State estimation and state feedback control in quasi-polynomial and quantum mechanical systems

PhD thesis

Written by: Attila Magyar

Supervisor: Dr. Katalin Hangos

University of Pannonia
Information Science Ph.D. School

MTA SzTAKI
Systems and Control Laboratory

2007

State estimation and state feedback control in quasi-polynomial and quantum mechanical systems

Értekezés doktori (PhD) fokozat elnyerése érdekében

Írta:
Magyar Attila

Készült a Pannon Egyetem Informatikai Tudományok Doktori Iskolája keretében

Témavezető: Dr. Hangos Katalin

Elfogadásra javaslom (igen / nem)

(aláírás)

A jelölt a doktori szigorlaton .....%-ot ért el

Veszprém .....
a Szigorlati Bizottság elnöke

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: ..... (igen / nem)

(aláírás)

Bíráló neve: ..... (igen / nem)

(aláírás)

A jelölt az értekezés nyilvános vitáján .....%-ot ért el

Veszprém, ...
a Bíráló Bizottság elnöke

A doktori (PhD) oklevél minősítése .....

.....
Az EDT elnöke

Tartalmi kivonat

Állapotbecslés és állapotvisszacsatolás kvázi-polinomiális és kvantummechanikai rendszerekre

A rendszer- és irányításelmélet állapottér reprezentáción alapuló módszereit manapság olyan területeken használják, mint a robusztus, LPV, és LQ irányítás. Ezek az eszközök olyan nehezen kezelhető rendszerekre is alkalmazhatóak, mint a folyamatrendszerek, nukleáris rendszerek, stb. Mindazonáltal, széles működési tartománnyal rendelkező erősen nemlineáris rendszerekre, mint a biomechanikai, biokémiai, vagy kvantum rendszerek, jelenleg sincsenek jól alkalmazható technikák.

A szerző a disszertációban két - a fizika különböző területeiről származó rendszerosztályon alkalmazta a modern irányításelmélet eszközeit. Kihasználva a speciális rendszerosztályok nyújtotta lehetőségeket, sikerült gyakorlatilag megvalósítható módszereket adni olyan problémákra, amelyek az általános esetben nehéznek, illetve számításigényesnek bizonyultak.

A folyamatrendszerek globális stabilitásvizsgálatát nemlineáris jellegük teszi nehéz feladattá. Olyan speciális nemlineáris rendszermodell osztályok alkalmazása, melyek elég általánosak a folyamatrendszerek dinamikus viselkedésének leírására, megkönnyítené ezen rendszerek irányítását. A szerző a disszertációban az úgynevezett kvázipolinomiális (QP) rendszerosztályt alkalmazza a fenti célra. Kihasználva, hogy QP rendszerekre a Ljapunov függvény alakja ismert, leegyszerűsödik az általános folyamatrendszerek globális stabilitásvizsgálata. A kvázipolinomiális rendszerosztály segítségével olyan szabályozótervezési feladatot is felír, amely biztosítja a zárt rendszer globális stabilitását egy adott Ljapunov függvény családra nézve.

Mindezidáig csupán néhány szerző próbálta meg a kvantummechanikai rendszereket rendszerelméleti oldalról megközelíteni. A disszertációban kitűzött probléma kvantuminformáció kiolvasásával kapcsolatos. Kvantummechanikában a mérés valószínűségi jellegű művelet, ami a teljes rendszert sztochasztikussá teszi, ezért megbízható állapotbecslési módszerekre van szükség. Az egyik út a bayesi módszertan alkalmazása, amely egy teljes valószínűségi modellt használ, és az ez alapján számolt becslő sok statisztikai információval szolgál a kérdéses állapotról. A másik irányt egy egyszerű pontbecslő kifejlesztése jelenti, amely a kvantumrendszerek egy szélesebb osztályára alkalmazható, továbbá a számításigénye nem túl nagy.

Abstract

State estimation and state feedback control in quasi-polynomial and quantum mechanical systems

The goal of this dissertation is to apply system and control theory to two system classes originating from different fields of physics: process systems and quantum mechanical systems.

Using the quasi-polynomial system representation it is possible to describe the dynamic behavior of general nonlinear process systems. It is shown in the work, that global stability analysis for such systems can be performed algorithmically, moreover, a globally stabilizing state feedback design method is given.

Only a few authors tried to apply control theory in quantum mechanical fields. The aimed problem of the second part of the dissertation is quantum state estimation, where the state of a quantum system is to be determined from quantum measurement. The problem is solved in two different ways.

Zusammenfassung

Zustandschätzung und Rückkoppelung für quasi-polynomischen und quantenmechanischen Systemen

Das Primärziel dieser Dissertation ist die Anwendung der Ergebnisse der System- und Regelungstheorie auf zwei verschiedene Systemsklassen der Physik: Prozess-Systeme und quantenmechanische Systeme.

Mit der Hilfe der quasi-polinomischen Systemklassen kann man die dynamischen Verhaltensweise des allgemeinen Prozesssysteme beschreiben. Der Autor will in der Dissertation zeigen, dass die globale Stabilitätsprüfung des quasipolinomischen Systemen algorithmisch ausgeführt werden kann. Ausserdem demonstriert er eine global stabilisierende Rückkoppelungs-Planungsmethode.

Bisher haben nur Wenige versucht die Systemtheorie im quantummachanischen Bereich anzuwenden. In dem zweiten Teil der Dissertation sind Lösungsmöglichkeiten des Problems der Zustandschätzung vom quantummechanischen Messungen gezeigt. Zur Erzielung dieser Lösung muss der Systemzustand vom Quantenmessungen auswertet werden. Der Autor zeigt zwei Methoden für die Lösung der Aufgabe.

Acknowledgement

First of all I would like to express my gratitude to Dr. Katalin Hangos, my supervisor, for her help. She gave me a lot of advice and guidance in my research, helped me with my studies, and gave me inspiration for my future work.

I would also like to thank Dr. József Bokor, Head of Systems and Control Laboratory, Computer and Automation Research Institute, for making my research possible, I am grateful to Dr. Gábor Szederkényi, the coauthor of most of my papers for his help and advices. I would like to thank Dr. Dénes Petz for his help in the quantum mechanical part of my work. I am also grateful to my research fellows, Dávid Csercsik, Csaba Fazekas, Dr. Erzsébet Németh, László Ruppert and Zsuzsanna Weinhandl for their cooperation. I am grateful for all the help I received from every member of our Laboratory.

Finally, I am also grateful to my family for the strong support they gave me in the past 28 years.

Contents

1 Introduction **10**
- 1.1 Background and motivation . . . 10
- 1.2 System- and control theory . . . 11
- 1.3 Problem statement and aims . . . 13
- 1.4 Thesis structure . . . 14
- 1.5 General notations . . . 15

I QP systems **17**

2 Basic notions on quasi-polynomial systems **19**
- 2.1 Quasi-polynomial and Lotka-Volterra systems . . . 19
  - 2.1.1 QP models . . . 19
  - 2.1.2 Lotka-Volterra models . . . 19
  - 2.1.3 Input-affine QP system models . . . 20
- 2.2 Embedding process systems into QP and LV forms . . . 21
  - 2.2.1 QP models of process systems . . . 22
  - 2.2.2 A simple fermentation example . . . 22
- 2.3 Earlier work on the representation of QP systems . . . 25

3 Stability analysis of quasi-polynomial systems **26**
- 3.1 Global stability analysis using linear matrix inequalities . . . 26
  - 3.1.1 Global stability analysis . . . 26
  - 3.1.2 Zero dynamics analysis . . . 27
  - 3.1.3 Zero dynamics of the simple fermentation process . . . 28
- 3.2 Time-reparametrization . . . 29
  - 3.2.1 The time-reparametrization transformation . . . 29
  - 3.2.2 Properties of the time-reparametrization transformation . . . 30
  - 3.2.3 The time-reparametrization problem as a BMI . . . 31
  - 3.2.4 Examples . . . 33
- 3.3 Summary . . . 34

4 Stabilizing control of quasi-polynomial systems **35**
- 4.1 The controller design problem . . . 35
- 4.2 Numerical solution of the controller design problem . . . 36
  - 4.2.1 Numerical solution based on bilinear matrix inequalities . . . 36
  - 4.2.2 An iterative LMI approach to controller design problem . . . 37

4.3 Equilibrium points . . . 39
4.3.1 Fully actuated case . . . 40
4.3.2 Partially actuated case . . . 40
4.3.3 Rank deficient (embedded) systems . . . 40
4.4 Feedback structure selection . . . 41
4.4.1 Fully actuated case . . . 41
4.4.2 Partially actuated case . . . 42
4.4.3 Degenerated case . . . 42
4.5 Examples . . . 42
4.5.1 Partially actuated process system example in QP-form . . . 43
4.5.2 Fully actuated process system example in QP-form . . . 44
4.6 Summary . . . 45

II Quantum systems 47

5 Introduction to finite quantum mechanical systems 49
5.1 States of quantum mechanical systems . . . 49
5.1.1 Quantum dynamics . . . 49
5.1.2 State representation: the density matrix . . . 50
5.1.3 Finite quantum systems . . . 51
5.1.4 Bloch vector . . . 51
5.1.5 Distances between quantum states . . . 53
5.2 Quantum measurement and state estimation . . . 54
5.2.1 von Neumann measurement . . . 54
5.2.2 Positive operator valued measurement . . . 55
5.2.3 The quantum state estimation problem . . . 56
5.3 Earlier work on quantum state estimation . . . 57

6 Bayesian state estimation of a quantum bit 58
6.1 Componentwise estimation of the Bloch vector . . . 58
6.1.1 Bayesian estimate . . . 58
6.1.2 Point estimate . . . 60
6.2 Estimation of the Bloch vector . . . 61
6.2.1 Unconstrained Bayesian estimation . . . 61
6.2.2 Constrained Bayesian estimation . . . 61
6.3 Comparison based on computer simulation results . . . 62
6.3.1 A simulation software for quantum systems . . . 62
6.3.2 The effect of the number of measurements . . . 62
6.3.3 The effect of the length of Bloch vector . . . 64
6.4 Summary . . . 66

7 Point estimation of N-level quantum systems 68
7.1 The unconstrained estimation scheme . . . 68
7.1.1 Measurement strategy . . . 68
7.1.2 State estimator for N-level quantum systems . . . 69
7.1.3 Properties of the estimate . . . 70

7.2 The constrained estimation scheme . . . 73
7.2.1 The constrained estimator and its properties . . . 73
7.2.2 Computing the constrained estimate . . . 74
7.3 Comparison with other state estimation methods . . . 77
7.3.1 The modified unconstrained estimator . . . 79
7.3.2 Standard qubit tomography . . . 79
7.3.3 Minimal qubit tomography . . . 80
7.4 Summary . . . 81

8 Conclusions **82**
8.1 Stability analysis and state estimation for state feedback . . . 82
8.2 New results . . . 83
8.3 Future work and applicability areas . . . 85
8.4 Own papers . . . 86

A Appendix **88**
A.1 Basics of system and control theory . . . 88
A.1.1 System classes, basic system properties . . . 89
A.1.2 Controller design . . . 91
A.1.3 State estimation . . . 94
A.1.4 System parameter estimation . . . 95
A.2 Bloch vector space in the N-level case . . . 97
A.3 Examples of QP feedback design . . . 98
A.3.1 Example with a rank deficient M matrix . . . 98
A.3.2 Feedback design for a simple fermentation process . . . 100
A.4 Applied mathematical tools . . . 101
A.4.1 Linear and bilinear matrix inequalities . . . 101
A.4.2 Tensor product and its properties . . . 102
A.5 Tables . . . 103
A.6 Spinsim function reference . . . 104

Chapter 1

Introduction

Discovering the world is interesting,
useful, delightful, terrifying or edifying;
discovering ourselves is the greatest journey,
the most terrifying discovery and the most edifying of encounters.
/Sándor Márai/

Analysis and control of general nonlinear and stochastic systems is a difficult area with many computationally hard problems. However, in special cases by using special system classes which exploit the physical characteristics of the examined system, feasible results can be achieved.

The present work connects two topics with different special system class and slightly different focus, that are originated from different fields of physics. However, they are connected through system- and control theory, and they represent two special, yet practically important nonlinear and stochastic system classes.

One of them is the class of lumped process systems with smooth nonlinearities that can be embedded into the class of quasi-polynomial systems. The other one is the class of finite quantum systems.

1.1 Background and motivation

Process systems appearing in practice [23] are difficult to handle since there are no *universal* methods which give a complete framework for their dynamical analysis, and synthesis [44]. That's why it would be of great importance to develop methods which are general enough to handle a wide range of process systems, or to find a representation that's suitable for describing almost all process systems. At the same time, process systems form a relatively simple nonlinear system class with smooth nonlinearities that are relatives of systems with polynomial nonlinearities. This is, why the quasi-polynomial system class has been selected as a case study for a class of "easy" nonlinear systems.

On the other hand, one of the biggest challenge of present days is the built of a *quantum computer* which would make some special problems easier to solve (e.g. prime factorization). These problems are tackled by quantum computation [1] and

quantum cryptography [5]. While classical computers manipulate classical information represented by systems obeying classical physics, quantum computers would modify, write, read quantum information which can be represented by quantum mechanical systems. It means, that in order to be able to *read quantum information* we must be able to guess the actual properties of a quantum mechanical system from measurements performed on it [24].

From the system theoretical point of view, even the simplest quantum systems are unusual stochastic systems, where the stochastic nature is caused by the measurements that act as a disturbance to the system. This property causes that quantum systems represent a real challenge for everyone who attempts to solve even the simplest control-related problem for them.

Modern control methods for nonlinear and/or stochastic systems rely on the concept of state and apply state-space models. Therefore, state estimation and state feedback controller design are key problems in system and control theory.

1.2 System- and control theory

The general notion of system allows us to treat physical objects originating from various fields of life: automotive systems, chemical processes, nuclear powerplants, etc. System- and control theory (see Appendix A.1 for introduction of some basic notions, [2], or [9] for a deeper insight) allows us to examine and modify systems with mathematical tools.

Nonlinear systems

A wide class of dynamical systems can be represented by the following state space model [64]:

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + \sum_{i=1}^{p} g_i(x(t))u_i(t) \\ y(t) &= h(x(t)) \qquad x(t_0) = x^0, \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) = (u_1(t), u_2(t), \ldots, u_p(t))^T \in \mathbb{R}^p$ and $y(t) \in \mathbb{R}^q$,

$$f : \mathbb{R}^n \to \mathbb{R}^n, \quad g_i : \mathbb{R}^n \to \mathbb{R}^n, \ i = 1, \ldots, p, \quad h : \mathbb{R}^n \to \mathbb{R}^q$$

are nonlinear functions. What makes (1) attractive is the fact that although the system is nonlinear in the states, it is linear in its inputs.

Dynamical analysis of nonlinear systems needs advanced mathematical tools [64], [29]. Global stability analysis of nonlinear systems calls for the searching of a suitable *Lyapunov function* V with the following properties:

- scalar valued function: $V : \mathbb{R}^n \to \mathbb{R}^+$
- positive: $V(x(t)) > 0$
- decreasing in time: $\frac{d}{dt}V(x(t)) < 0$

Although the form of the Lyapunov function is not known for a general nonlinear system (1), for some special system class (see Appendix A.1.1) it is possible to achieve results.

State feedback control

Applying control to the system makes it possible to modify its dynamical properties, and behavior. In most cases the control aim is reached by using *feedback*, i.e. the output signal is fed back to the input through a *controller* (see figure 1). In most cases, the structure of the controller is a state feedback, which means, that the control input is determined as a function of the states:

$$u(t) = k(x(t)), \quad k : \mathbb{R}^n \to \mathbb{R}^p. \tag{2}$$

This implies a problem in the general case, since it is only possible to measure the outputs of the system. It explains the additional unit (the state estimator/observer) before the controller that computes the state signal from the output, as it can be seen in figure 1.

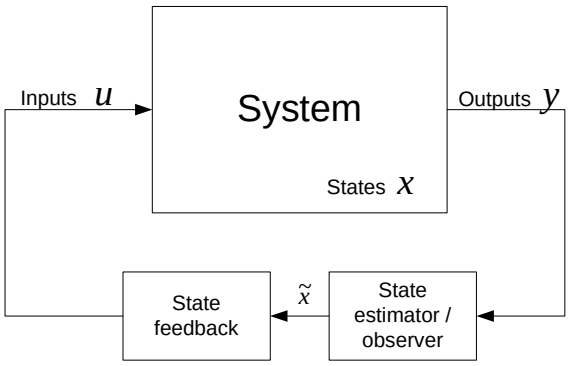

Figure 1: State feedback control

State estimation

It was mentioned above, that most control techniques apply state feedback. This calls for a method that determines the actual state x of the system using the supposed system model and measured input-output data corrupted by some noise according to measurement devices. Such a method is called state estimation.

The state estimator is a mapping from the set D of possible measurement data to the state space:

$$\hat{x} = l(u, \tilde{y}), \quad l : D \to \mathbb{R}^n, \tag{3}$$

where $\hat{x}$ denotes the estimate of the state x, $\tilde{y}$ stands for the measured (and possibly noisy) output. A well known, and widely used tool for state estimation is the *Kalman-filter* [32].

Of course, a good estimate $\hat{x}$ should meet some requirements. The first one is *unbiasedness* which means, that the expected value of the estimate $\hat{x}$ equals to the real state x. Moreover, a *consistent* estimate is needed, i.e. an estimate that converges to the real state as the number of measurements increases.

System parameter estimation

Most of the controller design and state estimation methods need the dynamical model of the system, since the parameters of the controller and/or the state estimator/observer are computed from the system parameters. Estimating the model parameters [52], [54] is also a basis of system diagnostics since different parameter values may refer to certain faults of the plant.

The two fundamental methods of parameter estimation are the least squares estimation and the bayesian parameter estimation, they are summarized in the Appendix A.1.4. Whilst the least squares (LS) estimation is based on minimizing the norm of the model error with a given parameter set, bayesian method uses a stochastic model of the system in the form of a conditional probability density function and gives more statistical information about the model parameters.

1.3 Problem statement and aims

The present thesis treats two different system classes originating from fields that are far away from each other. Tools of modern system and control theory are applied on them in such a way that their specialities are utilized to obtain practically feasible methods for problems that are computationally hard in the general case.

The nonlinear nature of general process systems makes their global stability analysis hard, however, in case of industrial process systems, like fermentation it is crucial to be able to prove the stability of a system to be implemented. Using nonlinear system model classes (more special than (1)) that is still general enough to describe the dynamics of them it might be easier to handle them.

In this work the so-called quasi-polynomial (QP) system class will be used for this purpose. QP systems has a very advantageous property, namely, the structure of their Lyapunov function is known. Using this fact will facilitate the global stability analysis of general process systems since it is only necessary to find suitable parameters of a Lyapunov function of a given form in order to prove global asymptotic stability.

As a next step, the QP system class will be used for synthesizing controllers which ensure the global stability of the closed loop system with respect to the given Lyapunov function family. Using the fact, that with a suitable feedback the closed loop system still belongs to the class of QP systems, the same type of Lyapunov function can be used.

Note, that the state variables of process systems are typically concentrations, temperatures which are measurable quantities so the feedback control of them does not require state observers or estimators.

So far, only a few people (e.g. [38]) has tried to handle quantum mechanical systems on the control theoretical basis. The aimed subproblem of reading quantum information asks for the design of state observers/estimators. As it will be shown later in chapter 5 the measurement of a quantum system has probabilistic nature and turns the whole system to be stochastic (see (134) in Appendix A.1.1), reliable state estimation methods must be developed.

One way is to apply the Bayesian methodology (see Appendix A.1.4 for the basic

notations of the Bayesian parameter estimation problem) to use a full probabilistic model and give a state estimate that holds a lot of information about the state to be estimated.

The other direction to quantum state estimation is to develop a simple estimator which is applicable for a wide range of quantum systems and moreover it is easy to compute.

1.4 Thesis structure

This work is structured as follows. Present chapter clarifies the notations used throughout the work, and a short overview on system- and control theory is also presented here.

The results presented in this thesis are structured in two main parts. Part I is dedicated to quasi-polynomial model based dynamical analysis and control of process systems. Within this, chapter 2 gives an introduction to *quasi-polynomial* systems and their connection to nonlinear process systems. Chapter 3 links the global stability analysis of quasi-polynomial systems to *linear matrix inequalities*. A bigger chance on proving global stability can be reached by the *time-reparametrization* transformation, which is also presented here. Chapter 4 uses the results of the previous chapter and deals with the design of a state feedback controller that globally stabilizes the system with respect to an entropy-like Lyapunov function.

The fundamental notions of quantum mechanics together with the proposed results pertaining to the state estimation of quantum systems are presented in part II. Within this, the basics of quantum mechanics used in the later chapters are summarized in chapter 5. The next two chapters concentrate on the state estimation of quantum mechanical systems. Chapter 6 applies Bayesian methodology to the determination of states of single *quantum bits*. A more general class of quantum systems is treated in chapter 7 by a simple but effective estimation procedure.

At the end of each chapter containing new results, a summary is presented. The own results are written in *italic* typeface. The corresponding publications are cited also in the title of each section.

Finally, chapter 8 summarizes all results, and the four suggested thesis points are also presented here together with my own papers. The future plans are also mentioned in chapter 8.

In the Appendix, one can find a brief summary of system- and control theory (Appendix A.1). Appendix A.2 the structure of N-level quantum systems' Bloch vector space is detailed. Afterwards some examples connected to quasi-polynomial feedback design are given (Appendix A.3). In Appendix A.4, linear matrix inequalities, bilinear matrix inequalities are discussed together with the basic properties of the tensor product. The function reference of the quantum system simulator called *Spinsim* can be found in Appendix A.6.

1.5 General notations

The notations and abbreviations used throughout the work are summarized in this section. Since we are treating two fields in a common system theoretical framework, it is important to use a strict notation which makes it easier to find the connections between the two parts. The most important notations are listed below.

Notation	-	Meaning
$x \in X$	-	x is an element of set X
$x \notin X$	-	x is not an element of set X
$X \subset Y$	-	X is a subset of Y
$\emptyset$	-	empty set
$\cup$	-	union
$\cap$	-	intersection
$\mathbb{R}$	-	set of real numbers
$\mathbb{C}$	-	set of complex numbers
$\mathbb{R}^n$	-	n dimensional real space
A	-	matrix
A^T	-	transpose of matrix A
A^*	-	conjugate transpose of matrix A
$A_{i,j}$	-	(i,j)-th entry of matrix A
A_k	-	k-th element in a series of matrices $A_1, A_2, \ldots$
I	-	unit matrix of appropriate size $I = \mathrm{diag}(1, \ldots, 1)$
E_{ij}	-	square matrix of appropriate size whose (i,j)-th entry is 1, all the others are 0
$\mathrm{Tr}A$	-	trace of matrix A
$\det A$	-	determinant of matrix A
$A \otimes B$	-	tensor product of matrices A and B
$A \circ B$	-	Hadamard product of matrices A and B, i.e. $(A \circ B)_{i,j} = A_{i,j} B_{i,j}$
c	-	scalar, or vector
c_i	-	i-th element of vector c
c^T	-	transpose of vector c
$\mathcal{H}$	-	Hilbert space
$\|x\rangle$	-	vector from a Hilbert space, a ket state
$\langle x\|$	-	a bra state according to the ket $\|x\rangle$
$\mathbf{S}$	-	system operator
$x(t)$, or x	-	state of a given system, or a Bloch vector of a quantum mechanical system
χ	-	density matrix

Notation	-	Meaning
$\mathrm{Prob}(\omega)$	-	probability of event ω
$\mathbf{E}\,\xi$	-	expectation value of random variable ξ
$u(t)$, or u	-	input of a given system
$y(t)$, or y	-	output of a given system
$\dot{x} = \frac{dx}{dt}$	-	time derivative of x
$\frac{\partial f(x)}{\partial x_i}$	-	i-th partial derivative of $f(x)$
$p(.)$	-	probability density function
D^k	-	measurement data obtained from k measurements

The abbreviations used in the sequel are the followings.

Abbreviation	-	Meaning
QP	-	quasi-polynomial
LV	-	Lotka-Volterra
GLV	-	generalized Lotka-Volterra
LS	-	least squares
RHS	-	right hand side
LMI	-	linear matrix inequality
ILMI	-	iterative LMI
BMI	-	bilinear matrix inequality
LTI	-	linear time invariant
CSTR	-	continuously stirred tank reactor
POVM	-	positive operator valued measurement
p.d.f.	-	probability density function
iff	-	if and only if
s.t	-	such that

Throughout the thesis, my own papers are cited in the form [Oi], while other publications are cited as [j].

Part I

QP systems

Process systems are highly nonlinear systems due to some special features taking place in them [23]. Quasi-polynomial and Lotka-Volterra models have proved to be one of the candidates for generally applicable canonical forms of nonlinear process system models since the majority of smooth nonlinear systems occurring in practice can be transformed into these forms.

The aim of the first part is to utilize the quasi-polynomial and Lotka-Volterra representation to stabilizing control of process systems. Before formulating the feedback design problem the global stability analysis will be solved.

Chapter 2

Basic notions on quasi-polynomial systems

Present chapter gives a theoretical summary and a literature overview of part I. Quasi-polynomial systems are introduced in section 2.1. Section 2.2 deals with the conditions and the algorithm of embedding general nonlinear process systems into quasi-polynomial form. At the end of the chapter a brief review of earlier works connected to the representation and stability analysis of quasi-polynomial systems is given (section 2.3).

2.1 Quasi-polynomial and Lotka-Volterra systems

The elementary notions in the field of quasi-polynomial (QP) and Lotka-Volterra (LV) systems are introduced in this chapter. In order to emphasize the similarity of QP and LV systems, QP systems are also called *generalized Lotka-Volterra (GLV) systems*.

2.1.1 QP models

Quasi-polynomial models are systems of ODEs of the following form

$$\dot{x}_i = x_i \left(\lambda_i + \sum_{j=1}^{m} A_{i,j} \prod_{k=1}^{n} x_k^{B_{j,k}} \right), \quad i = 1, \ldots, n. \tag{4}$$

where $x \in \mathrm{int}(\mathbb{R}^n_+)$, $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{m \times n}$, $\lambda_i \in \mathbb{R}$, $i = 1, \ldots, n$. Furthermore, $\lambda = [\lambda_1 \ \ldots \ \lambda_n]^T$. The above model belongs to the class of nonlinear systems (131), see Appendix A.1.1. Let us denote the equilibrium point of interest of (4) as $x^* = [x_1^* \ x_2^* \ \ldots \ x_n^*]^T$. Without the loss of generality we can assume that $\mathrm{Rank}(B) = n$ and $m \geq n$ (see [28]).

2.1.2 Lotka-Volterra models

The above family of models is split into classes of equivalence [27] according to the values of the products $M = B \cdot A$ and $N = B \cdot \lambda$. The *Lotka-Volterra form* known

from the field of population biology [42], [65], gives the representative elements of these classes of equivalence. If $rank(B) = n$, then the set of ODEs in (4) can be embedded into the following m-dimensional set of equations, the so called Lotka-Volterra model:

$$\dot{z}_j = z_j \left(N_j + \sum_{i=1}^{m} M_{j,i} z_i \right), \qquad j = 1, \ldots, m \tag{5}$$

where

$$M = B \cdot A, \quad N = B \cdot \lambda,$$

and each z_j represents a so called *quasi-monomial*:

$$z_j = \prod_{k=1}^{n} x_k^{B_{j,k}}, \qquad j = 1, \ldots, m. \tag{6}$$

2.1.3 Input-affine QP system models

An input-affine nonlinear system model (1) is in QP-form if all of the functions f, g and h are in QP-form. Then the general form of the state equation of an input-affine QP system model with p-inputs is:

$$\begin{aligned} \dot{x}_i &= x_i \left(\lambda_{0_i} + \sum_{j=1}^{m} A_{0_{i,j}} \prod_{k=1}^{n} x_k^{B_{j,k}} \right) + \\ &+ \sum_{l=1}^{p} x_i \left(\lambda_{l_i} + \sum_{j=1}^{m} A_{l_{i,j}} \prod_{k=1}^{n} x_k^{B_{j,k}} \right) u_l \end{aligned} \tag{7}$$

where

$$i = 1, \ldots, n, \quad A_0, A_l \in \mathbb{R}^{n \times m}, \quad B \in \mathbb{R}^{m \times n},$$
$$\lambda_0, \lambda_l \in \mathbb{R}^n, \quad l = 1, \ldots, p.$$

The corresponding input-affine Lotka-Volterra model is in the form

$$\dot{z}_j = z_j \left(N_{0_j} + \sum_{k=1}^{m} M_{0_{j,k}} z_k \right) + \sum_{l=1}^{p} z_j \left(N_{l_j} + \sum_{k=1}^{m} M_{l_{j,k}} z_k \right) u_l \tag{8}$$

where

$$j = 1, \ldots, m, \quad M_0, M_l \in \mathbb{R}^{m \times m}, \quad N_0, N_l \in \mathbb{R}^m, \quad l = 1, \ldots, p,$$

and the parameters can be obtained from the input-affine QP system's ones in the following way

$$\begin{aligned} M_0 &= B \cdot A_0 \\ N_0 &= B \cdot L_0 \\ M_l &= B \cdot A_l \\ N_l &= B \cdot \lambda_l \end{aligned} \qquad l = 1, \ldots, p. \tag{9}$$

2.2 Embedding process systems into QP and LV forms

A wide class of nonlinear autonomous systems with smooth nonlinearities can be embedded into QP-form [26] if they satisfy two requirements.

1. The set of nonlinear ODEs should be in the form:

$$\dot{x}_s = \sum_{i_{s1},\ldots,i_{sn},j_s} a_{i_{s1}\ldots i_{sn}j_s} x_1^{i_{s1}} \ldots x_n^{i_{sn}} f(\overline{x})^{j_s}, \tag{10}$$

$$x_s(t_0) = x_s^0, \qquad s = 1, \ldots, n$$

   where $f(\overline{x})$ is some scalar valued function, which is not reducible to *quasi-monomial form* containing terms in the form of $\prod_{k=1}^{n} x_k^{\Gamma_{j,k}}$, $j = 1, \ldots, m$ with Γ being a real matrix.

