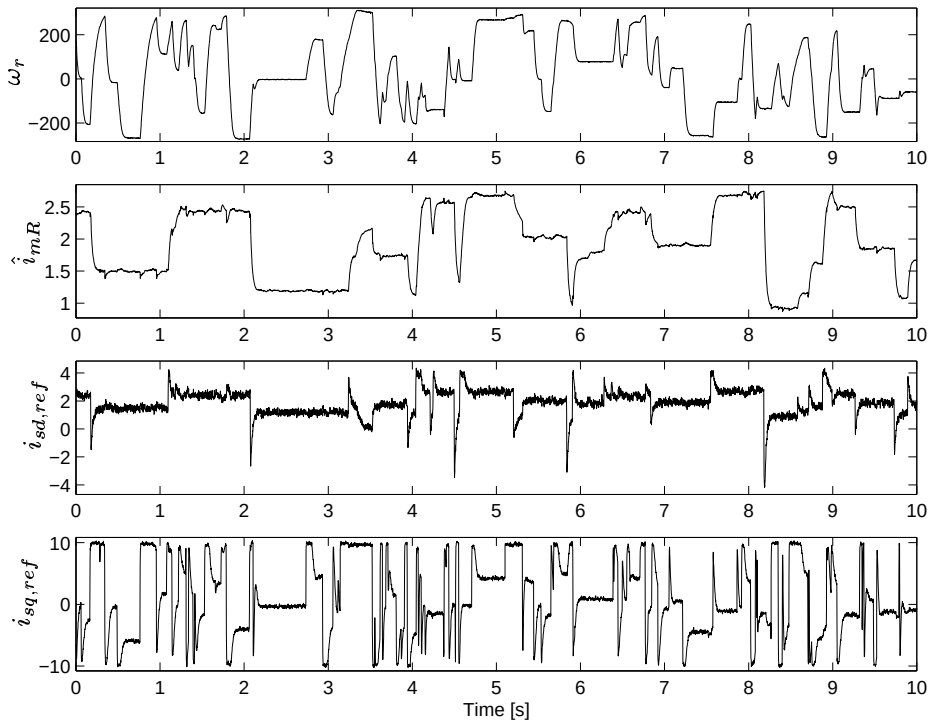


Figure 6.6: *6000 samples of the training set. First (top) figure: measured speed. Second figure: estimated magnitude of magnetising current. Third and fourth figures: references for the stator current controller.*

The control loop generating the data set operated at a sampling frequency of 3 kHz. The data was sampled at 600 Hz after being filtered by first order filters with bandwidths of 100 Hz.

A validation set was generated in exactly the same way. Both the training and the validation set consist of 22000 samples. The data was scaled before the training:

$$\begin{aligned}
\tilde{\omega}_r &\triangleq \sigma_\omega^{-1}(\omega_r - \omega_{r,0}),\\
\tilde{\hat{i}}_{mR} &\triangleq \sigma_i^{-1}(\hat{i}_{mR} - \hat{i}_{mR,0}),\\
\tilde{i}_{sd,ref} &\triangleq \sigma_d^{-1}(i_{sd,ref} - i_{sd,ref,0}),\\
\tilde{i}_{sq,ref} &\triangleq \sigma_q^{-1}(i_{sq,ref} - i_{sq,ref,0}),
\end{aligned}$$

where the σ's denote standard deviances of the respective signals, and the zero subscripts denote a known operating point.

Model type

Rather than using $i_{sq,ref}$ as an input to the model, we will attempt to precompensate for the known nonlinearity in (4.6) by instead using the input

$$i_\tau \triangleq i_{sq,ref}\hat{i}_{mR}.$$

As mentioned above we will attempt to model the system by a NARX MLP model as the one discussed in Section 2.5.2, i.e., we wish to predict the next value of ω_r based on old values of ω_r, $\hat{i}_{mR}$, $i_{sd,ref}$, and $i_{sq,ref}$. More specifically we wish to find weights Θ_1, Θ_2, and Θ_b such that

$$\hat{\tilde{\omega}}_{r,k} \triangleq M_m(z_k, \Theta_1, \Theta_2, \Theta_b), \approx \tilde{\omega}_{r,k}$$

where M_m is defined as in (2.21), and

$$z_k \triangleq \begin{bmatrix} \tilde{\omega}_{r,k-1} \\ \vdots \\ \tilde{\omega}_{r,k-n} \\ \tilde{\hat{i}}_{mR,k-1} \\ \vdots \\ \tilde{\hat{i}}_{mR,k-n_i} \\ \tilde{i}_{sd,ref,k-1} \\ \vdots \\ \tilde{i}_{sd,ref,k-n_u} \\ \tilde{i}_{\tau,k-1} \\ \vdots \\ \tilde{i}_{\tau,k-n_u} \end{bmatrix},$$

with $\tilde{i}_\tau \triangleq \sigma_\tau^{-1}(i_\tau - i_{sq,ref,0}\hat{i}_{mR,0})$, where σ_τ is the standard deviation of i_τ.

We choose the neuron functions to be tangent hyperbolic. Alternatively we could have chosen a combination of tangent hyperbolic and linear neuron functions, but the tangent hyperbolic neuron can yield almost linear behaviour simply by making the input weights small and the output weights large. The only thing remaining is now to specify the number of neurons, the order of the model n, and the number of delayed inputs n_u and n_i. These choices are left for the training, since some experimentation is usually necessary.

Training

Since the training of an MLP is nonconvex problem, the initial values of the parameters are important. This was found to be even more important, when the model is to be used

for designing a quasi-LPV controller. It was found that many different models gave the same performance in terms of summed squared prediction errors, but that in order to construct a controller yielding a good closed-loop performance it was necessary for the sector bounds (as discussed in Section 6.3.1) to be small. This is partially due to some conservatism discussed below in Section 6.4.3.

Define the *input matrix*

$$M_i \triangleq \begin{bmatrix} z_s & z_{s-1} & \dots & z_{n+1} \end{bmatrix}$$

and the *target matrix*

$$M_o \triangleq \begin{bmatrix} \tilde{\omega}_{r,s} & \tilde{\omega}_{r,s-1} & \dots & \tilde{\omega}_{r,n+1} \end{bmatrix},$$

where s is the number of samples in the training set.

The linear (ARX) model minimising the least-squares prediction error performance index

$$J_l \triangleq \frac{1}{s-n} \sum_{i=n+1}^{s} \epsilon_{l,i}^T \epsilon_{l,i},$$

where the prediction error ϵ_l is defined as

$$\epsilon_{l,k} \triangleq \tilde{\omega}_{r,k} - \hat{\tilde{\omega}}_{r,l,k},$$

is given by

$$\hat{\tilde{\omega}}_{r,l,k+1} = M_l z_k,$$

with the linear gain M_l defined as [Elbert, 1984]

$$M_l \triangleq M_o M_i^{\dagger}.$$

One could train the MLP to only learn the nonlinear part, i.e. defining the targets for the MLP as the prediction errors of the ARX model. This could potentially reduce the number of neurons needed as well as improving the chances of converging to a global minimum, but it was found that this lead to unpleasantly large sector bounds.

Instead the ARX model is used to generate the initial weights for the MLP training. We let the initial MLP model approximate the ARX model by letting $\Theta_b = 0$ and $\Theta_2 \Theta_1 = M_l$. We perform the factorisation of M_l such that Θ_2 is very large and Θ_1 is very small. Then $M_m(z) \approx M_l z$.

A number of MLPs were trained with various numbers of neurons and model orders using the Levenberg-Marquardt training algorithm to minimise the prediction error

$$J_m \triangleq \frac{1}{s-n} \sum_{i=n+1}^{s} \epsilon_{m,i}^T \epsilon_{m,i},$$

where the prediction error ϵ_m is defined as

$$\epsilon_{m,k} \triangleq \tilde{\omega}_{r,k} - \hat{\tilde{\omega}}_{r,k}.$$

A reasonable prediction error for both the training and validation sets was obtained with 4 neurons, a fourth order model ($n = 4$), $n_i = 1$, and $n_u = 2$.

Model validation

Figure 6.7 shows the prediction error for 500 samples of the training set. The top figure shows the prediction error for the MLP model, the bottom figure shows the prediction error for the ARX model. Figure 6.8 shows the same for the validation set. It may be difficult to see any significant difference in the plots but the unscaled prediction error performances show that the MLP predicts the system behaviour better than the ARX model:

Model	Training set	Validation set
MLP, $\sigma_\omega J_m =$	1.04	1.02
ARX, $\sigma_\omega J_l =$	1.71	1.82

The performances indicate that the MLP model has captured at least some of the non-linear behaviour of the system, and that, since the validation set prediction error is not larger than for the training set, it has not been overtrained.

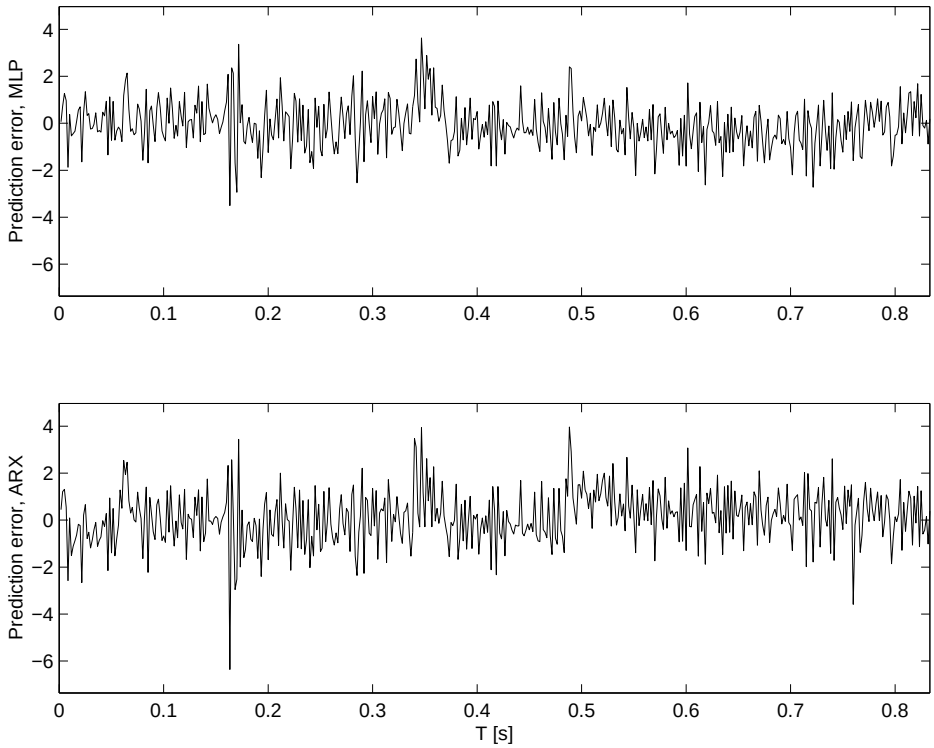


Figure 6.7: *Prediction error for 500 samples of the training set. The top figure shows the prediction error for the MLP model, the bottom figure shows the prediction error for the ARX model.*

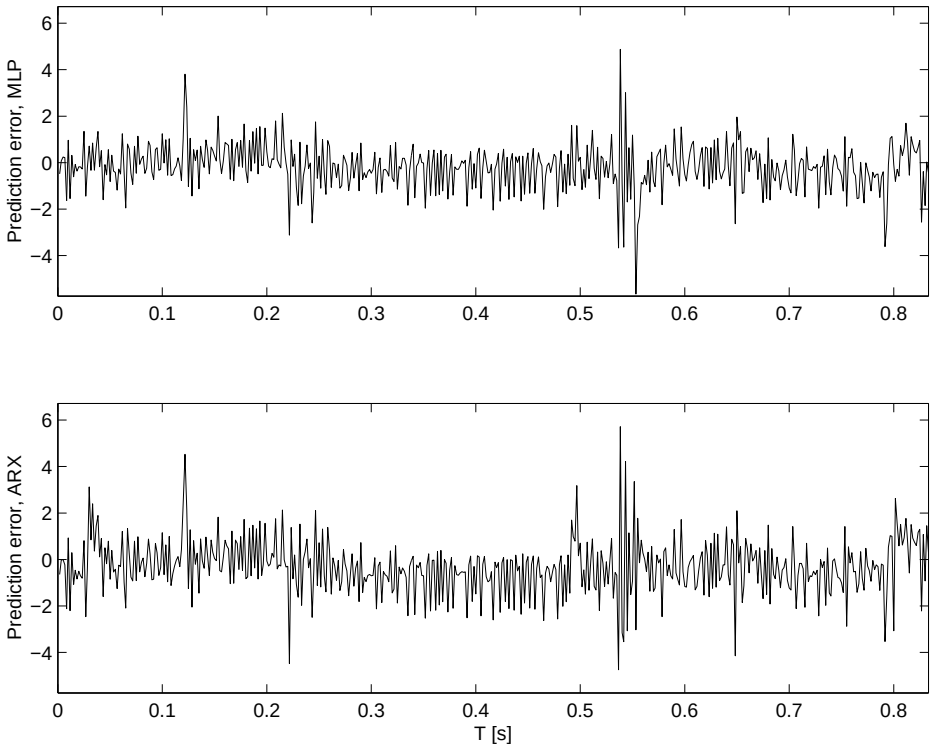


Figure 6.8: *Prediction error for 500 samples of the validation set. The top figure shows the prediction error for the MLP model, the bottom figure shows the prediction error for the ARX model.*

An alternative way of validating the model is to test the auto-correlation of the prediction error. Ideally there should be no correlation between the prediction errors. Figure 6.9 shows the scaled auto-correlation of the prediction error for the validation set. The plot indicates that the prediction error of the MLP model is not entirely uncorrelated, but on the other hand it is much better than for the linear model.

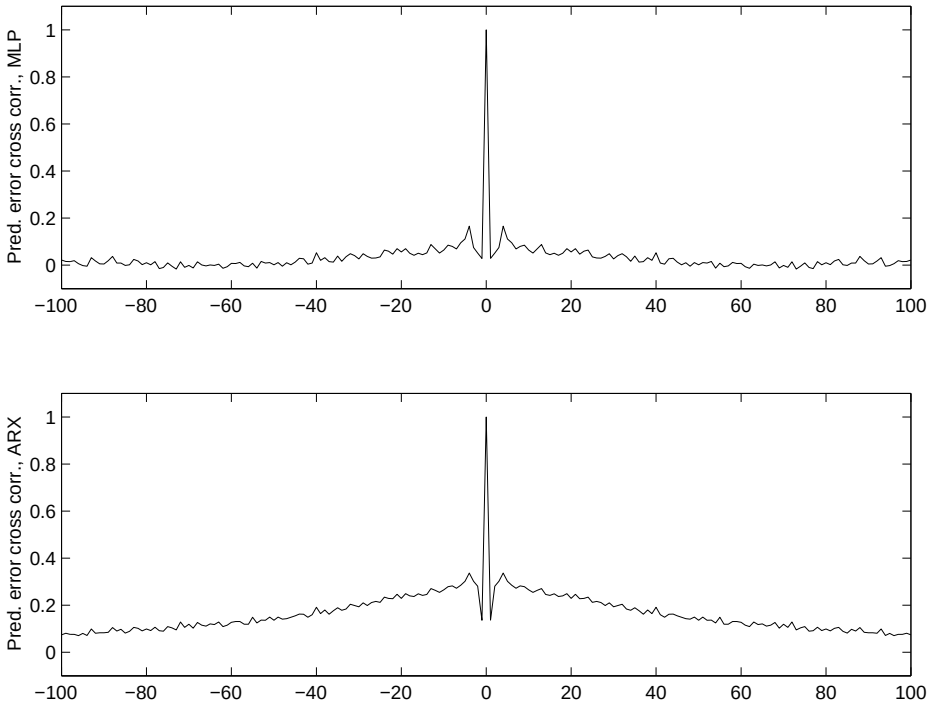


Figure 6.9: *Scaled auto-correlation of the prediction error for the validation set. The top figure shows the auto-correlation of the prediction error for the MLP model, the bottom figure shows the same for the ARX model.*

Another way of testing the obtained model is to use it as an open-loop simulator, i.e. replacing the delayed measurements of ω_r with the values predicted by the model. This is shown in Figure 6.10 for the training set and in Figure 6.11 for the validation set. The top figures show the simulation using the MLP model. The middle figures show the simulation using the ARX model. The bottom figures show the simulation error for the MLP model.

