Continuous-time System Identification

A Bilinear Optimization Approach

W.P. Krijgsman

Master of Science Thesis

TU Delft
Delft
University of
Technology

Delft Center for Systems and Control

Continuous-time System Identification

A Bilinear Optimization Approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

W.P. Krijgsman

November 28, 2018

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Delft University of Technology
Department of
Delft Center for Systems and Control (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis entitled

Continuous-time
System Identification

by

W.P. Krijgsman

in partial fulfillment of the requirements for the degree of

Master of Science Systems and Control

Dated: November 28, 2018

Supervisor(s):

prof.dr.ir. M. Verhaegen

R. Doelman LL.B. MSc.

ir. P.J. Piscaer

Reader(s):

dr. K. Batselier

Abstract

In the noiseless case, the identification of a grey-box model can be posed as a *feasibility problem*, i.e. determining if existent - and if so - finding a parameter vector such that the parametric model equals the actual model (or its associated input-output data). Being that in this thesis we are interested in continuous-time grey-box identification, we shall be dealing with models that allow for forming a direct relationship with physical meaningful quantities. Such models include the state space representation and the matrix differential equation. In general, identifying such grey-box models turns out to be a non-convex problem. In this thesis, we initially review a framework which allows us to solve feasibility problems which have bilinear constraints. It turns out that most of the aforementioned non-convexities can be captured into a single bilinear matrix equation. However, the resulting *feasibility problem*, including the bilinear matrix equations, makes the overall search for the actual parameter vector NP-hard. In order to come up with numerical tractable algorithms, we use a heuristic known as Sequential Convex Relaxation to relax the bilinear equality constraints. This iterative scheme is flexible enough to allow for additional (in)equality constraints, possibly resembling any other physical constraints. We explore two different approaches to identify both the state space model and the matrix differential equation; one, by directly identifying the model from the given frequency response function; two, by first identifying a black-box model before performing a small scale optimization problem, transforming the black-box model such that it fits the grey-box parameterization. In addition, we present a novel method which uses the Power Spectral Density to estimate a 2nd order model. All methods are numerically validated.

Table of Contents

Preface ix

Acknowledgements xiii

1 Preliminaries 1

1-1 Introduction 1

1-2 Identifiability 1

1-3 Continuous time vs. Discrete time 2

2 Solving Bilinear Equations 3

2-1 Introduction 3

2-2 Bilinear Equations 4

2-3 SDP 6

2-4 ADMM 6

2-5 Examples 7

2-5-1 Escaping local minima 8

2-5-2 Rosenbrock Function 8

2-6 Summary 10

3 State Space via Frequency Response 13

3-1 Introduction 13

3-2 State Space Model 14

3-3 Problem formulation 15

3-4 Overview 15

3-5 Identification Method 15

3-6 Simulation 17

3-7 Comparison 19

3-8 Summary 20

4 Matrix Differential Equation via Frequency Response 21
4-1 Introduction . . . 21
4-2 Matrix Differential Equation . . . 22
4-3 Problem formulation . . . 22
4-4 Overview . . . 23
4-5 Identification Method . . . 23
4-5-1 Special Cases . . . 24
4-6 Simulations . . . 25
4-6-1 Full set of Actuators Truss Structure . . . 25
4-6-2 General Case Mass Damper . . . 28
4-7 Future Work . . . 29
4-8 Summary . . . 29

5 Estimating 2nd Order Models via Power Spectral Density 31
5-1 Introduction . . . 31
5-2 Spectral Density . . . 32
5-3 Problem formulation . . . 32
5-4 Overview . . . 32
5-5 Identification Method . . . 33
5-6 Simulations . . . 34
5-7 Comparison . . . 37
5-8 Future Work . . . 38
5-9 Summary . . . 39

6 State Space via Similarity Transformation 41
6-1 Introduction . . . 41
6-2 Similarity Transformation . . . 42
6-3 Problem Formulation . . . 42
6-4 Overview . . . 43
6-4-1 Using the Similarity Transformation . . . 43
6-4-2 Minimize the model-matching Criterion . . . 43
6-4-3 Using the Hankel Matrix of Impulse Response . . . 43
6-5 Identification Method . . . 44
6-5-1 Construction of Bilinear form . . . 44
6-5-2 Standard form . . . 45
6-6 Simulation of an E-Nose model . . . 45
6-6-1 n4sid . . . 46
6-6-2 Identifiable model . . . 47
6-6-3 Non-Identifiable model . . . 48
6-7 Comparison study . . . 49
6-7-1 No a priori knowledge . . . 50
6-7-2 A priori knowledge . . . 52
6-8 Summary . . . 54