2. Furthermore, we require that the partial derivatives of the model (10) fulfill:

$$\frac{\partial f}{\partial x_s} = \sum_{e_{s1},..,e_{sn},e_s} b_{e_{s1}..e_{sn}e_s} x_1^{e_{s1}} \ldots x_n^{e_{sn}} f(\overline{x})^{e_s}$$

The embedding is performed by introducing a *new auxiliary variable*

$$\eta = f^q \prod_{s=1}^{n} x_s^{p_s}, \qquad q \neq 0. \tag{11}$$

Then, instead of the non-quasi-polynomial nonlinearity f we can write the original set of equations (10) into QP-form:

$$\dot{x}_s = \left(x_s \sum_{i_{s1},\ldots,i_{sn},j_s} \left(a_{i_{s1}\ldots i_{sn}j_s} \eta^{j_s/q} \prod_{k=1}^{n} x_k^{i_{sk}-\delta_{sk}-j_s p_k/q} \right) \right), \qquad s = 1, \ldots, n \tag{12}$$

where $\delta_{sk} = 1$ if $s = k$ and 0 otherwise. In addition, a new quasi-polynomial ODE appears for the new variable η:

$$\begin{aligned} \dot{\eta} \;=\; & \eta \left(\sum_{s=1}^{n} \left(p_s x_s^{-1} \dot{x}_s + \sum_{\substack{i_{s\alpha},j_s \\ e_{s\alpha},e_s}} a_{i_{s\alpha},j_s} b_{e_{s\alpha},e_s} q \eta^{(e_s+j_s-1)/q} \times \right.\right. \\ & \left.\left. \times \prod_{k=1}^{n} x_k^{i_{sk}+e_{sk}+(1-e_s-j_s)p_k/q} \right) \right), \qquad \alpha = 1, \ldots, n. \end{aligned} \tag{13}$$

It is important to observe that the embedding is not unique, because we can choose the parameters p_s and q in (11) in many different ways: the simplest is to choose $(p_s = 0,\ s = 1, ..., n;\ q = 1)$.

If we set the initial values of the newly introduced variables according to (11) then the dynamics of the embedded system is equivalent to the original non-QP

system described in (10). Since the embedded QP system includes the original differential variables x_i, $i = 1, \ldots, n$, it is clear that the stability of the embedded system (12)-(13) implies the stability of the original system (10).

It is important to note that QP models originate from embedding have some unusual dynamic properties because their trajectories range only a lower dimensional manifold of the QP state space. Thus they can be regarded as "hidden" differential-algebraic (DAE) system models with *rank deficient* A parameter matrices [51].

2.2.1 QP models of process systems

The nonlinearities of a lumped parameter process system model are of two types from the viewpoint of their QP-form representation. The nonlinearities originating from the sources (e.g. reaction or transfer rates) appear in the f function of the input-affine state-space model (133) and they are not necessarily in QP-form. Therefore, the above described embedding of such models into QP-form is of great practical importance.

The specialities of the input function g_i

The specialities of the input function g_i of the input-affine state-space model (1) originate from the fact that the inputs of process systems are most often realized through either inlet mass or component mass flow-rates, or alternatively, intensive variables at the inlet, like temperatures or concentrations. This means that they act through the inlet convection term [22] of the conservation balances that are transformed into state equations. As convection is bilinear in a mass flow-rate and an intensive variable (such as concentration, temperature or pressure), the nonlinear input function $g_i(x)$ is most often a simple homogeneous linear function of the corresponding state variable x_i:

1. $g_i(x) = const \cdot x_i$ when the mass flow-rates are the input variables, or
2. $g_i(x) = const^*$ when the intensive variables at the inlet are the inputs.

Case (1) implies that the parameters $A_l = 0$ in (7) and $M_l = 0$ in (8).

The above special form is, of course, not valid, when a QP state equation originates from variable embedding.

2.2.2 A simple fermentation example

A simple fermentation example illustrates the way of embedding non-QP system models into QP-form and the special properties of process system models in QP-form. Consider a simple fermentation process with non-monotonous reaction kinetics that is described by the non-QP input-affine state-space model

$$\begin{aligned} \dot{x}_1 &= \mu(x_2)x_1 + \frac{(X_F - x_1)F}{V} \\ \dot{x}_2 &= -\frac{\mu(x_2)x_1}{Y} + \frac{(S_F - x_2)F}{V} \\ \mu(x_2) &= \mu_{max}\frac{x_2}{K_2x_2^2 + x_2 + K_1} \end{aligned} \tag{14}$$

where the state variables x_1 and x_2 are the biomass- and the substrate concentrations respectively. The inlet substrate and biomass concentrations denoted by S_F and X_F, are the manipulated inputs. The variables and parameters of the model together with their units and parameter values are given in Table 5. The parameter values are taken from [41].

By introducing a new differential variable $\eta = \frac{1}{K_2 x_2^2 + x_2 + K_1}$ one arrives at a third differential equation

$$\dot{\eta} = -\frac{2K_2 x_2 + 1}{(K_2 x_2^2 + x_2 + K_1)^2} \cdot \dot{x}_2 \tag{15}$$

that completes the ones for x_1 and x_2. Thus the original system (14) can be represented by three differential equations in input-affine QP-form:

$$\begin{aligned}
\dot{x}_1 &= x_1 \left(\mu_{max} x_2 \eta - \frac{F}{V} \right) + x_1 \left(x_1^{-1} \frac{F}{V} \right) X_F \\
\dot{x}_2 &= x_2 \left(-\frac{\mu_{max}}{Y} x_1 \eta - \frac{F}{V} \right) + x_2 \left(x_2^{-1} \frac{F}{V} \right) S_F \\
\dot{\eta} &= \eta \left(\frac{2\mu_{max} K_2}{Y} x_1 x_2^2 \eta^2 + \frac{2K_2 F}{V} x_2^2 \eta + \frac{\mu_{max}}{Y} x_1 x_2 \eta^2 + \frac{F}{V} x_2 \eta \right) + \\
&\quad + \eta \left(-\frac{2K_2 F}{V} x_2 \eta - \frac{F}{V} \eta \right) S_F
\end{aligned} \tag{16}$$

The system has a locally stable equilibrium point in the positive orthant:

$$\begin{bmatrix} x_1^* \\ x_2^* \end{bmatrix} = \begin{bmatrix} 4.8906 \\ 0.2187 \end{bmatrix} \tag{17}$$

with steady-state inputs

$$\begin{bmatrix} X_F^* \\ S_F^* \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \end{bmatrix}. \tag{18}$$

Note, that there is also a so-called wash-out equilibrium where biomass concentration x_1 is zero.

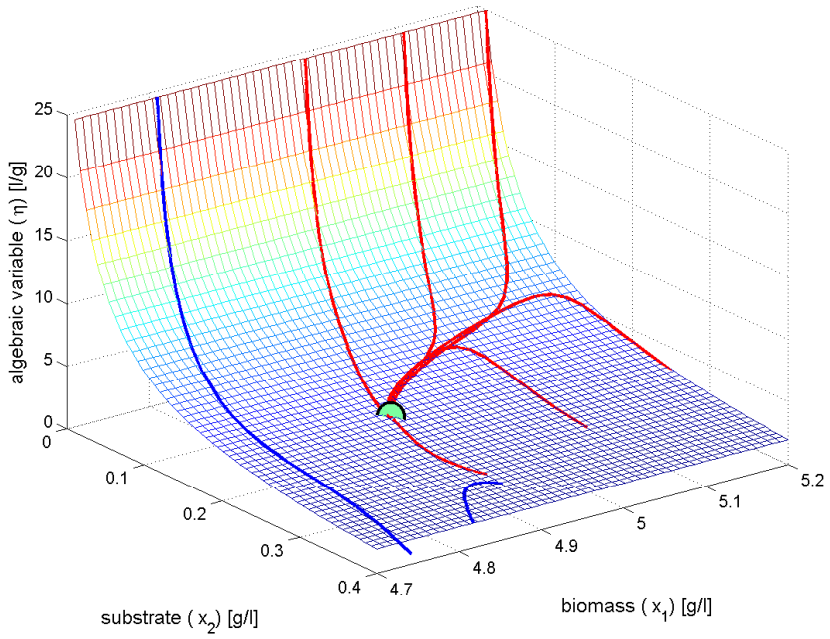

Figure 2: Some trajectories of the system (14) embedded into the QP model (16)

The system can be characterized by the following matrices:

$$A_0 = \begin{bmatrix} \mu_{max} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{\mu_{max}}{Y} & 0 & 0 & 0 & 0 & 0 \\ \frac{F}{V} & 0 & 0 & 0 & \frac{2\mu_{max}K_2}{Y} & \frac{2K_2F}{V} & \frac{\mu_{max}}{Y} & 0 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 0 & \frac{F}{V} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{F}{V} & 0 & 0 & 0 & 0 \\ -\frac{2K_2F}{V} & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{F}{V} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 2 & 2 \\ 0 & 2 & 1 \\ 1 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \quad \lambda_0 = \begin{bmatrix} -\frac{F}{V} \\ -\frac{F}{V} \\ 0 \end{bmatrix} \qquad \lambda_1 = \lambda_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{19}$$

The eight quasi-monomials of the QP system model given by the matrices (19) are

$$x_2\eta, \quad x_1^{-1}, \quad x_1\eta, \quad x_2^{-1}, \quad x_1x_2^2\eta^2, \quad x_2^2\eta, \quad x_1x_2\eta^2, \quad \eta.$$

The lower dimensional manifold and some trajectories of the system can be seen on Figure 2. (The inputs X_F and S_F are held constant.)

2.3 Earlier work on the representation and stability analysis of QP systems

The fundamental works on LV systems was proposed by Lotka [42] and Volterra [65] which put the LV form into a population biology framework.

From the $90's$ there are several works about the representation of general nonlinear systems having smooth nonlinearities by QP and LV models, e.g. [26], [27] [28]. [27] established the algebraic structure of the class of QP systems. They split into equivalence classes and each class of equivalence is represented by a Lotka-Volterra system.

The other branch of papers are engaged in the stability properties of Lotka-Volterra and quasi-polynomial systems. Local stability analysis of them can be found in [12], where the locally linearized system matrices can be determined directly from the QP or LV system's parameter matrices.
Several works investigate the global stability of Lotka-Volterra predator-prey models, especially with periodic solutions [59], [43]. However, there are also works on the global stability of quasi-polynomial systems [25], [14]. The main weakness of them is that only small (3-4) dimensional LV systems can be handled with these methods. An interesting numerical method for their stability analysis is given in [17].

An algorithmic method for finding invariants of quasi-polynomial systems is proposed in [51].

On the other hand, the utilization of Lotka-Volterra models for feedback control appears only in few papers [18], or [19], where the positive stabilizing control is proposed only for a subset of LV systems.

Chapter 3

Stability analysis of quasi-polynomial systems

Global asymptotic stability is a very strong property of all system classes discussed in section 1.2 and Appendix A.1.1. The global stability analysis of general nonlinear systems (131) is far from being trivial, and results can be obtained only for special system classes.

Based on the fundamental concepts of QP and LV systems presented in chapter 2, this chapter draws up own results for the global stability analysis of quasi-polynomial systems. Section 3.1 presents a method for global stability analysis, afterwards, section 3.2 generalizes its applicability.

3.1 Global stability analysis using linear matrix inequalities [O4]

This section reformulates the time-decreasing condition of a class of Lyapunov functions for Lotka-Volterra systems so that widespread numerical solvers can be used for their global stability analysis.

3.1.1 Global stability analysis

Henceforth it is assumed that x^* is a positive equilibrium point, i.e. $x^* \in \text{int}(\mathbb{R}^n_+)$ in the QP case and similarly $z^* \in \text{int}(\mathbb{R}^m_+)$ is a positive equilibrium point in the LV case. For LV systems there is a well known candidate Lyapunov function family [25],[14], which is in the form:

$$V(z) = \sum_{i=1}^{m} c_i \left(z_i - z_i^* - z_i^* \ln \frac{z_i}{z_i^*} \right), \tag{20}$$

$$c_i > 0, \quad i = 1 \dots m,$$

where $z^* = (z_1^*, \dots, z_m^*)^T$ is the equilibrium point corresponding to the equilibrium x^* of the original QP system (4). The time derivative of the of the Lyapunov function

(20) is:

$$\dot{V}(z) = \frac{1}{2}(z - z^*)(CM + M^T C)(z - z^*) \tag{21}$$

where $C = \mathrm{diag}(c_1, \ldots, c_m)$ and M is the invariant characterizing the LV form (5). Therefore the non-increasing nature of the Lyapunov function is equivalent to a feasibility problem over the following set of *linear matrix inequality* (LMI) constraints:

$$\begin{array}{rcl} CM + M^T C & \leq & 0 \\ C & > & 0 \end{array} \tag{22}$$

where the unknown matrix is C, which is diagonal and contains the coefficients of (20). (See Appendix A.4.1 for the properties and solution methods of LMIs.)

Note the similarity of the stability conditions with continuous time LTI systems (127): for a system with state matrix A to be asymptotically stable, there must be positive definite matrices P and Q such that $A^T P + PA = -Q$ (the *Lyapunov-equation*). If P is a diagonal matrix, A is said to be *diagonally stable* [33].

It is important to mention that the strict positivity constraint on c_i can be somewhat relaxed in the following way [14]: if the equations of the model (4) are ordered in such a way that the first n rows of B are linearly independent, then $c_i > 0$ for $i = 1, \ldots, n$ and $c_j = 0$ for $j = n+1, \ldots, m$ still guarantee global stability.

It is examined and proved in [14] and [25] that the global stability of (5) with Lyapunov function (20) implies the boundedness of solutions and global stability of the original QP system (4). It is stressed that *global stability is restricted to the positive orthant int*$(\mathbb{R}^n_+)$ *only for QP and LV models*, because it is their original domain (see the definition in (4)).

It is also important that the global stability of the equilibrium points of (4) with Lyapunov function (20) does not depend on the value of the vector L as long as the equilibrium points are in the positive orthant [14]. This fact will allow us to place the equilibrium point of the closed loop system during the stabilizing controller design (see section 4).

The possibilities to find a Lyapunov function that proves the global asymptotic stability of a QP system can be increased by using time-reparametrization [O4], that is described later in section 3.2.

3.1.2 Zero dynamics analysis

The results in this part indicate that a fortunate choice of a QP-type feedback can simplify the dynamics of a closed-loop system in such a way that the number of quasi-monomials may drastically decrease.

Let us consider a SISO input-affine QP-model in the form of (7) with $p = 1$ and with the simplest output $y = x_i - w^*$ for some i and $w^* > 0$, i.e. we want to keep the system's output being equal to a state variable at a positive constant value. Moreover, let us assume that the relative degree of the system equals one and $g_{i1}(x) = g_i(x) = \prod_{j=1}^n x_j^{\gamma_{ji}}$, i.e. the input function is of quasi-monomial type (see Appendix A.1.2). Then the output zeroing input is given in the form

$$u(t) = -\frac{L_f h(x)}{L_g h(x)} = -\frac{f_i(x)}{\prod_{j=1}^n x_j^{\gamma_{ji}}}. \tag{23}$$

It is seen that the output zeroing input above is in QP-form if $f_i(x)$ is in QP-form.

In order to obtain the zero dynamics (see Appendix A.1.2, or [29]), one has to substitute the input (23) to the state equation (7) to obtain an *autonomous* system model. It is easy to compute that the *resulting zero dynamics system model will remain in QP-form with an output zeroing input in QP-form* [O5].
Therefore the stability analysis of the zero dynamics can be investigated using the methods described earlier in section 3.1.1.

The above result can be easily generalized to the case of output functions in quasi-monomial form.

3.1.3 Zero dynamics of the simple fermentation process

In what follows a slightly different version of (14) is examined where the input is the flowrate F. The values of S_F, and X_F are the constant steady state values of them in (18). The zero dynamics analysis for the fermentation example can be performed e.g. by using the output

$$y = x_2 - x_2^*,$$

i.e. the centered substrate concentration. The output zeroing input can be easily computed:

$$F = \frac{\mu_{max} x_2^* V}{Y(S_F - x_2^*)} x_1 \eta \tag{24}$$

If the above equations are substituted into the QP-form, one gets the following zero dynamics

$$\dot{x_1} = x_1 \left(\frac{\mu_{max} x_2^*}{K_2 x_2^{*2} + x_2^* + K_1} - \frac{x_2^* \mu_{max}}{Y(S_F - x_2^*)(K_2 x_2^{*2} + x_2^* + K_1)} x_1 \right) \tag{25}$$

with QP matrices A', B' and λ' being the following ones:

$$A' = \left[-\frac{x_2^* \mu_{max}}{Y(S_F - x_2^*)(K_2 x_2^{*2} + x_2^* + K_1)} \right] = \left[-0.1640 \right],$$

$$B' = \left[1 \right], \quad \lambda' = \left[\frac{\mu_{max} x_2^*}{K_2 x_2^{*2} + x_2^* + K_1} \right] = \left[0.8022 \right],$$

Hence, the only monomial of the zero dynamics is

$$X$$

Note that *the number of quasi-monomials has been drastically reduced.*

In order to study the local stability of the zero dynamics, we first computed the eigenvalue (i.e. the value) of the Jacobian of the zero dynamics at the equilibrium point x_1^* that is

$$-0.8022$$

Thereafter the feasibility of the LMI (22) was investigated using the LMI Toolbox in Matlab [60] for global stability analysis. The result of the LMI is the following Lyapunov function parameter matrix:

$$C = \left[2.7642 \right]$$

Therefore the global stability of the zero dynamics is proved through the QP description. This result is in good agreement with [58] where the stability of the zero dynamics was proved through nonlinear coordinates-transformations.

3.2 Time-reparametrization [O4]

It was shown in [O4], that the significance of time-reparametrization is that it largely extends the possibilities to prove the global stability of a QP system (see e.g. [14]). As we will see on the examples in section 3.2.4, there are cases when the invariant matrix M of the system itself is not *diagonally stabilizable* (see section 3.1.1), but with an appropriate time-reparametrization, it is possible to find a Lyapunov function of the form (21) for the transformed (reparametrized) model.

3.2.1 The time-reparametrization transformation

Let $\omega = [\omega_1 \ \ldots \ \omega_n]^T \in \mathbb{R}^n$. It is shown e.g. in [14] that the following reparametrization of time

$$dt = \prod_{k=1}^{n} x_k^{\omega_k} dt' \tag{26}$$

transforms the original QP system into the following (also QP) form

$$\frac{dx_i}{dt'} = x_i \sum_{j=1}^{m+1} \tilde{A}_{i,j} \prod_{k=1}^{n} x_k^{\tilde{B}_{j,k}}, \ i = 1, \ldots, n \tag{27}$$

where $\tilde{A} \in \mathbb{R}^{n \times (m+1)}$, $\tilde{B} \in \mathbb{R}^{(m+1) \times n}$ and

$$\tilde{A}_{i,j} = A_{i,j}, \quad i = 1, \ldots, n; \ j = 1, \ldots, m \tag{28}$$
$$\tilde{A}_{i,m+1} = \lambda_i, \quad i = 1, \ldots, n \tag{29}$$

and

$$\tilde{B}_{i,j} = B_{i,j} + \omega_j, \quad i = 1, \ldots, m; \ j = 1, \ldots, n \tag{30}$$
$$\tilde{B}_{m+1,j} = \omega_j, \quad j = 1, \ldots, n. \tag{31}$$

It can be seen that the number of monomials is increased by one and vector $\tilde{\lambda}$ is zero in the transformed system.

Special case

A special case of the time-reparametrization or new time transformation occurs when the following relation holds:

$$\omega^T = -b_j, \quad 1 \leq j \leq m, \tag{32}$$

where b_j is an arbitrary row of the B matrix of the original system (4). From (30) and (31) we can see that in this case the j-th row of $\tilde{B}$ is a zero vector. This means

that the number of monomials in the transformed system (27) remains the same as in the original QP system (4) and a nonzero $\tilde{\lambda}$ vector that is equal to the j-th column of A appears in the transformed system (for an example, see [14]).

In this case, the ω vector can take only m possible different values (see (32)), therefore the stability analysis of the transformed system reduces to the feasibility check of m different LMIs of the form (22). However, our approach treats the ω vector as part of the unknowns to be determined, therefore from now on we will only consider the generic case discussed in section 3.2.

3.2.2 Properties of the time-reparametrization transformation

The most important properties of the time-reparametrization transformation that are used for analyzing local and global stability are as follows.

Monomials

The set of monomials $p_1, \ldots, p_{m+1}$ for the reparametrized system can be written up in terms of the original monomials:

$$p_j = \prod_{k=1}^{n} x_k^{\omega_k} \cdot \prod_{k=1}^{n} x_k^{B_{j,k}} = \prod_{k=1}^{n} x_k^{B_{j,k}+\omega_k}, \quad j = 1, \ldots, m$$

and

$$p_{m+1} = \prod_{k=1}^{n} x_k^{\omega_k}$$

or using a shorter notation:

$$p_j = r \cdot z_j, \quad j = 1, \ldots, m$$

$$p_{m+1} = r$$

where z_j is given in (6) and

$$r = \prod_{k=1}^{n} x_k^{\omega_k}$$

Equilibrium points

Since the equations of the reparametrized system (27) can be written as

$$\frac{dx_i}{dt'} = x_i \left(\lambda_i + \sum_{j=1}^{m} A_{i,j} \prod_{k=1}^{n} x_k^{B_{j,k}} \right) \prod_{k=1}^{n} x_k^{\omega_k}, \quad i = 1, \ldots, n \tag{33}$$

and we assume that $x_i > 0$, $i = 1, \ldots, n$, it is clear that the equilibrium point x^* of the original QP system (4) is also an equilibrium point of the reparametrized system (33).

Local stability

Let us denote the Jacobian matrix of the original QP system (4) at the equilibrium point by $J(x^*)$. Then the Jacobian matrix of the time reparametrized QP system at the equilibrium point can be computed by using the formula described in [12]:

$$\tilde{J}(x^*) = X^* \cdot \tilde{A} \cdot \tilde{Z}^* \cdot \tilde{B} \cdot (X^*)^{-1} = r^* \cdot J(x^*) = \prod_{k=1}^{n} {x_k^*}^{\omega_k} \cdot J(x^*), \tag{34}$$

where

$$\tilde{Z}^* = diag(p_1^*, \ldots, p_m^*, p_{m+1}^*) \quad , \quad X^* = diag(x_1^*, \ldots, x_n^*)$$

are the quasi-monomials of the time-reparametrized system and the system variables in the equilibrium point. From (34) one can see that (as we naturally expect) local stability is not affected by the time-reparametrization, because this transformation just multiplies the eigenvalues of the Jacobian by a positive constant r^*.

Global stability

Rewriting (26) gives

$$\frac{dt}{dt'} = \prod_{k=1}^{n} (x_k(t'))^{\omega_k} \tag{35}$$

from which we can see that t is a strictly monotonously increasing continuous and invertible function of t'. This means that global stability of the QP system in the reparametrized time t' is equivalent to global stability in the original time scale t.

3.2.3 The time-reparametrization problem as a bilinear matrix inequality

We denote an $n \times m$ matrix containing zero elements by $0^{n \times m}$. Let us define two auxiliary matrices by extending A with a zero column and B with a zero row, i.e.

$$\bar{A} = \left[\; A \; \middle| \; 0^{n \times 1} \; \right] \in \mathbb{R}^{n \times (m+1)}, \tag{36}$$

and

$$\bar{B} = \left[\begin{array}{c} B \\ \hline 0^{1 \times n} \end{array} \right] \in \mathbb{R}^{(m+1) \times n}. \tag{37}$$

Then $\tilde{A}$ and $\tilde{B}$ can be written as

$$\tilde{A} = [A|L] = \bar{A} + [0^{n \times m}|L], \tag{38}$$

and

$$\tilde{B} = \begin{bmatrix} b_1 + \omega^T \\ b_2 + \omega^T \\ \vdots \\ b_m + \omega^T \\ \omega^T \end{bmatrix} = \bar{B} + S \cdot \Omega \tag{39}$$

where

$$\Omega = \mathrm{diag}(\omega) \in \mathbb{R}^{n\times n} \tag{40}$$

and

$$S = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & & & \\ 1 & 1 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times n} \tag{41}$$

It can be seen from (38) and (39) that the invariant matrix of the reparametrized system is

$$\tilde{M} = \tilde{B}\cdot\tilde{A} = (\bar{B} + S\cdot\Omega)\cdot\tilde{A} \tag{42}$$

Therefore the matrix inequality for examining the global stability of the reparametrized system is the following

$$-C < 0 \tag{43}$$

$$\tilde{M}^T\cdot C + C\cdot\tilde{M} \leq 0 \tag{44}$$

i.e.

$$-C < 0 \tag{45}$$

$$\tilde{A}^T(\bar{B}^T + \Omega S^T)C + C(\bar{B} + S\Omega)\tilde{A} \leq 0 \tag{46}$$

which clearly has the same form as (22), but with the following set of unknowns:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{m+n+1} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{m+1} \\ \omega_1 \\ \vdots \\ \omega_n \end{bmatrix}, \tag{47}$$

that makes it a BMI (see Appendix A.4.1 for the properties of BMIs). Now we are ready to construct the parameter matrices in the BMI (158) starting with

$$G_0^1 = G_0^2 = 0^{(m+1)\times(m+1)}, \tag{48}$$

$$G_{ki,j}^1 = \begin{cases} -1, & i=j=k \\ 0, & \text{otherwise} \end{cases} \tag{49}$$
$$i,j,k = 1,\dots,m+1,$$

$$G_k^1 = 0^{(m+1)\times(m+1)}, \quad k = m+2,\dots,m+n+1 \tag{50}$$

and

$$K_{kl}^1 = 0^{(m+1)\times(m+1)}, \quad k,l = 1,\dots,m+n+1. \tag{51}$$

Furthermore, let us introduce the following notations

$$\begin{aligned} &P_k \in \mathbb{R}^{(m+1)\times(m+1)}, \\ &P_{ki,j} = \begin{cases} \bar{B} \cdot \tilde{A}_{i,j}, & i = k \\ 0, & i \neq k \end{cases}, \\ &i, j, k = 1, \ldots, m+1 \end{aligned} \tag{52}$$

and

$$\begin{aligned} &Q_{kl} \in \mathbb{R}^{(m+1)\times(m+1)}, \\ &Q_{kli,j} = \begin{cases} \tilde{A}_{l-m-1,j}, & i = k \\ 0, & i \neq k \end{cases}, \\ &i, j, k = 1, \ldots, m+1, \ l = m+2, \ldots, m+n+1. \end{aligned} \tag{53}$$

Then

$$G_k^2 = \begin{cases} P_k + P_k^T, & k = 1, \ldots, m+1 \\ 0^{(m+1)\times(m+1)}, & k = m+2, \ldots, m+n+1, \end{cases}, \tag{54}$$

and

$$\begin{aligned} &K_{kl} = \begin{cases} Q_{kl} + Q_{kl}^T, & k = 1, \ldots, m+1, \ l = m+2, \ldots, m+n+1 \\ 0^{(m+1)\times(m+1)}, & \text{otherwise} \end{cases} \\ &k, l = 1, \ldots, m+n+1. \end{aligned} \tag{55}$$

We note that in certain cases the feasibility of a BMI can be traced back to the feasibility of equivalent LMIs (see [6] or [56]), but in our case it is not possible because of the structural (diagonality) constraint on both of the unknown matrices Ω and C in (46).

3.2.4 Examples

In order to illustrate the above proposed method of finding time-reparametrization transformations for global stability analysis, two simple examples are presented.

Example with a full rank M matrix

Consider a QP system with the following matrices

$$A = \begin{bmatrix} \frac{2}{3} & -\frac{8}{3} \\ \frac{2}{3} & -\frac{7}{3} \end{bmatrix} \approx \begin{bmatrix} 0.6667 & -2.6667 \\ 0.6667 & -2.3333 \end{bmatrix} \tag{56}$$

$$B = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{8}{3} & \frac{16}{3} \end{bmatrix} \approx \begin{bmatrix} 0.6667 & -0.3333 \\ -2.6667 & 5.3333 \end{bmatrix} \tag{57}$$

$$L = \begin{bmatrix} 2 \\ \frac{5}{3} \end{bmatrix} \approx \begin{bmatrix} 2 \\ 1.6667 \end{bmatrix} \tag{58}$$

Its equilibrium point of interest is:

$$x^* = [1 \ \ 1]^T \tag{59}$$

The Jacobian matrix of the locally linearized system in x^* has the following eigenvalues: -0.1187, -4.9924. This shows that the investigated equilibrium point is at least locally asymptotically stable.

Using an appropriate LMI solver (e.g. Matlab's LMI Control Toolbox) it can be checked that the LMI (22) cannot be solved for $M = B \cdot A$. However, using the algorithm [36] for solving the corresponding BMI we find that a feasible solution of (46) is e.g.

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \omega = \begin{bmatrix} \frac{2}{3} & -\frac{5}{3} \end{bmatrix}^T \tag{60}$$

The eigenvalues of $\tilde{M}^T \cdot C + C \cdot \tilde{M}$ are

$$\lambda_1 = 0, \quad \lambda_2 \approx -0.2374, \quad \lambda_3 \approx -9.9848, \tag{61}$$

which shows that the examined system is globally stable.

An example in which time-reparametrization transformation is applied for a system for which the matrix M is rank deficient is given in Appendix A.3.1.

3.3 Summary

The main contribution of the chapter is first of all that *the non-increasing nature of a QP or LV system's Lyapunov function is equivalent to a linear matrix inequality, thus the stability analysis of these systems (and general nonlinear process systems embedded into QP form) is equivalent to the feasibility of a linear matrix inequality.* Even if a system is globally asymptotically stable, one not always finds a Lyapunov function proving the global stability.

It was also shown in section 3.2, that *the time-reparametrization transformation introduces a re-scaling in the QP system's quasi-monomials, such that the global stability of transformed QP system is equivalent to that of the original one. This way, the global stability analysis can be extended to a wider class of QP systems by embedding the parameters of the time-reparametrization transformation into the global stability analysis, this way one has to solve a bilinear matrix inequality.*

Chapter 4

Stabilizing control of quasi-polynomial systems

The most widely used method for the control of nonlinear process systems is the use of model predictive controllers [15], which offers an optimization based solution of the control problem. Meanwhile, techniques of modern nonlinear control plays only a limited role in the field of process engineering, although there are a few results on this topic (e.g. [40]).

Present chapter proposes results on globally stabilizing feedback design for QP systems utilizing the results of chapter 3.