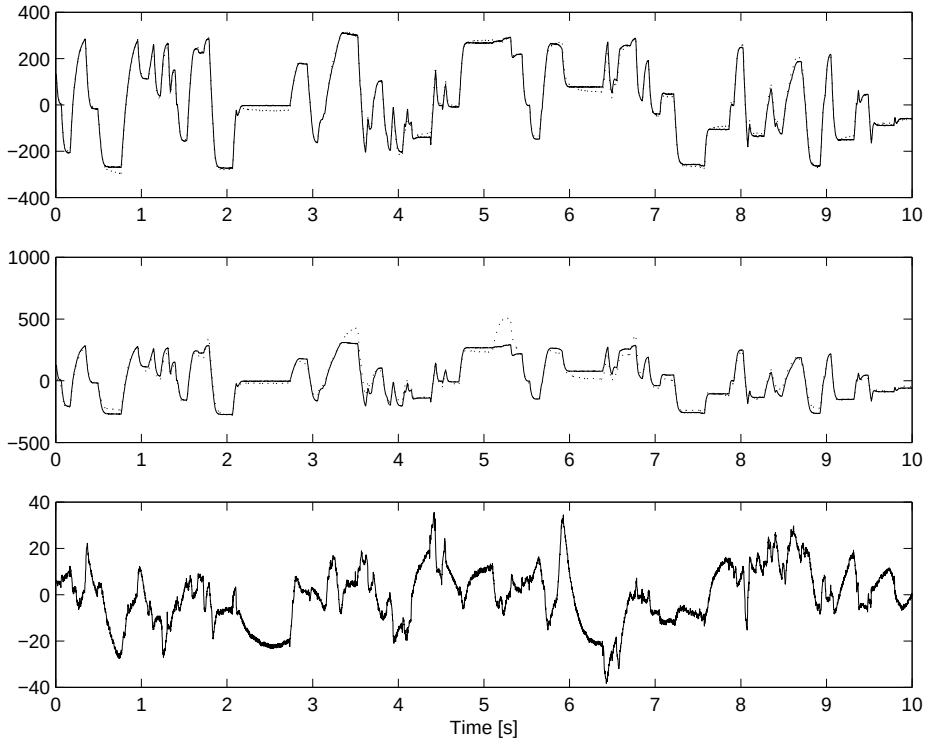


Figure 6.10: *Open-loop simulation of the speed using the MLP model (top figure) and the ARX model (middle figure) for 6000 samples of the training set. The solid lines show the simulated speed, the dotted lines show the actual (measured) speed. The bottom figure shows the simulation error for the MLP model.*

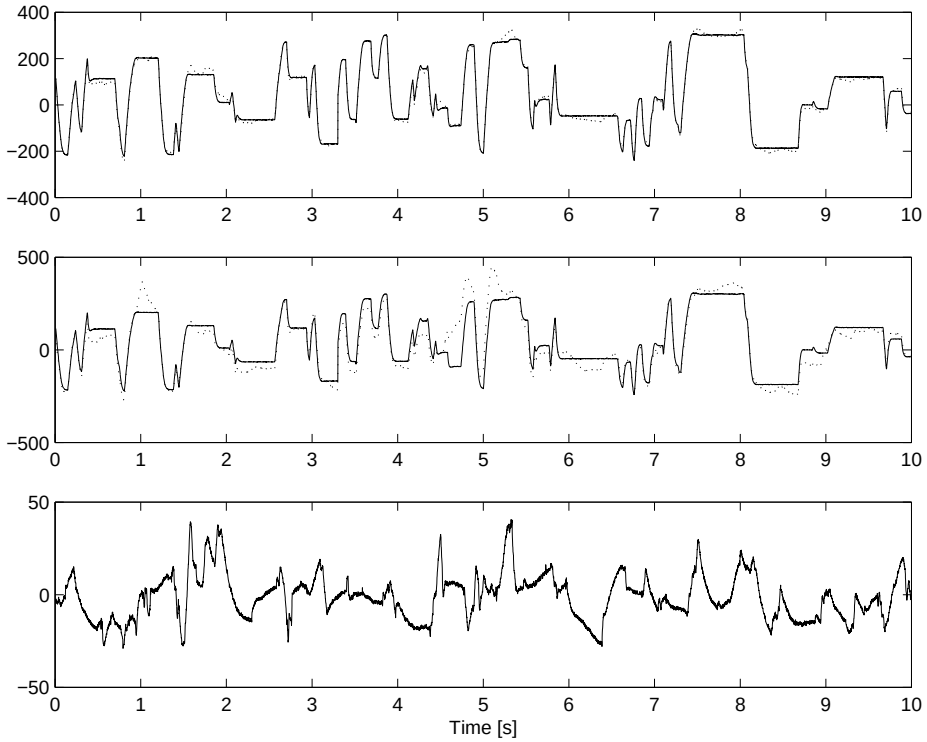


Figure 6.11: *Open-loop simulation of the speed using the MLP model (top figure) and the ARX model (middle figure) for 6000 samples of the validation set. The solid lines show the simulated speed, the dotted lines show the actual (measured) speed. The bottom figure shows the simulation error for the MLP model.*

The simulations show that the MLP model is a much better open-loop simulator than the ARX model. On the other hand, it does still show some systematic errors.

The overall conclusion of the validation procedure is that an MLP model of the system has been obtained which yields satisfactory performance both as a predictor and as an open-loop simulator.

6.4.3 Controller design

The aim is now to transform the MLP model into a quasi-LPV model on the LFT form suitable for LPV control design. The first step is to perform the transformation discussed in Section 6.3.1. The sector bounds were found by the method presented in Section 6.3.2 as

j	$k_{j,max} - k_{j,min}$
1	0.0114
2	0.0092
3	0.0062
4	0.0035

The transformation provided a model on the form

$$\begin{bmatrix} \zeta_{k+1} \\ z_{u,k} \end{bmatrix} = \begin{bmatrix} A_0 & B_u & B_i & B_1 & B_2 \\ C_u & 0 & D_i & D_{u1} & D_{u2} \end{bmatrix} \begin{bmatrix} \zeta_k \\ w_{u,k} \\ \tilde{\hat{i}}_{mR,k} \\ u_k \\ u_{k-1} \end{bmatrix},$$

where

$$\zeta_k \triangleq \begin{bmatrix} \tilde{\omega}_{r,k} \\ \tilde{\omega}_{r,k-1} \\ \tilde{\omega}_{r,k-2} \\ \tilde{\omega}_{r,k-3} \end{bmatrix} \text{ and } u_k \triangleq \begin{bmatrix} \tilde{i}_{sd,ref,k} \\ \tilde{i}_{\tau,k} \end{bmatrix},$$

and $w_{u,k} = \Omega(z_{u,k})$, where $\Omega(\cdot)$ is a static nonlinearity. This is also illustrated in Figure 6.12, where q^{-1} denotes the delay operator.

PSfrag replacements

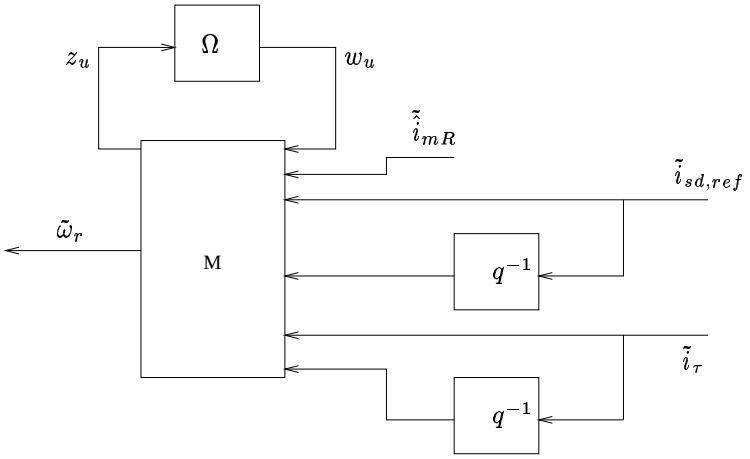
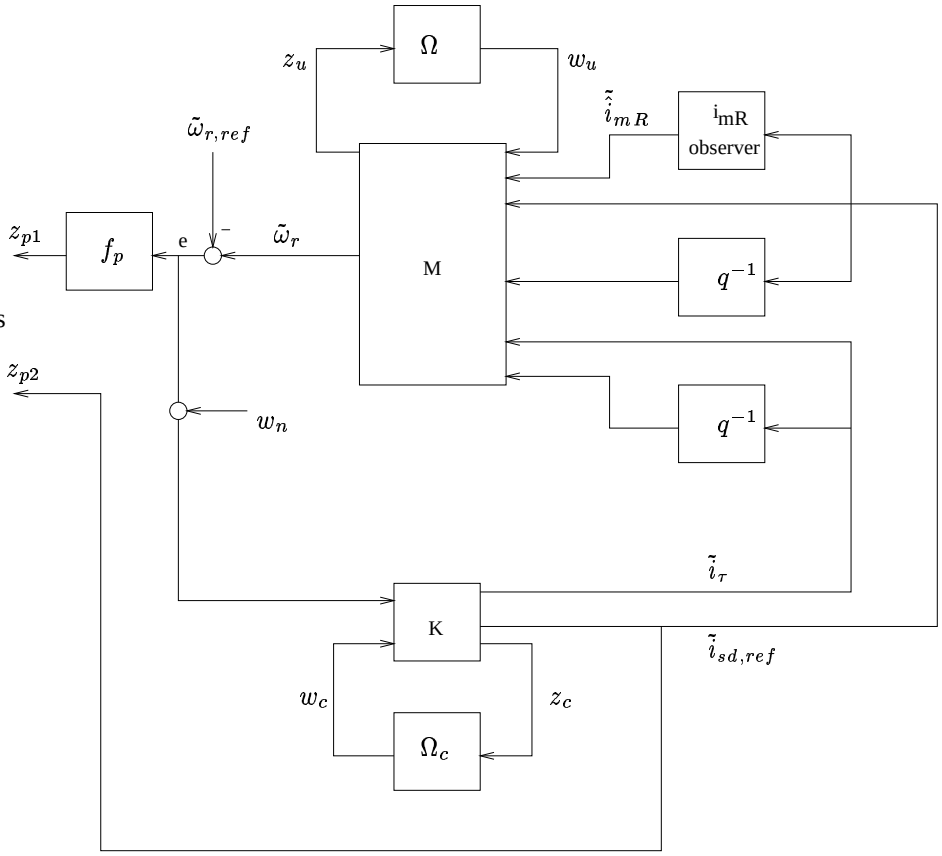


Figure 6.12: *MLP model on LFT form.*

In order to design the controller this system was expanded into the one in Figure 6.13. The i_{mR} observer block contains the discrete time version of equation (6.29) replacing i_{sd} with $i_{sd,ref}$. A reference signal $\tilde{\omega}_{r,ref}$ for the speed is subtracted from the actual

speed yielding a control error e. This control error is low-pass filtered by the filter f_p yielding part of the performance output. The performance filter f_p is a first order filter with a bandwidth of 5 Hz. Before feeding the control error to the controller K, measurement noise ω_n with variance $0.5\sigma_{\omega}^{-1}$ is added. The controller outputs the control signals $\tilde{i}_{sd,ref}$ and $\tilde{i}_{\tau}$.



PSfrag replacements

Figure 6.13: *Setup for LPV speed controller synthesis.*

The final thing to notice about the setup is that there is no reference signal to or performance signal from the magnetising current. Instead a penalty has been put on $\tilde{i}_{sd,ref}$ by outputting it as the performance signal z_{p2}. This is due to the fact that an early attempt to design a controller for both $\tilde{\omega}_r$ and $\tilde{\hat{i}}_{mR}$ resulted in a controller relying heavily on $\tilde{i}_{sd,ref}$ for speed control. From our knowledge of the nonlinearities, we know that this is not desirable, but this information is hidden in the residual gains Ω, and the control design procedure only employs knowledge of the bounds on the gains, thus ignoring this knowledge of the nonlinearities. The setup in Figure 6.13 results in a controller setting

$\tilde{i}_{sd,ref} \approx 0$. This control signal is then replaced by the control signal from a traditional magnetising current controller. This method is an ad hoc approach and most likely not the best approach. One problem is that the speed controller assumes $\tilde{i}_{sd,ref} \approx 0$, where it would probably be a better idea to include $\tilde{i}_{sd,ref}$ as a measurable disturbance. This should certainly be attempted in future versions.

The overall result is an eighth order model: four orders from the MLP model, two from the input delays, one from the observer, and one from the performance filter. A controller K and a scheduling function Ω_c were designed using the method described in Section 5.4 attempting to minimise the l_2-gain from $\tilde{\omega}_{r,ref}$ and ω_n to z_{p1} and z_{p2}. In order to obtain a controller which could be implemented in real-time it was necessary to disregard the two residual gains with the smallest sector bounds. It was possible to solve the LMI with an l_2-gain less than 0.016. However, due to numerical problems in solving the quadratic matrix inequality it was necessary to increase the bound to $\gamma_a = 0.018$ in order to obtain a controller. For comparison, designing a controller disregarding the residual gains, an l_2-gain less than $\gamma_l = 0.011$ could be achieved.

There are two sources of conservatism to the method employed here. First we consider the residual gains as a diagonal gain with no correlation between the individual gains. Finding a way to first obtain information of the correlation and secondly using this in a finite-dimensional scheme would be a lot more troublesome. Secondly, the rate of variation in the residual gains are not taken into account. It would seem a relatively straight-forward task to expand the procedure discussed in Section 6.3 in order to obtain bounds on the rate of variation. However, this is a matter for further research.

The obtained controller is on the form (5.57)-(5.58), where the scheduling subspace S_c depends on Δ_k, which is a diagonal gain matrix such that

$$\Delta_k z_{u,k} = \Omega(z_{u,k}).$$

This poses an algebraic loop problem, since $z_{u,k}$ depends on u_k. Thus, in order to compute S_c at sample k it is necessary to know u_k, and in order to compute u_k it is necessary to know S_c. If the sampling frequency is sufficiently high, and the controller has a reasonably low high-frequency gain, then the control signal can be expected to change only slightly from sample to sample, and u_{k-1} can be used as an estimate of u_k in computing Δ_k. Alternatively, an iterative scheme could be used to alternately compute u_k and Δ_k in an algebraic loop until the results (hopefully) converge. Since the sample rate is mainly limited by the computational power of the PC, this latter approach does not seem viable.

The controller was implemented using the first of these approaches, i.e. assuming $u_{k+1} \approx u_k$ when computing the scheduling subspace.

6.4.4 Closed-loop experiments

A closed-loop experiment was performed using the LPV speed controller. The result is shown in Figure 6.14, where the measured speed is shown by the solid line and the speed reference is shown by the dotted line. The reference moves in small steps from $-250 rad/s$ to $200 rad/s$ and then back to $-250 rad/s$ in one large step.

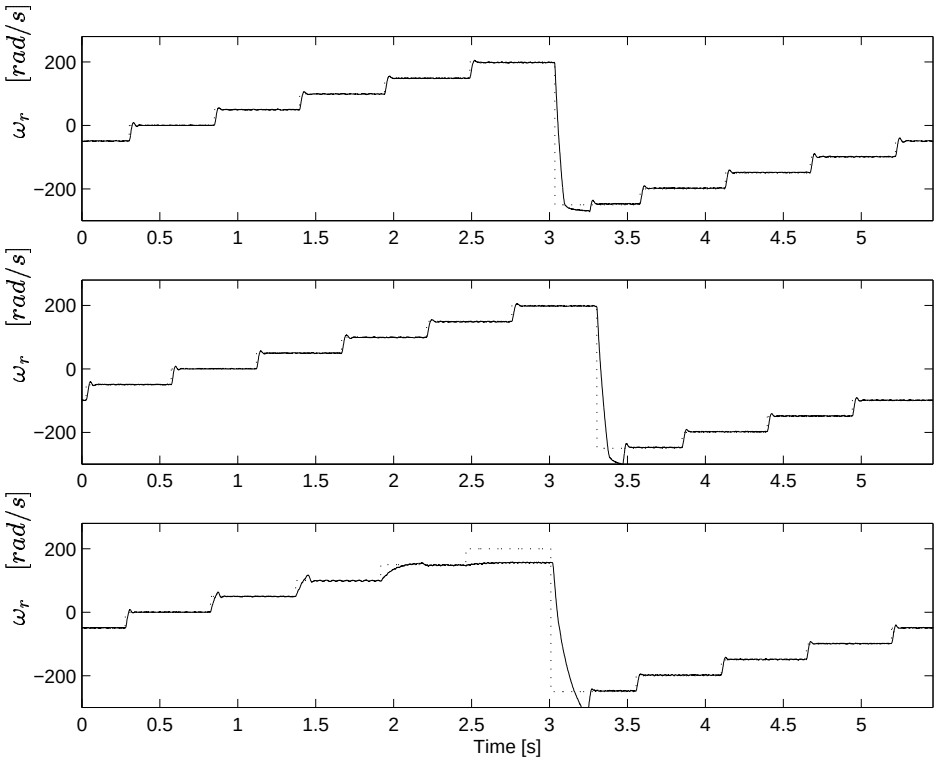


Figure 6.14: *Closed-loop experiment using the LPV speed controller. The speed reference is shown by the dotted line. The three figures show the behaviour with three different settings for the magnetising current. Top:* $i_{mR,ref} = 2.8A$. *Middle:* $i_{mR,ref} = 2.2A$. *Bottom:* $i_{mR,ref} = 1.0A$.