7 Differential Matrix Equation via Similarity Transformation 55

7-1 Introduction 55
7-2 Preliminaries 56
7-3 Problem Formulation 56
7-4 Overview 56
7-5 Identification Method 57
7-5-1 General Case 58
7-5-2 Special Cases 59
7-6 Simulation 61
7-6-1 Mass Damper 61
7-7 Future Work 63
7-7-1 Structure 63
7-7-2 Linear Mass Matrix parameterization 65
7-7-3 Measurements 65
7-8 Summary 65

8 Conclusion and Discussion 67

A Mathematical background 71

A-1 Vectorization 71
A-2 Singular Value Thresholding 71

B Least Square Updates 73

B-1 From Frobenius-norm to 2-norm 73
B-2 Constrained Least Squares 74

C Other Methods 75

C-1 Alternating Least Squares 75
C-2 Gradient Based Method 76

Bibliography 79

Glossary 83

List of Acronyms 83
List of Symbols 84

Index 85

List of Figures

2-1 Iterative estimation of α, β 9

2-2 Solving the Rosenbrock function using Sequential Convex Relaxation (SCR) for different regularization values 11

2-3 Slice of the first and second convex relaxation of the Rosenbrock function ($a = 0$), per iteration showing starting point and minimum. Note that the second iterate seems to be non-convex, however this due to the logarithmic scale, chosen to emphasize change in z-direction. 12

3-1 Bode plot of the driver belt (3-17) showing the used frequency samples for Identification. 18

3-2 Error versus iteration k. 18

4-1 [1] Truss structure with eight unrestrained Degree Of Freedom (DOF)'s (one horizontal and one vertical for each of the nodes denoted by 1,2,3,4) 26

4-2 Bode plot of the 8 DOF Truss structure from Figure 4-15 27

4-3 2DOF-Mass Damper system 28

4-4 Bode plot of the 2DOF mass-damper system Figure 4-3. The stemplots represent the 10 data points used for identification. Note that although the data points seem linearly spaced they are in fact log-spaced between 0.6 and 2 rad/s. 29

4-5 Convergence plot of the mass-damper problem. On the $x - axis$ the iteration number, on the $y - axis$ the difference of $f(x)$ w.r.t. the optimal value. Every traced line shows the the convergence for a certain initialization of x where each entry in x is take from a normal distribution with zero mean and unit variance. Out of the 10 initializations, the algorithm manages to converge to the optimal point 6 times within 200 iterations, the 2 traces around 20 are still converging and eventually do get to the global optimum after 258, two traces near 31 do not converge and are stuck, for this case $\hat{m}_1, \hat{k}_1, \hat{v}_1 = 0$. Also by looking at log-scaled plot: note the super-linear convergence near the optimal point. 30

5-1 Pink-noise, input power spectrum. 34

5-2 Figural representation of magnitude of the output power spectrum $\Phi^{yy}(\omega)$. The i, j'th subplot depicts the absolute value of $\Phi^{yy}_{i,j}(\omega)$ for $0.01 \leq \omega \leq 100$. 34

5-3 Figural representation of the phase angle of the output power spectrum $\Phi^{yy}(\omega)$. The i, j'th subplot depicts the phase of $\Phi^{yy}_{i,j}(\omega)$ for $0.01 \leq \omega \leq 100$. 35

5-4 Schematic depiction of a spring-mass-damper system in two dimensions. The problem discussed in (7-17) has 3 dimensions, so one should imagine an additional set of spring and damper C_z, k_z lateral to the paper. The input u are applied to the first mode, the outputs are measured in the x, y and z direction. 36

5-5 Error versus iteration k for 5 initializations. 4 out of 5 times maximum accuracy is attained within 7 iterations. 36

6-1 Three compartment e-nose sensor model 46

6-2 Bode plot of the real system (A, B, C) and the initial model $(\mathbb{A}, \mathbb{B}, \mathbb{C})$ 47

6-3 Performance comparison of different identifications when there is no prior knowledge 51

6-4 Plot depicting the error of the initial guess vs. the error of the estimated θ measured in the 2-norm. The dotted line resembles the equality of the initial error and final error. 53

7-1 Schematic depiction of a spring-mass-damper system in two dimensions. The problem discussed in (7-17) has 3 dimensions, so one should imagine an additional set of spring and damper C_z, k_z lateral to the paper. The inputs u_1, u_2 are applied diagonally to the mass in the xy-plane, the outputs are measured in the y and z direction. 62

7-2 Bode diagram of the three-dimensional spring-mass-damper system. In blue the original system, in purple the final estimate using using the estimated M, V, K matrices. 63

7-3 Error versus iteration k for different initial conditions, all continuous lines depict initializations where the entries in x are taken from the normal distribution, the dashed line is the case where S, T are initialized as the Identity matrix. The dotted line: S, T as the zero matrix. $p^\star$ is equal to $\|\mathbb{A}\|_\star + \lambda 6n$ and is obtained when bilinear equality holds in the relaxation of (7-10). Note that the error, given convergence is still around 10^{-2}, which is rather large. Also note the trace which after 4000 iterations still manages to converge. 64

Preface

Continuous-time

The laws of physics governing the world around us are continuous of nature; subsequently the mathematical models capturing this physical reality will also be continuous of nature.