Embedding the process system into QP form (see section 2.2), and applying state feedback that preserves the QP-form of the closed loop system, its global stability can be conveniently investigated by using LMIs if the feedback parameters are known and fixed. If the feedback parameters are not fixed, then a feedback design problem is defined that globally stabilizes the closed loop system, that is the subject of section 4.1.

Unfortunately, the solution of the feedback design problem does not automatically provide tools for the design of the steady-state point of the system. Therefore, the basic conditions of steady-state point placing are discussed in section 4.3. Finally, some additional structural feedback design results are presented in section 4.4.

4.1 The controller design problem [O5]

The globally stabilizing QP state feedback design problem for QP systems can be formulated as follows. Consider arbitrary quasi-polynomial inputs in the form:

$$u_l = \sum_{i=1}^{r} k_{il}\hat{q}_i, \quad l = 1\ldots, p \tag{62}$$

where $\hat{q}_i = \hat{q}_i(x_1, \ldots, x_n),\ i = 1, ..., r$ are arbitrary quasi-monomial functions of the state variables of (7) and k_{il} is the constant gain of the quasi-monomial function $\hat{q}_i$ in the l-th input u_l. The closed loop (autonomous) system will also be a QP system

with matrices

$$\hat{A} = A_0 + \sum_{l=1}^{p} \sum_{i=1}^{r} k_{il} A_{il}, \quad \hat{B}, \tag{63}$$

$$\hat{L} = L_0 + \sum_{l=1}^{p} \sum_{i=1}^{r} k_{il} L_{il}. \tag{64}$$

Note that the number of quasi-monomials in the closed-loop system (i.e. the dimension of the matrices) together with the matrix $\hat{B}$ may significantly change depending on the choice of the feedback structure, i.e. on the quasi-monomial functions $\hat{q}_i$.

Furthermore, the closed loop LV coefficient matrix $\hat{M}$ can also be expressed in the form:

$$\hat{M} = \hat{B} \cdot \hat{A} = M_0 + \sum_{l=1}^{p} \sum_{i=1}^{r} k_{il} M_{il}.$$

Then the global stability analysis of the closed loop system with unknown feedback gains k_{il} leads to the following bilinear matrix inequality (see (158) in Appendix A.4.1)

$$\hat{M}^T C + C\hat{M} = M_0^T C + C M_0 + \sum_{l=1}^{p} \sum_{i=1}^{r} k_{il} \left(M_{il}^T C + C M_{il} \right) \leq 0. \tag{65}$$

The variables of the BMI are the $p \times r$ k_{il} feedback gain parameters and the c_j, $j = 1, .., m$ parameters of the Lyapunov function. If the BMI above is feasible then there exists a globally stabilizing feedback with the selected structure.

4.2 Numerical solution of the controller design problem [O11]

This section deals with the numerical aspects of the globally stabilizing controller design problem.

4.2.1 Numerical solution based on bilinear matrix inequalities

There are just few software tools available for solving general bilinear matrix inequalities that is a computationally hard problem. In some rare fortunate cases with a suitable change of variables quadratic matrix inequalities can be rewritten as linear matrix inequalities (see e.g. [6]). Unfortunately, the structure of the matrix variable of (65) does not fall into this fortunate problem class, so the previously mentioned idea cannot be used.

Rewriting the above matrix inequality (65) in the form (158) one gets the following expression which can be directly solved by [37] as a BMI feasibility problem:

$$\begin{aligned} \sum_{j=1}^{m} c_j \bar{M}_{0,j} + \sum_{j=1}^{m}\sum_{l=1}^{p}\sum_{i=1}^{r} c_j k_{il} \bar{M}_{il,j} &\leq 0 \\ -c_1 &< 0 \\ &\vdots \\ -c_m &< 0. \end{aligned} \tag{66}$$

The two disjoint sets of BMI variables are the c_j parameters of the Lyapunov function and the k_{il} feedback parameters. The parameters of the problem $\bar{M}_{0,j}$ ($\bar{M}_{il,j}$, respectively) are the symmetric matrices obtained from M_0 (M_{il}, respectively) by adding the $m \times m$ matrix that contains only the j-th column of M_0 (M_{il}, respectively) to its transpose:

$$M = \begin{bmatrix} m_{11} & \cdots & m_{1j} & \cdots & m_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ m_{m1} & \cdots & m_{mj} & \cdots & m_{mm} \end{bmatrix}$$

$$\downarrow$$

$$\bar{M}_j = \begin{bmatrix} 0 & \cdots & m_{1j} & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ m_{1j} & \cdots & 2m_{jj} & \cdots & m_{mj} \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & m_{mj} & \cdots & 0 \end{bmatrix}.$$

Note that for low dimensions (i.e. for $m < 3$) there are practically feasible methods for circumventing the BMI feasibility problem [33] but these cannot be extended to the practically important higher dimensional case.

4.2.2 An iterative LMI approach to controller design problem

Because of the NP-hard nature of the general BMI solution problem, it is worthwhile to search for an approximate but numerically efficient alternative way of solution. As shown below, the special structure of the QP stabilizing feedback design BMI feasibility problem allows us to apply a computationally feasible method for its solution that solves an LMI in each of its iterative approximation step. The already existing iterative LMI (ILMI) algorithm of [8] used for static output feedback stabilization (see e.g. in [8]) will be used for this purpose.

In order to be able to use the ILMI algorithm, it is necessary to write up the QP stabilizing feedback design problem as a static output feedback stabilization problem for LTI systems. In what follows the globally stabilizing feedback design BMI (65) is used in the form

$$(M_0 + \Theta K)^T C + C(M_0 + \Theta K) < 0. \tag{67}$$

where

$$\Theta = \left[\overbrace{M_1, \ldots, M_p}^{1\text{st}}, \ldots, \overbrace{M_1, \ldots, M_p}^{r\text{th}} \right], \qquad K = \begin{bmatrix} k_{11} \cdot I_{m\times m} \\ \vdots \\ k_{1p} \cdot I_{m\times m} \\ \vdots \\ k_{r1} \cdot I_{m\times m} \\ \vdots \\ k_{rp} \cdot I_{m\times m} \end{bmatrix}.$$

The above problem is equivalent to a LTI output feedback stabilization problem

$$(A + BFC)^T P + P(A + BFC) < 0$$

with M_0 corresponding to the state matrix A, Θ playing the role of the input matrix B, and K serving as FC and P is the unknown matrix variable of the problem. It is apparent that the matrix parameters and variables have a special structure for quasi-polynomial systems.

The ILMI algorithm does not aim at finding the complete feasible set of the BMI (67) but computes an optimal solution point with minimal trace of C if the BMI is feasible. The ILMI algorithm solves a linear objective function minimizing LMI and a generalized eigenvalue problem in each step. The scheme of the algorithm is the following:

Step 1: Let $Q > 0$, the parameter of the algorithm. Solve the Riccati equation

$$M_0^T C + C M_0 - C\Theta\Theta^T C + Q = 0, \tag{68}$$

for C (not necessarily diagonal).

$$i = 1, \quad X_1 = C.$$

Step 2: Solve the following optimization problem for C_i, K and α_i:
Minimize α_i subject to the LMI constraint

$$\begin{bmatrix} M_0^T C_i + C_i M_0 - X_i\Theta\Theta^T C_i - C_i\Theta\Theta^T X_i + X_i\Theta\Theta^T X_i - \alpha_i C_i & (\Theta^T C_i + K)^T \\ \Theta^T C_i + K & -I \end{bmatrix} < 0,$$

$$C_i = \mathrm{diag}(c_{i1}, \ldots, c_{im}) > 0 \tag{69}$$

α_i^* denotes the minimized α_i.

Step 3: If $\alpha_i^* \leq 0$, K is a stabilizing feedback gain. STOP.

Step 4: Solve the following optimization problem for C_i and K:
Minimize trace(C_i) subject to the LMI constraints (69) using $\alpha_i = \alpha_i^*$. Denote C_i^* as the C_i that minimizes trace(C_i).

Step 5: If $\|X_i - C_i^*\| < \delta$, GOTO Step 6. Else set $i = i + 1$ and $X_i = C_i^*$ and GOTO Step 2.

Step 6: The system may not be stabilizable by a quasi-polynomial feedback. STOP.

It is important to note that for QP systems with rank deficient $M_0 = B \cdot A$ some additional techniques are needed because the algorithm fails for singular M_0 matrices. One possible way is using singular perturbation on M_0:

$$\tilde{M}_0 = M_0 - \varepsilon \cdot I_{m \times m}, \qquad \varepsilon > 0.$$

If this way $(\tilde{M}_0, \Theta)$ become stabilizable then the algorithm can be applied.

According to [8] the algorithm is convergent although sometimes we may not achieve a solution because α not always converges to its minimum. The proper selection of initial Q affects the convergence of the algorithm, a suitable selection of Q that guarantees the immediate convergence can be found in [8]. Based on the above, the algorithm is used as an off-the-shelf tool, that's why no numerical analysis is presented here.

It is important to emphasize here, that the computationally feasible ILMI algorithm can be used to test the feasibility of the associated BMI, and then the final design can be performed by a constrained optimization method using a suitable controller performance criterion in the feasible case.

4.3 Equilibrium points [O11]

After solving the globally stabilizing feedback design BMI the resulting Lotka-Volterra system has a globally asymptotically stable equilibrium point in the positive orthant. This steady-state equilibrium point x^* can be determined from the steady-state version of the closed loop quasi-polynomial system (4)

$$0 = x_i \left(\hat{\lambda}_i + \sum_{j=1}^{m} \hat{A}_{ij} \prod_{k=1}^{n} x_k^{\hat{B}_{jk}} \right), \quad i = 1, \ldots, n. \tag{70}$$

By excluding the non strictly positive equilibrium states one only has to deal with the equation:

$$0 = \hat{\lambda}_i + \sum_{j=1}^{m} \hat{A}_{ij} \prod_{k=1}^{n} x_k^{\hat{B}_{jk}}, \quad i = 1, \ldots, n \tag{71}$$

where the parameters $\hat{\lambda}_i$ and $\hat{A}_{ij}$ depend linearly on the feedback parameters according to the equations (63) and (64).

However, with the BMI (65) it is not possible to prescribe the equilibria of the closed loop system but only to globally stabilize it. So it is necessary to introduce extra parameters to the feedback in order to be able to place the positive steady state point anywhere in the positive orthant as needed. The feedback structure has to be constructed in a way that the parameters that are used in the *steady state point placing problem* appear in the vector $\hat{\lambda}$ of the closed loop quasi-polynomial

system. This way the parameters of the equilibrium placing are separated from the stabilizing feedback design BMI's parameters. The feedback has the form

$$u = K(k, x) + D(\delta, x) \tag{72}$$

where $K(k, y)$ is the feedback structure with the parameters for the BMI, and $D(\delta, y)$ has the form so that the components of the parameter vector δ appear in the vector λ of the closed loop QP system. It is important to note that the QP input (62) is linear in both of the parameters k and δ.

One can further simplify the QP input structure (62) for process systems if the input variables are selected to be the intensive variables at the inlet, i.e. $g_i(x) = const^*$ (see section 2.2.1). Then we can use a linear term $D_i(\delta_i, x_i) = \delta_i x_i$ in the feedback (72) to take care of the placing of the steady-state point, and the other term for stabilizing the closed loop system.

4.3.1 Fully actuated case

In this case the QP system has at least one designated input for each of the n state equations. The steady state point of these systems can be put anywhere in the positive orthant.

$$0 = \lambda_i(\delta) + \sum_{j=1}^{m} A_{ij} \prod_{k=1}^{n} x_k^{*B_{jk}}, \quad i = 1, \ldots, n \tag{73}$$

where $\lambda_i(\delta)$ is a linear function of the δ parameters of the problem and $x^* = (x_1^*, \ldots, x_n^*)^T$ is the desired equilibrium. That is, δ can be determined from a linear system of equations.

4.3.2 Partially actuated case

If the system has $k < n$ different inputs, then there are no general results for QP models. However, in the Lotka-Volterra case there is some possibility of shifting some components of the equilibrium point. If the LV coefficient matrix M can be transformed into an upper block triangular matrix by row and column changes then it means that the first k coordinates of the equilibrium point can be prescribed at will independently of the remaining $n - k$.

Note that if the system does not belong to the above two classes then it is not possible to redesign its equilibrium point with the above technique.

4.3.3 Rank deficient (embedded) systems

In case of systems that are not originally in quasi-polynomial form (see section 2.2 for embedding into QP-form) all the above hold with some specialities. It is known that for such QP systems that their trajectories range only a lower dimensional manifold of the QP state space and their parameter matrix A is rank deficient. With this understanding one has to design the equilibrium point of the system (if it is possible to design at all, see section 4.3.2) into this lower dimensional manifold.

4.4 Feedback structure selection [O11]

Of course, the feedback structure selection affects heavily the solution of the BMI. The results of zero dynamics analysis of QP-systems [O5] indicate that a fortunate choice of a QP-type feedback can simplify the dynamics of a closed-loop system in such a way that the number of quasi-monomials may drastically decrease. This way the dimension of the LV system, and the size of the BMI to be solved can also be drastically reduced.

In certain special cases it is possible to change the entire system dynamics to a desired one while this possibility depends on the number of available inputs.

4.4.1 Fully actuated case

Suppose, that we have an input affine QP system in the form:

$$\begin{aligned} \dot{x}_i &= f_i(x) + g_i(x)u_i = x_i \left(\lambda_{0_i} + \sum_{j=1}^{m} A_{0_{ij}} \prod_{k=1}^{n} x_k^{B_{jk}} \right) + \\ &+ x_i \left(\lambda_{i_i} + \sum_{j=1}^{m} A_{i_{ij}} \prod_{k=1}^{n} x_k^{B_{jk}} \right) u_i, \quad i = 1 \ldots, n, \end{aligned} \tag{74}$$

i.e. every equation has a *designated input*. Suppose in addition that *the desired closed-loop system dynamics is given in the form*:

$$\dot{x}_i = h_i(x), \quad i = 1, \ldots, n \tag{75}$$

where h_i are quasi-polynomial functions.

It is obvious that (74) can be transformed into (75) with the following feedback structure:

$$u_i = -\frac{f_i(x)}{g_i(x)} + \frac{h_i(x)}{g_i(x)}, \quad g_i(x) \neq 0. \tag{76}$$

It can be seen that in general case the expression fed back to the input is not a QP, but a rational function.

Fortunately, the input function g_i in the denominator of the above formulae (76) is a simple linear function $g_i(x) = const \cdot x_i$ or $g_i(x) = const^*$ for process systems (see section 2.2.1), therefore the feedback remains a QP function for process systems implying the closed-loop system dynamics to remain in the QP system class.

4.4.2 Partially actuated case

The other case is when there are not as many *different* inputs as equations i.e. the QP system can be arranged into the form

$$\begin{aligned}\dot{x}_p &= f_i(x) + g_p(x)u_p = x_p\left(\lambda_{0_p} + \sum_{j=1}^{m} A_{0_{pj}} \prod_{k=1}^{n} x_k^{B_{jk}}\right) + \\ &+x_p\left(\lambda_{p_p} + \sum_{j=1}^{m} A_{p_{pj}} \prod_{k=1}^{n} x_k^{B_{jk}}\right) u_p, \quad p = 1\ldots, k \\ \dot{x}_q &= f_q(x) = x_q\left(\lambda_{0_q} + \sum_{j=1}^{m} A_{0_{qj}} \prod_{k=1}^{n} x_k^{B_{jk}}\right), \quad q = k+1\ldots, n.\end{aligned} \tag{77}$$

This way only the first k equations can be modified freely:

$$u_p = -\frac{f_p(x)}{g_p(x)} + \frac{h_p(x)}{g_p(x)}, \quad g_p(x) \neq 0, \quad p = 1, \ldots, k.$$

The closed loop system with the above structure is

$$\begin{aligned}\dot{x}_p &= h_p(x), \quad p = 1, \ldots, k \\ \dot{x}_q &= f_q(x), \quad q = k+1, \ldots, n.\end{aligned} \tag{78}$$

4.4.3 Degenerated case

When there is an input that is assigned to more than one equations the above change of dynamics cannot be used in general. Choosing one equation to change with the input one can destroy the QP form of the other equations having the same input. Of course in special cases it is possible to have useful results, for example in the case of zero dynamics (see section 3.1.3 , or [O5]).

4.5 Examples

In the following, some simple process system examples are proposed for the BMI based stabilizing controller design problem discussed so far. The first two are simple continuously stirred tank reactor (CSTR) examples with second order chemical reactions where the system model is naturally in a QP-form.

A final example is described in Appendix A.3.2 that is similar to the fermentation example described in section 2.2.2 and has an embedded rank deficient QP model but in this case the reaction kinetics is a little bit simpler.

4.5.1 Partially actuated process system example in QP-form

The system of this example is a simpler variant of the fermentation process of section 2.2.2 with S_F being the manipulable input:

$$\begin{aligned} \dot{x}_1 &= \mu_{max} x_1 x_2 - \tfrac{F}{V} x_1 \\ \dot{x}_2 &= -\tfrac{\mu_{max}}{Y} x_1 x_2 + \tfrac{F}{V}(S_F - x_2). \end{aligned} \tag{79}$$

The parameter values can be seen in Table 7. The quasi-polynomial form of the model is:

$$\begin{aligned} \dot{x}_1 &= x_1 (S - 2) \\ \dot{x}_2 &= x_2 \left(-x_1 + 2x_2^{-1} S_F - 2\right). \end{aligned} \tag{80}$$

For a fixed value of the substrate concentration $S_F^* = 1$, the system has an asymptotically stable wash-out type equilibrium point

$$\begin{bmatrix} x_1^* \\ x_2^* \\ S_F^* \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}.$$

The feedback structure was chosen to be

$$S_F = k_1 x_2^2 + \delta_1 x_2.$$

The closed loop system with the above structure is

$$\begin{aligned} \dot{x}_1 &= x_1 (x_2 - 2) \\ \dot{x}_2 &= x_2 \left(-x_1 + 2k_2 x_2 + 2(\delta_1 - 1)\right). \end{aligned} \tag{81}$$

It is apparent, that the above QP model (81) is also the Lotka-Volterra model of the system. The LV matrices of the system are the following ones:

$$M = \begin{bmatrix} 0 & 1 \\ -1 & 2k_1 \end{bmatrix}, \quad N = \begin{bmatrix} -2 \\ 2(\delta_1 - 1) \end{bmatrix}.$$

It is noticeable that matrix M is not upper triangular, i.e. the equilibrium cannot be manipulated partially based on the results of section 4.3.2. However, with a fortunate choice of δ_1 (e.g. $\delta_1 = 2.5$) one can modify the value of the (non wash-out type) equilibrium of system (81). It is important to note, that in this case the equilibrium will be positive, but one cannot decide its value.
The other free parameter (k_1) can be used for stabilizing this equilibria. So k_1 and the two parameters of the Lyapunov function are given to the ILMI algorithm. It gives the following results:

$$k_1 = -0.0013, \quad C = \begin{bmatrix} 1.2822 & 0 \\ 0 & 1.2822 \end{bmatrix}.$$

Figure 3. shows the feasibility region of the globally stabilizing BMI problem and the solution given by the ILMI algorithm of section 4.2.2. The obtained feedback

with parameters k_1 and δ_1 globally stabilizes the system in the positive orthant. Indeed, the closed loop system has a unique equilibrium state

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} 2.9948 \\ 2.0000 \end{bmatrix}$$

in the positive orthant $\mathrm{int}(\mathbb{R}^2_+)$, for which the locally linearized system matrix has eigenvalues with strictly negative real part, this way at least local stability can be proved for the equilibrium.

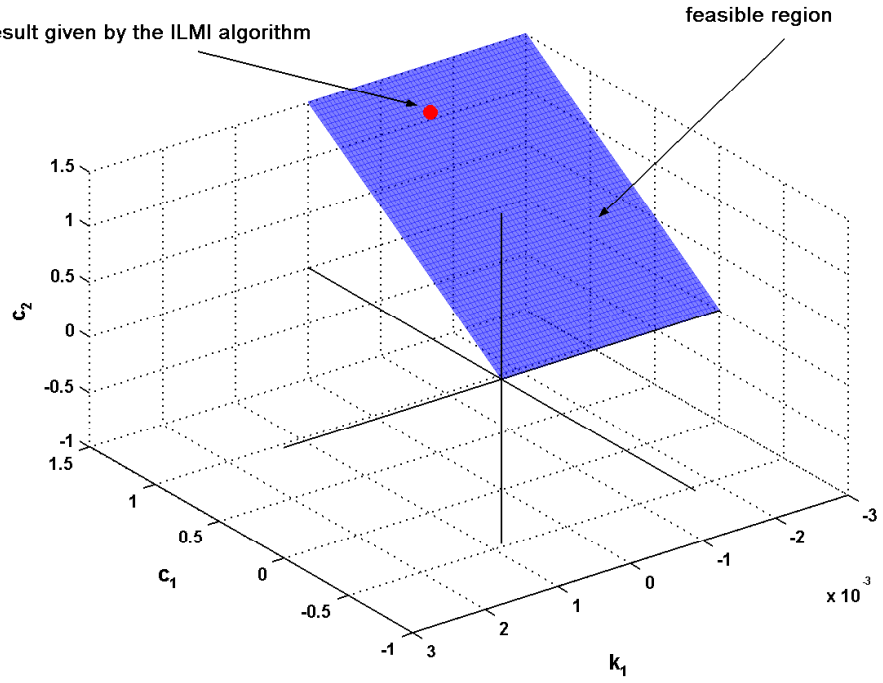

Figure 3: BMI feasibility region for Example 4.5.1

4.5.2 Fully actuated process system example in QP-form

The second process system example is the same fermentation process examined in section 4.5.1 but this time biomass is also fed to the reactor with manipulable inlet concentration X_F. The parameters of the system are the same as in the previous case.

$$\begin{aligned} \dot{x}_1 &= \mu_{max} x_1 x_2 + \tfrac{F}{V}(X_F - x_1) \\ \dot{x}_2 &= -\tfrac{\mu_{max}}{Y} x_1 x_2 + \tfrac{F}{V}(S_F - x_2) \end{aligned} \tag{82}$$

The quasi-polynomial form of the model is:

$$\begin{aligned} \dot{x}_1 &= x_1 \left(x_2 + 2 x_1^{-1} X_F - 2 \right) \\ \dot{x}_2 &= x_2 \left(-x_1 + 2 x_2^{-1} S_F - 2 \right). \end{aligned} \tag{83}$$

Note that (83) is also the Lotka-Volterra model of the system. The manipulable inputs are X_F and S_F. For fixed values of the input concentrations $X_F^* = 0$ and $S_F^* = 1$, the system has no equilibrium in the strictly positive orthant but has one asymptotically stable wash-out equilibrium on the boundary

$$\begin{bmatrix} x_1^* \\ x_2^* \\ X_F^* \\ S_F^* \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

The feedback structure is chosen to be

$$\begin{aligned} X_F &= k_1 x_1^2 + \delta_1 x_1 \\ S_F &= k_2 x_2^2 + \delta_2 x_2. \end{aligned}$$

Parameters k_1 and k_2 are to stabilize the system, δ_1 and δ_2 will be used to shift the equilibrium. The closed loop system is

$$\begin{aligned} \dot{x}_1 &= x_1 \left(2(\delta_1 - 1) + x_2 + 2k_1 x_1\right) \\ \dot{x}_2 &= x_2 \left(2(\delta_2 - 1) - x_1 + 2k_2 x_2\right). \end{aligned}$$

The iterative BMI algorithm yielded the following parameters for the feedback and the Lyapunov function:

$$k_1 = -1.0004, \quad k_2 = -1.0004, \quad C = \begin{bmatrix} 1.0004 & 0 \\ 0 & 1.0004 \end{bmatrix}.$$

We would like to prescribe a strictly positive equilibrium instead of the original one. Suppose that the desired equilibrium is at

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}.$$

Expressing the values of δ_1 and δ_2 from the state equations in which the desired equilibrium point is substituted in yields

$$\delta_1 = 1.2502, \quad \delta_2 = 1.7502.$$

Indeed, the closed loop system with the determined parameters $k_1, k_2, \delta_1, \delta_2$ has an asymptotically stable equilibria in $[\tilde{x}_1, \tilde{x}_2]^T$.

It is apparent that in this example with a higher degree of freedom it was possible to shift the steady state point of the system.

4.6 Summary

In this chapter a *globally stabilizing state feedback design problem was formulated based on the global stability analysis results of section 3.1. The problem was formulated and solved as a bilinear matrix inequality using the tools summarized in Appendix A.4.1. However, an alternative efficient method was also developed by reformulating the problem* so that an existent iterative LMI algorithm is suitable for

its' solution. It is important to note, that the control problem does not use the object function of the bilinear matrix inequality problem, so it is a possible point to introduce some performance or robustness criteria in the future.

The stabilizing state feedback may shift the closed loop system's equilibrium points compared to the original one, so *the conditions and methods of designing an additional feedback that moves the closed loop equilibrium were also developed and described.*

It is vital to take care of the feedback structure, too, since a fortunate choice can notable decrease the dimension of the closed loop Lotka-Volterra system (see section 3.1.3) and the size of the BMI (65) to be solved. Hence, *simple results regarding the selection of controller structure were also presented* here.

Part II

Quantum systems

As it was mentioned in section 1.1, reading quantum information from a quantum computer calls for the solution of a state estimation problem, that's why the second part of the thesis presents different state estimation methods for *quantum mechanical systems*. Most problems of quantum information and communication theory as well as quantum cryptography also needs the ability to determine the result of a computation or the received information. The unusual properties of quantum systems due to quantum measurement makes them very hard to handle with conventional methods. However, our aim is to apply the results of system- and control theory on them.

After a short introduction (chapter 5), the results regarding quantum state estimation of finite level quantum systems are presented. In chapter 6 a Bayesian parameter estimation method (discussed in Appendix A.1.4) is used for estimating the state of simple quantum mechanical systems, while chapter 7 gives an essentially different estimation scheme for general quantum systems.

Chapter 5

Introduction to finite quantum mechanical systems

This chapter summarizes some basic principles of quantum mechanics based on [55], [24]. The notions are formulated in such a way that the system- and control theoretical aspects are emphasized where it is possible.

Of course, it is not possible to give a well established introduction to quantum mechanics within the page constraints of this work. A good introduction to modern quantum mechanics can be found in [55], or [50]. Just the most necessary notions are presented here.

Section 5.1 discusses quantum dynamics together with different representations of quantum states. Section 5.2 deals with the difficulties occurring when a quantum system is being measured. Afterwards, the general state estimation problem (see section 1.2 or Appendix A.1.3) is revisited and relaxed for quantum mechanical systems in section 5.2.3. A short summary of quantum state estimation related works is given in section 5.3.

5.1 States of quantum mechanical systems

First of all, the classical Dirac notation is introduced together with the Schrödinger equation which determines the dynamical change of quantum systems in section 5.1.1. Afterwards, different representations of quantum states are introduced in section 5.1.2 and section 5.1.4 which will be used in the following two chapters.

5.1.1 Quantum dynamics

For quantum mechanical systems the state vectors are in a Hilbert space $\mathcal{H}$, and following Dirac, they are called *ket*s, and denoted by $|x\rangle$.

The evolution of the state ket in time is defined with the so-called *time-evolution operator* $\mathcal{U}(t, t_0)$ [55]:

$$|x(t)\rangle = \mathcal{U}(t, t_0)|x(t_0)\rangle$$

The fundamental differential equation of quantum mechanics, the *Schrödinger equa-*

tion describes the change of $\mathcal{U}(t,t_0)$ in time:

$$i\hbar\frac{d}{dt}\mathcal{U}(t,t_0) = H\mathcal{U}(t,t_0), \tag{84}$$

where H is the *Hamiltonian operator* of the system.

Note the similarity between (84) and (130). Thus, the system theoretical equivalent of the time-evolution operator is the *fundamental matrix* (see Appendix A.1.1) (disregarding $i\hbar$).

The Schrödinger equation (84) can also be written up for the state ket $|x\rangle$:

$$i\hbar\frac{d}{dt}|x(t)\rangle = H|x(t)\rangle \tag{85}$$

which reminds system theorists to the truncated state equation (128). Moreover, the Hamiltonian H acts as the system matrix A of an LTI system (127).

The form of $\mathcal{U}(t,t_0)$ for time independent Hamiltonian operator H is

$$\mathcal{U}(t,t_0) = \exp\left(-\frac{i}{\hbar}(t-t_0)H\right)$$

which also agrees with (129) from Appendix A.1.1.

5.1.2 State representation: the density matrix

For describing the states of *mixed quantum systems* that are prepared by statistically combining two different (pure) states $|x^{(1)}\rangle$ and $|x^{(2)}\rangle$ the state kets cannot be used, so another representation is needed.

The so-called density matrix describes the statistical state of a quantum system in this case. If the state ket vectors are $|x^{(j)}\rangle$, then the density matrix is given by

$$\chi = \sum_j p_j |x^{(j)}\rangle\langle x^{(j)}|, \quad p_j \geq 0, \quad \sum_j p_j = 1, \tag{86}$$

where $\langle x|$ stands for the so-called *bra* operator. (Together with the ket they represent the inner product, i.e. the *bra*-c-*ket*: $\langle x|x\rangle$.) Density matrices χ are statistical operators acting on the Hilbert space $\mathcal{H}$, they are positive semidefinite self-adjoint matrices having unit trace:

$$\chi \in \mathbb{C}^{n\times n}, \quad \chi \geq 0, \quad \chi = \chi^*, \quad \mathrm{Tr}\chi = 1 \tag{87}$$

A density operator describes a pure state if it is a rank one projection, i.e.

$$\chi = \chi^2.$$

From the geometrical point of view the set of mixed states is a convex set, since (86) defines a convex combination of pure states. The pure states are the extremal points of the set of mixed states.

The dynamical change of the system in the density matrix notation is given by

$$\chi(t) = \mathcal{U}(t,t_0)\ \chi(t_0)\ \mathcal{U}(t,t_0)^*.$$

5.1.3 Finite quantum systems

In what follows the dimension of $\mathcal{H}$ is assumed to be finite. The density matrix of an N-level quantum system is an $N \times N$ matrix with properties (87). The most important and the simplest special case is of the two level system, the so-called quantum bit, or *qubit* which is the quantum analogue of the classical bit.

For describing the state of a *quantum word*, i.e. more than one coupled qubits taking place in a register, it is necessary to join several quantum systems to form a composite system.