The experiment is performed for three different degrees of magnetisation, $i_{mR,ref} = 2.8A$, $i_{mR,ref} = 2.2A$, and $i_{mR,ref} = 1.0A$ respectively. The behaviour is satisfactory for all three situations, although for $i_{mR,ref} = 1.0A$ the performance is somewhat degraded for large (positive and negative) speeds. This is to be expected since the achievable torque is limited by the allowable stator current.

For comparisons an experiment is performed with a PI-controller tuned by the Ziegler-Nicholls relay method around $\omega_r = 0 rad/s$. The results are shown in Figure 6.15.

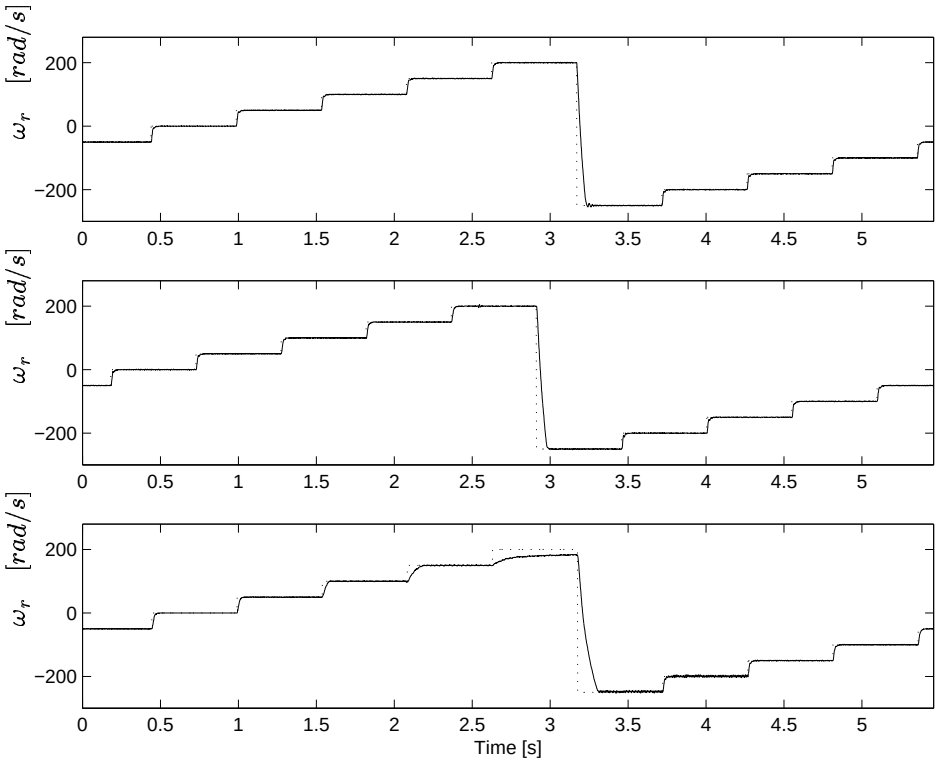


Figure 6.15: *Closed-loop experiment using a PI speed controller. The speed reference is shown by the dotted line. The three figures show the behaviour with three different settings for the magnetising current. Top:* $i_{mR,ref} = 2.8A$. *Middle:* $i_{mR,ref} = 2.2A$. *Bottom:* $i_{mR,ref} = 1.0A$.

As seen the performance of the PI-controller is somewhat better than that of the LPV controller. This is partially due to the PI-controller being implemented at a sample rate of $3kHz$, whereas the LPV controller was implemented at a sample rate of $600Hz$, although this cannot explain the entire difference. However, the main purpose of this section was to demonstrate how a controller could be designed for a nonlinear system using a systematic approach from MLP modelling to LPV controller design.

6.5 Summary

In this chapter the quasi-LPV approach was used to construct both a stator current controller and a speed controller. The quasi-LPV approach makes it possible to use LPV control methods on very general nonlinear systems.

A stator current controller was designed based on the model obtained in Chapter 3 and a satisfactory performance was achieved.

The speed controller was based on an MLP model. The MLP was used to form a non-linear state space model of a system including the induction motor, the stator current controller, and the flux observer. It was shown that the MLP model had captured the behaviour of the system. The MLP model was transformed into a quasi-LPV model on the LFT form, and a speed controller was designed yielding a reasonable performance.

Chapter 7

ROBUST LPV SPEED CONTROLLER

The LPV controller design methods described in Chapter 5 assumed that the time-varying parameters were fully known. However, very often the parameters are only known with some small uncertainty. This would for instance be the case if the parameters are measured under the influence of measurement noise. In the case of quasi-LPV control the parameters can depend on the system states, and then any uncertainty in the knowledge of the states gives rise to an uncertainty on the parameters.

Very little research has been done on the subject of synthesising controllers for LPV systems with small uncertainties on the time-varying parameters. In [Helmersson, 1995] a method is given for the situation where some parameters are fully known and others are completely unknown except for some bounds. The unknown parameters lead to non-convex rank constraints on the multipliers, i.e. we have to enforce for instance $P = \tilde{P}^{-1}$. Such constraints can in lucky cases be solved, for instance, using alternating projections, as discussed in for instance [Grigoriadis and Skelton, 1996] and [Beran and Grigoriadis, 1996]. This approach was used for a robust flux observer design in [Trangbæk, 2000]. This approach can also be used for the situation, where the time-varying parameters are known except for some small uncertainty, simply by splitting the

time-varying parameter into a known part and a small unknown part. It is however believed that the method presented below is simpler and will yield better result, possibly after further developments.

In the approach described in Chapter 5 we can achieve robustness by ensuring that the scheduling function is void. Consider the scheduling function in (5.36). If N_- has no rows or if V_+ has no columns then the scheduling function Δ_c is void, and the controller is actually robust to the parameter variations. These constraints correspond to $m_c = 0$ or $k_c = 0$, respectively. These are again equivalent to $P \geq \tilde{P}^{-1}$ or $P \leq \tilde{P}^{-1}$, respectively. Unfortunately this will rarely be the case when just solving the synthesis LMIs. Again some non-convex approach would have to be applied.

In Section 7.1 we will give an alternative approach for the case when the parameters are known except for some uncertainty, which is small compared to the actual parameter variations. We will end up with constraints on the multipliers, which are also non-convex, but are, however somewhat easier to satisfy than the rank constraints.

In order to simplify matters, we shall restrict ourselves to diagonal residual gains and diagonal multipliers. Diagonal residual gains arise, for instance, when using the transformation method for MLPs presented in Section 6.3. Restricting ourselves to diagonal multipliers restores the conservatism, which was removed by using the full block S-procedure rather than more traditional LPV control methods as discussed in Section 5.1. It is however hoped that this can serve as a first step towards a less conservative method.

In Section 7.2 the method is applied to the same speed controller problem as in Section 6.4 but this time taking uncertainty on the residual gains into account.

7.1 Robust LPV control of systems with diagonal variation

In this Section we will consider the problem of designing LPV controllers for the system (5.11), with

$$w_u(t) = \Delta(t) z_u(t),$$

where Δ is a diagonal matrix, which is known in real-time except for some small diagonal uncertainty. We shall assume that all matrices are real.

If for instance $\Delta(t)$ depends on parameters which can be measured in real-time, then noise on these measurements result in uncertainty on $\Delta(t)$, i.e. instead of using Δ to form the scheduling function, we have to use an estimate $\hat{\Delta}$.

Due to the LFT representation achieved through the full block S-procedure, the uncertainty only affects (5.9) in the analysis LMIs.

Hence, assuming both the system and the controller to be on the standard LFT form, instead of ensuring (5.9), we need to fulfill

$$\left[\begin{array}{cc|cc} \Delta & 0 & & \\ 0 & \Delta_c(\hat{\Delta}) & & \\ \hline I & 0 & & \\ 0 & I & & \end{array}\right]^T P_e \left[\begin{array}{cc|cc} \Delta & 0 & & \\ 0 & \Delta_c(\hat{\Delta}) & & \\ \hline I & 0 & & \\ 0 & I & & \end{array}\right] > 0. \tag{7.1}$$

If this can be fulfilled for all $(\Delta, \hat{\Delta})$, then the quasi-LPV controller will stabilise the system and achieve the required robust quadratic performance.

The problem that will be addressed in this section is thus to find additional constraints on the multipliers P and $\tilde{P}$ such that we can satisfy (7.1) even in the presence of uncertainty on $\Delta(t)$. To simplify the derivations, we will only consider constant, diagonal multipliers.

Theorem 7.1 (Robust diagonal LPV control)

Consider the LPV system (5.11) with

$$w_u = \Delta(t) z_u, \quad \Delta(t) \in \bar{\Delta},$$

where

$$\bar{\Delta} \triangleq \{\Delta : \Delta = \mathrm{diag}_{1 \le i \le n_u}\{\delta_i\}, |\delta_i| < 1\}.$$

Assume that $\Delta(t)$ is known in real-time except for some small diagonal uncertainty E, i.e.

$$\hat{\Delta}(t) = \Delta(t) + E(t), E = \mathrm{diag}_{1 \le i \le n_u}\{e_i\}, |e_i| < \bar{e}_i.$$

If there exist X, Y and

$$Q = \mathrm{diag}_{1 \le i \le n_u}\{q_i\} < 0 \tag{7.2}$$
$$R = \mathrm{diag}_{1 \le i \le n_u}\{r_i\} > 0 \tag{7.3}$$
$$\tilde{Q} = \mathrm{diag}_{1 \le i \le n_u}\{\tilde{q}_i\} < 0 \tag{7.4}$$
$$\tilde{R} = \mathrm{diag}_{1 \le i \le n_u}\{\tilde{r}_i\} > 0 \tag{7.5}$$

satisfying the inequalities (5.21)–(5.23) with

$$S = 0, \qquad \tilde{S} = 0 \tag{7.6}$$
$$q_i(1 + \bar{e}_i)^2 + r_i > 0, 1 \le i \le n_u \tag{7.7}$$
$$\tilde{q}_i^{-1}(1 + \bar{e}_i)^2 + \tilde{r}_i^{-1} > 0, 1 \le i \le n_u. \tag{7.8}$$

and for each $i = 1..n_u$ one *of the following three conditions holds*

1.

$$(q_i - \tilde{q}_i^{-1})(r_i - \tilde{r}_i^{-1}) > 0 \tag{7.9}$$

2.

$$(q_i > \tilde{q}_i^{-1}) \quad \textit{and} \quad (r_i < \tilde{r}_i^{-1}) \tag{7.10}$$

and

$$\begin{aligned}((1+\bar{e}_i)^2 q_i + \tilde{r}_i^{-1})^2 (\tilde{q}_i^{-1} + \tilde{r}_i^{-1})(q_i + r_i) > \\ ((1+\bar{e}_i) q_i - \tilde{r}_i^{-1})^2 \bar{e}_i^2 (\tilde{r}_i^{-1} - r_i)(q_i - \tilde{q}_i^{-1})\end{aligned} \tag{7.11}$$

3.

$$(q_i < \tilde{q}_i^{-1}) \quad \textit{and} \quad (r_i > \tilde{r}_i^{-1}) \tag{7.12}$$

and

$$\begin{aligned}((1+\bar{e}_i)^2 \tilde{q}_i^{-1} + r_i)^2 (q_i + r_i)(\tilde{q}_i^{-1} + \tilde{r}_i^{-1}) > \\ ((1+\bar{e}_i) \tilde{q}_i^{-1} - r_i)^2 \bar{e}_i^2 (r_i - \tilde{r}_i^{-1})(\tilde{q}_i^{-1} - q_i)\end{aligned} \tag{7.13}$$

then there exists a controller $K(\hat{\Delta})$ on the form (5.13) with $w_c = \Delta_c(\hat{\Delta}(t)) z_c(t)$ that yields robust quadratic performance with performance index P_p.

Proof: It is first of all noted that the inequalities in (5.20) are implied by (7.2)–(7.8), and that the estimated residual gains $\hat{\Delta}$ only appear in the inequalities involving the extended multiplier P_e. In order to prove the result, we hence need to show that with the extra requirements given above, (7.1) can be fulfilled. If that can be shown, then Theorem 5.10 ensures the existence of the desired controller $K(\hat{\Delta})$. We construct N, P_e and $\Delta_c(\Delta)$ as in the proof of Theorem 5.10, though U must be constructed in a particular way, which will be addressed below. Then (5.9) is satisfied, which is equivalent to

$$\begin{bmatrix} \begin{bmatrix} \Delta \\ I \end{bmatrix}^T P \begin{bmatrix} \Delta \\ I \end{bmatrix} & V_- \Delta_c + V_+ \\ \Delta_c^T V_-^T + V_+^T & \Delta_c^T N_-^{-1} \Delta_c + N_+^{-1} \end{bmatrix} > 0.$$

By a Schur argument, this is equivalent to

$$\begin{bmatrix} \begin{bmatrix} \Delta \\ I \end{bmatrix}^T P \begin{bmatrix} \Delta \\ I \end{bmatrix} & V_- \Delta_c + V_+ & 0 \\ \Delta_c^T V_-^T + V_+^T & N_+^{-1} & \Delta_c^T \\ 0 & \Delta_c & -N_- \end{bmatrix} > 0.$$

Via a congruence transformation, this expression can be rewritten as

$$\begin{bmatrix} \begin{bmatrix}\Delta\\ I\end{bmatrix}^T P \begin{bmatrix}\Delta\\ I\end{bmatrix} - V_- N_- V_-^T & V_+ & V_- N_- \\ V_+^T & N_+^{-1} & \Delta_c^T \\ N_- V_-^T & \Delta_c & -N_- \end{bmatrix} > 0. \tag{7.14}$$

Furthermore, since P fulfills (5.20) or equivalently

$$\begin{bmatrix}\Delta\\ I\end{bmatrix}^T P \begin{bmatrix}\Delta\\ I\end{bmatrix} = \begin{bmatrix}\Delta\\ I\end{bmatrix}^T P \begin{bmatrix}\Delta\\ I\end{bmatrix} - V_- N_- V_-^T + V_- N_- V_-^T > 0$$

we have by Schur complement that

$$\begin{bmatrix} \begin{bmatrix}\Delta\\ I\end{bmatrix}^T P \begin{bmatrix}\Delta\\ I\end{bmatrix} - V_- N_- V_-^T & V_- N_- \\ N_- V_-^T & -N_- \end{bmatrix} > 0.$$

This implies that we can apply the Schur complement to (7.14) and obtain the equivalent inequality

$$\begin{bmatrix} N_+^{-1} & \Delta_c^T \\ \Delta_c & -N_- \end{bmatrix} - \Pi \left(\begin{bmatrix}\Delta\\ I\end{bmatrix}^T P \begin{bmatrix}\Delta\\ I\end{bmatrix} - V_- N_- V_-^T \right)^{-1} \Pi^T > 0 \tag{7.15}$$

where $\Pi = \begin{bmatrix} V_+ & V_- N_- \end{bmatrix}^T$. With the diagonal structure of the multiplier, we can define the matrix

$$D \triangleq \Delta Q \Delta + R - V_- N_- V_-^T$$

and write (7.15) as

$$\begin{bmatrix} N_+^{-1} & \Delta_c^T \\ \Delta_c & -N_- \end{bmatrix} - \Pi D^{-1} \Pi^T > 0,$$

and (5.36) as $\Delta_c(\Delta) = N_- V_-^T D^{-1} V_+$, respectively. We will now choose U in the following way. Let $U = \begin{bmatrix} T_1 & T_2 \end{bmatrix}$ such that $V_- = \begin{bmatrix} \Delta & I \end{bmatrix} T_1$ and $V_+ = \begin{bmatrix} \Delta & I \end{bmatrix} T_2$. If necessary we can perturb $\tilde{P}$ such that it is nonsingular. Since the columns of U form an orthogonal basis of the image of $P - \tilde{P}^{-1}$ (if this matrix happens to be singular, we can again perturb $\tilde{P}$ such that it is nonsingular as well), it is possible to partition it such that

$$U = \left[\begin{array}{c|c} T_1 & T_2 \end{array}\right] = \left[\begin{array}{cc|cc} T_{1u} & 0 & T_{2u} & 0 \\ \hline 0 & T_{1l} & 0 & T_{2l} \end{array}\right]$$

in which the number of rows of U is $2l$ and the upper and lower parts each have l rows, and where each column contains exactly one 1 and $2l-1$ zeros. If L_1 is some $2l \times 2l$

diagonal matrix then the product $T_1^T L_1 T_1$ is a diagonal matrix with the elements of L corresponding to the negative entries in $P - \tilde{P}^{-1}$ in its main diagonal. Similarly, if L_2 is some diagonal matrix of appropriate dimensions then the product $T_1 L_2 T_1^T$ is a diagonal $2l \times 2l$ matrix with zero entries everywhere except for the entries corresponding to the negative entries in $P - \tilde{P}^{-1}$. T_2 has the corresponding effect for the positive entries in $P - \tilde{P}^{-1}$.