System Identification

Given a descriptive model of some process, one might be interested in the following question: If I apply input some u, what will be the output y at the other side? System Identification is however, concerned with the a different kind of problem: supposing I know the input u and output y at both sides, what is the underlying model?

Grey-Box

When using first principles to derive physical meaningful models, the result is models with structure. For example when dealing with state space models, knowing that state x_i doesn't directly influence state x_j, means that the state transition matrix contains a zero at the (j, i)'th position. Knowing that the change of some variable depends on gravity, will mean that the constant g will enter the equation at some specific position. This is what we mean by *grey-box* models; models with structure.

The identification of grey-box models is in general harder than the identification of black-box models. This is due to the fact that given an identifiable grey-box parametrization and its associated input-output data, we have to search for a unique parameter vector to describe the data, where as for a black-box model there are infinitely many such parameter vectors. The hardness of grey-box identification is also reflected in the non-linear non-convex solution methods proposed to solve them.

Bilinear

In the case of identifying linear time invariant grey box models, most of these non-convexities can be captured in a a single bilinear matrix equation. Formulating an identification problem as such has a great advantage, it means we can apply the method known Sequential Convex Relaxation to solve it. In this thesis we explore this method in a variety of system identification settings.

Goal of this thesis

The main goal we set for this thesis is the following:

> *To develop new algorithms for the purpose of identifying continuous-time systems.*

With *new* it is meant that we shall be exploring the utility of a fairly new bilinear optimization technique, SCR (2016), applied to problems in the realm system identification problems. As a direct consequence hereof, we shall be looking for new ways to fit the identification problems into the bilinear framework. Additionally, with new we mean that we will be looking into types of input/output data that have previously not been considered for identification of parametric models e.g. using the input and output spectra to estimate the parameters of a 2nd order model.

Prerequisites

The background required of the reader is a working knowledge of linear algebra and to a lesser degree calculus. Furthermore, the solution method we employ throughout this thesis, will use a (iterative) convex relaxation of a non-convex problem. To this extent, some basic knowledge of *convex optimization* could prove beneficial, [2] provides a thorough reference hereon, dealing with the theoretical, numerical and conceptual aspects of the topic.

Outline

Chapter 1 will review some of the basic concepts regarding the *Identifiability* of models and the relationship between continuous and discrete time systems. This chapter is intended to review some of the more general concepts regarding system identification. In the following chapters, theory regarding the specific problem at hand will be treated in separate sections.

In Chapter 2 we will describe the general framework used to pose the identification problems in. It turns out that the identification of a system - given noiseless data - is a *feasibility problem*. However, the concerned feasibility problems we will be dealing with are non-convex due to a bilinear matrix equation appearing in the constraints. In this chapter we review a method, known as SCR to solve such kind of problems. We will give two implementations to solve the general problem and we will give some illustrative examples.

Chapter 3 will asses the possibility of identifying a state space model given the Frequency Response Function (FRF). The method we employ, relies on the fact that we can rewrite the transfer function of the associated system as a set of two bilinear equations. Using this transformation we have the possibility to apply SCR. We shall see in the numerical example that we are able to identify a system using a very limited set of (non)-equidistant frequency data points. Furthermore we conduct a comparison with Prediction error methods, which are also capable of estimating parametric models using the FRF.

We will see that the theory developed in Chapter 3 easily extends to Matrix Differential Equation (MDE)'s, useful in structural identification which we cover in Chapter 4.

In Chapter 5 we again look at 2nd order MDE's introduced in Chapter 4. However now we explore the problem of identifying the given system using the input and output spectra only. Such an identification scheme could prove useful when it is not possible to directly measure the input, but the statistical properties are known, e.g. white noise. In order to do so, it will be assumed that we have a full set of sensors (which is not uncommon in literature for identifying the same system using other input-output data). We conduct a comparison analysis between SCR and a non-linear, non-convex solver to solve the identification problem. We will see that the flexibility of SCR (allowing us to impose additional Linear Matrix Inequality (LMI) constraints) makes all the difference in successfully identifying the system or not.

Chapter 6 again takes a closer look at the identification of state space models. An algorithm is introduced which initially identifies a black-box model and then performs a small scale optimization procedure to extract the grey-box model. The concept of this two step procedure is not new - as a benefit: we can conduct a comparison study between our proposed algorithm and others.