A quantum system $\mathbf{S}$ created as the composition of quantum systems $\mathbf{S_1}$, and $\mathbf{S_2}$ is described by the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$, where $\mathcal{H}_1$, and $\mathcal{H}_2$ are the Hilbert-spaces corresponding to $\mathbf{S_1}$ and $\mathbf{S_2}$, respectively. The tensor product of two matrices is defined in Appendix A.4.2.

By coupling k quantum systems which are of dimension N a special case of finite quantum systems is generated having dimension N^k. Specially, k coupled qubits can be treated as a 2^k-level quantum system.

5.1.4 Bloch vector

In the case of a qubit i.e. for the density matrices of size 2×2 the so-called *Pauli matrices* together with the 2×2 unit matrix can be used as a basis:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \sigma_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \sigma_2 = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix} \quad \sigma_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Using that σ_1, σ_2 and σ_3 are traceless, all 2×2 density matrices χ can be given in the following form:

$$\chi = \frac{1}{2}(I + x_1\sigma_1 + x_2\sigma_2 + x_3\sigma_3), \tag{88}$$

where $x = (x_1, x_2, x_3)^T \in \mathbb{R}^3$ is called *Bloch vector* (Figure 4). The unit trace of χ is ensured by construction since all 3 Pauli matrices are traceless. The positivity constraint from (87) transforms to a length constraint on x:

$$||x|| = \sqrt{x_1^2 + x_2^2 + x_3^2} \leq 1 \tag{89}$$

The distinction of pure states and mixed states is well defined for the Bloch representation of a qubit. The mixed states are in the bloch ball whilst the Bloch sphere i.e. the surface is populated by pure states, see Figure 4.

The parametrization (88) can be generalized for more qubits, or N-level quantum systems.

Bloch vector of k qubit systems

If the dimension of the system is 2^k, then a natural way of parameterizing the state uses the k-fold tensor product of the Pauli base

$$\chi = \frac{1}{2^k} \sum_{i_1,i_2,\ldots,i_k=0}^{3} x_{i_1,i_2,\ldots,i_k} \cdot \sigma_{i_1} \otimes \sigma_{i_2} \otimes \ldots \otimes \sigma_{i_k}, \qquad x_{0,0,\ldots,0} = 1$$

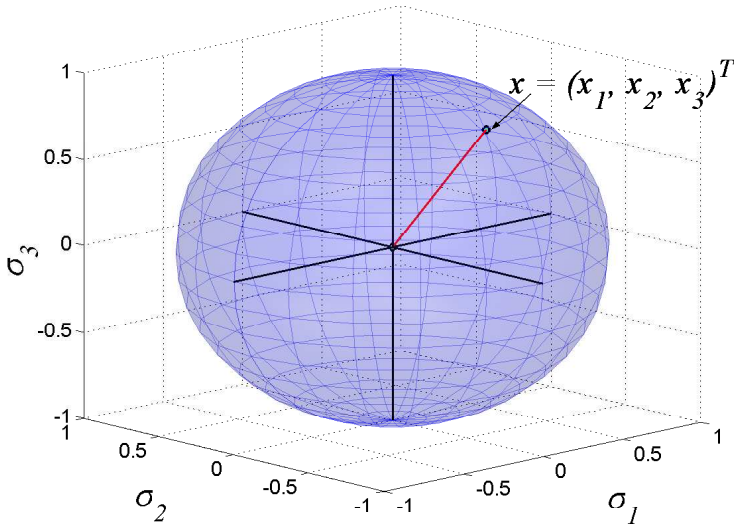

Figure 4: Bloch ball for a quantum bit

i.e it is a multidimensional extension of the Bloch-vector. For 4 level systems the dimension of the Bloch-matrix is 4×4:

$$x = \begin{bmatrix} 1 & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix}$$

That is, the state has $4 \times 4 - 1 = 15$ parameters [35].
The natural structure for the coefficients $x_{i_1,\ldots,i_k}$ of the basis elements is an N-dimensional hyper-matrix of size $4 \times 4 \times \ldots \times 4$, i.e. a *Bloch-hyper-matrix*:

$$x = (x_{i_1,\ldots,i_k})_{i_1,\ldots,i_k=0}^{3}$$

As before, the Bloch-parametrization ensures unit trace for χ since all the Pauli-tensor matrices are traceless except for the unity. The positivity of the density operator of size $2^k \times 2^k$ is still a constraint to be respected.

Bloch vector of N-level systems

The most general parametrization of the density matrix that suits for all quantum systems uses the matrix elements as parameters:

$$\chi = \sum_{k=1}^{N} x_{kk} E_{kk} + \sum_{i<j} \left(x_{ij}(E_{ij} + E_{ji}) + x_{ji}(iE_{ij} - iE_{ji}) \right), \tag{90}$$

where E_{ij} are the matrix units (full of zeros except for the i,j-th element which is one). This way, the state of an N-level system can be given with N^2 real parameters.

It can be seen that in contrast with the Bloch-parametrization this parametrization does not ensure the unit trace for the density, so a reasonable modification is:

$$\chi = \sum_{k=1}^{N-1} x_{kk} E_{kk} + \left(1 - \sum_{k=1}^{N-1} x_{kk}\right) E_{NN} + \sum_{i<j} \left(x_{ij}(E_{ij} + E_{ji}) + x_{ji}(iE_{ij} - iE_{ji})\right). \tag{91}$$

This offers the use of a generalization of the Bloch-vector space [35]. The Bloch-vector of an N-level quantum system is a vector of $\mathbb{R}^{N^2-1}$.

For 3-level systems, the density χ can be written up as

$$\chi = \begin{bmatrix} x_{11} & x_{12} - ix_{21} & x_{13} - ix_{31} \\ x_{12} + ix_{21} & x_{22} & x_{23} - ix_{32} \\ x_{13} + ix_{31} & x_{23} + ix_{32} & 1 - (x_{11} + x_{22}) \end{bmatrix}$$

The state of a qutrit thus can be represented by a vector $x \in \mathbb{R}^8$.

Of course, in the N-level case the Bloch vector space is not as simple as it was in the case of a qubit. The state space is a convex asymmetric real subset of a ball in $\mathbb{R}^{N^2-1}$. In this case, the boundary of the Bloch vector space includes only a subset of the sphere (which corresponds to the density matrices with rank 1) together with a subset of the ball (density matrices having not full rank, i.e. not invertible density matrices). The mixed states are still inside the boundary of the Bloch vector space.

More information about the structure of the Bloch vector space of N-level systems based on [35] (which slightly differs from the approach applied here) can be found in Appendix A.2.

5.1.5 Distances between quantum states

Throughout the work two basic quantities are used for describing the *distance* between two states of a quantum system. The first one is the usual ℓ_2 distance, the other one is the so-called fidelity.

ℓ_2 distance

Having two Bloch vectors x and r the ℓ_2 distance between them is defined as

$$d(x, r) = \left(\sum_j (x_j - r_j)^2\right)^{\frac{1}{2}} \tag{92}$$

The properties of ℓ_2 distance are:

- $d(x, r) \geq 0$
- $d(x, x) = 0$
- $d(x, s) \leq d(x, r) + d(r, s)$

Fidelity

Another useful quantity, the *fidelity* [30] is defined for two density matrices χ and ρ as

$$f_1(\chi,\rho) = \left(\mathrm{Tr}\left(\chi^{\frac{1}{2}}\rho\chi^{\frac{1}{2}}\right)^{\frac{1}{2}}\right)^2 \tag{93}$$

Note, that sometimes the above expression without square is called fidelity [46]:

$$f_2(\chi,\rho) = \mathrm{Tr}\left(\chi^{\frac{1}{2}}\rho\chi^{\frac{1}{2}}\right)^{\frac{1}{2}} \tag{94}$$

The basic properties of fidelity (defined as either (93), or (94)) are:

- $0 \leq f(\chi,\rho) \leq 1$
- $f(\chi,\rho) = f(\rho,\chi)$
- $f(\chi,\chi) = 1$.

5.2 Quantum measurement and state estimation

Measurement makes quantum mechanics hard to treat using control theory, since it also acts as a special stochastic input, or disturbance (see section **??**), that intervenes the system any time a measurement is done.

In what follows two types of measurement is introduced. The classical measurement is discussed in section 5.2.1 while a more general measurement is defined in section 5.2.2.

Because of the specialities of the measurements, the problem of state estimation as it was defined in Appendix A.1.3 is not easy to formulate for quantum systems so a simplified version commonly used in quantum information theory [24] is introduced in section 5.2.3.

5.2.1 von Neumann measurement

The measurable physical quantities of quantum systems (the so called *observables*) are represented by self-adjoint operators of $\mathcal{H}$ (i.e. self-adjoint matrices of $\mathbb{C}^{\dim\mathcal{H}\times\dim\mathcal{H}}$) [7].

The measurement of an observable O has a probabilistic nature. The possible outcomes of the measurement are the different λ eigenvalues of O, the corresponding probability is

$$\mathrm{Prob}(\lambda_i) = \mathrm{Tr}E_i\chi E_i^*,$$

where E_i are the projections onto the subspace spanned by the eigenvector corresponding to λ_i, i.e.

$$\sum_i E_i = I, \quad E_i^2 = E_i.$$

Moreover, the state of the system **S** after measuring O, and having the outcome λ_i changes to

$$\chi' = \frac{E_i \chi E_i^*}{\mathrm{Tr} E_i \chi E_i^*}$$

that means that the measurement has lost its good property of being a passive operation known from classical physics. *The measurement changes the actual state of the quantum system.*

The above measurement is called *von Neumann measurement*, the most popular example of it is the spin measurement, i.e. the measurement of the observable σ_1, σ_2, or σ_3.

Spin measurement of a qubit

In what follows, the observable being the i-th Pauli spin operator σ_i is detailed since some of the quantum state estimation methods presented in chapter 6 and chapter 7 are based on spin measurements.

In this case the observable to be measured is $O = \sigma_i$. The possible outcomes of this measurement i.e. the eigenvalues of σ_i are ± 1. Thus, the spectral decomposition of the observable is

$$\sigma_i = E_{+1} - E_{-1}, \tag{95}$$

where

$$E_{+1} = \frac{1}{2}(I + \sigma_i), \quad E_{-1} = \frac{1}{2}(I - \sigma_i). \tag{96}$$

The probability of having outcome ± 1 can be computed from

$$\mathrm{Prob}(\pm 1) = \mathrm{Tr} E_{\pm 1} \chi E_{\pm 1}^*. \tag{97}$$

Using that the cyclic permutation behind the trace operator is allowed, and that $E_{\pm 1}$ are projections, (97) can be written as

$$\begin{aligned} \mathrm{Prob}(\pm 1) &= \tfrac{1}{2}\mathrm{Tr}\chi(I \pm \sigma_i) = \\ &= \mathrm{Tr}\left(\tfrac{1}{2}(I + x_1\sigma_1 + x_2\sigma_2 + x_3\sigma_3)\tfrac{1}{2}(I \pm \sigma_1)\right) = \\ &= \tfrac{1}{2}(1 \pm x_i). \end{aligned}$$

5.2.2 Positive operator valued measurement

A more general measurement type is the so-called POVM (positive operator valued measurement). It is defined by a set of self-adjoint positive semidefinite operators F_i of $\mathcal{H}$ such that

$$\sum_i F_i = I \in \mathcal{H}.$$

The difference between von Neumann measurement (section 5.2.1) and positive operator valued measurement is that as opposed to the von Neumann measurement, F_i-s are not necessarily orthogonal, i.e. the $F_i F_j \neq 0$ in general. That's why the

number of elements of a POVM can be greater than the dimension of the Hilbert space $\mathcal{H}$. The probability of an outcome belonging to F_i is

$$\mathrm{Prob}(i) = \mathrm{Tr} F_i \chi,$$

where χ is the actual state of the system. The value of the density matrix after the measurement is

$$\chi' = \frac{M_i \chi M_i^*}{\mathrm{Tr} M_i \chi M_i^*},$$

where M_i are the so-called Kraus operators:

$$F_i = M_i^* M_i.$$

POVM generalization of the spin measurements

The Pauli spin measurements can also be generalized to a POVM, so that the operators of the measurements are related to (95) and (96) defined in the previous section:

$$F_{+i} = \frac{1}{6}(I + \sigma_i), \quad F_{-i} = \frac{1}{6}(I - \sigma_i), \quad i = 1, 2, 3. \tag{98}$$

Of course, the outcomes of a POVM cannot be identified by the eigenvalues of an observable. It is enough to assign different symbols to the different operators (98). The simples choice is

$$-3, \quad -2, \quad -1, \quad +1, \quad +2, \quad +3,$$

where the pair $(-i, +i)$ corresponds to the possible outcomes of the spin measurement (95), $i = 1, 2, 3$. The probability of the i-th outcome is

$$\mathrm{Prob}(+i) = \frac{1}{6}(1 + x_i), \quad \mathrm{Prob}(-i) = \frac{1}{6}(1 - x_i), \quad i = 1, 2, 3.$$

5.2.3 The quantum state estimation problem

As it was construed in section 5.2, the measurement act as a special disturbance in the quantum mechanical domain, moreover it is probabilistic. That's why if one measures a certain quantum system it is not possible to have enough information from a single measurement. On the other hand, a second identical measurement on the same system cannot be performed since the state has been changed with the first measurement so the problem statement of state estimation for quantum systems [24] needs some simplification.

In order to avoid the above difficulties it will be assumed in the sequel that *sufficiently many identical copies of the system is available for measurement.* We perform measurements on the quantum systems one after another, that is, a system is measured only once, the next measurement is performed on the next copy, so the state of the quantum systems after the measurement are irrelevant from our viewpoint. This way the system dynamics is omitted and the state estimation is reduced to a parameter estimation problem (see section 1.2).

The goal of state estimation is to determine the density operator ρ of a quantum system by measurements on n copies of the quantum system which are all prepared according to ρ [4, 11, 34]. The number n corresponds to the sample size in classical mathematical statistics. An estimation scheme means a collection of measurements and an estimate for every n. The estimate is a mapping defined on the measurement data and its values are density operators. For a reasonable scheme, we expect the estimation error to tend to 0 when n tends to infinity (i.e. we expect to have an asymptotically unbiased estimate) as a consequence of the law of large numbers.

5.3 Earlier work on quantum state estimation

The method of estimating the state of a system based on measurements has different names in different fields. The term *state estimation* is used by the engineering community (see section 1.2) while physicists use the expression *state tomography*. In the following any of them might be used.

Authors from physics related fields, e.g. [53] put more stress on the experimental implementation of a method (e.g. quantum optics, interferometers, etc.). [66] gives an estimation scheme (i.e. a POVM with 4 operators and an estimator). An adaptive extension of the method is also given.

State estimation based on classical statistical estimation methods are given in [24], [10]. A mutually unbiased measurements scheme is proposed by [67].

There only a few papers in which engineering tools are attempted to use for quantum systems. In [38] a optimal experiment design for quantum systems was treated as a convex optimization problem, while [39] handles the parameter estimation of quantum systems with convex optimization tools.

The available tools on the quantum state estimation using engineering methods suffer from the fact, that they give result on the distribution and variance of the estimate only in the asymptotic sense. On the other hand the physics related literature provides only *ad hoc* methods without further mathematical analysis.

Chapter 6

Bayesian state estimation of a quantum bit

In this chapter a state estimation method for a single qubit is presented that uses Bayesian approach described in section 1.2 and Appendix A.1.4 in more details. Throughout the chapter the Bloch parameterization of quantum states (section 5.1.4) is used, so the aim will be to estimate the value of the Bloch vector. Using the assumptions of section 5.2.3, i.e. neglecting the dynamics defined by (84) the state estimation problem (see section 1.2) turns to a parameter estimation problem, see Appendix A.1.4.

Section 6.1 handles the problem componentwise i.e. a Bayesian estimation scheme is given for all 3 components of the Bloch vector.

Section 6.2 connects the results of section 6.1, and gives an estimate for the Bloch vector, however it may lay out of the state space i.e. the Bloch ball. This problem will be solved in section 6.2.2.

The two different estimators are compared in section 6.3 using a software tool for simulating quantum mechanical systems, which is presented in section 6.3.1.

6.1 Componentwise estimation of the Bloch vector [O7]

First the simplest case is considered when a single component of a qubit is to be estimated. This can be regarded as two related components of the so-called standard six-state-tomography [24].

The method of estimation is discussed for the component in the direction of the Pauli operator σ_1, but any other direction can be treated similarly. The speciality of a component-wise estimation is that one can apply the methods of classical statistics, because the outcomes of a measurement constitute a probability distribution.

6.1.1 Bayesian estimate

In order to be able to apply the Bayesian parameter estimation formula (145) defined in section A.1.4, the parametrized system model $p(y(k)|D^{k-1}, \theta)$ needs to be

constructed. The parameter to be estimated is $\theta = x_1$ that is a single parameter. It is important to note that the methods of classical statistics can now be applied because the estimations in the different directions are treated separately.

Measuring the spin in the σ_1-direction

Concentrating on the measurements for the observable $A = \sigma_1$ (see section 5.2.1) one can associate the measured value $+1$ or -1 at a discrete time instance k to the system output $y(k)$. Due to the measurement strategy laid down in section 5.2.3, there is no dynamics associated to the system (in this case to the qubit source) and the measurements are independent, therefore

$$p(y(k)|D^{k-1}, x_1) = \begin{cases} P_{+1} = \frac{1}{2}(1 + x_1), & \text{if} \quad y(k) = 1, \\ \\ P_{-1} = \frac{1}{2}(1 - x_1), & \text{if} \quad y(k) = -1. \end{cases} \tag{99}$$

It is important to note that the parametrized system model $p(y(k)|D^{k-1}, x_1)$ depends now on the parameter x_1 only, thus

$$p(y(k) \mid D^{k-1}, \theta) = p(y(k) \mid D^{k-1}, x_1)$$

in this case.

Recursive estimation

One can construct a Bayesian parameter estimation scheme for the parameter x_1 using only the measured values for the observable σ_1 from (145):

$$p(x_1|D^k) = \frac{p(y(k)|x_1)p(x_1|D^{k-1})}{\int P(y(k)|\nu)p(\nu|D^{k-1})d\nu}\,. \tag{100}$$

The above estimation is recursive and applies for updating the estimate $p(x_1|D^{k-1})$ by using a single measured value $y(k)$, i.e. from step $(k-1)$ to step k.

Prior estimate

As it was mentioned in Appendix A.1.4, Bayesian estimation scheme is able to take any initial a'priori knowledge about the state to be estimated into account in the form of a prior probability density function, $p^0(x_1) = p(x_1|D^0)$.

One-shot estimation

The non-recursive one-shot version of the Bayesian estimate can be developed from the following ingredients, having k measured values.

1. Assume that we start our estimation with a prior probability density function p^0.

2. The value $+1$ is observed l times and the value -1 $(k-l)$ times.

Then the above recursive formulae (100) with (99) becomes:

$$p(x_1|D^k)(t) = \frac{(\frac{1}{2}(1+t))^l(\frac{1}{2}(1-t))^{k-l}p^0(t)}{\int(\frac{1}{2}(1+\nu))^l(\frac{1}{2}(1-\nu))^{k-l}p^0(\nu)\,d\nu} \tag{101}$$

Part of this probability density function suggests the *β-distribution* after the variable transformation

$$s = \frac{1}{2}(1+t) \tag{102}$$

(see e.g. [20] for the properties of the β-distribution). It seems to be worthwhile to assume p^0 in a similar form,

$$p^0(t) = C\left(\frac{1+t}{2}\right)^{\lambda}\left(\frac{1-t}{2}\right)^{\kappa-\lambda} \qquad (t \in [-1,1])\,.$$

(Behind this assumption, one can think about previously performed measurements with λ times the value $+1$ and $\kappa - \lambda$ times the value -1.) So the conditional probability density is

$$p(x_1|D^k)(t) = C\left(\frac{1+t}{2}\right)^{l+\lambda}\left(\frac{1-t}{2}\right)^{k+\kappa-l-\lambda} \tag{103}$$

on the interval $[-1,1]$. (Here the normalization constant C depends on the parameters in the distribution.) The RHS of (103) is called *transformed β-distribution* and properties of the usual β-distribution can be used to work with it.

Based on the idea of the above derivation, similar Bayesian estimation schemes for each component of the Bloch vector can be constructed. The estimates of the individual components will be stochastically unrelated, because the observables σ_1, σ_2 and σ_3 are incompatible. (Compatible observables O_1 and O_2 commute, i.e. $O_1 \cdot O_2 = O_2 \cdot O_1$, thus form a set of classical random variables.) The Bayesian estimate is then given separately for the parameters x_i, $i = 1,2,3$ by the formula (101).

6.1.2 Point estimate

Equation (101) shows that the estimate has a β-distribution with parameters $(l + 1 + \lambda)$ and $(k - l + 1 + \kappa - \lambda)$ in the transformed variable s.

Bayesian point estimate

Then one can choose the mean value of this distribution

$$m_1 = \frac{l+1+\lambda}{k+\kappa+2} \tag{104}$$

as a point estimate and use the variance

$$\sigma_1^2 = \frac{(l+1+\lambda)(k-l+1+\kappa-\lambda)}{(k+\kappa+2)^2(k+\kappa+3)} \tag{105}$$

as a measure of uncertainty.

It is important to note that the above estimates are computed from two information sources:

1. the overall number of measurements k and the number of measurements l with the outcome $+1$ and

2. the prior distribution that is expressed in terms of the *fictive number* of measurements $(\kappa - 1)$ with the number of measurements $(\lambda - 1)$ of the outcome $+1$.

The above statistics (104) can be used to construct an unbiased estimate for x_1 in the form

$$\hat{x}_1 = 2\frac{l + 1 + \lambda}{k + \kappa + 2} - 1 \tag{106}$$

using the variable transformation (102).

6.2 Estimation of the Bloch vector [O7]

Turning to the problem of estimating the state of a qubit as a Bloch vector the parameters to be determined are $x = (x_1, x_2, x_3)^T$. It is crucial to note that the observables σ_1, σ_2 and σ_3 are incompatible in the sense of quantum mechanics. Therefore the estimation of x_1, x_2 and x_3 is treated separately but then the algebraic inequality constraint (89) that relates the three component of a Bloch vector cannot be taken into account.

6.2.1 Unconstrained Bayesian estimation

The maximum-likelihood state estimation of quantum systems (see [10] or [38]) also needs the joint probability density function (abbreviated as p.d.f.) to maximize. Without the above consideration, one could interpret the incompatibility of the observables as the independency of the measurements, thus the joint p.d.f. is constructed as the product of the marginal p.d.f.'s. Note that this approach is consistent in the limit, when the number of measurements is large, but may fail elsewhere. With this caution *the product form of the componentwise p.d.f.'s (103) is used here*, too.

For the sake of simplicity it is assumed that the same number of measurements k is applied in all of the σ_1, σ_2 and σ_3 directions but one observes the measured value $+1$ in the σ_1-direction l_1 times, and in the σ_2 and σ_3-directions l_2 and l_3 times, respectively. The measurement record D^k is formed from this integers and from k, and the joint conditional p.d.f. of the state estimate is formed from the marginal p.d.f.'s in the form of (103):

$$p(x_1, x_2, x_3|D^k)(t, u, v) = C^* p(x_1|D^k)p(x_2|D^k)p(x_3|D^k) \tag{107}$$

6.2.2 Constrained Bayesian estimation

In the Bayesian state estimation framework, the common way of respecting an inequality constraint for the elements of a parameter vector is to choose the prior p.d.f. accordingly [48] and then the recursive estimation formula (100) will transfer this property to any posterior p.d.f. With this approach one looses the β-form of

the posterior p.d.f.'s, because the prior is not in β-form. Therefore, *the inequality (89) is respected separately by setting the value of the joint p.d.f. to zero outside the valid region*. Thus the proposed state estimation consists of two steps:

(i) estimating the componentwise p.d.f.'s of the elements of the Bloch vector x_1, x_2 and x_3 separately as described in section 6.1 and

(ii) regularization of the estimate.

Having estimated the conditional p.d.f.'s separately and formed p.d.f. (107) from them, a constrained p.d.f. $\overline{p}(x_1, x_2, x_3|D^k)$ is calculated as follows:

$$\overline{p}(x_1, x_2, x_3|D^k)(t, u, v) = \begin{cases} 0 & \text{if } t^2 + u^2 + v^2 > 1 \\ p(x_1, x_2, x_3|D^k)(t, u, v) & \text{elsewhere} \end{cases} \tag{108}$$

The constrained p.d.f. (108) is no longer of β-type, therefore its mean value vector $\underline{m}$ should be computed by numerical integration according to the definition:

$$\underline{m} = \int_{t^2+u^2+v^2 \leq 1} \underline{\tau} \cdot \overline{p}(x_1, x_2, x_3|D^k)(\underline{\tau}) \cdot d\underline{\tau} \tag{109}$$

where $\underline{\tau} = [t, u, v]^T$.

6.3 Comparison based on computer simulation results [O7]

In what follows, the qubit state estimator presented in section 6.2.1 and its constrained extension (section 6.2.2) is compared via computer simulation. The difference between the two estimators depends on the number of measurements and on the length of the Bloch vector, (i.e. the *degree of purity* of the state).

6.3.1 A simulation software for quantum systems

In order to be able to analyze and compare the different estimation schemes it was necessary to perform experiments. For the sake of this, a simulation software was written [O8] in *Matlab* environment [60]. It is suitable for the dynamical simulation of simple finite dimensional quantum mechanical systems, however this function of the simulator is not used here. Different types of measurements can also be implemented which makes it possible to compare not only different estimators but also similar estimators using different kinds of measurements. The stochastic nature of quantum measurement is modelled by random number generator. Some basic functions of the simulator are briefly described in Appendix A.6.

6.3.2 The effect of the number of measurements

The effect of the constraining is investigated by estimating the state of a qubit in a *pure state* with density matrix χ and Bloch vector x as follows:

$$\chi = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad , \quad x = (1,\ 0,\ 0)^T$$

The dependency on the number of measurements (k) is examined through two cases. In the first case 10 measurements are performed in each of the three directions. The componentwise p.d.f.'s of the first case can be seen in Figure 5. After

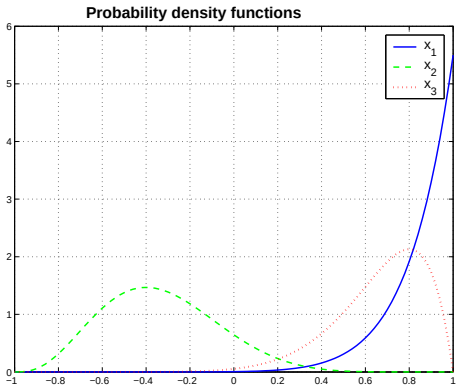

Figure 5: Componentwise p.d.f.'s after 10 measurements

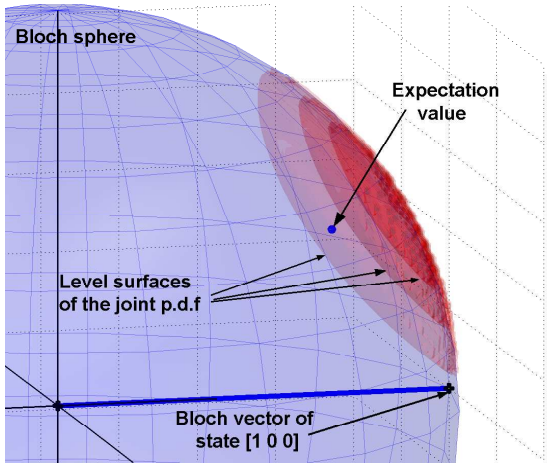

Figure 6: Joint p.d.f. after 10 measurements

constraining the joint p.d.f. (Figure 6) it is possible to determine the expectation value of the Bloch vector x. The expectation values originating from the marginal p.d.f.'s and that from the joint p.d.f.'s are summarized in (Table 1). It is important

$k = 10$	**componentwise**	**joint**	**real value**
x_1	0.8333	0.6494	1
x_2	-0.3333	-0.2088	0
x_3	0.6667	0.4557	0
length	1.1180	0.8204	1

Table 1: Comparison of the two estimates ($k = 10$)

to note that using only the separate estimates for x_1, x_2, and x_3 it is possible to get

an estimation for the Bloch vector with a *length* being greater than 1 which is not in the Bloch ball (first column of Table 1). The constraining of the product type joint p.d.f. solves this problem, it cannot give a false estimate of this kind.

The situation is quite similar for the case with 100 measurements. First the separate marginal probability density functions (Figure 7) are determined. The constrained joint p.d.f. (Figure 8) is calculated by formula (108). As it was ex-

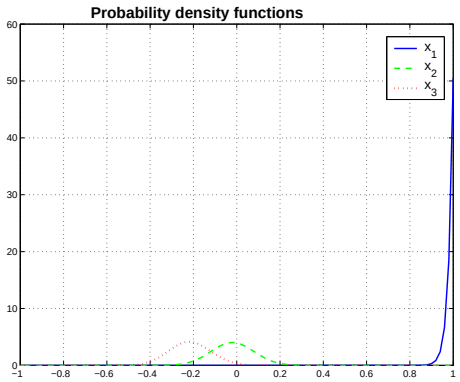

Figure 7: Componentwise p.d.f.'s after 100 measurements

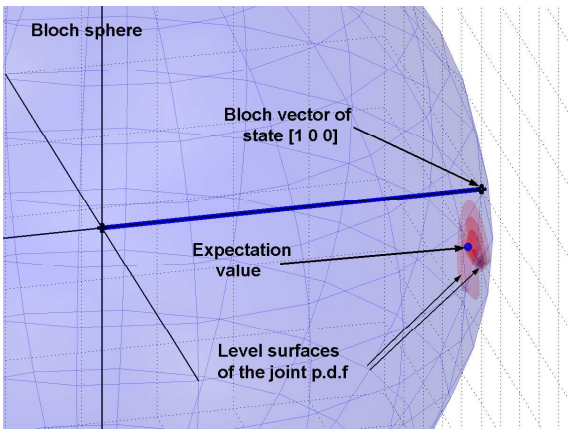

Figure 8: Joint p.d.f. after 100 measurements

pected, the precision of the estimate increased with the number of measurements. Table 2 shows that the separate marginal estimations of x still can violate the constraint $x_1^2 + x_2^2 + x_3^2 \leq 1$.
Of course, if the Bloch vector to be estimated is somewhere in the middle of the Bloch ball then the constraining has no effect, so the two estimates for the Bloch vector (i.e. the one originating from the separate componentwise p.d.f.'s and the one from the joint p.d.f.) gives the same expectation value.