Tedious calculations based on equations (5.27) and (5.29) show that D is of the form

$$D = \text{diag}_{1 \leq i \leq l} \left\{ \delta_i^2 \max\{q_i, \tilde{q}_i^{-1}\} + \max\{r_i, \tilde{r}_i^{-1}\} \right\} \tag{7.16}$$

since U rearranges the negative and positive diagonal elements of $P - \tilde{P}^{-1}$ into N_- and N_+, respectively, which means that N_- contains exactly those elements where $q_i - \tilde{q}_i^{-1} < 0, r_i - \tilde{r}_i^{-1} < 0$. With perfect knowledge about Δ it is then easy to choose $\Delta_c(\Delta)$ such that (7.15) is fulfilled, for instance as in (5.36) where the off-diagonal blocks are made to vanish, leaving a positive definite block diagonal matrix on the left hand side.

However, as stated above we are not scheduling the controller based on the exact Δ, but rather on the estimate $\hat{\Delta}$. This prompts us to define the diagonal matrices $\hat{D}, \hat{V}_-$ and $\hat{V}_+$ analogously with (7.16) and (5.29) (replacing δ_i with $\hat{\delta}_i$) and rewrite (7.15) with $\Delta_c(\hat{\Delta}) = N_- \hat{V}_-(\hat{\Delta})^T \hat{D}(\hat{\Delta})^{-1} \hat{V}_+(\hat{\Delta})$ instead of $N_- V_-(\Delta)^T D(\Delta)^{-1} V_+(\Delta)$:

$$\begin{bmatrix} N_+^{-1} & \hat{V}_+^T \hat{D}^{-1} \hat{V}_- N_- \\ N_- \hat{V}_-^T \hat{D}^{-1} \hat{V}_+ & -N_- \end{bmatrix} - \begin{bmatrix} V_+^T D^{-1} V_+ & V_+^T D^{-1} V_- N_- \\ N_- V_-^T D^{-1} V_+ & N_- V_-^T D^{-1} V_- N_- \end{bmatrix} > 0. \tag{7.17}$$

Let $\tilde{D}$ denote the matrix

$$\tilde{D} \triangleq \begin{bmatrix} \hat{\Delta} \\ I \end{bmatrix} \hat{D}^{-1} \begin{bmatrix} \hat{\Delta} \\ I \end{bmatrix}^T - \begin{bmatrix} \Delta \\ I \end{bmatrix} D^{-1} \begin{bmatrix} \Delta \\ I \end{bmatrix}^T.$$

This allows us to rewrite (7.17) as

$$\begin{bmatrix} T_2^T (N^{-1} - \begin{bmatrix} \Delta \\ I \end{bmatrix} D^{-1} \begin{bmatrix} \Delta \\ I \end{bmatrix}^T) T_2 & T_2^T \tilde{D} T_1 T_1^T N T_1 \\ T_1^T N T_1 T_1^T \tilde{D} T_2 & -N_-(N_-^{-1} + V_-^T D^{-1} V_-) N_- \end{bmatrix} > 0.$$

Some straightforward computations reveal that $\tilde{D}$ consists of diagonal submatrices

$$\tilde{D} = \begin{bmatrix} \tilde{D}_{11} & \tilde{D}_{12} \\ \tilde{D}_{12} & \tilde{D}_{22} \end{bmatrix}$$

given by

$$\begin{aligned}
\tilde{D}_{11} &= \mathrm{diag}_{1\le i\le l}\left\{\frac{(\hat{\delta}_i^2-\delta_i^2)r_{mi}}{(\hat{\delta}_i^2 q_{mi}+r_{mi})(\delta_i^2 q_{mi}+r_{mi})}\right\},\\
\tilde{D}_{12} &= \mathrm{diag}_{1\le i\le l}\left\{\frac{(\hat{\delta}_i\delta_i^2-\hat{\delta}_i^2\delta_i)q_{mi}+(\hat{\delta}_i-\delta_i)r_{mi}}{(\hat{\delta}_i^2 q_{mi}+r_{mi})(\delta_i^2 q_{mi}+r_{mi})}\right\},\\
\tilde{D}_{22} &= \mathrm{diag}_{1\le i\le l}\left\{\frac{(\delta_i^2-\hat{\delta}_i^2)q_{mi}}{(\hat{\delta}_i^2 q_{mi}+r_{mi})(\delta_i^2 q_{mi}+r_{mi})}\right\},
\end{aligned}$$

where

$$q_{mi} \triangleq \max\{q_i, \tilde{q}_i^{-1}\} \text{ and } r_{mi} \triangleq \max\{r_i, \tilde{r}_i^{-1}\}, 1 \le i \le l. \tag{7.18}$$

Applying the Schur complement lemma to the inequality above and simplifying gives the following equivalent matrix inequality:

$$\begin{aligned}
T_2^T\left(N^{-1}-\begin{bmatrix}\Delta\\ I\end{bmatrix}D^{-1}\begin{bmatrix}\Delta\\ I\end{bmatrix}^T\right)T_2 \\
+ T_2^T\tilde{D}T_1\left(N_-^{-1}+V_-^TD^{-1}V_-\right)^{-1}T_1^T\tilde{D}T_2 > 0.
\end{aligned} \tag{7.19}$$

Let

$$G_- \triangleq N^{-1}-\begin{bmatrix}\Delta\\ I\end{bmatrix}D^{-1}\begin{bmatrix}\Delta\\ I\end{bmatrix}^T \text{ and } G_+ \triangleq N^{-1}+\begin{bmatrix}\Delta\\ I\end{bmatrix}D^{-1}\begin{bmatrix}\Delta\\ I\end{bmatrix}^T,$$

such that (7.19) can be written as

$$T_2^TG_-T_2 + T_2^T\tilde{D}T_1(T_1^TG_+T_1)^{-1}T_1^T\tilde{D}T_2 > 0 \tag{7.20}$$

in which, using (7.16), it is seen that G_- and G_+ must be of the form

$$G_- = \begin{bmatrix}G_{-11} & G_{-12}\\ G_{-12} & G_{-22}\end{bmatrix} \text{ and } G_+ = \begin{bmatrix}G_{+11} & G_{+12}\\ G_{+12} & G_{+22}\end{bmatrix},$$

where

$$G_{\pm 11} = \mathrm{diag}_{1\le i\le l}\left\{\frac{1}{q_i-\tilde{q}_i^{-1}} \pm \frac{\delta_i^2}{\delta_i^2 q_{mi}+r_{mi}}\right\}, \tag{7.21}$$

$$G_{\pm 12} = \mathrm{diag}_{1\le i\le l}\left\{\pm\frac{\delta_i}{\delta_i^2 q_{mi}+r_{mi}}\right\}, \tag{7.22}$$

$$G_{\pm 22} = \mathrm{diag}_{1\le i\le l}\left\{\frac{1}{r_i-\tilde{r}_i^{-1}} \pm \frac{1}{\delta_i^2 q_{mi}+r_{mi}}\right\}. \tag{7.23}$$

Now Lemma B.4 (in the Appendix) implies that

$$T_2^T \tilde{D} T_1 (T_1^T G_+ T_1)^{-1} T_1^T \tilde{D} T_2 = T_2^T \Lambda T_2,$$

where

$$\Lambda = \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix},$$

which means that (7.20) is equivalent to

$$T_2^T (G_- + \Lambda) T_2 > 0. \tag{7.24}$$

Λ is diagonal and $\lambda_{1,i} = \lambda_{2,i} = 0$ for the i's for which $q_i > \tilde{q}_i^{-1}$ and $r_i > \tilde{r}_i^{-1}$. We also know from Lemma B.4 that $\Lambda_{1,i} = \tilde{d}_{12,i}^2 / g_{+22,i}$ for the i's for which $q_i > \tilde{q}_i^{-1}$ and $r_i < \tilde{r}_i^{-1}$ and $\Lambda_{i+l,i+l} = \tilde{d}_{12,i}^2 / g_{+11,i}$ for the i's for which $q_i > \tilde{q}_i^{-1}$ and $r_i < \tilde{r}_i^{-1}$ (lower-case letters with subscript i refer to the i'th diagonal element of the matrix denoted by the corresponding upper-case letter). Furthermore, the pre- and postmultiplication by T_2^T and T_2, respectively, eliminates the elements for which $q_i < \tilde{q}_i^{-1}$ and $r_i < \tilde{r}_i^{-1}$.

By a permutation (7.24) can then be seen to be equivalent to the fulfilment of a number of 1×1 or 2×2 matrix inequalities of the form

$$q_i > \tilde{q}_i^{-1}, r_i > \tilde{r}_i^{-1} \quad : \quad \begin{bmatrix} g_{-11,i} & g_{-12,i} \\ g_{-12,i} & g_{-22,i} \end{bmatrix} + \begin{bmatrix} \lambda_{1,i} & 0 \\ 0 & \lambda_{2,i} \end{bmatrix} > 0 \tag{7.25}$$

$$q_i > \tilde{q}_i^{-1}, r_i < \tilde{r}_i^{-1} \quad : \quad g_{-11,i} + \lambda_{1,i} > 0 \tag{7.26}$$

$$q_i < \tilde{q}_i^{-1}, r_i > \tilde{r}_i^{-1} \quad : \quad g_{-22,i} + \lambda_{2,i} > 0. \tag{7.27}$$

As mentioned above we have $\lambda_{1,i} = \lambda_{2,i} = 0$ for the i's for which $q_i > \tilde{q}_i^{-1}, r_i > \tilde{r}_i^{-1}$. Furthermore, the submatrix of G_- can be seen to be positive definite by combining equations (7.21)–(7.23) with the basic assumptions (7.8), which imply that $\delta^2 q_i + r_i > 0, \delta^2 \tilde{q}_i^{-1} + \tilde{r}_i^{-1} < 0$. Hence, (7.25) is automatically satisfied.

This leaves us with (7.26) and (7.27), which represent a set of simple scalar inequalities. By combining (7.21) and (7.23) with the definition of $\tilde{D}_{12}$ we can rewrite these inequalities as

$$\frac{1}{q_i - \tilde{q}_i^{-1}} - \frac{\delta_i^2}{\delta_i^2 q_i + \tilde{r}_i^{-1}} + \mu_1 \left(\frac{1}{r_i - \tilde{r}_i^{-1}} + \frac{1}{\delta_i^2 q_i + \tilde{r}_i^{-1}} \right)^{-1} > 0$$

for $q_i > \tilde{q}_i^{-1}, r_i < \tilde{r}_i^{-1}$, and

$$\frac{1}{r_i - \tilde{r}_i^{-1}} - \frac{\delta_i^2}{\delta_i^2 \tilde{q}_i^{-1} + r_i} + \mu_2 \left(\frac{1}{q_i - \tilde{q}_i^{-1}} + \frac{1}{\delta_i^2 \tilde{q}_i^{-1} + r_i} \right)^{-1} > 0$$

for $q_i < \tilde{q}_i^{-1}, r_i > \tilde{r}_i^{-1}$, in which

$$\begin{aligned}\mu_1 &\triangleq \frac{\delta_i\hat{\delta}_i(\delta_i-\hat{\delta}_i)q_i+(\hat{\delta}_i-\delta_i)\tilde{r}_i^{-1}}{(\hat{\delta}_i^2 q_i+\tilde{r}_i^{-1})(\delta_i^2 q_i+\tilde{r}_i^{-1})},\\ \mu_2 &\triangleq \frac{\delta_i\hat{\delta}_i(\delta_i-\hat{\delta}_i)\tilde{q}_i^{-1}+(\hat{\delta}_i-\delta_i)r_i}{(\hat{\delta}_i^2\tilde{q}_i^{-1}+r_i)(\delta_i^2\tilde{q}_i^{-1}+r_i)}.\end{aligned}$$

Finally, applying Lemma B.5 to each of these inequalities shows that they are satisfied if

$$\begin{aligned}&((1+\bar{e}_i)^2 q_i+\tilde{r}_i^{-1})^2(\tilde{q}_i^{-1}+\tilde{r}_i^{-1})(q_i+r_i)-\\ &\qquad((1+\bar{e}_i)q_i-\tilde{r}_i^{-1})^2\bar{e}_i^2(\tilde{r}_i^{-1}-r_i)(q_i-\tilde{q}_i^{-1})>0\end{aligned}$$

if $q_i > \tilde{q}_i^{-1}$ and $r_i < \tilde{r}_i^{-1}$ or

$$\begin{aligned}&((1+\bar{e}_i)^2\tilde{q}_i^{-1}+r_i)^2(q_i+r_i)(\tilde{q}_i^{-1}+\tilde{r}_i^{-1})>\\ &\qquad((1+\bar{e}_i)\tilde{q}_i^{-1}-r_i)^2\bar{e}_i^2(r_i-\tilde{r}_i^{-1})(\tilde{q}_i^{-1}-q_i)\end{aligned}$$

if $q_i < \tilde{q}_i^{-1}$ and $r_i > \tilde{r}_i^{-1}$. Hence, (7.24) will be fulfilled if for each $i = 1..n_u$ one of the three conditions in equations (7.9)-(7.13) is satisfied, which is what we wanted to show. ◁

Remark 7.2 Normally, if $P - \tilde{P}^{-1}$ loses rank, i.e. $(q_i - \tilde{q}_i^{-1})(r_i - \tilde{r}_i^{-1}) = 0$ for some i, it would be more efficient to construct an extended multiplier of lower dimension. However, to keep the proof simple, it was chosen to ignore this possibility, since there is no loss of generality in assuming that $P - \tilde{P}^{-1}$ is indeed invertible. If necessary, it is always possible (due to the strictness of the matrix inequalities) to perturb $P - \tilde{P}^{-1}$ in the right direction, such that there is no need to schedule according to the particular diagonal elements which are the cause of loss of rank, i.e. Δ_c will be independent of these elements.

The conditions will usually not hold automatically when just solving the LMI (5.21)–(5.23) so it is necessary to find some additional convex constraints which will guarantee the fulfilment of one of the three conditions.

Of course the convex constraints

$$\begin{bmatrix} R & I \\ I & \tilde{R}\end{bmatrix} > 0, \quad Q + \tilde{Q} > -2I \tag{7.28}$$

would guarantee (7.9), but would also be far too conservative.

The three conditions in (7.9), (7.10), and (7.12) discriminate between the signs of $q_i - \tilde{q}_i^{-1}$ and $r_i - \tilde{r}_i^{-1}$. If these have the same sign for all i, then there will be no scheduling function

and the controller will be robust. To assure (7.11) or (7.13) in the other cases, convex constraints of the form

$$R + (I + \mathcal{E})Q > 0 \tag{7.29}$$

$$\tilde{Q} + (I + \mathcal{E})\tilde{R} < 0 \tag{7.30}$$

$$\mathcal{E} = \mathrm{diag}_{1 \le i \le n_u} \epsilon_i, \quad \epsilon_i > 2\bar{e}_i + \bar{e}_i^2 \tag{7.31}$$

can be used.

The best way to synthesise the robust controller is probably to keep increasing the relevant elements of $\mathcal{E}$ until the solution of the LMI leads to a solution fulfilling the conditions (7.11) or (7.13). At each iteration, the multipliers should be kept small, for instance by minimising the trace of $R - \tilde{Q}$. This is a linear objective minimisation problem as discussed in Section 2.3.1. Alternatively the following lemmas can be used to provide sufficient condition beforehand, thereby only requiring one solution of the LMI.

Lemma 7.3 Assume that $ar \ge \tilde{r}^{-1} > r > 0$ *and* $\tilde{q}^{-1} < q < 0$*, and let* ϵ *be larger than the largest real root of the polynomial*

$$P_a(\epsilon_p) = a^2\epsilon_p^3 + \left(\bar{e}^2(a^2 - a^3 - a) + a^2 - a - 2a\bar{e}\right)\epsilon_p^2 + 2\bar{e}^2(a - a^3)\epsilon_p + \bar{e}^2(a - a^3 - a^2 + 1) \tag{7.32}$$

in ϵ_p *and assume also* $1 + \epsilon > (1 + \bar{e})^2$.

Then

$$r + (1 + \epsilon)q > 0, \quad \tilde{r}^{-1} + (1 + \epsilon)\tilde{q}^{-1} > 0 \tag{7.33}$$

implies

$$((1 + \bar{e})^2 q + \tilde{r}^{-1})^2(\tilde{q}^{-1} + \tilde{r}^{-1})(q + r) > ((1 + \bar{e})q - \tilde{r}^{-1})^2\bar{e}^2(\tilde{r}^{-1} - r)(q - \tilde{q}^{-1}) \tag{7.34}$$

Proof: Given in the appendix on page 177. $\lhd$

Lemma 7.3 can be used to assure (7.11). The convex constraint $a_i r_i > \tilde{r}_i^{-1}$ can be implemented as

$$\begin{bmatrix} R & G \\ G & \tilde{R} \end{bmatrix} > 0, \tag{7.35}$$

where

$$G = \mathrm{diag}_{1 \le i \le n_u}\{a_i^{-1/2}\}. \tag{7.36}$$

Lemma 7.3 can of course also be used for guaranteeing (7.13) with the substitution $q \leftrightarrow \tilde{q}_i^{-1}, \tilde{q}^{-1} \leftrightarrow q_i, r \leftrightarrow \tilde{r}_i^{-1}, \tilde{r}^{-1} \leftrightarrow r_i$. Unfortunately the constraint $a_i\tilde{r}_i^{-1} > r_i$ is not jointly convex in r and $\tilde{r}$. However, there seems to be no way around this kind of constraint. A convex conservative constraint is given by the following proposition

Proposition 7.4 Assume $a > 0, b > 0, d > 0$, and $\alpha > 0$. Then

$$\alpha^2 a + db < 2\alpha d \Rightarrow ab < d \tag{7.37}$$

By this proposition $a_i \tilde{r}_i^{-1} > r_i$ is implied by $\alpha_i^2 r_i + a_i \tilde{r}_i < \alpha_i a_i$.

Corollary 7.5 Consider the system (5.11) with $w_u = \Delta(t) z_u$ where $\Delta = \mathrm{diag}_{1 \le i \le n_u}\{\delta_i\}, |\delta_i| < 1$. Assume that $\Delta(t)$ is known except for some small uncertainty e, i.e. $\hat{\Delta}(t) = \Delta(t) + E(t), E = \mathrm{diag}_{1 \le i \le n_u}\{e_i\}, |e_i| < \bar{e}_i$.

Choose some $A_f = \mathrm{diag}_{1 \le i \le n_u}\{a_i\} > 0, \alpha = \mathrm{diag}_{1 \le i \le n_u}\{\alpha_i\} > 0$ and $\mathcal{E} = \mathrm{diag}_{1 \le i \le n_u}\{\epsilon_i\} > 0$ such that for each $i = 1..n_u$, ϵ_i is larger than the largest real root of the polynomial

$$P_{a_i}(\epsilon_p) = a_i^2 \epsilon_p^3 + \left(\bar{e}^2(a_i^2 - a_i^3 - a_i) + a_i^2 - a_i - 2a_i\bar{e}\right)\epsilon_p^2 + 2\bar{e}^2(a_i - a_i^3)\epsilon_p + \bar{e}^2(a_i - a_i^3 - a_i^2 + 1) \tag{7.38}$$

and larger than $\bar{e}_i^2 + 2\bar{e}_i$.

If there exist X, Y and

$$P = \begin{bmatrix} Q & S \\ S^* & R \end{bmatrix} \qquad \tilde{P} = \begin{bmatrix} \tilde{Q} & \tilde{S} \\ \tilde{S}^* & \tilde{R} \end{bmatrix} \tag{7.39}$$

satisfying the inequalities (5.21)–(5.23) with

$$\begin{gathered} Q = \mathrm{diag}_{1 \le i \le n_u}\{q_i\} < 0 \\ R = \mathrm{diag}_{1 \le i \le n_u}\{r_i\} > 0, S = 0 \\ R + (I + \mathcal{E})Q > 0 \\ \tilde{Q} = \mathrm{diag}_{1 \le i \le n_u}\{\tilde{q}_i\} < 0 \\ \tilde{R} = \mathrm{diag}_{1 \le i \le n_u}\{\tilde{r}_i\} > 0, \tilde{S} = 0 \\ \tilde{R}^{-1} + (I + \mathcal{E})\tilde{Q}^{-1} > 0 \\ \begin{bmatrix} R & A_f^{-1/2} \\ A_f^{-1/2} & \tilde{R} \end{bmatrix} > 0 \\ \alpha^2 R + A_f \tilde{R} < 2\alpha A_f. \end{gathered} \tag{7.40}$$

Then there exists a controller $K(\hat{\Delta})$ on the form (5.13) with $w_c = \Delta_c(\hat{\Delta}(t)) z_c(t)$ that yields robust quadratic performance with performance index P_p.

If A_f and α are chosen a priori then we are left with conditions that can easily be implemented in an LMI solver. If the $\bar{e}_i$'s are small then any large choice of A_f should work. Notice that if a_i is large and $\bar{e}_i$ is small then the roots of (7.38) are approximately equal to the roots of

$$\epsilon_p^3 + (1 - e_a)\epsilon_p^2 - 2e_a\epsilon_p - e_a \tag{7.41}$$

where $e_a = a_i \bar{e}_i^2$. This would suggest choosing a_i proportional to $\bar{e}_i^{-2}$.

Example 7.6 With $\bar{e}_i = 0.01$ and $a_i = 100$, $\epsilon_i \geq 0.11$ will suffice.

Remark 7.7 In LMI problems, slacking on one condition will often increase the range of the feasibility set drastically. So for instance by allowing a small degradation of performance, it will often by possible to greatly increase the robustness.

In [Bendtsen and Trangbæk, 2000a] an example is given of a control design for a simple system, where the above method significantly improves the performance.

Remark 7.8 Since the requirements for robustness given above are only related to the multipliers, the theory works equally well for continuous and discrete time.

7.2 Robust speed controller

In this section we will use the robust LPV synthesis presented in Section 7.1 to design a robust version of the speed controller presented in Section 6.4.

Using the MLP model obtained in Section 6.4, the first step is to obtain an estimate of the uncertainty on the residual gains. As discussed in Section 6.4.3 we consider two residual gains with sector bounds 0.0114 and 0.0092 respectively. We will assume that ω_r is known except for a measurement noise bounded by 0.5 rad/s. Using the method described in Section 6.3.3 we find that this results in uncertainties on these gains of 3.1% and 2.5% respectively.

The same synthesis as in Section 6.4.3 was performed but now the multipliers were chosen as

$$P = \begin{bmatrix} -(1+\epsilon_1)^{-1} r_1 & 0 & 0 & 0 \\ 0 & -(1+\epsilon_2)^{-1} r_2 & 0 & 0 \\ 0 & 0 & r_1 & 0 \\ 0 & 0 & 0 & r_2 \end{bmatrix}$$

$$\tilde{P} = \begin{bmatrix} -(1+\epsilon_1)\tilde{r}_1 & 0 & 0 & 0 \\ 0 & -(1+\epsilon_2) r_2 & 0 & 0 \\ 0 & 0 & \tilde{r}_1 & 0 \\ 0 & 0 & 0 & \tilde{r}_2 \end{bmatrix}.$$

By iteratively performing the synthesis, checking for robustness using (7.9)-(7.13), and increasing the corresponding ϵ if robustness was not achieved, a robust controller was designed. It was necessary to increase the bound the l_2-gain, γ, from 0.018 to 0.034.

The resulting multipliers were given by $r_1 = 3.8$, $r_2 = 3.0$, $\tilde{r}_1 = 4.0$, $\tilde{r}_2 = 10.7$, $\epsilon_1 = 0.15$, and $\epsilon_2 = 0.16$.

The same experiment as in Section 6.4.4 was performed now using the robust speed controller. The result is shown in Figure 7.1.

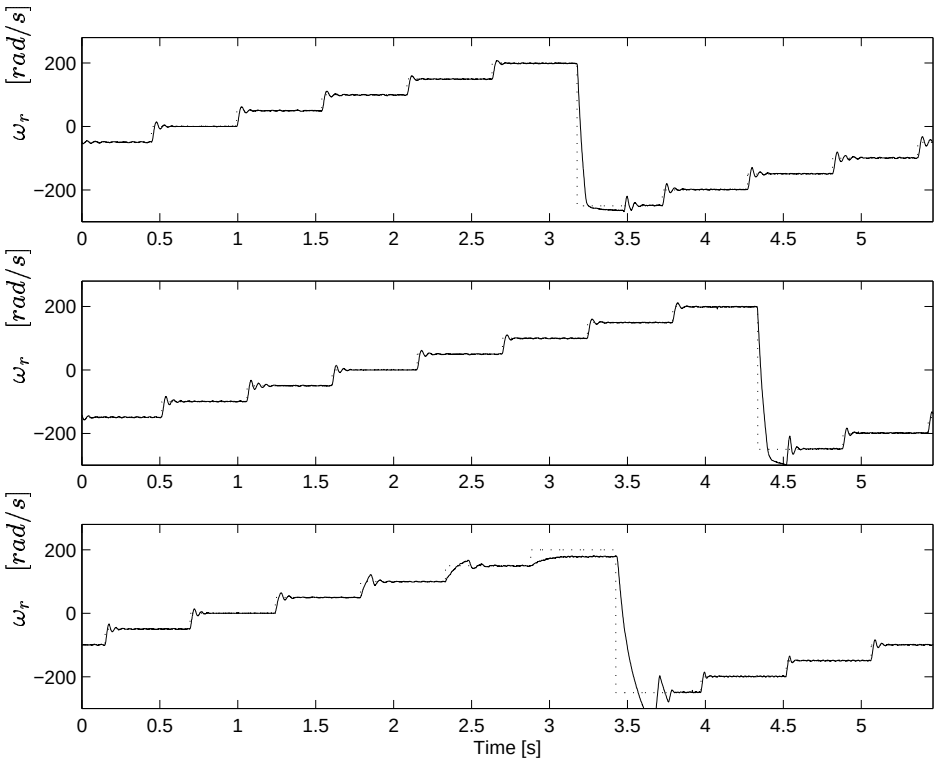


Figure 7.1: *Closed-loop experiment using the robust LPV speed controller. The speed reference is shown by the dotted line. The three figures show the behaviour with three different settings for the magnetising current. Top:* $i_{mR,ref} = 2.8A$. *Middle:* $i_{mR,ref} = 2.2A$. *Bottom:* $i_{mR,ref} = 1.0A$.

Comparing with Figure 6.14 it is seen that there are only small differences in the performance. If anything, the robust LPV controller is slightly harder tuned than the nominal LPV controller. This is due to the fact that the bound on the l_2-gain for the nominal LPV controller had to be increased due to numerical problems in solving the quadratic matrix inequality. This was not necessary for the robust LPV controller, since increasing the ϵ's improved the numerics.

7.3 Summary

In this chapter a novel method was presented for robust LPV design for systems where the time-varying parameters are uncertain. It was assumed that the time-varying gain was diagonal and that the diagonal elements were known except for some small uncertainty. A theorem was presented guaranteeing the existence of an LPV controller yielding robust quadratic performance under the condition of a new type of constraints on the multipliers. Even though these constraints were nonconvex, it was demonstrated how the theorem could be used as a basis for controller synthesis.

The method was applied to the design of a speed controller, which was robust to measurement errors of the speed. There was no significant difference compared to the controller designed in Chapter 6.

Chapter 8

CONCLUSIONS

This thesis demonstrated how the theory of linear parameter varying (LPV) systems could be applied to several subproblems in induction motor control resulting in a novel flux observer and novel current and speed controllers. Various contributions to the field of LPV control theory were also presented. This chapter summarises and concludes on the work presented in this thesis and gives recommendations for further work. Section 8.1 gives a summary of the thesis. Section 8.2 concludes on the work in general. Finally, Section 8.3 gives suggestions for further work.

8.1 Summary of the thesis

Chapter 3 described the dynamic model the induction motor. The part of the model describing the currents was written as a complex second order state space model with the shaft speed as a time-varying parameter. The laboratory setup on which experiments were performed was also discussed.

In Chapter 4 the rotor flux oriented control scheme for the induction motor was described. First it was discussed how the dynamical equations of the motor could be simplified by writing them in a reference system following the angle of the rotor flux. Then the rotor flux oriented control method was described. A short discussion of flux and speed observers was also given.

Chapter 5 reviewed a recently developed LPV synthesis method found in [Scherer, 2001],

the so-called full block S-procedure. An equivalent discrete-time version of the theory was developed. Then the special symmetry of the current equations of the induction motor was investigated. It was shown that controllers and observers for LPV systems with this type of symmetry can be assumed to have the same type of symmetry without loss of performance. Finally a discrete-time flux observer was designed and tested on the laboratory setup. A good performance was achieved with very little need for tuning.

In Chapter 6 it was described how the quasi-LPV approach allows the use of LPV theory for a very general class of nonlinear systems. The approach was then applied to the design of a stator current controller. Again, a good performance was achieved with very little need for tuning. A new method for transforming a neural network state space model into a quasi-LPV model suitable for control design was then presented. This method was then applied to the design of a speed controller based on a neural network model. The main purpose of the speed controller design was to demonstrate how a controller could be designed for a nonlinear system using a systematic approach from neural network modelling to LPV controller design.

In Chapter 7 a novel method was presented for robust LPV design for systems where the time-varying parameters are uncertain. The method was applied to the design of a speed controller, which was robust to measurement errors of the speed.

8.2 Conclusions

The following general conclusions on LPV controller design can be drawn from the work in this thesis:

- The LPV control theory provides a systematic way to approach controller design for nonlinear systems. Once an LPV model of the system to be controlled has been obtained, it is straightforward task to design the controller.
- Good results could be achieved without adding a large number of filters as is often necessary with robust $\mathcal{H}_\infty$ techniques such as μ-synthesis. Consequently the resulting controllers were of low order and very little tuning was needed.

With respect to the application of LPV control to induction motor control the following conclusions can be drawn:

- LPV controllers are computationally heavy considering the fast dynamics of small and medium sized induction motors. Thus actual industrial implementation may still be premature except for large motors. However, it is expected that the coming decade will bring faster and cheaper processors making implementation of LPV controllers a viable option.

- The described synthesis method provides numerical solutions to the control design rather than solutions given directly in terms of the motor parameters. This is a disadvantage of LPV methods compared to for instance feedback linearisation as in [Rasmussen et al., 1997] or backstepping design as in [Rasmussen et al., 1999].
- Good results could be obtained for the flux observer and the stator current controller using only a minimum amount of trial and error.

8.3 Recommendations for further work

The following topics would benefit from further examinations:

Robustness The robustness of the proposed flux observer and stator current controller to parameter variations should be further investigated, for instance through simulation studies. Since the LPV methods do not rely on inversion or exact cancellation of the nonlinearities, good robustness properties should be expected.

Numerics Constructing the LPV controllers involved solving the quadratic matrix inequality in Lemma 2.9. This can be numerically problematic even for small problems. A suggestion for improving the numerics was given in Lemma 5.12, but even better results could probably be achieved by finding an alternative to the construction given in the proof of Lemma 2.9.

Saturation The stator voltage saturates, constituting an input saturation for the stator current controller. In addition the stator current has to be limited to protect the motor. This constitutes an input saturation for the speed controller. These saturations were not included in the LPV models, mainly because it is difficult to do this in a meaningful way. In [Scorletti and Ghaoui, 1998] it is suggested to make the controller robust to the difference between the commanded input and the actual input. However, this will probably yield very conservative solutions.

In addition the work could be extended in the following directions:

Rates of change If bounds on the rates of variation of the time-varying parameters are known, then this can be exploited by making the Lyapunov matrix parameter-dependent, see [Rugh and Shamma, 2000] and the references therein. This can be used to reduce conservatism and thus obtain a better performance. For the presented flux observer and stator current controller, a bound on the rate of variation of the shaft speed could be achieved through assumptions on the load torque and by considering the bound on the stator current. It would probably also be fairly simple to obtain bounds on the rate of variations for the quasi-LPV model obtained from a neural network model, as described in Section 6.3, through a gridding of the parameter space.

Multipliers for robust LPV It would probably be a straightforward though tedious task to replace the requirements (7.6) with requirements similar to (7.2)-(7.5), i.e. allowing S to be diagonal. This would in some cases reduce conservatism greatly.

One unified controller It may be possible to avoid the cascaded controllers and instead attempt to design one large LPV controller for the entire system. The success of this attempt would probably depend on the choice of quasi-LPV representation for the system.

Neural network implementation The main source of computational complexity of the implemented LPV controllers was the computation of the scheduling function and the subsequent inversion due to D_{c22} being non-zero (see Section 5.4.4). As an alternative a neural network could be trained to mimic these functions. If a sufficiently small network yielding a good approximation in the entire range of operation could be obtained, then it would only be necessary to implement the multiplications and tangent hyperbolic functions of the neural network, which in many cases could reduce the computational complexity.

Appendix A

EXPERIMENTAL SETUP

In this thesis several experiments will be performed on a laboratory induction motor system illustrated in Figure A.1. The system was designed in the student project described in detail in [Skougaard and Wenzel, 1997].