6.3.3 The effect of the length of Bloch vector

The aim of the next experiment was to investigate the performance of the constrained and the unconstrained estimators when Bloch vectors with different length (i.e. different purity) are estimated.

$k = 100$	componentwise	joint	real value
x_1	0.9804	0.9609	1
x_2	−0.0196	−0.0139	0
x_3	−0.2157	−0.1512	0
length	1.0040	0.9729	1

Table 2: Comparison of the two estimates ($k = 100$)

For this, the number of measurement is fixed to $k = 40$ and two extreme states are to be estimated. The first state is the totally mixed one:

$$\chi^{\text{mixed}} = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix} \quad , \quad x^{\text{mixed}} = (0,\ 0,\ 0)^T$$

The two estimators were expected to be identical since the state is at the middle of the Bloch ball and the number of measurements is not extremely small.

Figure 9 shows the probability density functions for the Bloch vector components, the joint p.d.f. is depicted in Figure 10. It is apparent, that the joint p.d.f. is not constrained (i.e. it has no nonzero value outside the state space), and in the middle of the ball the β-distribution is symmetric, so it gives concentric spheres as level surfaces (Figure 10). Table 3 contains the point estimates computed from the

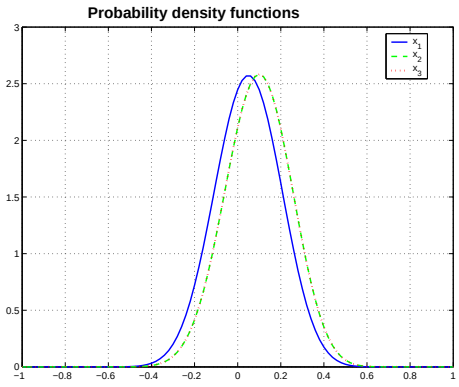

Figure 9: Componentwise p.d.f.-s for χ^{mixed}

componentwise and the joint p.d.f.-s, it can be seen that the two estimators are identical for this case.

The other extreme state is a pure state laying on the Bloch sphere (on the surface of the Bloch ball):

$$\chi^{\text{pure}} = \begin{bmatrix} 0.7906 & 0.2179 - 0.3436i \\ 0.2179 + 0.3436i & 0.2094 \end{bmatrix} \quad , \quad x^{\text{pure}} = (0.4359,\ 0.6872,\ 0.5812)^T$$

The componentwise Bayesian estimates of the Bloch vector elements can be seen in Figure 11, the constrained Bloch vector estimate is in Figure 12. The constraining step described in section 6.2.2 was necessary for this state. It can be seen in Table

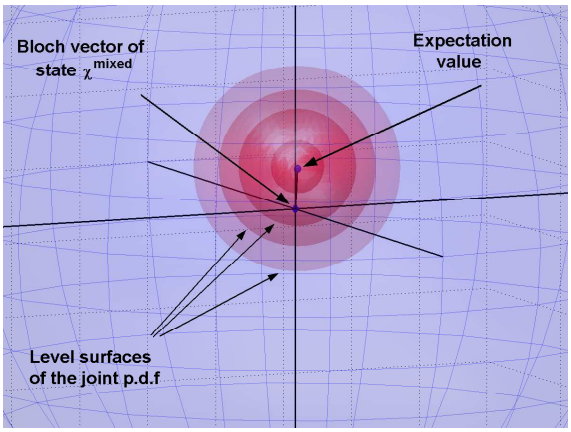

Figure 10: Joint p.d.f. for χ^{mixed}

$k = 40$	**componentwise**	**joint**	**real value**
x_1	0.0476	0.0476	0
x_2	0.0952	0.0952	0
x_3	0.0952	0.0952	0
length	0.1429	0.1429	0

Table 3: Comparison of the two estimates for χ^{mixed}

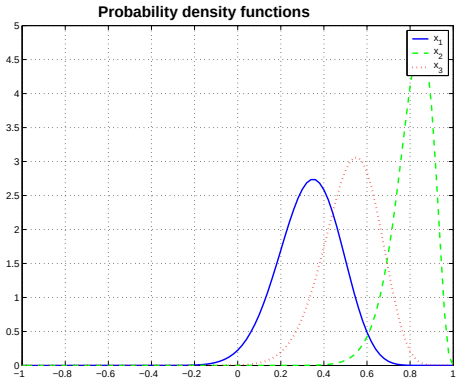

Figure 11: Componentwise p.d.f.-s for χ^{pure}

4, that the unconstrained Bloch vector estimate would be outside the Bloch ball (its' length is 1.0202). This example demonstrates the main drawback of the used constraining method: it is practically impossible for the constrained estimator to result in a pure state.

6.4 Summary

A slightly modified state estimation problem defined in section 1.2 was solved for a quantum bit using the results of Bayesian parameter estimation introduced in section 1.2 and Appendix A.1.4. The estimation was based on the Bloch vector representa-

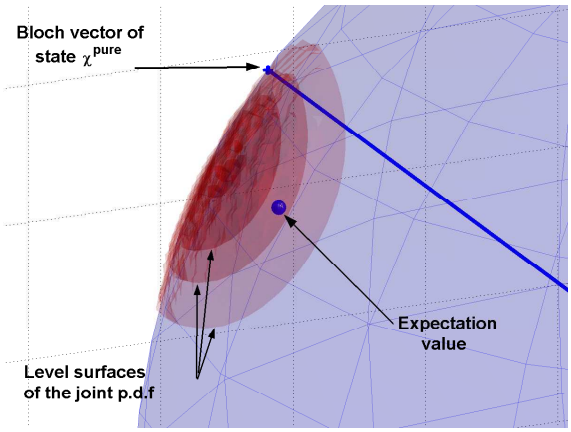

Figure 12: Joint p.d.f. for χ^{pure}

$k = 40$	**componentwise**	**joint**	**real value**
x_1	0.3333	0.2718	0.4359
x_2	0.8095	0.7432	0.6872
x_3	0.5238	0.4391	0.5812
length	1.0202	0.9051	1.0000

Table 4: Comparison of the two estimates for χ^{pure}

tion of quantum states and on the von Neumann measurements of the three Pauli spin operator.

Since the three measurements were assumed to be independent (because they are incompatible), *the problem was simplified to componentwise estimations of the three Bloch vector. As the componentwise estimates were independent the total estimate was obtained by multiplying the three probability density functions.*

The obtained Bayesian state estimator was expected to perform weak for estimating pure states and/or for estimating from small measurement data, so *an additional constraint was added to the problem. This step resulted in an estimator that always gives a physically meaningful result, however it's computation is more difficult.*

Afterwards, the *the expectations were supported by simulation results and the behavior of the two estimators was investigated under different circumstances, in the case of small and medium sample sizes and for pure and mixed states.*

Chapter 7

Point estimation of N-level quantum systems

The subject of this chapter is the state estimation of N-level quantum systems [24]. As it was mentioned before in section 5.1.3, in this case the boundary of the state space is not the set of pure states but the non-invertible density matrices.

Section 7.1 defines the observables to be measured and derives the estimator that maps the set of measurement outcomes to the state space (actually to a set contains the state space). The estimator is modified in section 7.2 in order that the range of it is identical to the state space. The modification is done by introducing a positivity constraint to be respected.

Finally, in section 7.3 the effectiveness of the estimator is examined and compared to two different quantum state estimation schemes available in the literature.

At the end of the chapter the results are summarized in section 7.4.

7.1 The unconstrained estimation scheme [O10]

In order to define an estimation scheme one needs to define a measurement strategy and afterwards, an estimator is constructed that maps the set of possible outcomes onto the set of quantum states (density matrices, or Bloch vectors, see section 5.1).

In what follows, a set of observables are defined for finite quantum mechanical systems, that will be the basis of the estimation scheme.

7.1.1 Measurement strategy

In chapter 6 the quantum system of interest was the single qubit so the von Neumann measurement of the Pauli spin was applicable as a measurement scheme. Since this chapter treats quantum systems more general than a qubit, another set of observables is necessary.

The measurement scheme used is based on the Bloch parametrization (91) of N-level quantum systems. In the qubit case the elements of the Pauli basis were used as observables, in this case the same strategy is used, i.e. the observables are the $N \times N$ self-adjoint matrices in (91) are applied. The three set of observables are

in the form

$$\begin{aligned} X_{ij} &= E_{ij} + E_{ji}, \quad i < j \\ Y_{ij} &= -\mathrm{i}E_{ij} + \mathrm{i}E_{ji}, \quad i < j \\ Z_{ii} &= E_{ii}, \quad i = 1, \ldots, N-1. \end{aligned} \tag{110}$$

Note, that for the 2-level case, $X_{12} = \sigma_1$, and $Y_{12} = \sigma_2$ holds. The spectrum of Z_{ii} is $\{0, 1\}$, i.e. the possible outcomes of its measurement are 0 and 1. The spectrum of X_{ij} and Y_{ij} is $\{-1, 0, 1\}$.

The measurement scheme (110) means that the observables Z_{ii}, X_{ij} and Y_{ij} are measured on the r copies of the original system, i.e the complete set of observables used in the measurement scheme (110) is measured r times. Since the trace of a density matrix is 1, the measurement of the first $N-1$ diagonal observables (Z_{ii}) gives enough information about the diagonal entries of χ, i.e. Z_{NN} can be removed from the set of observables. It means, that $N^2 - 1$ observables form the collection of measurements. Hence the overall number of measurements performed is $n = r(N^2 - 1)$. This way a measurement data of size r holds the information about each entry of χ.

7.1.2 State estimator for N-level quantum systems

The aim is to estimate the $N \times N$ density matrix χ of a quantum system. The parametrization is naturally given by the entries of the matrix. In what follows, we are given several copies of a N-level quantum system in the same state. We perform measurements on the systems one after another, that is, a copy of the system is measured only once, the next measurement is performed on the next copy of the system, so the states of the systems after the measurement are irrelevant from our viewpoint.

If the task is to estimate the real part of the (i, j)-th entry of the density matrix χ, then the observables $X_{ij} = E_{ij} + E_{ji}$ are measured . Their spectral decomposition is

$$1 \cdot \frac{1}{2}(E_{ii} + E_{ij} + E_{ji} + E_{jj}) + 0 \cdot \sum_{i \neq m \neq j} E_{mm} - 1 \cdot \frac{1}{2}(E_{ii} - E_{ij} - E_{ji} + E_{jj})$$

and its measurement has three different outcomes, ± 1 and 0. The probabilities of the outcomes ± 1 are

$$\mathrm{Prob}(X_{ij} = \pm 1) = \frac{1}{2}(\chi_{i,i} \pm \chi_{i,j} \pm \chi_{j,i} + \chi_{j,j}) = \frac{1}{2}(\chi_{i,i} + \chi_{j,j}) \pm \mathrm{Re}\ \chi_{i,j}\,. \tag{111}$$

To estimate the imaginary part, observables of form $Y_{ij} = \mathrm{i}E_{ij} - \mathrm{i}E_{ji}$ are measured with spectral decomposition

$$1 \cdot \frac{1}{2}(E_{ii} + \mathrm{i}E_{ij} - \mathrm{i}E_{ji} + E_{jj}) + 0 \cdot \sum_{i \neq m \neq j} E_{mm} - 1 \cdot \frac{1}{2}(E_{ii} - \mathrm{i}E_{ij} + \mathrm{i}E_{ji} + E_{jj}).$$

The probabilities of the outcomes ± 1 and 0 are

$$\mathrm{Prob}(Y_{ij} = \pm 1) = \frac{1}{2}(\chi_{i,i} \pm \mathrm{i}\chi_{i,j} \mp \mathrm{i}\chi_{j,i} + \chi_{j,j}) = \frac{1}{2}(\chi_{i,i} + \chi_{j,j}) \pm \mathrm{Im}\ \chi_{i,j}\,. \tag{112}$$

Finally, for the diagonal entries one has

$$\mathrm{Prob}(Z_{ii} = +1) = \chi_{i,i}. \tag{113}$$

Altogether there are N^2-1 different measurements and each of them is repeated r times. The measurement outcomes form a set D^n and this is the domain of the state estimator. To determine the estimate one needs only the relative frequencies of the outcomes of the N^2-1 different measurements, all of them are performed r times. If M is one of the observables which has an outcome t, then the relative frequency of t when the measurement is performed r times will be denoted by $\nu(r, M, t)$. According to the **law of large numbers**, $\nu(r, M, t) \to \mathrm{Prob}(M = t)$ as $r \to \infty$. (Of course, $\mathrm{Prob}(M = t)$ depends on the true state χ of the system.)

The following estimate is natural:

$$\begin{aligned}
\hat{\chi}^{\mathrm{un}}_{i,i} &= \nu(r, Z_{ii}, +1) \text{ for } (1 \le i < N) \\
\hat{\chi}^{\mathrm{un}}_{N,N} &= 1 - \textstyle\sum_{i=1}^{N-1} \nu(r, Z_{ii}, +1), \\
\mathrm{Re}\hat{\chi}^{\mathrm{un}}_{i,j} &= \tfrac{1}{2}(\nu(r, X_{ij}, +1) - \nu(r, X_{ij}, -1)), \quad i < j \\
\mathrm{Im}\hat{\chi}^{\mathrm{un}}_{i,j} &= \tfrac{1}{2}(\nu(r, Y_{ij}, +1) - \nu(r, Y_{ij}, -1)), \quad i < j.
\end{aligned} \tag{114}$$

where $\hat{\chi}_{i,j}$ denotes the (i, j)-th entry of the estimated density matrix.

Notation, "*un*" is an abbreviation of the word "*unconstrained*". It may happen that $\hat{\chi}^{\mathrm{un}}$ is not a positive semidefinite matrix, hence it is not an estimate in the really strict sense. By definition, the range of the above estimator is the set of all self-adjoint $N \times N$ matrices of trace 1. On the other hand, the density matrices are self-adjoint $N \times N$ *positive* matrices, thus, they form an open subset of the range of the estimator.

7.1.3 Properties of the estimate

Because of construction, the above unconstrained estimate $\hat{\chi}^{\mathrm{un}}$ defined by (114) is unbiased. It follows from the fact that the expectation value of the relative frequencies involved are the probabilities. The expectation values of the elements of the estimator can be computed as

$$\begin{aligned}
\mathbf{E}\ \hat{\chi}^{\mathrm{un}}_{i,i} &= \mathbf{E}\ \nu(r, Z_{ii}, +1) = \mathrm{Prob}(Z_{ii} = +1) = \chi_{i,i} \\
\mathbf{E}\ \mathrm{Re}\ \hat{\chi}^{\mathrm{un}}_{i,j} &= \mathbf{E}\ \tfrac{1}{2}(\nu(r, X_{ij}, +1) - \nu(r, X_{ij}, -1)) \\
&= \tfrac{1}{2}(\mathrm{Prob}(X_{ij} = +1) - \mathrm{Prob}(X_{ij} = -1)) = \mathrm{Re}\ \chi_{i,j} \\
\mathbf{E}\ \mathrm{Im}\ \hat{\chi}^{\mathrm{un}}_{i,j} &= \mathbf{E}\ \tfrac{1}{2}(\nu(r, Y_{ij}, +1) - \nu(r, Y_{ij}, -1)) \\
&= \tfrac{1}{2}(\mathrm{Prob}(Y_{ij} = +1) - \mathrm{Prob}(Y_{ij} = -1)) = \mathrm{Im}\ \chi_{i,j}
\end{aligned}$$

Thus, the expectation value of $\hat{\chi}^{\mathrm{un}}$ is the true state χ of the system, so it is an unbiased estimate.

It was seen in chapter 6 that the properties of the unconstrained estimator depend very much on the true state, i.e. it sometimes failed when the state χ to be estimate was on the boundary of the state space, i.e. it was a pure state in the qubit case. The same phenomena can be observed in the case of the estimator (114), when the number of measurement data is small (n is small), or the real state χ is not invertible (for N-level systems the notion of pure state is not equal to the boundary of the state space, see section 5.1.4).

For invertible states, it can be shown [O10] using large deviation theorem, that the probability of having a false estimate exponentially decreases with the number of measurements:

$$\mathrm{Prob}(\hat{\chi}_n^{\mathrm{un}} \notin G) \leq c_1 \exp(-c_2 n)$$

where c_1 and c_2 are constants [13], [O9], and G stands for the set of density matrices.

Theorem 1 *Assume that χ is an invertible density matrix. The probability of that $\hat{\chi}_n^{\mathrm{un}}$ is not a density matrix converges exponentially to 0 as $n \to \infty$.*

Proof. The expectation value of $\hat{\chi}_1^{\mathrm{un}}$ is $\chi \in \mathcal{M}_k$. Cramér's theorem tells us that there is a function $I : \mathcal{M}_k \to \mathbb{R}^+ \cup \{+\infty\}$ such that for any open set containing χ

$$\limsup_{n\to\infty} \frac{1}{n} \log \mathrm{Prob}(\hat{\chi}_n^{\mathrm{un}} \notin G) \leq -\inf\{I(D) : D \in \mathcal{M}_n \setminus G\}$$

The RHS is strictly negative and if ρ is invertible, then we can choose G such that it consists of density matrices (that is, its elements are positive definite). This gives the proof.

The computation of the rate function I is theoretically possible, but we do not need its concrete form. □

The situation is different when the state to be estimated is not invertible, i.e. it is on the boundary of the state space. For example, consider the pure state

$$\rho = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} = \frac{1}{2}(\sigma_0 + \sigma_3), \tag{115}$$

and measure the following observables:

$$Z_{11} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad X_{12} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y_{12} = \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}.$$

The measurement of Z_{11} gives 1 (with probability 1). For the the others we have

$$\mathrm{Prob}(X_{12} = \pm 1) = \mathrm{Prob}(Y_{12} = \pm 1) = \frac{1}{2}.$$

Let us introduce the binary valued random variable γ with

$$\gamma := \begin{cases} 1 & \text{with probability } 1/2, \\ -1 & \text{with probability } 1/2. \end{cases} \tag{116}$$

Then evidently $\gamma^2 = 1$ with probability 1.

The estimator can be regarded as a matrix-valued random variable in the form

$$\hat{\chi}^{\mathrm{un}} = \begin{bmatrix} 1 & ((\beta_1 + ... + \beta_r) - \mathrm{i}(\gamma_1 + ... + \gamma_r))/2r \\ ((\beta_1 + ... + \beta_r) + \mathrm{i}(\gamma_1 + ... + \gamma_r))/2r & 0 \end{bmatrix},$$

where γ_i and β_j are identically distributed independent random variables with the same distribution as γ (116).

It is easy to see that the expectation value of the determinant of $\hat{\chi}^{\mathrm{un}}$ is $-1/2r$. This shows, that although the expectation value of the determinant is zero in the limit (as a consequence of unbiasedness of the estimate), but the estimate has always a negative eigenvalue with positive probability. Therefore, in this example $\hat{\chi}^{\mathrm{un}}$ is a rather bad estimate.

Figure 13 shows a simulation with another pure state $(\sigma_0+\sigma_1)/2$. The ℓ_2 distance of the estimator $\hat{\chi}^{\mathrm{un}}$ and the true density matrix has been depicted as a function of n. It is seen that the convergence is very slow, not of an exponential type.

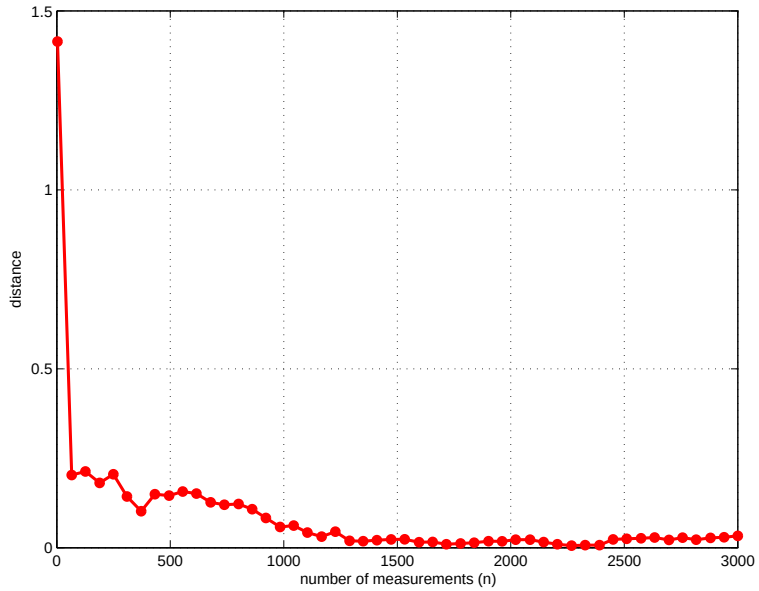

Figure 13: The ℓ_2 distance between the true pure state $(\sigma_0+\sigma_1)/2$ and $\hat{\chi}^{\mathrm{un}}$ converges very slowly to 0 as $n \to \infty$.

The above example shows the weak point of the unconstrained estimator (114) namely, the case of non invertible states. If the eigenvalues of the true state are strictly positive (and not very small), then the estimate is rather good. Note, that the computations are essentially simpler in the 2×2 case, when the boundary of the state space consists of pure states and the positivity of the estimate can be seen from the length of the Bloch vector. In the 3×3 case the boundary is more complicated, it consists of the non-invertible densities.

7.2 The constrained estimation scheme [O9],[O10]

As it was seen in section 7.1.3, problems arise when the estimator (114) faces with data obtained from a small number of measurements, or when it has to estimate a non-invertible state. To avoid the above problem, the estimator should be forced to give a physically meaningful estimate.

The basis of the estimator presented in this section is the unconstrained estimator (114). The measurement strategy is also the same as it was described in section 7.1.1.

7.2.1 The constrained estimator and its properties

Formally, the constrained state estimator $\hat{\chi}$ can be determined by solving the following optimization problem

$$\hat{\chi} := \mathrm{argmin}_\omega \mathrm{Tr}(\hat{\chi}^{\mathrm{un}} - \omega)^2 = \mathrm{argmin}_\omega \sum_{i,j} (\hat{\chi}^{\mathrm{un}})_{i,j} - \omega_{i,j})^2 \,, \tag{117}$$

where ω runs over the density matrices. The density matrices form a closed convex set G, therefore the minimizer is unique. Note that for a qubit the closest positive semidefinite matrix is easy to find. When the values of the estimate are 2×2 matrices, they can be identified by vectors in $\mathbb{R}^3$. When the estimate is unconstrained, it may happen that the values go out of the Bloch ball.

If the values of the estimates are simply the Bloch vectors, then the solution of (117) is simply

$$\hat{\chi} = \begin{cases} \hat{\chi}^{\mathrm{un}} & \text{if } \|\hat{\chi}^{\mathrm{un}}\| \leq 1, \\ \\ \frac{\hat{\chi}^{\mathrm{un}}}{\|\hat{\chi}^{\mathrm{un}}\|} & \text{otherwise.} \end{cases} \tag{118}$$

Due to the constraining, the estimator $\hat{\chi}$ is biased, however it is still unbiased in the asymptotic sense, i.e. when the number of measurements ($n = r(N^2-1)$) goes to ∞. In other words, the constrained estimator converges to the unconstrained one as $n \to \infty$.

Theorem 2 *The constrained estimate $\hat{\chi}_n$ is asymptotically unbiased.*

Proof. We can use the fact that $\hat{\chi}_n^{\mathrm{un}}$ is unbiased and to show that $\hat{\chi}_n^{\mathrm{un}}$ is an asymptotically unbiased estimate we study their difference. Let $p(y)$ be the probability of the measurement result y and Y is the set of outcomes such that $\hat{\chi}_n^{\mathrm{un}}(y) \neq \hat{\chi}_n(y)$, then evidently

$$\sum_y \hat{\chi}_n^{\mathrm{un}}(y)p(y) - \sum_y \hat{\chi}_n(y)p(y) = \sum_{y\in Y} (\hat{\chi}_n^{\mathrm{un}}(y) - \hat{\chi}_n(y))p(y) \,. \tag{119}$$

If $\mathcal{D}_k \subset \mathcal{M}_k$ is the set of density matrices, then Y is the set of outcomes y such that $\hat{\chi}_n^{\mathrm{un}}(y) \notin \mathcal{D}_k$. Let us fix a norm on the space $\mathcal{M}_k$. (Note that all norms are equivalent.)

Let $\varepsilon > 0$ be arbitrary. We split Y into two subsets:

$$Y_1 = \{y \in Y : \mathrm{distance}(\hat{\chi}_n^{\mathrm{un}}(y), \mathcal{D}_k) \leq \varepsilon\} \quad \text{and} \quad Y_2 = Y \setminus Y_1.$$

Note that distance$(\hat{\chi}_n^{\rm un}(y), \mathcal{D}_k) = \|\hat{\chi}_n^{\rm un}(y) - \hat{\chi}_n(y)\|$. Then

$$\sum_{y\in Y} \|\hat{\chi}_n^{\rm un}(y) - \hat{\chi}_n(y)\|p(y) \leq \sum_{y\in Y_1} \|\hat{\chi}_n^{\rm un}(y) - \hat{\chi}_n(y)\|p(y) + \sum_{y\in Y_2} \|\hat{\chi}_n^{\rm un}(y) - \hat{\chi}_n(y)\|p(y)\,.$$

The first term is majorized by ε and the second one by $C\text{Prob}(Y_1)$. Since the first is arbitrary small and the latter goes to 0, we can conclude that (119) goes to 0. □

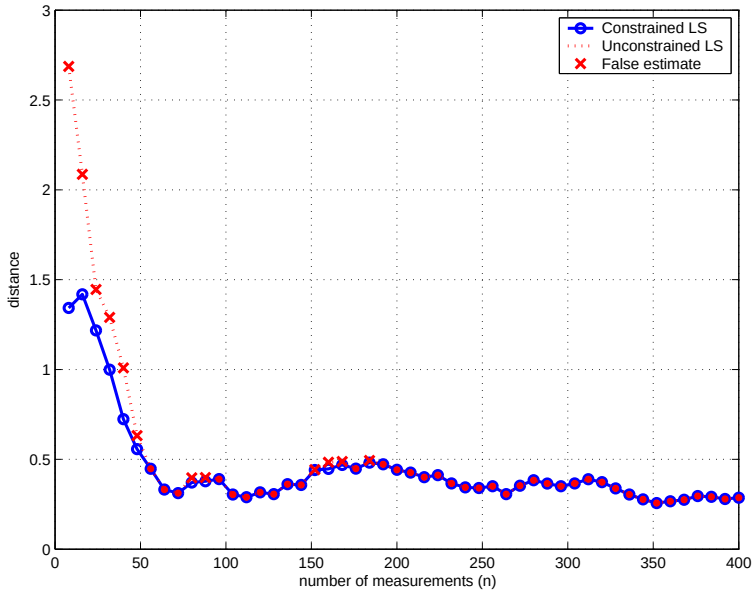

Figure 14: The ℓ_2 distance between the true 3×3 state with eigenvalues 0.1186, 0.2871, 0.5943 and the estimate. When the number of measurements is more than 200, the unconstrained estimate gives really a positive semidefinite matrix.

The difference between the two estimators (114) and (117) were examined also via computer simulation, using the tool introduced in section 6.3 (see also Appendix A.6, or [O8]). The results for a 3 level quantum system can be seen in Figure 14, where the two estimators were to estimate an invertible state. In Figure 14 the ℓ_2 norm of the difference between the estimates and the real states are depicted. Figure 15 and Figure 16 shows the behavior of the two estimators for an invertible (mixed) state and a non invertible (pure) state of a $N = 2$ level system. In this case, the fidelity (93) of the difference between the real and the estimated state was the qualifying quantity. It is apparent, that for a mixed state (Figure 15) the two estimates are different only for extremely few measurements (denoted by $\times$).

On the other hand, if the state to be estimated is non invertible, the difference remains, moreover the unconstrained estimate almost always gives an indefinite $\hat{\chi}^{\rm un}$, that results in a fidelity greater, than 1 (Figure 16).

7.2.2 Computing the constrained estimate

In what follows, two essentially different methods are proposed for determining the constrained estimate $\hat{\chi}$. The first is a simple algebraic method, the second is geometrical, and uses the Bloch representation of N-level states (section 5.1.4).

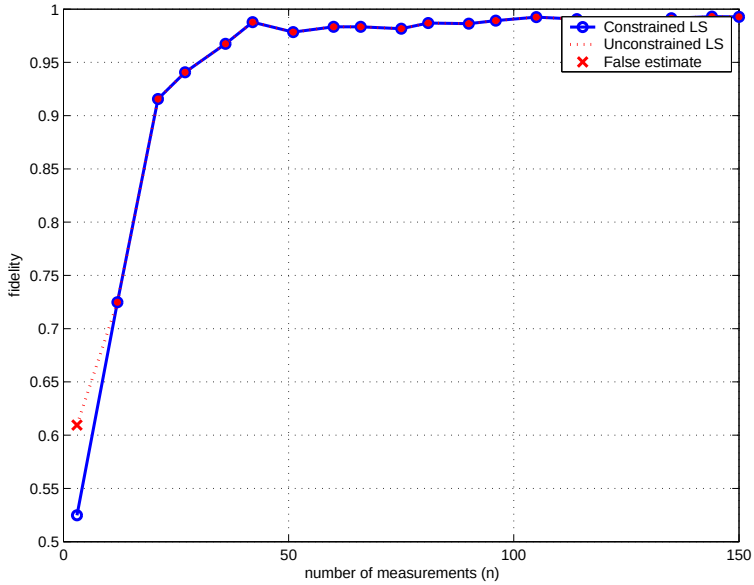

Figure 15: The fidelity between the true 2×2 mixed state with eigenvalues 0.1235, 0.8765 and the estimate. When the number of measurements is more than 10, the unconstrained and the constrained estimates are the same.