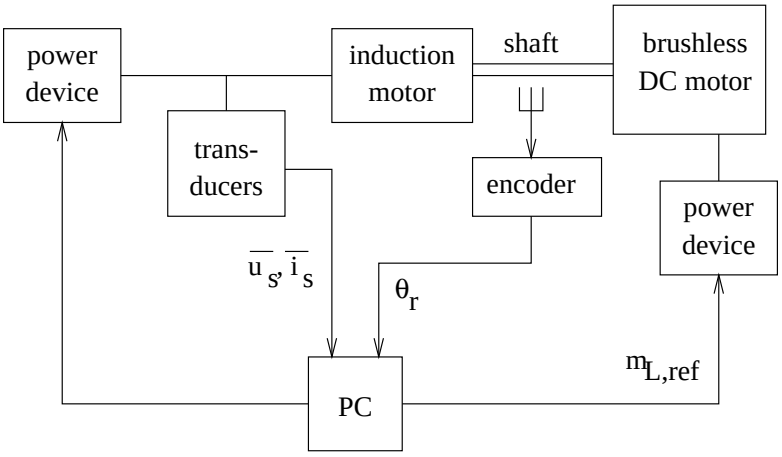


Figure A.1: *Laboratory induction motor system.*

The PC runs the control program to be tested providing a reference voltage for the induction motor power device and a torque reference to the DC motor power device. The brushless DC motor can be used to simulate a load torque on the shaft. Since the PC receives measurements of the rotor angular position from the encoder this can for instance be a position or speed dependent load.

A.1 Induction motor

The induction motor is a Bauknecht ATB drive with two pole pairs ($Z_p = 2$), a squirrel-cage rotor, and three star connected stator windings. Its nominal data are

Nominal power	1.5 kW
Nominal speed	1420 rpm
Nominal torque	10 Nm
Nominal current at 380 V	3.6 A

In [Rasmussen, 1995] the parameters of the induction motor were identified at standstill at $20°C$ under the assumption $L_s = L_r$ as

$$L_s = L_r = 0.352H, \quad L_m = 0.341H, \quad R_s = 5.0\Omega, \quad R_r = 3.3\Omega. \tag{A.1}$$

A.2 Power device

The power device is of the VSI-type described in Section 3.5. It is a VLT5003 from Danfoss A/S. It has been customised, so that the voltage references can be set via the ISA bus of the PC. The voltages are generated by pulse-width modulation (PWM) with a switching frequency of 15 kHz.

A.3 Current transducers

The three stator currents are measured by LEM-modules, which are essentially transformers. A current transducer is illustrated in Figure A.2. The stator current induces a smaller current in the LEM-module, which is converted to an equivalent voltage through a resistor. This voltage can then be measured by the AD-converter in the PC.

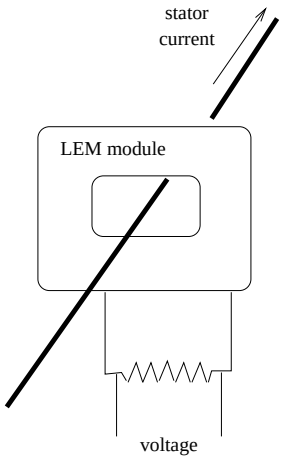


Figure A.2: *LEM module for stator current measurement.*

A.4 Voltage transducers

The stator voltages are measured through a star connected resistor bridge as shown in Figure A.3. Each leg of the bridge consists of two resistors providing a voltage division. The resistors are chosen so large that no significant current is drawn. The neutral is isolated and is therefore equivalent to the stator neutral. Since the stator voltages are generated by PWM, it is necessary to filter the measurements before feeding them to the AD-converter in the PC. A second-order analog hardware filter with a cut-off frequency of 1 kHz was chosen.

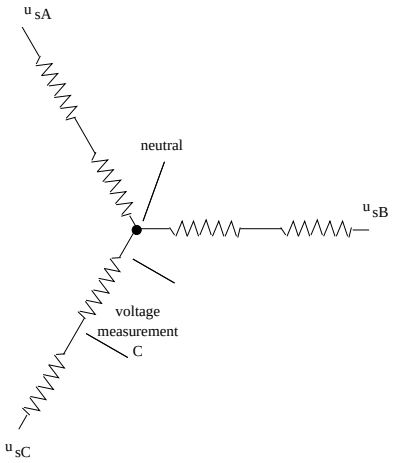


Figure A.3: *Resistor bridge for stator voltage measurement.*

A.5 Encoder

An encoder from Heidenhahn is connected to the shaft making it possible to measure the position or speed. The encoder essentially consists of a disc with alternately opaque and transparent sectors and two pairs of light transmitters and receivers. As the disc rotates, the light from the transmitters will either be interrupted be the disc or allowed to pass though to the receiver depending on the position. By measuring the light at two slightly offset positions it is possible not only to know the number of sectors which have passed but also the direction. Counting the number of sectors that have passed in a sampling period gives an estimate of the speed. This particular encoder has 1024 transparent sectors. At a sampling frequency of 3kHz this gives a resolution of the mechanical speed of approximately 4.6 rad/s at each sample. By low pass filtering a more accurate estimate can be obtained.

A.6 DC motor

The DC motor used to simulate a load torque is an Indramat permanent magnet motor from Mannesmann/Rexroth. It is driven by a power device containing a current controller receiving its references from a torque controller. The reference load torque is supplied as an analog signal from the DA-converter in the PC.

A.7 PC

The PC controlling the entire system is a Siemens-Nixdorff Scenic Pro M6 with a 200 MHz Pentium Pro processor and 32 MB of RAM. To allow measurements of stator currents and voltages, a DataTranslation DT2829 I/O-card has been inserted. The sampling frequency of the control system has been chosen as 3 kHz.

The control system is designed in MatLab Simulink [MathWorks, Inc., 1993] with Real Time Workshop [MathWorks, Inc., 1994] (RTW). The RTW converts the Simulink program to a C-program, which is then compiled to work with a VxWorks kernel and downloaded over the internet to the PC using a Tornado/VxWorks software package from WindRiver [WindRiver Systems, 1995]. When the program is running on the PC, data can be collected over the internet using the Stethoscope package from Real-Time Innovations [Real-Time Innovations,]. The system allows selected parameters of the control system to be changed in real-time from within the Simulink environment.

Appendix B

LEMMAS AND PROOFS

This appendix contains lemmas with no appropriate place in the main thesis as well as proofs which were deemed too long and tedious for the main thesis.

B.1 Lemmas

B.1.1 Lemma B.1: Matrix Inversion Lemma

Lemma B.1 (Matrix Inversion Lemma) e.g. [Helmersson, 1995]

Let A and D be non-singular. Then

$$(D + CAB)^{-1} = D^{-1} - D^{-1}C(BD^{-1}C + A^{-1})^{-1}BD^{-1} \tag{B.1}$$

B.1.2 Lemma B.2: Partitioned matrix inversion

Lemma B.2 (Partitioned matrix inversion) [Horn and Johnson, 1985]

Let A be a square matrix partitioned as

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix},$$

where A_{11} and A_{22} are square. If A_{11} is nonsingular then A is nonsingular if and only if the Schur complement $\Delta \triangleq A_{22} - A_{21}A_{11}^{-1}A_{12}$ *is nonsingular. Furthermore the inverse is then*

$$A^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1}A_{12}\Delta^{-1}A_{21}A_{11}^{-1} & -A_{11}^{-1}A_{12}\Delta^{-1} \\ -\Delta^{-1}A_{21}A_{11}^{-1} & \Delta^{-1} \end{bmatrix}.$$

Dually, if A_{22} is nonsingular then A is nonsingular if and only if the Schur complement $\hat{\Delta} \triangleq A_{11} - A_{12}A_{22}^{-1}A_{21}$ *is nonsingular. Furthermore the inverse is then*

$$A^{-1} = \begin{bmatrix} \hat{\Delta}^{-1} & -\hat{\Delta}^{-1}A_{12}A_{22}^{-1} \\ -A_{22}^{-1}A_{21}\hat{\Delta}^{-1} & A_{22}^{-1} + A_{22}^{-1}A_{21}\hat{\Delta}^{-1}A_{12}A_{22}^{-1} \end{bmatrix}.$$

B.1.3 Lemma B.3: Full rank multiplier extension

Lemma B.3 Let $X, Y, Z \in \mathbb{H}^{n\times n}$ and assume that Y, Z, and $X - Y^{-1}$ are non-singular and choose

$$\mathcal{X} = \begin{bmatrix} X & Z \\ Z^* & \left[Z^{-1}(X - Y^{-1})Z^{-*}\right]^{-1} \end{bmatrix}. \tag{B.2}$$

Then $\mathcal{X}$ is non-singular and

$$\mathcal{X}^{-1} = \begin{bmatrix} Y & * \\ * & * \end{bmatrix} \tag{B.3}$$

Proof: It is seen that Y is the Schur complement of $\left[Z^{-1}(X - Y^{-1})Z^{-*}\right]^{-1}$ in $\mathcal{X}$. Thus the nonsingularity of $\mathcal{X}$ is implied, and the upper left part of $\mathcal{X}^{-1}$ is Y as seen from the Partitioned matrix inversion lemma (B.2). $\triangleleft$

B.1.4 Lemma B.4

Lemma B.4 Consider the index sets $\mathcal{I}_-, \mathcal{I}_+, \mathcal{J}_-$ and $\mathcal{J}_+$ with cardinalities $n_I, l-n_I, n_J$ and $l - n_J$, respectively, defined such that $\mathcal{I}_- \cup \mathcal{I}_+ = \mathcal{J}_- \cup \mathcal{J}_+ = \{1, \ldots, l\} \subset \mathbb{N}$.

Let $e_i, 1 \leq i \leq l$, denote the i'th unit coordinate vector of $\mathbb{R}^l$. Let $T_1 = \begin{bmatrix} T_{1u} & 0 \\ 0 & T_{1l} \end{bmatrix} \in \mathbb{R}^{2l \times n_I + n_J}$ and $T_2 = \begin{bmatrix} T_{2u} & 0 \\ 0 & T_{2l} \end{bmatrix} \in \mathbb{R}^{2l \times 2l - (n_I + n_J)}$, where the columns of T_1 and T_2 are unit coordinate vectors of $\mathbb{R}^{2l}$, be defined by

$$T_{1u}^T e_i = 0 \Leftrightarrow i \in \mathcal{I}_+, \quad T_{1l}^T e_i = 0 \Leftrightarrow i \in \mathcal{J}_+,$$
$$T_{2u}^T e_i = 0 \Leftrightarrow i \in \mathcal{I}_-, \quad T_{2l}^T e_i = 0 \Leftrightarrow i \in \mathcal{J}_-.$$

Furthermore, let D and G be any two matrices such that

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \in \mathbb{R}^{2l \times 2l}$$

and

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{12} & G_{22} \end{bmatrix} \in \mathbb{R}^{2l \times 2l}$$

where each of the submatrices $D_{11}, D_{12}, \ldots, G_{22} \in \mathbb{R}^{l \times l}$ are diagonal.

Then, assuming that $T_1^T G T_1$ is invertible, we have

$$T_2^T D T_1 (T_1^T G T_1)^{-1} T_1^T D T_2 = T_2^T \Lambda T_2$$

where

$$\Lambda = \begin{bmatrix} \mathrm{diag}_{1 \leq i \leq l}\{\lambda_{1i}\} & 0 \\ 0 & \mathrm{diag}_{1 \leq i \leq l}\{\lambda_{2i}\} \end{bmatrix}.$$

Furthermore, those elements of Λ that do not vanish by the pre- and postmultiplication by T_2 are given by

$$\begin{aligned} i \in \mathcal{I}_+ \cap \mathcal{J}_+ &\Rightarrow \lambda_{1i} = \lambda_{2i} = 0, \\ i \in \mathcal{I}_+ \cap \mathcal{J}_- &\Rightarrow \lambda_{1i} = d_i^2 / g_{2i}, \\ i \in \mathcal{I}_- \cap \mathcal{J}_+ &\Rightarrow \lambda_{2i} = d_i^2 / g_{1i}, \end{aligned} \tag{B.4}$$

in which d_i, g_{1i} and g_{2i} are the i'th diagonal elements of D_{12}, G_{11} and G_{22}, respectively.

Proof: First of all it is noticed that $T_2^T D T_1 = T_2^T \begin{bmatrix} 0 & D_{12} \\ D_{12} & 0 \end{bmatrix} T_1$ since

$$\begin{bmatrix} T_{2u}^T & 0 \\ 0 & T_{2l}^T \end{bmatrix} \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} \begin{bmatrix} T_{1u} & 0 \\ 0 & T_{1l} \end{bmatrix} =$$
$$\begin{bmatrix} T_{2u}^T D_{11} T_{1u} & T_{2u}^T D_{12} T_{1l} \\ T_{2l}^T D_{12} T_{1u} & T_{2l}^T D_{22} T_{1l} \end{bmatrix} = \begin{bmatrix} 0 & T_{2u}^T D_{12} T_{1l} \\ T_{2l}^T D_{12} T_{1u} & 0 \end{bmatrix} \tag{B.5}$$

because T_1 and T_2 do not have non-zero entries on the same rows and hence these particular products must vanish.

Let us define

$$\Gamma \triangleq T_1(T_1^T G T_1)^{-1} T_1^T = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} \\ \Gamma_{12} & \Gamma_{22} \end{bmatrix},$$

where $\Gamma_{11}, \Gamma_{12}, \Gamma_{22} \in \mathbb{R}^{l \times l}$. It is deduced that each of the submatrices Γ_{11}, Γ_{12} and Γ_{22} are diagonal with

$$\begin{aligned} \Gamma_{11,i} &= g_{11,i}^{-1} \quad \text{for} \quad i \in \mathcal{I}_- \cap \mathcal{J}_+, \\ \Gamma_{11,i} &= 0 \quad \text{for} \quad i \in \mathcal{I}_+, \\ \Gamma_{22,i} &= g_{22,i}^{-1} \quad \text{for} \quad i \in \mathcal{I}_+ \cap \mathcal{J}_-, \\ \Gamma_{22,i} &= 0 \quad \text{for} \quad i \in \mathcal{J}_+, \\ \Gamma_{12,i} &= 0 \quad \text{for} \quad i \in \mathcal{I}_+ \cup \mathcal{J}_+. \end{aligned} \tag{B.6}$$

This is seen by noticing that $T_1^T G T_1$ is equivalent, via a permutation, to a block diagonal matrix where each sub-block is either of dimension 2×2 arising from elements corresponding to $i \in \mathcal{I}_- \cap \mathcal{J}_-$, or 1×1 arising from elements corresponding to $i \in (\mathcal{I}_+ \cap \mathcal{J}_-) \cup (\mathcal{I}_- \cap \mathcal{J}_+)$. Matrix inversion preserves this equivalence, and pre- and postmultiplying by T_1 and T_1^T then produces a matrix where the newly formed elements are rearranged back to the corresponding positions in G.

In light of (B.5) it can then be seen that

$$T_2^T D T_1 (T_1^T G T_1)^{-1} T_1^T D T_2 = \begin{bmatrix} T_{2u}^T D_{12} \Gamma_{22} D_{12} T_{2u} & T_{2u}^T D_{12} \Gamma_{12} D_{12} T_{2l} \\ T_{2l}^T D_{12} \Gamma_{12} D_{12} T_{2u} & T_{2l}^T D_{12} \Gamma_{11} D_{12} T_{2l} \end{bmatrix}.$$

The off-diagonal blocks in this matrix can furthermore be seen to be zero, since $\Gamma_{12,i} = 0$ for $i \in \mathcal{I}_+ \cup \mathcal{J}_+$ and pre- and postmultiplying by T_{2l}^T and T_{2u} eliminates the elements corresponding to $i \in \mathcal{I}_- \cup \mathcal{J}_-$. That leaves us with

$$\begin{aligned} T_2^T D T_1 (T_1^T G T_1)^{-1} T_1^T D T_2 &= \begin{bmatrix} T_{2u}^T D_{12} \Gamma_{22} D_{12} T_{2u} & 0 \\ 0 & T_{2l}^T D_{12} \Gamma_{11} D_{12} T_{2l} \end{bmatrix} = \\ &T_2^T \begin{bmatrix} D_{12} \Gamma_{22} D_{12} & 0 \\ 0 & D_{12} \Gamma_{11} D_{12} \end{bmatrix} T_2 = T_2^T \Lambda T_2. \end{aligned}$$

Looking at the diagonal elements, we see that pre- and postmultiplying by T_{2u}^T and T_{2u} eliminates the elements corresponding to $i \in \mathcal{I}_-$, while pre- and postmultiplying by T_{2l}^T and T_{2l} eliminates the elements corresponding to $\mathcal{J}_-$. This implies that only those diagonal elements in $D_{12}\Gamma_{11}D_{12}$ and $D_{12}\Gamma_{22}D_{12}$ corresponding to $i \in \mathcal{J}_+$ and $i \in \mathcal{I}_+$, respectively, will not vanish by this operation. Since Γ_{11} and Γ_{22} have the structures given in (B.6) we deduce that the non-vanishing elements in $T_2^T \Lambda T_2$ must be of the form (B.4), which completes the proof. $\lhd$

B.1.5 Lemma B.5

Lemma B.5 Consider the inequality

$$\frac{1}{q-\tilde{q}^{-1}} - \frac{\delta^2}{\delta^2 q+\tilde{r}^{-1}} + \mu\left(\frac{1}{r-\tilde{r}^{-1}} + \frac{1}{\delta^2 q+\tilde{r}^{-1}}\right)^{-1} > 0, \tag{B.7}$$

where $\mu = \frac{\delta\hat{\delta}(\delta-\hat{\delta})q+(\hat{\delta}-\delta)\tilde{r}^{-1}}{(\hat{\delta}^2 q+\tilde{r}^{-1})(\delta^2 q+\tilde{r}^{-1})}$. *Assuming that* $|\delta| < 1, |\delta-\hat{\delta}| < e, 0 > q > \tilde{q}^{-1}, (1+e^2)q + r > 0$, *and* $\tilde{r}^{-1} > r > 0$, *(B.7) is satisfied if*

$$((1+e)^2 q+\tilde{r}^{-1})^2(\tilde{q}^{-1}+\tilde{r}^{-1})(q+r) > ((1+e)q-r)^2 e^2(\tilde{r}^{-1}-r)(q-\tilde{q}^{-1}).$$

Proof: Rewriting (B.7) as a single fraction gives

$$\frac{t_1+t_2+t_3}{(q-\tilde{q}^{-1})(\hat{\delta}^2 q+\tilde{r}^{-1})^2(\delta^2 q+\tilde{r}^{-1})(\delta^2 q+r)} > 0,$$

where

$$\begin{aligned} t_1 &= (\hat{\delta}^2 q+\tilde{r}^{-1})^2(\delta^2 q+\tilde{r}^{-1})(\delta^2 q+r), \\ t_2 &= -(q-\tilde{q}^{-1})(\hat{\delta}^2 q+\tilde{r}^{-1})^2(\delta^2 q+r)\delta^2, \\ t_3 &= (\delta\hat{\delta}(\delta-\hat{\delta})q+(\hat{\delta}-\delta)r)^2(r-\tilde{r}^{-1})(q-\tilde{q}^{-1}). \end{aligned}$$

It is seen that the denominator is positive since $q > \tilde{q}^{-1}$ and $q+\tilde{r}^{-1} > q+r > 0$ by assumption. Furthermore it is obvious that the inequality is hardest to satisfy for $\delta \to 1$, so we will let $\delta = 1$. Similarly, the worst case for $\hat{\delta}$ is for $\hat{\delta} \to \delta + e$, so we will let $\hat{\delta} = 1 + e$ and examine the numerator inequality $t_1 + t_2 > -t_3$ or:

$$\begin{aligned} &((1+e)^2 q+\tilde{r}^{-1})^2((q+\tilde{r}^{-1})(q+r) - (q-\tilde{q}^{-1})(q+r)) \\ &\qquad > ((1+e)q-r)^2 e^2(\tilde{r}^{-1}-r)(q-\tilde{q}^{-1}). \end{aligned} \tag{B.8}$$

In other words, if (B.8) is satisfied, then (B.7) will be satisfied as well. It is now easy to see that (B.8) can be simplified to

$$((1+e)^2 q+\tilde{r}^{-1})^2(\tilde{q}^{-1}+\tilde{r}^{-1})(q+r) > ((1+e)q-r)^2 e^2(\tilde{r}^{-1}-r)(q-\tilde{q}^{-1})$$

which was what we wanted to show. $\lhd$

B.2 Proofs

B.2.1 Proof of Lemma 2.9

Proof: [Scherer, 2001].

($\Rightarrow$): First observe that

$$\operatorname{im}\begin{bmatrix} I \\ A^*XB+C \end{bmatrix}^{\perp} = \operatorname{im}\begin{bmatrix} -(A^*XB+C)^* \\ I \end{bmatrix} \tag{B.9}$$

for any X. Using this with Lemma 2.7 we see that (2.11) is equivalent to

$$\begin{bmatrix} -(A^*XB+C)^* \\ I \end{bmatrix}^* P^{-1} \begin{bmatrix} -(A^*XB+C)^* \\ I \end{bmatrix} > 0. \tag{B.10}$$

Pre- and postmultiplication by $B^{\vdash}$ in (2.11), and by $A^{\vdash}$ in (B.10) we arrive at (2.12) and (2.13).

($\Leftarrow$): First define the nonsingular matrices K and L so that $AK = \begin{bmatrix} A_1 & 0 \end{bmatrix}$ and $BL = \begin{bmatrix} B_1 & 0 \end{bmatrix}$, where A_1 and B_1 have full column rank. With

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \triangleq K^*CL \tag{B.11}$$

(2.11) can be written as

$$\begin{bmatrix} I & 0 \\ 0 & I \\ A_1^*XB_1+D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}^* \underbrace{\begin{bmatrix} L & 0 \\ 0 & K^{-*} \end{bmatrix}^* P \begin{bmatrix} L & 0 \\ 0 & K^{-*} \end{bmatrix}}_{\Pi} \begin{bmatrix} I & 0 \\ 0 & I \\ A_1^*XB_1+D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} < 0.$$

With the definitions

$$R \triangleq \begin{bmatrix} I & 0 \\ 0 & 0 \\ 0 & I \\ D_{21} & 0 \end{bmatrix}, S \triangleq \begin{bmatrix} 0 \\ I \\ D_{12} \\ D_{22} \end{bmatrix}, T \triangleq \begin{bmatrix} -D_{21}^* \\ -D_{22}^* \\ 0 \\ I \end{bmatrix},$$

$$Z \triangleq A_1^*XB_1 + D_{11} \in \mathbb{C}^{r_Z \times c_Z},$$

this can be written as

$$\begin{bmatrix} R\begin{bmatrix} I \\ Z \end{bmatrix} & S \end{bmatrix}^* \Pi \begin{bmatrix} R\begin{bmatrix} I \\ Z \end{bmatrix} & S \end{bmatrix} < 0. \tag{B.12}$$

Since A_1 and B_1 are of full column rank, the mapping $X \to A_1^*XB_1 + D_{11}$ is surjective, and we can now consider Z in (B.12) as the unrestricted unknown. By observing $BLL^{-1}B^{\vdash} = BB^{\vdash} = 0$ we conclude that $L^{-1}B^{\vdash} = \begin{bmatrix} 0 \\ * \end{bmatrix}$. Using this along with $C = K^{-*}DL^{-1}$ and $P = \begin{bmatrix} L^{-1} & 0 \\ 0 & K^* \end{bmatrix}^* \Pi \begin{bmatrix} L^{-1} & 0 \\ 0 & K^* \end{bmatrix}$ we can rewrite (2.12) as

$$S^*\Pi S < 0. \tag{B.13}$$

Similarly, (2.13) can be written as

$$T^*\Pi^{-1}T > 0. \tag{B.14}$$

Performing the multiplication in (B.12) and using a Schur complement argument we find that it is equivalent to

$$\begin{bmatrix} I \\ Z \end{bmatrix}^* \left(R^*\Pi R - R^*\Pi S(S^*\Pi S)^{-1}S^*\Pi R \right) \begin{bmatrix} I \\ Z \end{bmatrix} < 0. \tag{B.15}$$

We now only need to show that this has a solution in Z. Since

$$\operatorname{im} T = \operatorname{im} \begin{bmatrix} R & S \end{bmatrix}^{\perp}, \tag{B.16}$$

Lemma 2.7 gives

$$\begin{aligned} \operatorname{in}_- \left(\begin{bmatrix} R & S \end{bmatrix}^* \Pi \begin{bmatrix} R & S \end{bmatrix} \right) = \\ \operatorname{in}_-(\Pi) - \operatorname{in}_-(T^*\Pi^{-1}T) = \operatorname{in}_-(P) - 0 = m. \end{aligned} \tag{B.17}$$

Applying Lemma 2.6 then yields

$$\begin{aligned} \operatorname{in}_- \left(R^*\Pi R - R^*\Pi S(S^*\Pi S)^{-1}S^*\Pi^* R \right) = \\ \operatorname{in}_- \left(\begin{bmatrix} R & S \end{bmatrix}^* \Pi \begin{bmatrix} R & S \end{bmatrix} \right) - \operatorname{in}_-(S^*\Pi S) = m - (m - c_Z) = c_Z. \end{aligned} \tag{B.18}$$

It is thus possible to find a nonsingular $Z_1 \in \mathbb{C}^{c_Z \times c_Z}$ and a Z_2 fulfilling

$$\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}^* \left(R^*\Pi R - R^*\Pi S(S^*\Pi S)^{-1}S^*\Pi R \right) \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} < 0. \tag{B.19}$$

A solution to (B.15) is then $Z = Z_2 Z_1^{-1}$. $\lhd$

B.2.2 Proof of Lemma 5.17

Proof:

Denote M as

$$M = \begin{bmatrix} M_r & -M_i \\ M_i & M_r \end{bmatrix}$$

and partition M^{-1} as

$$M^{-1} = \begin{bmatrix} \hat{M}_{r1} & -\hat{M}_{i1} \\ \hat{M}_{i2} & \hat{M}_{r2} \end{bmatrix}$$

comformably to the partitioning of M. From $MM^{-1} = I$ we have

$$\begin{aligned} M_r\hat{M}_{r1} - M_i\hat{M}_{i2} &= I, && \text{(B.20)}\\ M_i\hat{M}_{r1} + M_r\hat{M}_{i2} &= 0, && \text{(B.21)}\\ M_r\hat{M}_{r2} - M_i\hat{M}_{i1} &= I, && \text{(B.22)}\\ -M_i\hat{M}_{r2} - M_r\hat{M}_{i1} &= 0. && \text{(B.23)} \end{aligned}$$

Define

$$X \triangleq \hat{M}_{i2} - \hat{M}_{i1}, \qquad Y \triangleq \hat{M}_{r1} - \hat{M}_{r2}.$$

Subtracting (B.22) from (B.20) yields

$$\begin{bmatrix} M_r & M_i \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = 0. \tag{B.24}$$

Adding (B.21) and (B.23) yields

$$\begin{bmatrix} M_r & M_i \end{bmatrix} \begin{bmatrix} Y \\ -X \end{bmatrix} = 0. \tag{B.25}$$

Now (B.24) and (B.25) imply

$$\begin{bmatrix} M_r & -M_i \\ M_i & M_r \end{bmatrix} \begin{bmatrix} Y & X \\ X & -Y \end{bmatrix} = 0.$$

Since M is nonsingular this implies $X = 0$ and $Y = 0$ or equivalently $\hat{M}_{i2} = \hat{M}_{i1}$ and $\hat{M}_{r1} = \hat{M}_{r2}$, which is exactly what we needed to show. $\lhd$

B.2.3 Proof of Lemma 5.20

Proof: From the proof of Lemma 5.17 we have

$$M^{-1} = \begin{bmatrix} \hat{M}_r & -\hat{M}_i \\ \hat{M}_i & \hat{M}_r \end{bmatrix}.$$

From $MM^{-1} = I$ and $M^{-1}M = I$ we have

$$\begin{aligned} M_r\hat{M}_r - M_i\hat{M}_i &= I, && \text{(B.26)}\\ M_i\hat{M}_r + M_r\hat{M}_i &= 0, && \text{(B.27)}\\ \hat{M}_rM_r - \hat{M}_iM_i &= I, && \text{(B.28)}\\ \hat{M}_iM_r + \hat{M}_rM_i &= 0. && \text{(B.29)} \end{aligned}$$

Then we immediately have

$$T_{\mathcal{C}}(M)\,T_{\mathcal{C}}\left(M^{-1}\right) = (M_r + jM_i)(\hat{M}_r + j\hat{M}_i) = \\ M_r\hat{M}_r - M_i\hat{M}_i + j(M_i\hat{M}_r + M_r\hat{M}_i) = I + 0 \quad \text{(B.30)}$$

as well as

$$T_{\mathcal{C}}(M)\,T_{\mathcal{C}}\left(M^{-1}\right) = (\hat{M}_r + j\hat{M}_i)(M_r + jM_i) = \\ \hat{M}_rM_r - \hat{M}_iM_i + j(\hat{M}_iM_r + \hat{M}_rM_i) = I + 0 \quad \text{(B.31)}$$

which are equivalent to (5.66). $\lhd$

B.2.4 Proof of Lemma 7.3

Proof: Due to (7.33), it is then seen that

$$((1+\bar{e})^2q + \tilde{r}^{-1})^2(q+r)\frac{\epsilon}{1+\epsilon}\tilde{r}^{-1} \\ > ((1+\bar{e})q - \tilde{r}^{-1})^2\bar{e}^2(\tilde{r}^{-1} - r)\left(q + \frac{1}{1+\epsilon}\tilde{r}^{-1}\right) \quad \text{(B.32)}$$

implies (7.34). Since

$$\frac{(1+\bar{e})^2q + \tilde{r}^{-1}}{q + \frac{1}{1+\epsilon}\tilde{r}^{-1}} = (1+\epsilon)\frac{(1+\bar{e})^2q + \tilde{r}^{-1}}{(1+\epsilon)q + \tilde{r}^{-1}} > 1 + \epsilon$$

we observe that (B.32) is implied by

$$((1+\bar{e})^2q + \tilde{r}^{-1})(q+r)\tilde{r}^{-1}\epsilon > \bar{e}^2((1+\bar{e})q - \tilde{r}^{-1})^2(\tilde{r}^{-1} - r) \quad \text{(B.33)}$$

which is again implied by (due to (7.33))

$$\left(\frac{(1+\bar{e})^2}{1+\epsilon}(-r) + \tilde{r}^{-1}\right)\frac{\epsilon^2}{1+\epsilon}r\tilde{r}^{-1} \geq \bar{e}^2\left(\frac{1+\bar{e}}{1+\epsilon}r + \tilde{r}^{-1}\right)^2(\tilde{r}^{-1} - r). \quad \text{(B.34)}$$

With $b \triangleq \tilde{r}^{-1}/r$ this is equivalent to

$$b\left(b(1+\epsilon) - (1+\bar{e})^2\right)\epsilon^2 - \bar{e}^2\left(b(1+\epsilon) + 1\right)^2(b-1) \geq 0. \quad \text{(B.35)}$$

The left hand side can be written as a third order polynomial in ϵ with the coefficient for the third order term being positive:

$$P_b(\epsilon) = b^2\epsilon^3 + \left[\bar{e}^2(b^2 - b^3 - b) + b^2 - b - 2b\bar{e}\right]\epsilon^2 \\ + 2\bar{e}^2(b - b^3)\epsilon + \bar{e}^2(b - b^3 - b^2 + 1). \quad \text{(B.36)}$$

Then (B.35) is equivalent to $P_b(\epsilon) \geq 0$. Now choose ϵ larger than the largest real root of $P_a(\epsilon)$. Then $P_b(\epsilon)$ is positive for $b = a$. All we have to show now is that $P_b(\epsilon) > 0$ for $1 > b > a$. With ϵ fixed, P_b is a third degree polynomial in $b_n \triangleq b - 1$ with a negative coefficient for the third order term and a positive constant term. The only way that P_b can be negative for some $0 > b_n > a - 1$ is then if all the roots of P_b in b_n are real and positive. This requires the coefficients for the second order and first order terms being positive and negative respectively. But the coefficient of the first order term can be shown to be positive. $\lhd$

BIBLIOGRAPHY

[Apkarian, 1997] Apkarian, P. (1997). On the discretization of LMI-synthesized linear parameter-varying controllers. *Automatica*, 33(4):655–661.

[Apkarian and Gahinet, 1995] Apkarian, P. and Gahinet, P. (1995). A convex characterization of gain-scheduled $\mathcal{H}_\infty$ controllers. *IEEE Transactions on Automatic Control*, 40(5):853–864.

[Apkarian et al., 1995] Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example. *Automatica*, 31(9):1251–1262.

[Becker et al., 1993] Becker, G., Packard, A., Philbrick, D., and Balas, G. (1993). Control of parametrically-dependent linear systems: A single quadratic Lyapunov approach. *Proc. of the American Control Conference*, pages 2795–2799.

[Benchaib and Edwards, 2000] Benchaib, A. and Edwards, C. (2000). Nonlinear sliding mode control of an induction motor. *International Journal of Adaptive Control and Signal Processing*, 14(2-3):201–221.

[Bendtsen, 1999] Bendtsen, J. D. (1999). *Practical application of neural networks in state space control*. PhD thesis, Aalborg University.

[Bendtsen and Trangbæk, 2000a] Bendtsen, J. D. and Trangbæk, K. (2000a). Robust quasi-LPV control based on neural state space models. *submitted in 2000 to IEEE Transactions on Neural Networks*.

[Bendtsen and Trangbæk, 2000b] Bendtsen, J. D. and Trangbæk, K. (2000b). Transformation of neural state space models into LFT models for robust control design. *Proc. of the Sixth International Conference on Control, Automation, Robotics and Vision*.

[Bendtsen and Trangbæk, 2001a] Bendtsen, J. D. and Trangbæk, K. (2001a). Comments on "Lur'e systems with multilayer perceptron and recurrent neural networks: Absolute stability and dissipativity". *Transactions on Automatic Control*, 46(4).

[Bendtsen and Trangbæk, 2001b] Bendtsen, J. D. and Trangbæk, K. (2001b). Discrete-time lpv current control of an induction motor. *submitted in May 2001 to IEEE Control Systems Technology*.