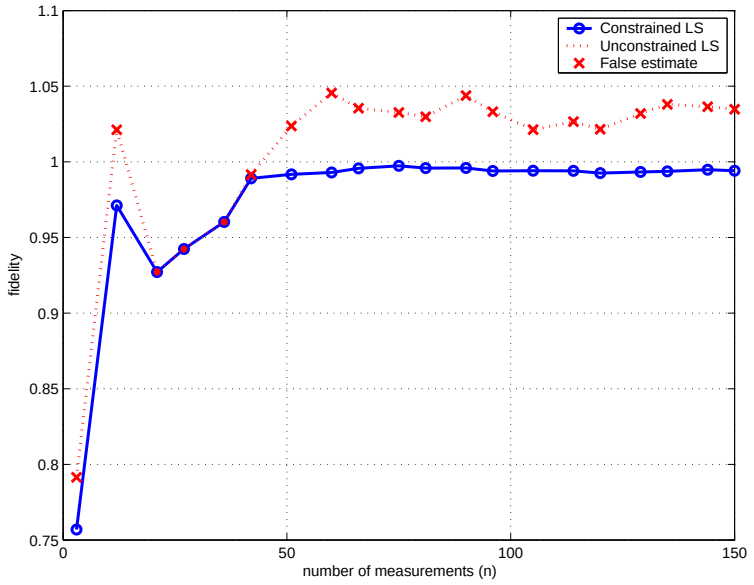

Figure 16: The fidelity between the true 2×2 pure state and the estimates. The unconstrained estimate is often outside of the Bloch ball and in this case the (real part of the complex) fidelity can be bigger than 1. The constrained estimate converges to the true state.

Algebraic way

The computation of the minimizer of (117) is easier if $\hat{\chi}^{\mathrm{un}}$ is diagonal. Since $\hat{\chi}^{\mathrm{un}}$ is self-adjoint, one can change the basis so that it can be assumed that $\hat{\chi}^{\mathrm{un}} =$

$\mathrm{Diag}(e_1, e_2, \ldots, e_N)$ and $e_1, e_2, \ldots, e_k < 0$ and $e_{k+1}, e_{k+2}, \ldots, e_N \geq 0$. The minimizer is obviously diagonal, hence it is enough to solve

$$\mathrm{argmin}_{f_i} \sum_i (e_i - f_i)^2$$

under the constraint $f_i \geq 0$ and $\sum_i f_i = 1$. According to the inequality between the quadratic and arithmetic means, one has

$$\begin{aligned} \sum_{i=1}^{n} (e_i - f_i)^2 &\geq \sum_{i=1}^{k} e_i^2 + \sum_{i=k+1}^{N} (e_i - f_i)^2 \geq \sum_{i=1}^{k} e_i^2 + \frac{1}{N-k} \left(\sum_{i=k+1}^{N} (e_i - f_i) \right)^2 \\ &= \sum_{i=1}^{k} e_i^2 + \frac{1}{N-k} \left(\sum_{i=1}^{k} f_i - e_i \right)^2 . \end{aligned}$$

If

$$f_i = e_i + c \quad \left(i = k+1, k+2, \ldots, N, \quad c = \frac{1}{N-k} \sum_{i=1}^{k} e_i, \quad c < 0 \right)$$

are positive, then the minimizer is $(f_1, f_2, \ldots, f_N)$, where $f_1 = f_2 = \cdots = f_k = 0$ and the other f_i's are defined above. If the N-tuple $(f_1, f_2, \ldots, f_N)$ contains negative entries, then the procedure must be repeated, the negative entries are replaced with 0 and the actual value of c is added to the other entries. After finitely many steps the minimizer will be found. Figure 17 shows the details for $N = 3$.

In the general case, it is possible to change the basis such that $\hat{\chi}^{\mathrm{un}}$ becomes diagonal, since the ℓ_2 distance is invariant under this transformation. So let

$$U \hat{\chi}^{\mathrm{un}} U^* = \mathrm{Diag}(e_1, e_2, \ldots, e_N)$$

for a unitary U. Then compute the minimizer $\mathrm{Diag}(f_1, f_2, \ldots, f_N)$ using the above procedure and

$$\hat{\chi} = U^* \mathrm{Diag}(f_1, f_2, \ldots, f_N) U \, .$$

Geometrical way of constraining

Having determined the value of the unconstrained estimator $\hat{\chi}^{\mathrm{un}}$ it is possible to convert it to a Bloch vector estimate using the Bloch representation (91) introduced in section 5.1.4. The Bloch representation gives a clear geometric view on the state space of quantum mechanical systems so geometric methods can be used to determine the constrained estimate $\hat{\chi}$. Similarly to the qubit case, the constraining means, that a Bloch vector being longer than 1, is shrunk onto the surface of the Bloch ball. This idea applied for N-level systems. In this case, the boundary of the Bloch vector space is difficult to formulate (see e.g. [35]).

The algorithm consists of 2 steps:

1. Determine the Bloch vector corresponding to the unconstrained estimate $\hat{\chi}^{\mathrm{un}}$. According to (91), it is an $N^2 - 1$ dimensional vector in $\mathbb{R}^{N^2-1}$. If the vector lies outside the Bloch vector space , then the corresponding density matrix is indefinite, i.e. it has negative eigenvalues as well.

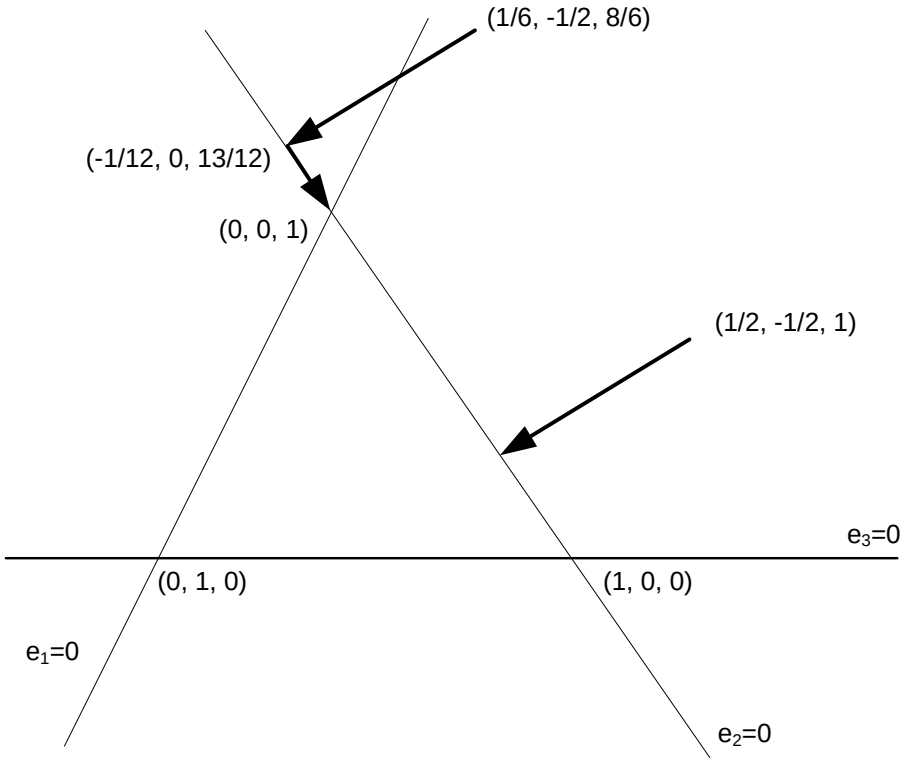

Figure 17: The constrained estimate for 3×3 matrices. The plain $e_1+e_2+e_3 = 1$ of $\mathbb{R}^3$ is shown. The triangle $\{(e_1, e_2, e_3) : e_1, e_2, e_3 \geq 0\}$ corresponds to the diagonal density matrices. Starting from the unconstrained estimate $\mathrm{Diag}(1/2, -1/2, 1)$, the constrained $\mathrm{Diag}(1/4, 0, 3/4)$ is reached in one step. Starting from $\mathrm{Diag}(1/6, -1/2, 8/6)$, two steps are needed.

2. Now decrease the length of the estimated Bloch-vector, until the boundary of the Bloch vector space is reached. It can be found using the fact, that the density matrix has positive eigenvalues inside the state space, has a rank deficiency (zero eigenvalue) on the boundary, and it has at least one negative eigenvalue outside the state space (of course, it is not fortunate to call it density matrix, since it is not). With a simple bisection algorithm it is possible to find the Bloch vector length that corresponds to the boundary. The 2-level case can be drawn, since it gives 3 dimensional Bloch vector space, a sketch of the algorithm can be seen in Figure 18.

Numerical simulations show only a negligible difference between the two different constraining methods, however the algebraic one is much easier to compute.

7.3 Comparison with other state estimation methods based on statistical properties [O10]

In this section different unbiased estimators of 2-level quantum systems (qubits) will be used together with a slightly modified version of the unconstrained estimator

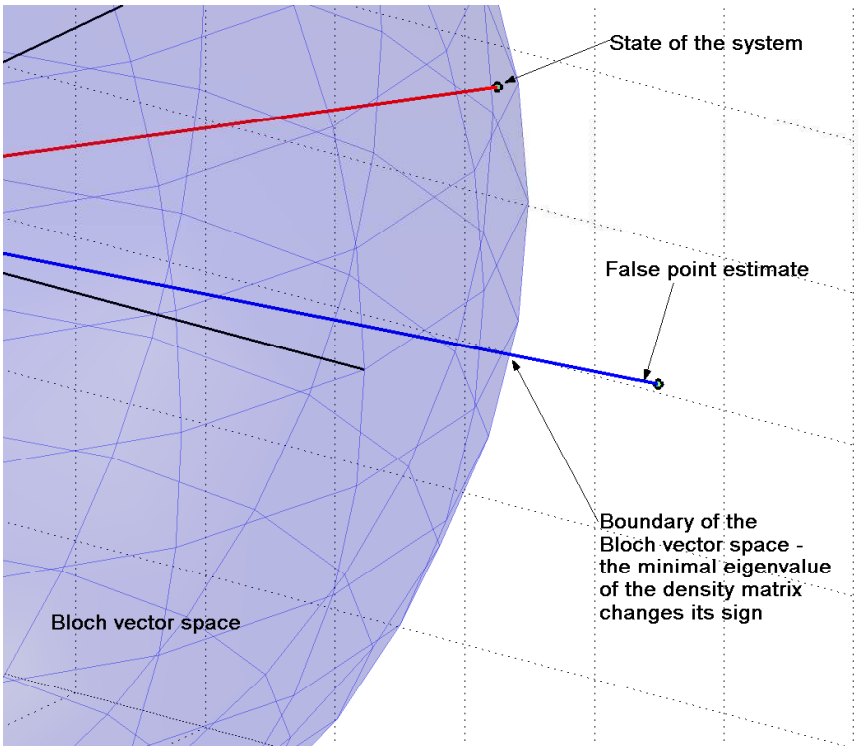

Figure 18: An indefinite estimate in the Bloch vector representation for a pure state of a qubit.

(114). The Bloch parametrization (88) of qubits will be used, i.e. the estimators give an estimate $\hat{x}$ of the Bloch vector x that describes the actual state of the system. Note, that density matrix-valued estimators can be converted to a Bloch vector without difficulty, using the formula (88), any estimate $\hat{\chi}$ can be expressed as $\hat{x}$.

The *mean quadratic error matrix* [57] is used to measure the efficiency of the different estimators. If the Bloch vector of the unknown real state of the quantum system is $x = (x_1, x_2, x_3)^T$, then the mean quadratic error is a 3×3 matrix defined as

$$V_n(x)_{i,j} := \sum_{y \in \mathcal{Y}_n} (\hat{x}(y)_i - x_i)(\hat{x}(y)_j - x_j)\, p(y) \qquad (1 \leq i, j \leq 3).$$

where $\mathcal{Y}_n$ is the set of all possible measurement outcomes from $n = 3r$ measurements, $\hat{x}(y)$ denotes the value of the estimator $\hat{x}$ if the concrete measurement data is y. (As it was described in Appendix A.1.3 and section 5.2.3, the estimator is a mapping from the set of possible measurement data to the state-space.) The probability of the outcome y is denoted by $p(y)$. Note, that since the following estimators are unbiased, the mean quadratic error matrix is identical to the covariance matrix [57].

Two estimators can be compared using their mean quadratic error matrix. If $V^1(x) < V^1(x)$, then the estimator 1 is more effective, than estimator 2. Note, that a matrix A is greater, than matrix B, if matrix $A - B$ is positive (semi)definite. Of course, two matrices are not always comparable since it is possible that their difference is indefinite.

7.3.1 The modified unconstrained estimator

In what follows, a modified version of the unconstrained estimator (114) is introduced, and the mean quadratic error matrix is determined for it.

The main difference is in the measurement scheme, since in this case the Pauli matrices will be used as observables. Note, that only Z_{ii} is changed to σ_3 in the measurement scheme (110), since $X_{ij} = \sigma_1$ and $Y_{ij} = \sigma_2$ in the 2 level case.

Based on the slightly modified measurement strategy, the form of the Bloch vector valued estimator is

$$\hat{x}^{\mathrm{un}} = \begin{pmatrix} 2\nu(r, \sigma_1, +1) \\ 2\nu(r, \sigma_2, +1) \\ 2\nu(r, \sigma_3, +1) \end{pmatrix} \tag{120}$$

The calculation of the mean quadratic error matrix leads to

$$V_n^{\mathrm{un}}(x) = \frac{3}{n} \begin{bmatrix} 1 - x_1^2 & 0 & 0 \\ 0 & 1 - x_2^2 & 0 \\ 0 & 0 & 1 - x_3^2 \end{bmatrix}. \tag{121}$$

When each measurement is performed r times, then

$$V_n^{\mathrm{un}}(x) = \frac{3}{n} V_3^{\mathrm{un}}(x),$$

where $n = 3r$. Note, that the diagonal form of the above matrix is in good accordance with the independency of the Pauli measurements. The estimator (120) also uses only one relative frequency for each component of the Bloch vector estimate.

In [O10] a more general version of the above estimator was investigated by applying general (and not necessarily orthogonal) observables. The results showed, that the mean quadratic matrix is minimal if the three observables are orthogonal. It is similar to the result of [67], however in that approach the mean quadratic error was minimized but the information gain was maximized.

7.3.2 Standard qubit tomography

The basis of the following estimator is the POVM introduced in section 5.2.2, or [53]. The measurement has 6 outcomes with probabilities

$$\mathrm{Prob}(+i) = \frac{1}{6}(1 + x_i), \quad \mathrm{Prob}(-i) = \frac{1}{6}(1 - x_i), \quad i = 1, 2, 3.$$

It is important to note that (opposed to the previous section) there is only one kind of measurement in this case, i.e. $n = r$.

Based on the above POVM, the qubit state estimate (based on $r = n = 1$ measurement) is defined as

$$\hat{\chi}^{stand}(+i) = \frac{1}{2}(I + 3\sigma_i) = -I + 3F_{+i}, \quad \hat{\chi}^{stand}(-i) = \frac{1}{2}(I - 3\sigma_i) = -I + 3F_{-i}\,. \tag{122}$$

The symbols $\pm i$ stand for the different outcomes of the measurement corresponding to the operators defined in section 5.2.2. It can be shown, that the estimator (122) is unbiased.

The quadratic error matrix for n independent measurements is

$$V_n^{\text{stand}}(x) = \frac{1}{n}\begin{bmatrix} 3-x_1^2 & -x_1x_2 & -x_1x_3 \\ -x_1x_2 & 3-x_2^2 & -x_2x_3 \\ -x_1x_3 & -x_2x_3 & 3-x_3^2 \end{bmatrix}. \tag{123}$$

To compare the efficiency of the standard qubit tomography and estimator (120), the mean quadratic error matrices (121) and (123) are studied. The difference $V_n^{\text{stand}}(x) - V_n^{\text{un}}(x)$ has the form

$$\frac{1}{n}\begin{bmatrix} 3-x_1^2 & -x_1x_2 & -x_1x_3 \\ -x_1x_2 & 3-x_2^2 & -x_2x_3 \\ -x_1x_3 & -x_2x_3 & 3-x_3^2 \end{bmatrix} - \frac{3}{n}\begin{bmatrix} 1-x_1^2 & 0 & 0 \\ 0 & 1-x_2^2 & 0 \\ 0 & 0 & 1-x_3^2 \end{bmatrix}$$

$$= \frac{1}{n}\begin{bmatrix} 2x_1^2 & -x_1x_2 & -x_1x_3 \\ -x_1x_2 & 2x_2^2 & -x_2x_3 \\ -x_1x_3 & -x_2x_3 & 2x_3^2 \end{bmatrix} = \frac{1}{n}\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \circ \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \cdot \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \right)$$

where $\circ$ stands for the Hadamard product. Since the Hadamard product of two positive semidefinite matrices is positive semidefinite, we have $V_n^{\text{stand}}(x) \geq V_n^{\text{un}}(x)$. The measurement scheme defined in section 7.3.1 is more effective, than the standard one .

7.3.3 Minimal qubit tomography

Consider the following Bloch vectors

$$a_1 = \frac{1}{\sqrt{3}}(1,1,1), \qquad a_2 = \frac{1}{\sqrt{3}}(1,-1,-1),$$
$$a_3 = \frac{1}{\sqrt{3}}(-1,1,-1), \quad a_4 = \frac{1}{\sqrt{3}}(-1,-1,1).$$

and define the positive operators

$$F_i = \frac{1}{4}(\sigma_0 + a_i \cdot \sigma) \qquad (1 \leq i \leq 4). \tag{124}$$

They determine a positive operator valued measurement, $\sum_{i=1}^4 F_i = I$. The probability of the outcome i is

$$\text{Prob}(i) = \text{Tr} F_i \chi = \frac{1}{4}(1 + a_i \cdot x).$$

The above POVM is used and called **minimal qubit tomography** in [53]. In this case $n = r$, again, i.e. the estimator is capable to give an estimate from 1 measurement as opposed to (120), which needs at least 3 measurements to give an estimate.

The density matrix-valued estimator using $n = 1$ measurement is

$$\hat{\chi}^{min}(i) = -I + 6F_i \qquad (1 \leq i \leq 4).$$

is unbiased. If the measurement is performed n times, then the average (written in Bloch vector-valued form) is

$$\hat{x}^{min} = 3\sum_{i=1}^{4} \nu(n, A, i)a_i \tag{125}$$

where $\nu(n, A, i)$ is the relative frequency of the outcome i from the n measurements. The mean quadratic error matrix is

$$V_n^{\min}(x) = \frac{1}{n}\begin{bmatrix} 3 - x_1^2 & \sqrt{3}x_3 - x_1x_2 & \sqrt{3}x_2 - x_1x_3 \\ \sqrt{3}x_3 - x_1x_2 & 3 - x_2^2 & \sqrt{3}x_1 - x_2x_3 \\ \sqrt{3}x_2 - x_1x_3 & \sqrt{3}x_1 - x_2x_3 & 3 - x_3^2 \end{bmatrix}. \tag{126}$$

Unfortunately, the above matrix is not comparable with the mean quadratic error matrix (121), i.e. their difference is indefinite. However, $\mathrm{Tr}V_n^{\mathrm{un}} \leq \mathrm{Tr}V_n^{\min}$.

7.4 Summary

The quantum state estimation problem defined in section 5.2.3 was solved in this chapter for general finite quantum systems. First of all, the observables were defined which provide the basis of the estimation scheme. In this case, one observable was defined for each parameter of the density matrix using the Bloch parametrization (91).

The estimator gives a point estimate based on the relative frequencies of certain outcomes of the previously defined observables. Just like for the Bayesian estimator (chapter 6) the false estimates caused a problem that was solved by a modification of the unconstrained estimator. *The constrained estimate can be computed using two different methods, however their results are practically the same* (their difference is negligible).

For the unconstrained estimator it was possible to compare its effectiveness to two different estimation schemes known from the literature, called standard qubit tomography and minimal qubit tomography. The comparison was based on their mean quadratic error matrices of the estimators. The results shows that the unconstrained estimator is more effective, than the other two.

Chapter 8

Conclusions

The aim of this chapter is to sum up the results of the previous four chapters. Section 8.1 places the results of part I and part II in the framework of state feedback control. The new results proposed by the thesis are collected to four thesis points in section 8.2. The possible directions in future work are sketched in section 8.3.

8.1 Stability analysis and state estimation for state feedback control

The results presented in this work are based on a suitable parametrized state-space model of two physically motivated challenging system classes: the class of quasi-polynomial systems and that of finite dimensional quantum systems. Both the practically feasible stability analysis and the globally stabilizing feedback control design of process systems in QP form, and quantum state estimation has been possible using the framework of system- and control theory.

In part I, the quasi-polynomial system representation was used to describe general nonlinear process systems. The state equation for an autonomous (truncated or closed-loop) QP system (4) is parametrized in such a way that its stability depends only on a quadratic matrix $M = BA$. Using a well-known physically motivated Lyapunov function candidate family suitably designed for the QP model class, the global stability analysis and also the globally stabilizing feedback design were formulated as optimization problems. In particular, global stability analysis of QP systems requires to solve an LMI for which practically feasible solution methods exist. Thus, the otherwise hard problem of global stability analysis falls into a solvable category by utilizing the special structure of QP systems.

Similarly, a suitably parametrized state representation has been used to solve the simplest state-feedback control related task, the state estimation of finite dimensional quantum mechanical systems in part II, where two essentially different methods were developed and examined. Here the state is described by density matrices that are positive self-adjoint matrices with unit trace. They have been parametrized by using real vectors (Bloch vector parametrization, section 5.1.4). This parametrization enabled us to use standard parameter estimation methods, the Bayesian and the LS methods for quantum state estimation. A complete es-

timation scheme was given in both cases. The difficulties arising from the nature of quantum measurement was avoided by a "measure-and-throw" assumption, i.e. sufficiently many quantum systems was used being in the same state.

8.2 New results

The new scientific results presented in this work are summarized in this section. They are arranged in four thesis points as follows.

Thesis 1. *The global stability analysis of nonlinear process systems being in quasi-polynomial representation has been formulated as a linear matrix inequality. The chance to prove global stability has been extended by time-reparametrization where the scaling factors were determined and the global stability is proved by solving a bilinear matrix inequality.* (Chapter 3)
([**O1**], [**O2**], [**O3**], [**O4**])
It was shown, that the negative definiteness condition of the Lyapunov function of QP and LV systems is equivalent to a linear matrix inequality, thus the stability analysis of QP systems (and general nonlinear process systems embedded into QP form) is equivalent to the feasibility of a LMI. The LMI is non-strict if the model has been obtained by embedding.

It has been shown, that the time-reparametrization transformation introduces a re-scaling in the QP system's quasi-monomials, such that the global stability of transformed QP system is equivalent to that of the original one. This way the global stability analysis has been extended to a wider class of QP systems by embedding the parameters of the time-reparametrization transformation into the global stability analysis, when one has to solve a bilinear matrix inequality.

Thesis 2. *The globally stabilizing quasi-polynomial state feedback design problem for quasi-polynomial systems has been expressed as a bilinear matrix inequality. The problem has been reformulated so that it can be solved by an existing iterative LMI algorithm.*
A supplementary feedback controller that shifts some coordinates of the closed loop systems's steady state has been computed from a linear set of equations. Conditions on the number of shiftable coordinates were also given. (Chapter 4)
([**O5**], [**O11**])
A globally stabilizing state feedback design problem was formulated using the global stability analysis results of **Thesis 1**. The problem has been solved as a bilinear matrix inequality feasibility problem, having two groups of variables, one for the parameters of the Lyapunov function and another for the feedback gains. The proposed method does not utilize the objective function of the BMI optimization problem (158), thus it is a possible point to introduce some performance or robustness specifications.
If one is to solve just the BMI feasibility without additional criteria, the prob-

lem has been reformulated so that an existent iterative LMI algorithm is suitable for its' solution.

The stabilizing state feedback may shift the closed loop system's equilibrium points into unwanted values that's why the possibilities of designing an additional feedback that (partially) sets back the original steady state were proposed. It was shown that under certain conditions on the closed loop system's Lotka-Volterra coefficient matrix it is possible to design such a controller. It's parameters were determined from a linear set of equations. In most cases, however, it is only possible to redesign the steady state for only a few number of state coordinates.

Thesis 3. *A Bayesian state estimation scheme was developed for a single quantum bit. As a measurement scheme, the von Neumann measurement of the Pauli spin operators was used. Using the independency of the applied measurements, the problem was solved componentwise.*
The estimator was improved in order to avoid estimates laying out of the state space by an additional constraint. (Chapter 6)
([**O6**], [**O7**])
The relaxed state estimation problem defined in section 5.2.3 was solved for a quantum bit using Bayesian methodology (Appendix A.1.4). The estimation was based on the Bloch vector representation of quantum states and on the von Neumann measurements of the three Pauli spin operator.

Since the three measurements are incompatible, the problem was regarded to be an independent estimation of the three Bloch vector components. The total estimate was obtained by multiplying the three probability density functions. The obtained Bayesian state estimator performed weak for estimation pure states, so an additional constraint was added to the problem. This step resulted in an estimator that always gives a physically meaningful result, however it's computation is more difficult.

Using the simulator Spinsim the constrained and the unconstrained Bayesian state estimation methods were compared. Their difference was outstanding in the case of estimating a pure state, or estimating based on a small number of measurement data.

Thesis 4. *A novel, componentwise quantum state estimation scheme was developed for N-level quantum mechanical systems. The measurement data were obtained from the von Neumann measurement of N^2-1 independent observables. The estimator uses the measurement data of the above measurement to determine the N^2-1 parameters of the density matrix.*
An algebraic and a geometric method was proposed to force the estimator to produce physically meaningful result.
The effectiveness of the estimator was compared to other estimation schemes using the mean squared error matrix. (Chapter 7)
([**O8**], [**O9**], [**O10**])
The quantum state estimation problem refined for quantum systems was solved for N-level quantum systems, not only for qubits.

The collection of observables consists of 3 group of von Neumann measurements. The basis of the estimation scheme is the Bloch parametrization used for general finite quantum systems. The estimator (114) consists of $N^2 - 1$ equations for the $N^2 - 1$ parameters of the density matrix, and gives a point estimate based on the relative frequencies of certain outcomes of the observables.

It has been proven that the estimator is unbiased but suffers from the tendency to give false estimates so a modification was necessary to respect the positivity constraint (87).

It was shown that for invertible states the constrained estimator converges to the unconstrained one when the size of the measurement data increases. If the real state is on the boundary of the state space, then the unconstrained estimator is useless, since it is always necessary to correct its result by one of the two constraining methods proposed.

The effectiveness of the unconstrained estimator was compared to two different estimation schemes available in the literature. The comparison was based on their mean quadratic error matrices. It has been shown that the proposed scheme is more efficient than the other two.

8.3 Future work and applicability areas

Based on the results presented in section 8.2 the aimed future directions are summarized in this section. The areas of possible applicability is also presented here.

Quasi-polynomial system representation (4) is a good tool for describing biochemical systems given in the form of reaction kinetic networks. The state variables of such systems are typically concentrations, i.e. they are also positive systems. These reaction kinetic networks are given by their *mass action law* description. This special form enables to apply the results of classical reaction kinetics together with the results of thesis points 1 and 2.

On the other hand, mixed mechanical-thermodynamical systems (e.g. gas turbines) can also be embedded into QP representation, and with a Lyapunov function (21) their global stability can be investigated. Note, that using a *quadratic Lyapunov function*, the region of their (local) stability can be conveniently determined by solving LMIs.

By formulating robustness and/or performance specifications as an objective function it will be possible to prescribe the quality of the controller to be designed. The selection of the feedback controller structure is also an important question since a wise choice can decrease the size of the BMI to be solved. That's why controller structure selection based on graph theoretic methods is another direction of future work.

The controller design BMI with the built-in robustness specifications and the controller structure design together would extend the controller design problem to a complete methodology for the stabilizing control of nonlinear process systems given in QP representation.

The measure-and-throw philosophy applied in the problem statement of quantum state estimation is in good agreement with the measurement of the polarization of photons in a photon beam, so the quantum state estimation methods presented in thesis points 3 and 4 can be applied for photon source identification. This way, the state of the system corresponds to the photon polarization, and since the polarization of photons emitted by source is not varying, there is no need to deal with the dynamics of the system.

After developing reliable state estimation methods for the quantum state estimation problem supposing no dynamics the next step would be to modify the developed methods for quantum process tomography. Its problem statement is as follows: known quantum states are sent through a quantum channel with unknown parameters and the states leaving the channel are measured. Give an estimate of the channel's parameters.

Another possible way is to include quantum dynamics to the system whose state is to be estimated. Of course, in this case a totally different kind of measurement should be applied that influences the system not as rough as the von Neumann measurement. The drawback of such measurement scheme might be the fact that it would not provide as much information as the von Neumann, or POVM type.

After having a correct method for quantum state estimation that involves also the dynamical model of the quantum system, the way is clear towards quantum control.

As for the system theoretic description of the jump-like effect of the measurement in the state trajectory, the class of impulsive systems [3] is a promising candidate. The only difficulty is that the impulses caused by the measurement are stochastic, i.e. it leads to the field of stochastic impulsive system.

8.4 Own papers

[O1] **A. Magyar** and K. M. Hangos. Lotka-Volterra representation of process systems for stability analysis. In *Proceedings of 14th International Conference on Process Control (PC'03), Štrbské Pleso, Slovakia*, 2003. http://www.ka.chtf.stuba.sk/pc03.

[O2] **A. Magyar** and K. M. Hangos. Lotka-Volterra representation of process systems for stability analysis. In *Proceedings of 4th International PhD Workshop on Systems and Control, Libverda, Czech Republic*, 2003. on CD.

[O3] **A. Magyar**, G. Szederkényi, and K. M. Hangos. Quadratic stability of process systems in generalized Lotka-Volterra form. In *Proceedings of 6th IFAC Symposium on Nonlinear Control (NOLCOS 2004), Stuttgart, Germany*, 2004. http://www.nolcos2004.uni-stuttgart.de.

[O4] G. Szederkényi, K.M. Hangos, and **A. Magyar**. On the time-reparametrization of quasi-polynomial systems. *Physics Letters A*, 334:288–294, 2005. **impact factor: 1.550**.

[O5] **A. Magyar**, G. Szederkényi, and K. M. Hangos. Quasi-polynomial system representation for the analysis and control of nonlinear systems. In *Proceedings of 16th IFAC World Congress, Prague, Czech Republic*, 2005. on CD.

[O6] **A. Magyar**, K. M. Hangos, and D. Petz. Bayesian qubit tomography. In *Proceedings of 6th International PhD Workshop on Systems and Control, Izola - Simonov zaliv, Slovenia*, 2005. on CD, ISBN: 961-6303-74-0.

[O7] **A. Magyar**, D. Petz, and K. M. Hangos. Bayesian qubit state estimation. In *Proceedings of 14th IFAC Symposium on System Identification (SYSID-2006), Newcastle, Australia*, pages 949–954, 2006. on CD.

[O8] D. Petz, K. M. Hangos, **A. Magyar**, and L. Ruppert. State estimation of N-level quantum systems. Research report no. SCL-007/2006, Computer and Automation Research Institute, 2006.

[O9] L. Ruppert and **A. Magyar**. The effect of constraints on LS state estimators for a qubit. In *Proceedings of 7th International PhD Workshop, Hruba Skala, Czech Republic*, 2006. on CD, ISBN: 80-903834-1-6.

[O10] D. Petz, K. M. Hangos, and **A. Magyar**. Point estimation of states of finite quantum systems. *Journal of Physics A: Math. Theor.*, 40:7955–7969, 2007. **impact factor: 1.566**.

[O11] **A. Magyar**, G. Szederkényi and K.M. Hangos,. Globally stabilizing feedback control of process systems in generalized Lotka-Volterra form. *Journal of Process Control*, 2007. in print.

Appendix A

Appendix

A.1 Basics of system and control theory

In general, the word *system* (Figure 19) is used for various concepts and features of the world. The common properties of them are the following [2]:

- The system is a well defined part of the world and it is isolated from its environment.
- The interactions between the system and its environment are carried out using signals that are time dependent variables. Through the *input* signal it is possible for the environment to affect the system, while the responses of the system are fed back to the environment via the *output* signal.
- All information about the system's behavior up to the present is contained in the actual *state* of the system. The state is also a signal just like the input and the output.

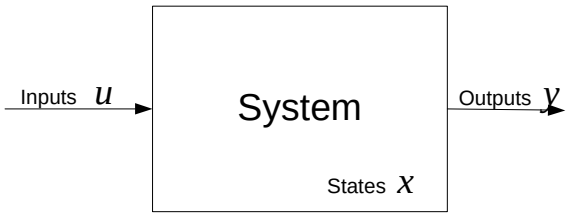

Figure 19: Scheme of a system

The system can be given by an abstract operator **S** that maps the set of input signals onto the set of output signals.

Systems can be represented with various mathematical tools. Some of them use only the inputs and the outputs, hence they are called *input-output representations*. Another class is of *state space representations* which uses the states, in addition to the inputs and outputs, for describing the behavior of the system.

If the time is regarded to be a continuous variable $t \in \mathbb{R}^+$, then the representation is a *continuous time* one. The other possibility is to choose the time set to be discrete

$(t \in \{t_0, t_1, \dots t_k, \dots\})$, which results in a *discrete time* system model [2]. Throughout this thesis, we restrict ourselves to the class of finite dimensional systems where the state of the system at any time is described by a vector.

A.1.1 System classes, basic system properties

In what follows some of the most important state space models of different system classes are introduced in brief, together with the most important system properties.

Continuous time linear time invariant systems

If the system operator **S** is both linear and time invariant, it is possible to describe the system by a higher order linear differential equation with constant coefficients [31]. Another way is to use the so-called *state space representation*:

$$\begin{array}{rcll} \dot{x}(t) & = & Ax(t) + Bu(t) & x(t_0) = x^0, \\ y(t) & = & Cx(t) & \end{array} \tag{127}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, and $y \in \mathbb{R}^q$. The matrices are accordingly

$$A \in \mathbb{R}^{n \times n}, \quad B \in \mathbb{R}^{n \times p}, \quad C \in \mathbb{R}^{q \times n}.$$

The above representation consists of a system of first order (linear) differential equations (*state equation*) and an algebraic equation (*output equation*). The important dynamical properties of (127) such as *controllability, observability*, and *stability* can be conveniently examined using tools of linear algebra.

System (127) is termed *asymptotically stable*, if

$$\lim_{t \to \infty} x(t) = 0$$

holds for the solution of

$$\dot{x}(t) = Ax(t), \quad x(t_0) \neq 0, \quad t > t_0. \tag{128}$$

A system (127) is asymptotically stable iff all of the eigenvalues of matrix A have strictly negative real part:

$$\mathrm{Re}\lambda_i(A) < 0, \quad i = 1 \dots, n.$$

Fundamental matrix of continuous time LTI systems The solution (128) can be expressed in the following from

$$x(t) = \Phi(t, t_0)x(t_0)$$

where $\Phi(t, t_0)$ is the so-called *fundamental matrix* of the LTI system, [9]:

$$\Phi(t, t_0) = \exp\left((t - t_0)A\right) \tag{129}$$

The basic properties of the fundamental matrix are as follows.

- $\Phi(t_0, t_0) = I$
- $\Phi(t_2, t_0) = \Phi(t_2, t_1)\Phi(t_1, t_0)$, where $t_0 \leq t_1 \leq t_2$
- It satisfies the matrix differential equation

$$\dot{\Phi}(t, t_0) = A\Phi(t, t_0) \tag{130}$$

Continuous time nonlinear (time invariant) systems

A more general class of systems can be represented by the following state space model:

$$\begin{array}{rcll} \dot{x}(t) & = & f(x(t),u(t)) & x(t_0)=x^0, \\ y(t) & = & g(x(t)) & \end{array} \tag{131}$$

where

$$f:\mathbb{R}^{n+p}\to\mathbb{R}^n, \quad g:\mathbb{R}^n\to\mathbb{R}^q$$

are nonlinear functions.

Analysis of nonlinear systems needs more sophisticated methods [64], [29]. The stability analysis of (131) is twofold. Local stability around equilibrium points of the state equation is performed by local linearization i.e. it is traced back to the analysis of (127). On the other hand, global stability analysis calls for the searching of a suitable *Lyapunov function* V with the following properties:

- scalar valued function: $V:\mathbb{R}^n\to\mathbb{R}^+$
- positive: $V(x(t))>0$
- decreasing in time: $\frac{d}{dt}V(x(t))<0$

Theorem 3 *A system* $\boldsymbol{S}$ *is asymptotically stable if there exists a Lyapunov function with the above properties.*

Note, that the above theorem is not constructive, i.e. the form of the Lyapunov function is not known for a general nonlinear system (131). Moreover, the Lyapunov function of a system is not unique. However for certain system classes its general form is well-known. E.g. for linear systems, it is a quadratic function

$$V(x(t))=x^TPx, \quad P>0. \tag{132}$$

In spite of the fact that (132) is a Lyapunov function for (127), it can be used for analyzing the stability of nonlinear systems: it is suitable for determining regions of local stability around its equilibrium point(s) [47].

Nonlinear input affine systems

A special subset of systems in form (131) is the family of so-called nonlinear input affine systems [64]. What makes it attractive is the fact that although the system is nonlinear in the states, it is linear in its inputs.

$$\begin{array}{c} \dot{x}(t)=f(x(t))+\sum_{i=1}^{p}g_i(x(t))u_i(t) \\ y(t)=h(x(t)) \qquad x(t_0)=x^0, \end{array} \tag{133}$$

where $u(t)=(u_1(t),u_2(t),\ldots,u_p(t))^T$,

$$f:\mathbb{R}^n\to\mathbb{R}^n, \quad g_i:\mathbb{R}^n\to\mathbb{R}^n,\ i=1,\ldots,p, \quad h:\mathbb{R}^n\to\mathbb{R}^q$$

are nonlinear functions.

Control methods available for nonlinear systems usually assume a state space model of the form (133), see e.g. [29], [47].

Discrete time linear time invariant stochastic systems

For the state space models (127), (131), (133) presented above the time t was a continuous variable, hence the state equations were differential equations [2]. Choosing the time set to be discrete, one arrives at a state space model where the state equation is a difference equation.

In practice, there are some inputs of the system that are not controlled. On the other hand, the sensors measuring the output of the system also generate some noise. This leads to the following system model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v(k) \\ y(k) &= Cx(k) + e(k) \end{aligned} \tag{134}$$

where $\{v(k)\}_0^\infty$, and $\{e(k)\}_0^\infty$ are zero mean discrete time independent white noise processes, $v(k)$ is called system noise, $e(k)$ is termed sensor noise:

$$\begin{aligned} &E[v(k)v(k)^T] = R^v, \quad E[v(k)v(j)^T] = 0, \quad k \neq j \\ &E[v(k)e(j)^T] = 0, \quad \forall k, j \\ &E[e(k)e(k)^T] = R^e, \quad E[e(k)e(j)^T] = 0, \quad k \neq j, \end{aligned}$$

and the initial conditions of (134) are given by

$$E[x(0)] = m^0, \quad \text{cov}[x(0)] = R^0$$

A.1.2 Controller design

The *general problem statement of control* is as follows. Given a system, and a control aim, modify the system such that the modified one fulfills the control aim.

In most cases the control aim is reached by using *feedback*, i.e. the output signal is fed back to the input through a subsystem called *controller* (see figure 20). It is

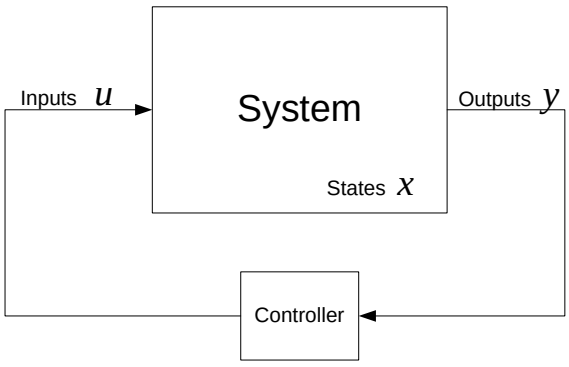

Figure 20: Scheme of feedback control

possible to categorize the feedback controllers based on the

- type of the signal used (*output feedback* vs *state feedback*)
- dynamical properties of the controller. If it also contains the derivatives of the signals then it is a *dynamic feedback*, otherwise it is called *static feedback*.

- type of the function implemented in the controller. E.g. a *linear feedback* controller's output is a linear function of its input signal. A wider class of feedback controllers is covered by the term *nonlinear feedback* - but their analysis and synthesis needs a deeper mathematical knowledge.

In what follows a few different feedback controller types are summarized in brief.

Linear quadratic (LQ) state feedback controller

For LTI systems given by their state space model (127), let us define the following functional:

$$J(x,u) = \int_0^T \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt, \tag{135}$$

where matrices $Q \geq 0$ and $R > 0$ are weighting the states and the inputs. Find an input signal $u(t),\ 0 < t < T$ that minimizes (135).
If the control energy available is *cheap*, then R is smaller than Q (supposed that they are comparable). This setup leads to a control where the state- and output signals have no *overshoot*, and they *settle down* relatively fast.
On the other hand, if the relation between Q and R was turned, then one would get a closed loop system with weaker dynamical properties, but the control energy consumption would be less.

Assuming, that $T \to \infty$ in (135), and that the system is *jointly controllable and observable* (see e.g. [21]), the solution of the above problem is a *static state feedback* controller of the form

$$u(t) = -R^{-1}B^T K x(t), \tag{136}$$

where K is the solution of a so-called control algebraic Riccati equation:

$$KA + A^T K - KBR^{-1}B^T K + Q = 0$$

It is necessary to emphasize, that the obtained controller (136) is a state feedback, and in the general case it is only possible to measure the outputs of the system. It means, that the scheme of figure 20 needs to be extended by a unit before the controller that computes the state signal from the output, as it can be seen in figure 21. In case of deterministic signals, this unit is called *state observer* while in the

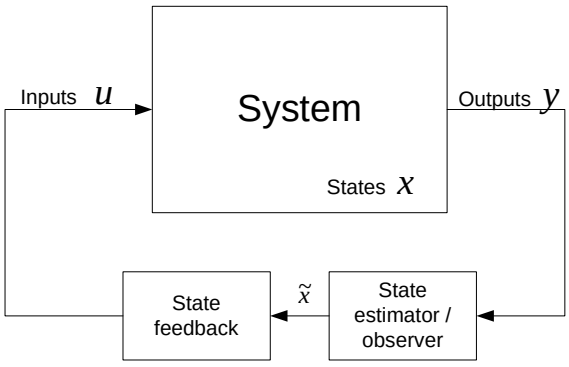

Figure 21: State feedback and state estimator

stochastic case the term *state estimator* is used.

Input-output linearization via state feedback

The aim of this method is to apply a nonlinear state feedback in order to get a closed loop system that is a linear one. Given a system in the form (133) where the input- and output signals are both scalar, i.e. the general input-affine system model (133) specializes to

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{137}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$. Suppose, that the *relative degree* of system (137) is r at point x_0, i.e.

1. $L_g L_f^k h(x) = 0$ for all x in a neighborhood of x_0 and $k < r - 1$
2. $L_g L_f^r h(x) \neq 0$

where $L_f h(x)$ stands for the so-called *Lie derivative* [29]

$$L_f h(x) = \sum_{i=1}^{n} \frac{\partial h(x)}{\partial x_i} f_i(x).$$

The relative degree r equals to the number of times one has to differentiate $y(t)$ in order that $u(t)$ explicitly appear in $y^{(r)}(t)$.
Using the local coordinates transformation (138) in the neighborhood of x_0,

$$z = T(x) = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{r-1} h(x) \\ \phi_1(x) \\ \vdots \\ \phi_{n-r}(x) \end{bmatrix} \tag{138}$$

where $\phi_i(x)$ are chosen such a way that $L_g \phi_i(x) = 0$ around x_0, $i = 1, \ldots, n - r$, and applying the input

$$u = \frac{1}{L_g L_f^{r-1} h(x)} \left(-L_f^r h(x) + v \right),$$

the system (137) can be rewritten in the form

$$\begin{aligned} \dot{z}_1 &= z_2 \\ &\vdots \\ \dot{z}_{r-1} &= z_r \\ \dot{z}_r &= v \\ \dot{z}_{r+1} &= q_{r+1}(z) \\ &\vdots \\ \dot{z}_n &= q_n(z) \\ y &= z_1 \end{aligned} \tag{139}$$

It is apparent that the first r states of (139) form a linear subsystem with a new input v, and there is an additional nonlinear dynamics, the so-called *zero dynamics* [29]. The zero dynamics describes the systems behavior when its output y is forced to be constantly zero.

The applicability condition of the above method is the asymptotic stability of the zero dynamics. If it holds, then any suitable control method from the area of LTI systems (e.g. an LQ type) can be applied to stabilize the linear subsystem, and the overall dynamics will be stable.

A.1.3 State estimation

Most of the modern controllers (e.g. LQ, or $\mathcal{H}_\infty$ optimal controllers [68]) use state feedback. This calls for a method that determines the actual state x of the system using the supposed system model and measured input-output data corrupted by measurement noise. Such a method is called state estimation.

There are some requirements a good estimate $\hat{x}$ should meet. The first one is *unbiasedness* which means, that the expected value of the estimate $\hat{x}$ equals to the real state x. Among the unbiased estimates one strains after finding an estimate with minimal *variance*. Moreover, a *consistent* estimate is needed, i.e. an estimate that converges to the real state as the number of measurements increases.

Kalman-filter

As an example, the famous Kalman-filter [32] can be mentioned. The discrete time version of the estimator for discrete time stochastic LTI systems is in the form

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + K(k)(y(k) - C\hat{x}(k)), \quad E\left[\hat{x}(0)\right] = m^0. \tag{140}$$

As it can be seen in (140), it uses a discrete time stochatic LTI system (134) where the system noise is being modelled by the measurement noise. Of course, $e(k)$ and $v(k)$ supposed to have similar statistical properties. The time dependent matrix $K(k)$ is the so-called *Kalman-gain* which is to be determined for each time instance k.
Note, that the above estimator is unbiased, since both $e(k)$ and $v(k)$ are zero mean random variables of the same distribution, $k = 0, 1, 2, \ldots$.
Defining the reconstruction error process as

$$\tilde{x}(k) = x(k) - \hat{x}(k),$$

the problem to be solved is the minimization of the norm of the covariance function of $\tilde{x}(k)$:

$$P(k) = E\left[\left(\tilde{x}(k) - E\left[\tilde{x}(k)\right]\right)\left(\tilde{x}(k) - E\left[\tilde{x}(k)\right]\right)^T\right]$$

The minimization leads to the following pair of formulas:

$$\begin{aligned} K(k) &= AP(k)C^T(R^e + CP(k)C^T)^{-1} \\ P(k+1) &= AP(k)A^T + R^v - AP(k)C^T(R^e + CP(k)C^T)^{-1}CP(k)A^T \end{aligned},$$

where R^v and R^e are coming from (134).
The above estimator is optimal for Gaussian white noise processes only. The initially assumed LTI system model is also a crucial point, and it is not recommended to differ from it. However, the Kalman-filter can be extended both for nonlinear systems and colored noise processes.

A.1.4 System parameter estimation

As it was seen in section A.1.2 and section A.1.3, most of the controller design and state estimation problems need the dynamical model of the system. Estimating the model parameters is also a basis of system diagnostics since different parameter values may refer to certain faults of the plant. Thus, parameter estimation, or *identification* of dynamical systems is an important field of system- and control theory [52], [54].

The basic problem of system parameter estimation can be formulated as follows:
Given

- a parametric dynamical system model

$$\hat{y} = \mathcal{M}(x; \theta) \tag{141}$$

  where $\mathcal{M}$ is the model, θ is the vector of model parameters, and $\hat{y}$ is the predicted output computed by the model.

- a record of measurement data

$$D^k = \{(x(i), y(i)), \quad i = 0, \ldots, k\},$$

  where $y(i)$ is corrupted by measurement noise.

- a suitable *signal norm* $||.||$ to measure the difference between the model output y^M and the measured output y of the system. The loss function describing the quality of the parameter estimation can be defined as

$$L = ||y - \hat{y}||.$$

Find the estimate for the parameters θ of (141) that minimizes the loss function L.

Least squares parameter estimation

The basis of least squares (LS) parameter estimation is the direct minimization of the prediction error. Its' simplicity and good statistical properties make this estimation method very popular [52], [62], [45].

The 2-norm of the prediction error signal based on N measurements is defined as

$$V_N(\theta, D^N) = \frac{1}{N} \sum_{k=1}^{N} \left(y(k) - \hat{y}(k|\theta)\right)^2.$$

The above quantity is also called the loss function of the estimation. If the parameterized system model is assumed to be

$$\hat{y}(k) = f(k, \theta),$$

then the loss function is:

$$V_N(\theta, D^N) = \frac{1}{N} \sum_{k=1}^{N} \left(y(k) - f(k, \theta)\right)^2 . \tag{142}$$

For a given set of measurement data D^N, the loss function $V_N(\theta, D^N)$ is a function that maps the parameter space onto $\mathbb{R}^+$. The determination of the parameter estimate is equivalent with the minimization of (142) with respect to θ, it can be performed e.g. by *gradient method*.

Note, that if the model is a linear function of the parameters, then the minimizer can be given explicitly, without searching.

The above introduced method uses all the available measurement data and the estimate is determined in one step. Such parameter estimation methods are called *off-line*. If the measurement data is not available together but only one at a time (e.g. it is not possible to store a large amount of data), then *online* methods are preferred.

Bayesian parameter estimation

The basic idea of Bayesian parameter estimation is that *any unknown quantity (including parameters θ) of a system is regarded as a random variable* [48]. The measured values that are available from the system are used to determine the *conditional probability density function* $p(\theta|D^k)$ of the unknown parameters conditioned on the measured values $D^k = \{y(j) : \ j = 0, 1, ..., k\}$. In the case of *discrete time stochastic systems with measurable output* $\{y(j) : \ j = 1, 2, \dots\}$ that vary in time, one may separate the current ($\tau = k$) data from the previously measured ones in D^{k-1} to have

$$p(\theta \mid D^k) = p(\theta \mid y(k), \ D^{k-1}) . \tag{143}$$

Then we can use the Bayes formula developed for conditional probability density functions:

$$p(a|b, c) = \frac{p(b|a, c)p(a|c)}{\int p(b|\nu, c)p(\nu|c)\, d\nu} \tag{144}$$

with the correspondence

$$b \sim y(k) \quad , \quad a \sim \theta \quad \text{and} \quad c \sim D^{k-1}.$$

This way we obtain the following *recursive formula* for parameter estimation:

$$p(\theta \mid D^k) = \frac{p(y(k) \mid D^{k-1}, \ \theta)p(\theta \mid D^{k-1})}{\int p(y(k) \mid D^{k-1}, \ \nu)p(\nu \mid D^{k-1})d\nu} \tag{145}$$

where $p(y(k) \mid D^{k-1}, \ \theta)$ is the parametrized system model describing the dynamics of the system.

There are two properties of the parameter estimation formula (145) that are important to emphasize. Firstly, it needs a stochastic system model in the form of a conditional probability density function. The other property is that it is an *online* method, i.e. because of the recursive formula it is not necessary to have all the measurement data at the starting of the estimation procedure. However, it is possible to express (145) in a non-recursive way, that would result in an off-line estimation method.

Another important speciality of Bayesian parameter estimation is that it can take into account any initial a'priori knowledge about a parameter to be estimated in the form of the so called *prior probability density*. With the prior probability density, we can restrict the range of the parameter or give information about the precision of the initial guess in the form of a variance.

A.2 Bloch vector space in the N-level case

The N-dimensional generalization of the Bloch vector space (a slightly different version of the one described in section 5.1.4) is given in [35]. The basis matrices of that approach are the generators G_i of $SU(N)$. (Note, that this work uses a different basis , see (91)).

The basic properties (87) of the density matrices are transformed to Bloch vectors using the following theorem (see [35] for the proof)

Theorem 4 *Let $a_i(x)$ be the coefficients of the characteristic polynomial $\det(tI_N - \chi)$, where χ is a density matrix with properties (87) and define*

$$\mathcal{B}(\mathbb{R}^{N^2-1}) = \{x \in \mathbb{R}^{N^2-1} : a_i(x) \geq 0, i = 1, \ldots, N\}.$$

Then a map

$$x \in \mathcal{B}(\mathbb{R}^{N^2-1}) \to \chi = \frac{1}{N} I_N + \frac{1}{2} x_i G_i$$

is a bijection from $\mathcal{B}(\mathbb{R}^{N^2-1})$ to the space of density matrices.

The concrete form of the coefficients $a_i(x)$ are as follows:

$$\begin{aligned}
1!a_1 &= 1, \\
2!a_2 &= \left(\tfrac{N-1}{N} - \tfrac{1}{2}|x|^2\right), \\
3!a_3 &= \left(\tfrac{(N-1)(N-2)}{N^2} - \tfrac{3(N-2)}{2N}|x|^2 + \tfrac{1}{2} g_{ijk} x_i x_j x_k\right), \\
4!a_4 &= \left(\tfrac{(N-1)(N-2)(N-3)}{N^3} - \tfrac{3(N-2)(N-3)}{N^2}|x|^2 + \tfrac{3(N-2)}{4N}|x|^4 + \right. \\
&\quad \left. + \tfrac{2(N-2)}{N} g_{ijk} x_i x_j x_k - \tfrac{3}{4} g_{ijk} g_{klm} x_i x_j x_l x_m\right), \\
&\quad \ldots
\end{aligned} \tag{146}$$

where g_{ijk} stands for the structure constants of $SU(N)$

For example, the Bloch vector space of a 3-level quantum system is defined by the first three constraints of (146), which gives, that it is the ball of radius $\frac{3}{\sqrt{3}}$ subject to the additional constraint

$$36 - 9|x|^2 + 9g_{ijk}x_i x_j x_k \geq 0.$$

Substituting the values of g_{ijk}, the above equation reads

$$-8 + 18|x|^2 - 27x_3(x_4^2 + x_5^2 - x_6^2 - x_7^2) + 6\sqrt{3}x_8^3 -$$

$$-9\sqrt{3}[2(x_1^2 + x_2^2 + x_3^2) - (x_4^2 + x_5^2 + x_6^2 + x_7^2)] -$$

$$-54(x_1x_4x_6 + x_1x_5x_7 + x_2x_5x_6 - x_2x_4x_7) \geq 0.$$

It defines a proper convex asymmetric subset of the ball in $\mathbb{R}^{N^2-1}$ having radius $\frac{3}{\sqrt{3}}$.

It is easy to see, that by increasing the dimension of the state space, the number of additional conditions to be regarded increases, which makes it less tractable. For numerical simulations and experiments, the approach for handling Bloch vector space of N level systems presented in this work (section 5.1.4) is easier, and more straightforward.

A.3 Examples of QP feedback design

A.3.1 Example with a rank deficient M matrix

Consider the following open generalized mass-action law system

$$\begin{aligned} \dot{x}_1 &= 0.5x_1 - x_1^{2.25} - 0.5x_1^{1.5}x_2^{0.25} + u_1 \\ \dot{x}_2 &= x_2 - 0.5x_2^{1.75} + u_2 \end{aligned} \tag{147}$$

where x_1 and x_2 are the concentrations of chemical species $\mathcal{A}_1$ and $\mathcal{A}_2$ ($[\frac{moles}{m^3}]$), while u_1 and u_2 (the manipulable inputs) are their volume-specific component mass inflow rates ($[\frac{moles}{m^3 sec}]$). The above two differential equations originate from the component mass conservation equations constructed for a perfectly stirred balance volume [23] under the following modeling assumptions:

1. constant temperature and overall mass,
2. constant physico-chemical properties (e.g. density),
3. presence of an inert solvent in a great excess,
4. presence of the following reaction network:
   - autocatalytic generation of the species $\mathcal{A}_1$ and $\mathcal{A}_2$ (e.g. by polymer degradation when they are the monomers and the polymers are present in a great excess) giving rise to the reaction rates $0.5x_1$ and x_2 (the first terms in the right-hand sides) respectively,

- a self-degradation of these species described by the reaction rates $-x_1^{2.25}$ and $-0.5x_2^{1.75}$ (the second terms on the right-hand sides) respectively,
- a catalytic degradation of the specie $\mathcal{A}_1$ catalyzed by specie $\mathcal{A}_2$ that corresponds to $-0.5x_1^{1.5}x_2^{0.25}$ in the first equation only (the third term).

The control aim is to drive the system to a positive equilibrium

$$x_1^* = 2.4082 \ \frac{moles}{m^3}, \quad x_2^* = 16.3181 \ \frac{moles}{m^3}.$$

This goal can be achieved e.g. by the following nonlinear feedback:

$$\begin{array}{rcl} u_1 & = & 0.5x_1x_2^{0.75} \\ u_2 & = & 0.5x_1^{1.25}x_2 + 0.5x_1^{0.5}x_2^{1.25}. \end{array} \tag{148}$$

The above inputs being the component mass flow rates fed to the system (they are both positive) are needed for compensating for the degradation of the specie $\mathcal{A}_1$ and $\mathcal{A}_2$.

By substituting (148) into (147), we obtain the controlled system that is a QP system with the following matrices

$$A = \begin{bmatrix} -1 & 0.5 & -0.5 \\ 0.5 & -0.5 & 0.5 \end{bmatrix} \tag{149}$$

$$B = \begin{bmatrix} 1.25 & 0 \\ 0 & 0.75 \\ 0.5 & 0.25 \end{bmatrix}, \quad L = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} \tag{150}$$

The eigenvalues of the Jacobian matrix of the system at the equilibrium point are -6.4076 and -0.7768.

Since the rank of $M = B \cdot A$ in this case is only 2, we can only use the algorithm described in [17] to prove that the LMI (22) is not feasible in this case.

However, by solving (46) using again the algorithm described in [36] we find that we can use the following time-reparametrization:

$$\omega = \begin{bmatrix} -0.25 & -0.5 \end{bmatrix}^T \tag{151}$$

and the diagonal matrix containing the coefficients of the Lyapunov function is:

$$C = \mathrm{diag}([1 \ \ 2 \ \ 2 \ \ 2]) \tag{152}$$

The eigenvalues of $\tilde{M}^T \cdot C + C \cdot \tilde{M}$ in this case are

$$\lambda_1 = 0, \ \lambda_2 = 0, \ \lambda_3 = -4.5, \ \lambda_4 = -2.5 \tag{153}$$

which again proves the global stability of the system.

The above example demonstrates how time-reparametrization can be used in the design of suitable globally stabilizing static feedbacks for nonlinear process systems.

A.3.2 Feedback design for a simple fermentation process

This example presents a fermentation process similar to the one presented in section 2.2.2, just the reaction kinetics (i.e. function $\mu(x_2)$) is different. This one is a monotonous function of the substrate concentration x_2, that results in a simpler nonlinearity. The system is described by the non-QP input-affine state-space model

$$\begin{aligned} \dot{x}_1 &= \mu(x_2)x_1 + \tfrac{(X_F - x_1)F}{V} \\ \dot{x}_2 &= -\tfrac{\mu(x_2)x_1}{Y} + \tfrac{(S_F - x_2)F}{V} \\ \mu(x_2) &= \mu_{max}\tfrac{x_2}{K_S + x_2}, \end{aligned} \tag{154}$$

where the inlet substrate and biomass concentrations denoted by S_F and X_F, are the manipulated inputs. The variables and parameters of the model together with their units and parameter values are given in Table 6.