[Beran and Grigoriadis, 1996] Beran, E. and Grigoriadis, K. (1996). A combined alternating projections and semidefinite programming algorithm for low-order control design. *Proceedings of the 13th Triennial World Congress*, C:85–90.

[Billings et al., 1992] Billings, S., Jamaluddin, H., and Chen, S. (1992). Properties of neural networks with applications to modelling nonlinear dynamical systems. *International Journal of Control*, 55(1):193–224.

[Blaabjerg et al., 1996] Blaabjerg, F., Pedersen, J. K., and Kaźmierkowski, M. P. (1996). DSP based current regulated PWM inverter-fed induction motor drive without speed sensor. *ISIE '96. Proceedings of the IEEE International Symposium on Industrial Electronics*, 2(2):659–664.

[Boyd et al., 1994] Boyd, S., Ghaoui, L. E., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*. Society for for Industrial and Applied Mathematics (SIAM), Philadelphia, Pennsylvania.

[Chai et al., 1996] Chai, J. S., Tan, S., and Hang, C. C. (1996). Gain scheduling control of nonlinear plant using a RBF neural network. *Proc. of the IEEE International Symposium on Intelligent Control*.

[Chen and Khalil, 1995] Chen, F. C. and Khalil, H. K. (1995). Adaptive control of a class of nonlinear discrete-time systems using neural networks. *IEEE Transactions on Automatic Control*, 40(9):791–801.

[Chilali et al., 1999] Chilali, M., Gahinet, P., and Apkarian, P. (1999). Robust pole placement in LMI regions. *IEEE Transactions on Automatic Control*, 44(12):2257–2270.

[Choi and Sul, 1998] Choi, J.-W. and Sul, S.-K. (1998). Generalized solution of minimum time current control in three-phase balanced systems. *IEEE Transactions on Industrial Electronics*, 45(5):738–44.

[Darengosse et al., 2000] Darengosse, C., Chevrel, P., and Doeuff, R. L. (2000). A linear parameter-varying flux observer: design and experimentation. *IEEE 31st Annual Power Electronics Specialists Conference Proceedings*, 3:1112–1117.

[Doyle et al., 1991] Doyle, J., Packard, A., and Zhou, K. (1991). Review of LFTs, LMIs and μ. *Proceedings of the 30th IEEE Conference on Decision and Control*, pages 1227–1232.

[Doyle et al., 1989] Doyle, J. C., Glover, K., Khargonekar, P. P., and Francis, B. A. (1989). State-space solutions to $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–846.

[Elbert, 1984] Elbert, T. F. (1984). *Estimation and Control of Systems*. Van Nostrand Reinhold Company Inc., New York.

[Franklin et al., 1994] Franklin, G. F., Powell, J. D., and Emami-Naeini, A. (1994). *Feedback Control of Dynamic Systems, 3rd Ed.* Addison-Wesley.

[French and Rogers, 1998] French, M. and Rogers, E. (1998). Input/output stability theory for direct neuro-fuzzy controllers. *IEEE Transactions on Fuzzy Systems*, 6(3):331–345.

[Gahinet and Apkarian, 1994] Gahinet, P. and Apkarian, P. (1994). A linear matrix inequality approach to $\mathcal{H}_\infty$ control. *International Journal of Robust and Nonlinear Control*, 4:421–448.

[Gahinet et al., 1995] Gahinet, P., Nemirovski, A., Laub, A., and Chilali, M. (1995). *LMI Toolbox*. MatLab, The MathWorks Inc.

[Gorter, 1997] Gorter, R. J. (1997). *Grey-box Identification of Induction Machines*. PhD thesis, Technische Universiteit Eindhoven.

[Grigoriadis and Skelton, 1996] Grigoriadis, K. M. and Skelton, R. E. (1996). Low-order control design for LMI problems using alternating projection methods. *Automatica*, 32:1117–1125.

[He et al., 1998] He, S., Reif, K., and Unbehauen, R. (1998). A neural approach for control of nonlinear systems with feedback linearization. *IEEE Transactions on Neural Networks*, 9(6):1409–1421.

[Helmersson, 1995] Helmersson, A. (1995). *Methods for Robust Gain Scheduling*. PhD thesis, Linköping University.

[Helmersson, 1999] Helmersson, A. (1999). IQC synthesis based on inertia constraints. *Proc. 14th IFAC World Congres, Bejing, China*, pages 163–168.

[Horn and Johnson, 1985] Horn, R. A. and Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press, New York, USA.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feed-forward networks are universal approximators. *Neural Networks*, 2:359–366.

[Iwasaki and Skelton, 1994] Iwasaki, T. and Skelton, R. E. (1994). All controllers for the general $\mathcal{H}_\infty$ control problem: LMI existence conditions and state space formulas. *Automatica*, 30(8):1307–1317.

[Jansen and Lorenz, 1992] Jansen, P. and Lorenz, R. (1992). A physically insightful approach to the design and accuracy assessment of flux observers for field oriented induction machine drives. *Research report, Wisconsin electric machines and power electronics consortium*.

[Jung et al., 1997] Jung, J., Lim, S., and Nam, K. (1997). PI type decoupling control scheme for high speed operation of induction motors. *PEPSC97 Record. 28th Annual IEEE Power Electronics Specialists Conference*, 2(2):1082–1085.

[Kaźmierkowski and Dzieniakowski, 1994] Kaźmierkowski, M. P. and Dzieniakowski, M. A. (1994). Review of current regulators for three-phase PWM inverters. *Proc. of the 20th International Conference on Industrial Electronics, Control and Instrumentation*, 1:567–575.

[Kaźmierkowski and Malesani, 1998] Kaźmierkowski, M. P. and Malesani, L. (1998). Current control techniques for three-phase voltage source PWM converters - a survey. *IEEE Transactions on Industrial Electronics*, 45(5):691–703.

[Kaźmierkowski and Tunia, 1994] Kaźmierkowski, M. P. and Tunia, H. (1994). *Automatic Control of Converter-fed Drives*. Polish Scientific Publishers PWN Ltd, Warszawa.

[Kim et al., 1997] Kim, Y. H., Lewis, F. L., and Abdallah, C. T. (1997). A dynamic recurrent neural-network-based adaptive observer for a class of nonlinear systems. *Automatica*, 33(8):1539–1543.

[Korbicz and Janczak, 1996] Korbicz, J. and Janczak, A. (1996). A neural network approach to identification of structural systems. *ISIE '96. Proceedings of the IEEE International Symposium on Industrial Electronics*, 1:98–103.

[Kubota et al., 1993] Kubota, H., Matsuse, K., and Nakano, T. (1993). DSP-based speed adaptive flux observer of induction motor. *IEEE Trans. Ind. Appl.*, vol. 29 no. 2:344–348.

[Lee et al., 1996] Lee, T. T., Jeng, J. T., and Shih, C. L. (1996). Using neural networks to improve gain scheduling techniques for linear parameter-varying systems. *Proc. of the International Workshop on Advanced Motion Control*.

[Leonhard, 1990] Leonhard, W. (1990). *Control of Electrical Drives*. Springer-Verlag, Berlin, Heidelberg.

[Levin and Narendra, 1993] Levin, A. U. and Narendra, K. S. (1993). Control of nonlinear dynamical systems using neural network: Controllability and stabilization. *IEEE Transactions on Neural Networks*, 4(2):192–205.

[Lightbody and Irwin, 1996] Lightbody, G. and Irwin, G. W. (1996). Multi-layer perceptron based modelling of nonlinear systems. *Fuzzy Sets and Systems*, (79):93–112.

[Liu et al., 1998] Liu, Y.-H., Chen, C.-L., and Tu, R.-J. (1998). A novel space-vector current regulation scheme for a field-oriented-controlled induction motor drive. *IEEE Transactions on Industrial Electronics*, 45(5):730–737.

[Lu and Basar, 1998] Lu, S. and Basar, T. (1998). Robust nonlinear system identification using neural network models. *IEEE Transactions on Neural Networks*, (9):407–429.

[Manes et al., 1996] Manes, C., Parasiliti, F., and Tursini, M. (1996). DSP based field-oriented control of induction motor with a nonlinear state observer. *Power Electronics Specialists Conference Proceedings*, 2:1254–1259.

[Marino et al., 1996] Marino, R., Peresada, S., and Tomei, P. (1996). Adaptive observer-based control of induction motors with unknown rotor resistance. *International Journal of Adaptive Control and Signal Processing*, 10:345–363.

[Martin and Rouchon, 2000] Martin, P. and Rouchon, P. (2000). Two simple flux observers for induction motors. *International Journal of Adaptive Control and Signal Processing*, 14(2-3):171–176.

[MathWorks, Inc., 1993] MathWorks, Inc. (1993). *SIMULINK User's guide*. MatLab, The MathWorks Inc.

[MathWorks, Inc., 1994] MathWorks, Inc. (1994). *Real-Time Workshop: User's guide*. MatLab, The MathWorks Inc.

[Mears and Polycarpou, 1999] Mears, M. J. and Polycarpou, M. M. (1999). Stable local neural control of uncertain systems. *Proc. of the IEEE International Conference on Decision and Control*.

[Megretski and Rantzer, 1997] Megretski, A. and Rantzer, A. (1997). System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830.

[Nemirovski and Gahinet, 1994] Nemirovski, A. and Gahinet, P. (1994). The projective method for solving linear matrix inequalities. *Proc. of the American Control Conference*, pages 840–844.

[Nesterov and Nemirovski, 1994] Nesterov, Y. and Nemirovski, A. (1994). *Interior–Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia.

[Nikoukhah et al., 1995] Nikoukhah, R., Delebecque, F., and Ghaoui, L. E. (1995). LMITOOL: a Package for LMI Optimization in Scilab, User's Guide . Technical report.

[Nilsen and Kaźmierkowski, 1992] Nilsen, R. and Kaźmierkowski, M. (1992). New reduced-order observer with parameter adaptation for flux estimation in induction motors. *Power Electronics Specialists Conference Proceedings*, 1:245–252.

[Ortega et al., 1996] Ortega, R., Nicklasson, P. J., and Espinosa-Perez, G. (1996). On speed control of induction motors. *Automatica*, 32(3):455–460.

[Packard, 1994] Packard, A. (1994). Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22:79-92.

[Packard and Kantner, 1996] Packard, A. and Kantner, M. (1996). Gain scheduling the LPV way. *Proc. IEEE CDC*, pages 3938–3941.

[Packard et al., 1991] Packard, A., Zhou, K., Pandey, P., and Becker, G. (1991). A collection of robust control problems leading to LMI's. *Proc. IEEE CDC*, 2:1245–1250.

[Petersen and Pulle, 1997] Petersen, I. R. and Pulle, D. W. (1997). Robust Kalman filtering in direct torque control. *Proceedings of the 7th European Conference on Power Electronics and Applications*, 4:4.585–4.590.

[Petersen and Pulle, 1998] Petersen, I. R. and Pulle, D. W. (1998). Kalman filtering applied to induction motors: A deterministic viewpoint. *Proceedings of the Conference on Power Electronics and Variable Speed Drives*, pages 489–493.

[Rantzer and Megretski, 1994] Rantzer, A. and Megretski, A. (1994). System analysis via integral quadratic constraints. *Proceedings of the IEEE Conference on Decision and Control*, pages 3062–3067.

[Rasmussen, 1995] Rasmussen, H. (1995). *Self-tuning Torque Control of Induction Motors for High Performance Applications*. PhD thesis, Aalborg University.

[Rasmussen et al., 1995] Rasmussen, H., Tønnes, M., and Knudsen, M. (1995). Inverter and motor model adaption at stand-still using reference voltages and measured currents. *Proc. EPE*, pages 1367–1372.

[Rasmussen et al., 1997] Rasmussen, H., Vadstrup, P., and Børsting, H. (1997). Nonlinear decoupling of torque and field amplitude in an induction motor. *Finnish Workshop on Power and Industrial Electronics (FINPIE)*.

[Rasmussen et al., 1999] Rasmussen, H., Vadstrup, P., and Børsting, H. (1999). Nonlinear field oriented control of induction motors using the backstepping design. *Proc. European Control Conference (ECC)*.

[Real-Time Innovations,] Real-Time Innovations. *Stethoscope User's Manual 4.3*. Real-Time Innovations, Inc.

[Rugh and Shamma, 2000] Rugh, W. and Shamma, J. S. (2000). Research on gain scheduling. *Automatica*, 36:1401–1425.

[Scherer, 1999] Scherer, C. W. (1999). *Robust Mixed Control and LPV Control and Full Block Scalings, to appear in L. El Ghaoui, S. Niculescu:Recent Advances in LMI Methods in Control*. SIAM.

[Scherer, 2001] Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica*, 37:361–375.

[Scorletti and Ghaoui, 1995] Scorletti, G. and Ghaoui, L. E. (1995). Improved linear matrix inequality conditions for gain scheduling. *Proceedings of the 34th Conference on Decision and Control*, pages 3626–3631.

[Scorletti and Ghaoui, 1998] Scorletti, G. and Ghaoui, L. E. (1998). Improved LMI conditions for gain scheduling and related control problems. *Int. J. Robust Nonlinear Control*, (8):845–877.

[Shamma and Athans, 1990] Shamma, J. S. and Athans, M. (1990). Analysis of gain scheduled control for nonlinear plants. *IEEE Transactions on Automatic Control*, 35(8):898–907.

[Shamma and Athans, 1992] Shamma, J. S. and Athans, M. (1992). Gain scheduling:potential hazards and possible remedies. *IEEE Control Systems Magazine*, 12(3):101–107.

[Shiau and Lin, 2001] Shiau, L.-G. and Lin, J.-L. (2001). Stability of sliding-mode current control for high performance induction motor position drives. *IEE Proceedings of Electric Power Applications*, 148(1):69–75.

[Shyu and Shieh, 1995] Shyu, K.-K. and Shieh, H.-J. (1995). Variable structure current control for induction motor drives by space voltage vector PWM. *IEEE Transactions on Industrial Electronics*, 42(6):572–578.

[Siegelmann et al., 1997] Siegelmann, H. T., Horne, B. G., and Giles, C. L. (1997). Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 27(2):208–215.

[Skougaard and Wenzel, 1997] Skougaard, J. and Wenzel, J. M. (1997). Flexible test bench for design of high performance induction motor controllers. Master's thesis, Aalborg University.

[Su and Annaswamy, 1998] Su, Y. and Annaswamy, A. M. (1998). Stable neural controllers for nonlinear dynamic systems. *Automatica*, 34(5):641–650.

[Suykens et al., 1995a] Suykens, J. A. K., Vandewalle, J., and De Moor, B. (1995a). Nonlinear system identification using neural state space models, applicable in robust control design. *International Journal of Control*, (1):129–152.

[Suykens et al., 1996] Suykens, J. A. K., Vandewalle, J., and De Moor, B. (1996). *Artificial neural networks for modelling and control of non-linear systems*. Kluwer Academic Publishers, Netherlands.

[Suykens et al., 1999] Suykens, J. A. K., Vandewalle, J., and De Moor, B. (1999). Lur'e systems with multilayer perceptron and recurrent neural networks: Absolute stability and dissipativity. *IEEE Transactions on Automatic Control*, (4):770–774.

[Suykens et al., 1995b] Suykens, J. A. K., Vandewalle, J., and Moor, B. D. (1995b). Global asymptotic stability criteria for multilayer recurrent neural networks with applications to modelling and control. *Proc. of the IEEE International Conference on Neural Networks*.

[Trangbæk, 2000] Trangbæk, K. (2000). LMI-based gain scheduled robust flux observer for induction motors. *Proceedings of the 14th International Conference on Mathematical Theory of Networks and Systems (MTNS)*.

[Vas, 1998] Vas, P. (1998). *Sensorless Vector and Direct Torque Control*. Oxford University Press, New York.

[Vas, 1999] Vas, P. (1999). *Artificial-Intelligence Based Electrical Machines and Drives. Application of Fuzzy, Neural, Fuzzy-Neural and Genetic Algorithm-Based Techniques*. Oxford University Press, New York.

[WindRiver Systems, 1995] WindRiver Systems (1995). *Tornado: User's Guide (UNIX version)*. WindRiver Systems.

[Wu and Boyd, 1996] Wu, S.-P. and Boyd, S. (1996). sdpsol, a parser/solver for semidefinite programming and determinant maximization problems with matrix structure. Technical report.

[Yang and Lee, 1999] Yang, S.-M. and Lee, C.-H. (1999). A current control technique for voltage-fed induction motor drives. *IECON'99. Conference Proceedings. 25th Annual Conference of the IEEE Industrial Electronics Society*, 3(3):1380–1385.

[Zhou et al., 1996] Zhou, K., Doyle, J. C., and Glover, K. (1996). *Robust and Optimal Control*. Prentice-Hall, Inc, Upper Saddle River, New Jersey.

[Zhou et al., 1992] Zhou, K., Khargonekar, P. P., Stoustrup, J., and Niemann, H. H. (1992). Robust stability and performance of uncertain systems in state space. *Proc. of the 31st Conference on Decision and Control*, pages 662–667.

INDEX