The system has a unique locally stable equilibrium point in the positive orthant:

$$\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} 0.6500 \\ 0.4950 \end{bmatrix} \tag{155}$$

with steady-state inputs

$$\begin{bmatrix} \bar{X}_F \\ \bar{S}_F \end{bmatrix} = \begin{bmatrix} 0.6141 \\ 4.3543 \end{bmatrix}.$$

By introducing a new differential variable $\eta = \frac{1}{K_S + x_2}$ one arrives at a third differential equation

$$\begin{aligned} \dot{\eta} &= -\tfrac{1}{(K_S+x_2)^2} \cdot \tfrac{dx_2}{dt} = -\eta^2 \cdot \left(-\tfrac{\mu_{max}}{Y} x_1 x_2 \eta + \tfrac{(S_F - x_2)F}{V} \right) = \\ &= \eta \left(\tfrac{\mu_{max}}{Y} x_1 x_2 \eta^2 + \tfrac{F}{V} x_2 \eta - S_F \tfrac{F}{V} \eta \right) \end{aligned} \tag{156}$$

that completes the ones for x_1 and x_2. Thus the original system (154) can be represented by the following three quasi-polynomial differential equations:

$$\begin{aligned} \dot{x}_1 &= x_1 \cdot \left(-\tfrac{F}{V} + \mu_{max} x_2 \eta + \tfrac{F}{V} x_1^{-1} X_F \right) \\ \dot{x}_2 &= x_2 \cdot \left(-\tfrac{F}{V} - \tfrac{\mu_{max}}{Y} x_1 \eta + \tfrac{F}{V} x_2^{-1} S_F \right) \\ \dot{\eta} &= \eta \cdot \left(\tfrac{F}{V} x_2 \eta + \tfrac{\mu_{max}}{Y} x_1 x_2 \eta^2 - \tfrac{F}{V} \eta S_F \right). \end{aligned}$$

Using a wise choice of the feedback structure, the quasi-monomials of the closed loop system may decrease. In our case the feedback structure is chosen to be

$$\begin{aligned} X_F &= k_1 x_1 x_2 \eta + \delta_1 x_1 \\ S_F &= k_2 x_1 x_2 \eta + \delta_2 x_2. \end{aligned}$$

The closed loop QP system is then

$$\begin{aligned} \dot{x}_1 &= x_1 \cdot \left(-\tfrac{F}{V} + \left(\mu_{max} + k_1 \tfrac{F}{V} \right) x_2 \eta \right) \\ \dot{x}_2 &= x_2 \cdot \left(-\tfrac{F}{V} + \left(-\tfrac{\mu_{max}}{Y} + k_2 \tfrac{F}{V} \right) x_1 \eta \right) \\ \dot{\eta} &= \eta \cdot \left(\tfrac{F}{V} x_2 \eta + \left(\tfrac{\mu_{max}}{Y} - k_2 \tfrac{F}{V} \right) x_1 x_2 \eta^2 \right). \end{aligned}$$

Note, that for the globally stabilizing feedback design phase parameters δ_1, and δ_2 are set to zero, since they will be used for shifting the equilibrium of the closed loop system to the original fermenter's one. It is apparent that the closed loop system has only 3 quasi-monomials: $x_2\eta, x_1\eta, x_1x_2\eta^2$.
The solution of the BMI problem gives the feedback gain parameters

$$\begin{aligned} k_1 &= -1.5355 \\ k_2 &= 43.6516, \end{aligned}$$

which makes the system globally asymptotically stable (in the positive orthant) with the Lyapunov function (20) having parameters:

$$c_1 = 0.0010, \quad c_2 = 0.0761, \quad c_3 = 0.0760.$$

The equilibrium (155) of the open loop fermenter can be reset by expressing δ_1, and δ_2 from the steady-state equations. This gives $\delta_1 = 1.7152$, $\delta_2 = -20.9293$, so the equilibrium point (155) of the fermentation process (154) is globally stabilized.

A.4 Applied mathematical tools

In this section some of the mathematical tools applied throughout the thesis are described.

A.4.1 Linear and bilinear matrix inequalities

In what follows, linear- and bilinear matrix inequalities are defined as special tools applied by system- and control theory.

Linear matrix inequality

A (non-strict) linear matrix inequality (LMI) is an inequality of the form

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \leq 0, \tag{157}$$

where $x \in \mathbb{R}^m$ is the variable and $F_i \in \mathbb{R}^{n\times n}$, $i = 0, \ldots, m$ are given symmetric matrices. The inequality symbol in (157) stands for the negative semi-definiteness of $F(x)$. If the equality is not allowed, then the LMI is termed *strict*.

One of the most important properties of LMIs is the fact, that they form a convex constraint on the variables, i.e. the set $\mathcal{F} = \{x \mid F(x) \leq 0\}$ is convex and thus many different kinds of convex constraints can be expressed in this way [6], [56]. It is important to note that a particular point from the convex solution set $\mathcal{F}$ can be selected using additional criteria (e.g. different kinds of objective functions) [6]. Standard LMI optimization problems are e.g. linear function minimization, generalized eigenvalue problem, etc.

Various problems in system- and control theory can be written up as a set of linear matrix inequalities. For example, the Lyapunov equation connected to the global

stability of LTI systems. But they also appear in the context of *linear parameter-varying* (LPV) systems, or within μ-analysis there are also LMIs solved.

There are several software tools available for solving linear matrix inequalities. The most widespread ones are in the *Matlab LMI Control Toolbox* [16]. In spite of its great popularity it has problems when a non-strict LMI is to be solved. On the other hand, *Scilab* (an open source platform for numerical computation, see http://www.scilab.org) performs far much better for the non-strict case. The algorithm of [36] is also able to handle the rank deficiency of matrices F_i in (157). A good survey on the available solvers can be found in [63].

Bilinear matrix inequality

A bilinear matrix inequality (BMI) is a diagonal block composed of q matrix inequalities of the following form

$$G_0^i + \sum_{k=1}^{p} x_k G_k^i + \sum_{k=1}^{p}\sum_{j=1}^{p} x_k x_j K_{kj}^i \leq 0, \quad i = 1, \ldots, q \tag{158}$$

where $x \in \mathbb{R}^p$ is the decision variable to be determined and G_k^i, $k = 0, \ldots, p$, $i = 1, \ldots, q$ and K_{kj}^i, $k, j = 1, \ldots, p$, $i = 1, \ldots, q$ are symmetric, quadratic matrices.

The main properties of BMIs are that they are non-convex in x (which makes their solution numerically much more complicated than that of linear matrix inequalities), and their solution is NP-hard [63], so the size of the tractable problems is limited. However, there exist practically applicable and effective algorithms for BMI solution [36], [61], or [8]. In Matlab environment the TomLab/PENBMI solver [37] can be used effectively to solve bilinear matrix inequalities. Similarly to the LMIs, additional criteria can be used to select a preferred solution point of a feasible BMI from its solution set.

BMIs are mostly applied in the field of robust control, many problems can be formulated in the form (158).

A.4.2 Tensor product and its properties

The tensor product [49] of matrices A and B is denoted by $A \otimes B$, where

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{bmatrix} \in \mathbb{C}^{m \times n}$$

and $B \in \mathbb{C}^{p \times q}$. The value of $A \otimes B$ is the block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \ldots & a_{1n}B \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1}B & a_{m2}B & \ldots & a_{mn}B \end{bmatrix} \in \mathbb{C}^{mp \times nq}$$

The most important properties of tensor product are as follows.

- $A \otimes (B + C) = A \otimes B + A \otimes C$, if B and C are of the same size
- $(A + B) \otimes C = A \otimes C + B \otimes C$, if A and B are of the same size
- $(cA) \otimes B = A \otimes (cB) = c(A \otimes B)$
- $(A \otimes B) \otimes C = A \otimes (B \otimes C)$
- $(A \otimes B)^T = A^T \otimes B^T$

Additional properties of tensor product for quadratic matrices:

- if $A \in \mathbb{C}^{n \times n}$ and $B \in \mathbb{C}^{k \times k}$ with eigenvalues $\lambda_1, \ldots, \lambda_n$ and $\mu_1, \ldots, \mu_k$, respectively, then the eigenvalues of $A \otimes B$ are in the form $\lambda_i \mu_j, i = 1, \ldots, n,\ j = 1, \ldots, k$
- $\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\mathrm{Tr}(B)$
- $\det(A \otimes B) = (\det(A))^k (\det(B))^n$
- $A \otimes B$ is invertible iff A and B are both invertible, and $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$

A.5 Tables

x_1	biomass concentration		$[\frac{g}{l}]$
x_2	substrate concentration		$[\frac{g}{l}]$
S_F	substrate feed concentration		$[\frac{g}{l}]$
X_F	biomass feed concentration		$[\frac{g}{l}]$
F	inlet feed flow-rate	3.2089	$[\frac{l}{h}]$
V	volume	4.0000	$[l]$
Y	yield coefficient	0.5000	-
μ_{max},	kinetic parameter	1.0000	$[\frac{1}{h}]$
K_1	kinetic parameter	0.0300	$[\frac{g}{l}]$
K_2	kinetic parameter	0.5000	$[\frac{l}{g}]$

Table 5: Variables and parameters of the fermenter model with non-monotonous kinetics (14)

x_1	biomass concentration		$[\frac{g}{l}]$
x_2	substrate concentration		$[\frac{g}{l}]$
S_F	substrate feed concentration		$[\frac{g}{l}]$
X_F	biomass feed concentration		$[\frac{g}{l}]$
F	inlet feed flow-rate	1.0000	$[\frac{l}{h}]$
V	volume	97.8037	$[l]$
Y	yield coefficient	0.0097	-
μ_{max},	kinetic parameter	0.0010	$[\frac{1}{h}]$
K_s	kinetic parameter	0.5	$[\frac{l}{g}]$

Table 6: Variables and parameters of the fermenter model (154)

x_1	biomass concentration		$[\frac{g}{l}]$
x_2	substrate concentration		$[\frac{g}{l}]$
F	inlet feed flow-rate	2	$[\frac{l}{h}]$
V	volume	1	$[l]$
S_F	substrate feed concentration		$[\frac{g}{l}]$
Y	yield coefficient	1	-
μ_{max},	kinetic parameter	1	$[\frac{1}{h}]$

Table 7: Variables and parameters of the fermenter model (79)

A.6 Spinsim function reference

The syntax of the most important functions used in Spinsim is listed here.

density2vector.m

Converts a density matrix to a Bloch vector.
Syntax: b=density2vector(D)
Input:

D: density matrix

Output:

b: Bloch vector

vector2density.m

Converts a Bloch vector to a density matrix.
Syntax: D=vector2density(b)
Input:

b: Bloch vector

Output:

D: density matrix

measure.m

Performs a measurement on a quantum system.
Syntax: [Dt,r]=measure(D0,E1,E2)
Input:

D0: density matrix before the measurement

E1,E2: projection operators of the (von Neumann) measurement

Output:

Dt: density matrix after the measurement

r: outcome of the measurement (± 1)

spinsim.m

Function spinsim is for simulating spin $\frac{1}{2}$ quantum systems. The function plots the trajectory of the system in a Bloch sphere (see Figure 23) . Measurement is also implemented in the simulator. Syntax: [xf,Df]=spinsim(Dx0,stepsize,u,c)
Input:

Dx0: Inital state of the system. $Dx0$ can be either a 3 dimensional column vector with absolute value smaller than 1 or a density matrix.

stepsize: The stepsize of the simulation (since it simulates discrete time quantum systems)

u: A double matrix with 3 rows. Column i contains the value of the magnetic field in each direction at timestep i (see Figure 22). For example, $u(1,2)$ is the strength of the magnetic field in direction x at the second timestep. It is very important to note that the input vector also contains information about measurement. If the value of $u(i,j)$ equals to π exactly, then the spin in direction i is measured at time j. This causes jumps in the trajectory. They are marked with yellow/orange.

c: Optional parameter. If the value of c is '*manual*', then the user has to click on the figure for the next simulation step. If it is '*auto*', then the simulator is stepping automatically. If the value is '*none*' then the simulation runs in one step. The default value is '*auto*'.

Output:

xf: The value of the Bloch vector at the end of the simulation.

Df: Density matrix of the qubit at the end of the simulation.

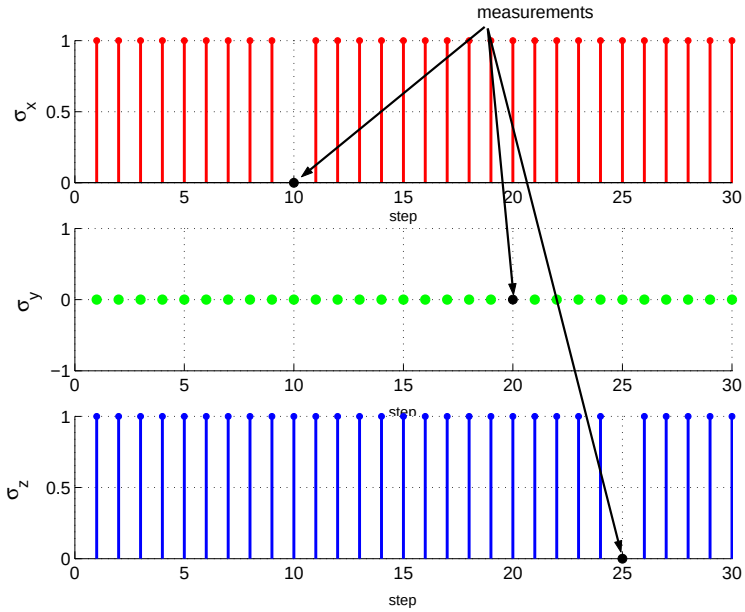

Figure 22: Input for spinsim

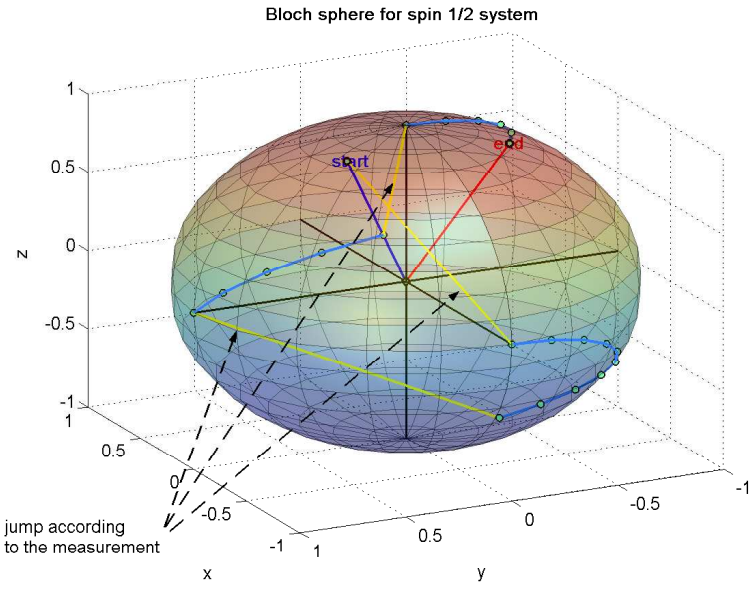

Figure 23: State trajectory of a qubit in spinsim

spinsim2.m

Function for simulating 2 coupled spin 1/2 particles. The function plots the trajectory of the system in seven Bloch spheres. Measurements are also implemented in the simulator. Syntax: [Df]=spinsim2(D0,H0,stepsize,u1,u2,c)
Input:

D0: Inital state of the system. $D0$ can only be a density matrix of size 4×4.

H0: Inner Hamiltonian function of the system. $H0$ must be a self-adjoint matrix

of size 4×4.

stepsize: The stepsize of the simulation (since it simulates discrete time quantum systems)

u1: A double matrix of size $3 \times steps$. Column i contains the value of the magnetic field applied on the first spin in each direction at timestep i. It is very important to note that the input vector also contains information about measurement. If the value of $u1(i, j)$ equals to π exactly, then the spin in direction i is measured at time j on the first spin. This causes jumps in the trajectory - they are marked with yellow.

u2: A double matrix of size $3 \times steps$. Column i contains the value of the magnetic field applied on the first spin in each direction at timestep i.

c: Optional parameter. If the value of c is '*manual*', then the user has to click on the figure for the next simulation step. If it is '*auto*', then the simulator is stepping automatically. If the value is '*none*' then the simulation runs in one step. The default value is '*auto*'.

Output:

Df: Density matrix of the qubit at the end of the simulation.

qbayes.m

Function qbayes performs a bayesian parameter estimation on a quantum system (spin $\frac{1}{2}$) having no dynamics. The function computes and plots the probability density function for the three spin component x_1, x_2 and x_3 (Figure 24). The joint (3 dimensional) probability density function is also computed and plotted , see Fihure 25.

Syntax: `est=qbayes(n,D)`

Input:

n: number of measurements to be performed in each spin direction (i.e. $3n$ measurements are performed)

D: density matrix representing the state of the spin $\frac{1}{2}$ quantum system to be estimated

Output:

est: estimated Bloch vector

qls.m

Function qls performs a least squares parameter estimation on a quantum system (spin $\frac{1}{2}$) having no dynamics. The function computes and plots the least squares point estimates for the three spin component x_1, x_2 and x_3. In the Bloch-sphere plot (Figure 26), the red vector is the original Bloch-vector and the blue is the estimated

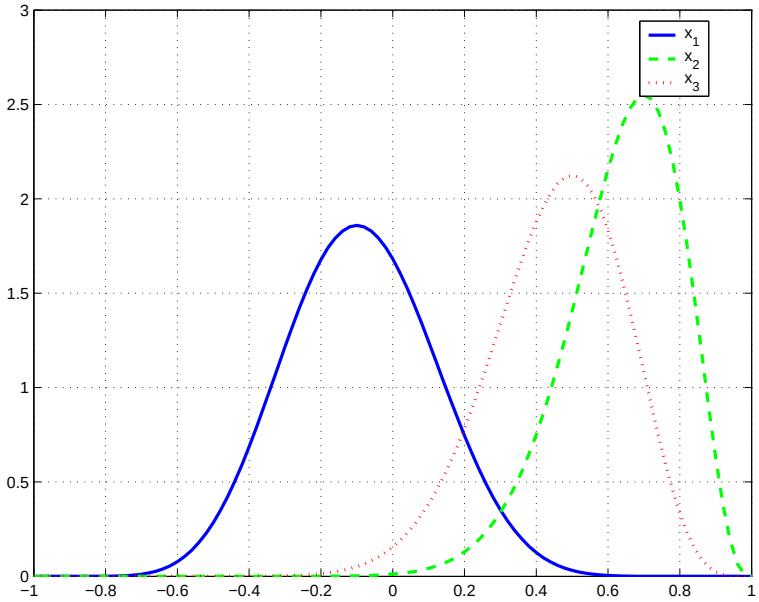

Figure 24: Probability density functions for the Bloch vector components

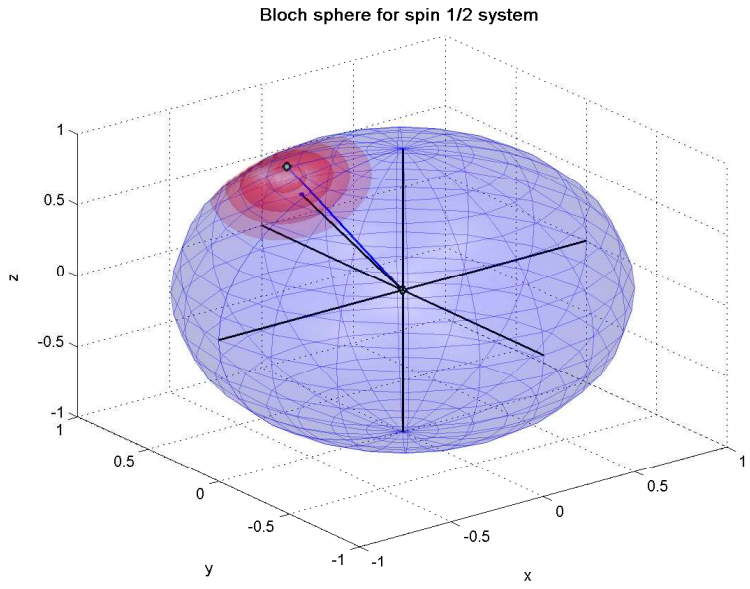

Figure 25: Joint p.d.f. for as the Bayesian estimate

one.

Syntax: est=qls(n,D)

Input:

n: number of measurements to be performed in each spin direction (i.e. $3n$ measurements are performed)

D: density matrix representing the state of the spin $\frac{1}{2}$ quantum system to be estimated

Output:

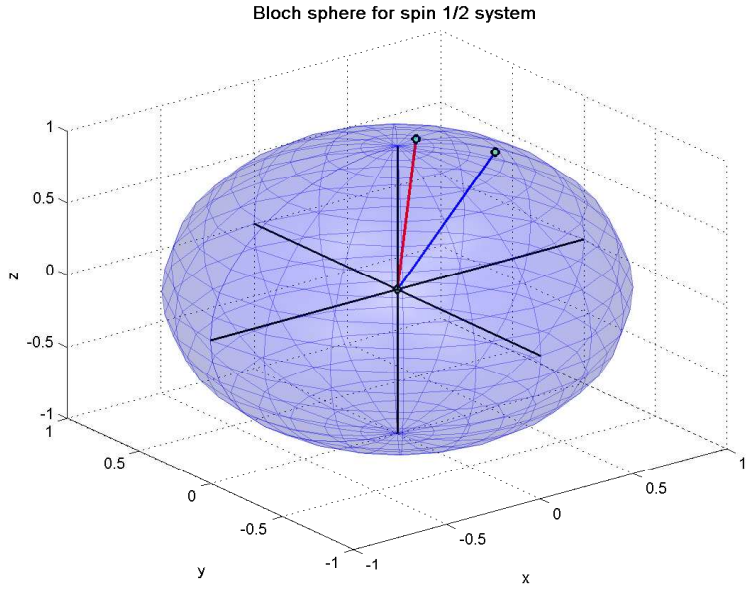

Figure 26: Point estimate of the LS estimator

est: estimated Bloch vector

Bibliography

[1] G. Alber, T. Beth, M. Horodecki, P. Horodecki, R. Horodecki, M. Rötteler, H. Weinfurter, R. Werner, and A Zeilinger. *Quantum Information*. Springer, 2001.

[2] K. J. Aström and B. Wittenmark. *Computer-controlled systems*. Prentice Hall, 1997.

[3] D. D. Bainov and P. S. Simeonov. *Systems with impulse effect: stability, theory and applications*. Ellis Horwood Limited, 1989.

[4] J. A. Bergou, U. Herzog, and M. Hillery. Discrimination of quantum states. *Lect. Notes Phys.*, 649:417–465, 2004. in Quantum State Estimation, eds. M. Paris and J. Rehácek,.

[5] D Bouwmeester, A.K. Ekert, and A. Zeilinger. *The Physics of Quantum Information: Quantum Cryptography, Quantum Teleportation, Quantum Computation*. Springer, 2007.

[6] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, Philadelphia, 1994.

[7] V. B. Braginsky, F. Y. Khalili, and K. S. Thorne. *Quantum Measurement*. Cambridge University Press, 1995.

[8] Y.-Y. Cao, J. Lam, and Y.-X. Sun. Static output feedback stabilization: An ILMI approach. *Automatica*, 12:1641–1645, 1998.

[9] F. Csáki. *Fejezetek a szabályozástechnikából - Állapotegyenletek*. Műszaki Könyvkiadó, 1973.

[10] G. M. D'Ariano, M. G. A. Paris, and M. F. Sacchi. Quantum tomography. *arXiv*, quant-ph/0302028:v1, 2003.

[11] G. M. D'Ariano, M. G. A. Paris, and M. F. Sacchi. Quantum tomographic methods. *Lect. Notes Phys.*, 649:7–58, 2004. in Quantum State Estimation, eds. M. Paris and J. Rehácek,.

[12] R. Díaz-Sierra, B. Hernández-Bermejo, and V. Fairén. Graph-theoretic description of the interplay between non-linearity and connectivity in biological systems. *Mathematical Biosciences*, 156:229–253, 1999.

[13] R. Ellis. *Entropy, large deviations, and statistical mechanics*. Springer-Verlag, 1985.

[14] A. Figueiredo, I. M. Gleria, and T. M. Rocha. Boundedness of solutions and Lyapunov functions in quasi-polynomial systems. *Physics Letters A*, 268:335–341, 2000.

[15] P. Findeisen, L. Imsland, F. Allgöver, and B.A. Foss. State and output feedback nonlinear model predictive control: An overview. *European Journal of Control*, 9:No. 2–3, 2003.

[16] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali. *LMI Control Toolbox Users Guide*. The MathWorks, Inc., 1995.

[17] I.M. Gléria, A. Figueiredo, and T.M. Rocha Filho. A numerical method for the stability analysis of quasi-polynomial vector fields. *Nonlinear Analysis*, 52:329–342, 2003.

[18] J.L. Gouzé. Global stabilization of n-dimensional population models by a positive control. In *Proceedings of the 33rd IEEE Conf. on Decision and Control, Orlando, USA*, pages 1335–1336, 1994.

[19] F. Grognard and J.L. Gouzé. Positive control of Lotka-Volterra systems. In *Proceedings of 16th IFAC World Congress, Prague, Czech Republic*, 2005. on CD.

[20] A. K. Gupta and S. Nadarajah. *Handbook of Beta distribution and its applications*. Dekker, 2004.

[21] K. Hangos, J. Bokor, and G. Szederkényi. *Computer controlled systems*. Veszprémi Egyetemi Kiadó, Veszprém, 2002.

[22] K. M. Hangos, J. Bokor, and G. Szederkényi. *Analysis and Control of Nonlinear Process Systems*. Springer-Verlag, 2004.

[23] K.M. Hangos and I.T. Cameron. *Process modelling and model analysis*. Academic Press, London, 2001.

[24] C. W. Helstrom. *Quantum decision and estimation theory*. Academic Press, New York, 1976.

[25] B. Hernández-Bermejo. Stability conditions and Lyapunov functions for quasi-polynomial systems. *Applied Mathematics Letters*, 15:25–28, 2002.

[26] B. Hernández-Bermejo and V. Fairén. Nonpolynomial vector fields under the Lotka-Volterra normal form. *Physics Letters A*, 206:31–37, 1995.

[27] B. Hernández-Bermejo and V. Fairén. Lotka-Volterra representation of general nonlinear systems. *Math. Biosci.*, 140:1–32, 1997.

[28] B. Hernández-Bermejo, V. Fairén, and L. Brenig. Algebraic recasting of nonlinear ODEs into universal formats. *J. Phys. A, Math. Gen.*, 31:2415–2430, 1998.

[29] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, 1995.

[30] R. Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41:2315–2324, 1994.

[31] T. Kailath. *Linear systems*. Prentice Hall, 1980.

[32] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering (ASME)*, 82D:35–45, 1960.

[33] E. Kaszkurewicz and A. Bhaya. *Matrix Diagonal Stability*. Birkhauser, Boston, 2000.

[34] M. Keyl and R.F. Werner. Estimating the spectrum of a density operator. *Phys. Rev. A*, 64:052311, 2001.

[35] Gen Kimura. The bloch vector for n-level systems. *arXiv:quant-ph/0301152 v2*, 2003.

[36] M. Kocvara and M. Stingl. A code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 8:317–333, 2003.

[37] M. Kocvara and M. Stingl. TOMLAB/PENBMI solver (Matlab Toolbox), 2005. PENOPT Gbr.

[38] R. L. Kosut, I. Walmsley, and H. Rabitz. Optimal experiment design for quantum state and process tomography and hamiltonian parameter estimation. *arXiv*, quant-ph/0411093:v1, 2004.

[39] R. L. Kosut, I. Walmsley, and H. Rabitz. Identification of quantum systems: Maximum likelihood and optimal experiment design for state tomography. *IFAC World Congress*, 1.1:Prague (Czech Republic), 2005.

[40] C. Kravaris and J.C. Kantor. Geometric methods for nonlinear process control: 2. controller synthesis. *Ind. & Eng. Chem. Res.*, 29:2310–2323, 1990.

[41] Ch. Kuhlmann, I. D. L. Bogle, and Z. S. Chalabi. Robust operation of fed batch fermenters. *Bioprocess Engineering*, 19:53–59, 1998.

[42] A. J. Lotka. *Elements of physical biology*. William and Wilkens, Baltimore, 1925.

[43] Y. Luo and Z. Lu. Stability analysis for lotka volterra systems based on an algorithm of real root isolation. *Journal of Computational and Applied Mathematics*, 201:367–373, 2007.

[44] W. Marquardt and M. Mönnigmann. Constructive nonlinear dynamics in process systems engineering. *Computers & Chemical Engineering*, 29:1265–1275, 2005.

[45] V. Nalbantoğlu, J. Bokor, G. Balas, and P. Gáspár. System identification with generalized orthonormal basis functions: an application to flexible structures. *Control Engineering Practice*, 11:245–259, 2003.

[46] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[47] H. Nijmeijer and A. J. van der Schaft. *Nonlinear dynamical control systems*. Springer-Verlag, New York, 1990.

[48] V. Peterka. *'Bayesian system identification' in Trends and Progress in System Identification, P. Eykhoff, Ed.* Pergamon Press, Oxford, 1981.

[49] D. Petz. *Lineáris Analízis*. Akadémiai Kiadó, 2002.

[50] A. C. Phillips. *Introduction to quantum mechanics*. Wiley, 2003.

[51] B. Pongrácz, G. Szederkényi, and K.M. Hangos. An algorithm for determining a class of invariants in quasi-polynomial systems. *Computer Physics Communications*, 175:204–211, 2006.

[52] J. R. Raol, G. Girija, and J. Singh. *Modelling and Parameter Estimation of Dynamic Systems*. IEE, 2004.

[53] J. Rehácek, B. Englert, and D. Kaszlikowski. Minimal qubit tomography. *Physical Review A*, 70:052321, 2004. in Quantum State Estimation, eds. M. Paris and J. Rehácek,.

[54] A. P. Sage and J. L. Melsa. *System Identification*. Academic Press, NY, 1971.

[55] J. J. Sakurai. *Modern quantum mechanics*. Addison-Wesley, 1994.

[56] C. Scherer and S. Weiland. *Linear Matrix Inequalities in Control*. DISC, http://www.er.ele.tue.nl/sweiland/lmi.pdf, 2000.

[57] J. Shao. *Mathematical Statistics*. Springer-Verlag New York, 1999.

[58] G. Szederkényi, N.R. Kristensen, and S.B. Joergensen. Nonlinear analysis and control of a continuous fermentation process. *Computers and Chemical Engineering*, 26:659–670, 2002.

[59] Y. Takeuchi. *Global Dynamical Properties of Lotka-Volterra Systems*. World Scientific, Singapore, 1996.

[60] Inc. The MathWorks. Matlab environment. Copyright 1984-2007 The MathWorks, Inc.
http://www.mathworks.com.

[61] H.D. Tuan, P. Apkarian, and Y. Nakashima. A new Lagrangian dual global optimization algorithm for solving bilinear matrix inequalities. *International Journal of Robust and Nonlinear Control*, 10:561–578, 2000.

[62] P. M. J. van den Hof, P. S. C. Heuberger, and J. Bokor. System identification with generalized orthonormal basis functions. *Automatica*, 31:1821–1834, 1995.

[63] J.G. VanAntwerp and R.D. Braatz. A tutorial on linear and bilinear matrix inequalities. *Journal of Process Control*, 10:363–385, 2000.

[64] M. Vidyasagar. *Nonlinear systems analysis*. Prentice Hall, 1993.

[65] V. Volterra. Variazioni flultuazioni del numero d'invididui in specie convirenti. *Men Acad Lincei*, 2:31–113, 1926.

[66] J. Řehaček, B.-G. Englert, and D. Kraszlikowski. Minimal qubit tomography. *arXiv*, quant-ph/0405084:v2, 2004.

[67] W.K. Wootters and B.D. Fields. Optimal state determination by mutually unbiased measurements. *Annals of Physics*, 191:363–381, 1989.

[68] K. Zhou, J. C. Doyle, and K. Glover. *Robust and optimal control*. Prentice Hall, New Jersey, 1996.