In chapter 7 we look at MDE's for the third and final time. The chapter forms a broadening of chapter 6 and investigates the possibility to identify a 2nd order MDE from an existing black-box state space representation - once more by virtue of finding a similarity transformation matrix which relates the black box model to the grey-box parametrization.

Finally in Chapter 8, we summarize and conclude.

Acknowledgements

This document - all 85 pages of it - forms the apotheosis of 7-years of academic education and concludes the master *Systems and Control* at the TU Delft. If all goes well, on the 29'th of November I will be receiving the 'ir.' title. An achievement I will be truly proud of, not because of the title in itself, but for the effort which I have put in the past few years. The Latin way of putting this: *"Luctor et Emergo"*. It is evident, that this would not have been possible without the help of some important people along the way; obviously all the professors and tutors during my education at the TU Delft. Regarding the past year, I would like to give special thanks to:

First and foremost, Professor Michel Verhaegen, for giving me the opportunity to conduct my thesis under his supervision, for all of his input and new ideas. In our biweekly meetings he came up with new paths to explore, in my opinion Chapter 5 is a direct consequence thereof.

Secondly, my daily supervisors: Pieter Piscaer and Reinier Doelman. Reinier wasn't initially one of the supervisors, but after I came to him with a short question about a rank constraint, he soon was. He showed me the ropes in optimization, and explained me the workings - and the way to implement an algorithm he developed, an algorithm I have made extensive use of. He was always willing to answer any question I had on the subject. In return, I hope I have showed him the wide applicability of his developed algorithm. Further, I would like to thank him for proof reading and correcting all kinds of grammar, capitalization and other kinds of mistakes from my behalf.

Furthermore, I would like to thank my co-student Stijn Bosma. I especially remember the countless days and coffees which we spend in the computer room Zebra III at Industrieel Ontwerpen, a room which we soon hijacked and rebranded to the "The Simulation Room". This because at several times, Stijn, also working on his thesis had five computers running simultaneously, trying to optimize his Artificial Intelligence code. And me, doing a big validation study, having the same number of computers crunching at the numbers.

Off coarse, I would like to thank my mum, for supporting me along the way. Finally, I dedicate this thesis to the memory of my father, whom as a 5 year old toddler armed with a "*Encyclopedie voor de jeugd*", I barraged with questions.

Delft, University of Technology
November 28, 2018

W.P. Krijgsman

“Alle betere ideeën in de wereld kennen het rijpingsproces van een overjarige brokkelkaas.“

— *Dimitri Verhulst*

Chapter 1

Preliminaries

1-1 Introduction

This brief chapter is intended to touch upon the very general topics regarding grey-box identification; what does it mean to say that a system is identifiable, and what are differences between continuous time and discrete time identification? The chapters hereafter, cover specific theory regarding the topic at hand in separate sections.

outline

This chapter is organized as follows: §1-2 handles the concept of identifiability. §1-3 treats continuous-time versus discrete-time models.

1-2 Identifiability

Much of the literature on physical modeling has concentrated on the question of identifiability of the parameters, before even estimating them. This relates to the more fundamental question of whether the parameters are structurally identifiable. That is, whether the parameters of a noise-free model can be determined uniquely from the noise-free data, [3, 4, 5, 6, 7, 8, 9]. Distinctions have been made between global identifiability, structural identifiability and local identifiability.

Global Identifiability We say that a model is identifiable at $\theta^\star$ if the mapping $\theta^\star \to \mathcal{M}(\theta^\star)$ is one-to-one, i.e.

Definition 1.1. *[10] A model structure $\mathcal{M}$ is globally identifiable at θ^* if*

$$\mathcal{M}(\theta) = \mathcal{M}(\theta^\star) \Rightarrow \theta = \theta^\star \tag{1-1}$$

Local Identifiability A system that is locally identifiable may have a set of parameters that cannot uniquely be estimated from the data. However the number of solutions is finite [8].

Non-Identifiability For non-identifiable parameters, the number of solutions θ is infinite.

For the rest of this text we mostly handle globally identifiable models, if this is not the case, this is explicitly stated. For a state space model: a necessary condition is that there are less than $n(p+m)$ free parameters [8]. As an extreme case, if all system matrices are fully parameterized then they are unidentifiable. However less than $n(p+m)$ does not necessarily result in the identifiability of the system model. Identifiability also depends on the model structure. In [4] it is shown that state space model described by the following parameterized system matrices is unidentifiable even though the number of parameters is strictly less than $n(p+m)$

$$A(\theta) = \begin{bmatrix} -\theta_1 & \theta_3 & 0 \\ \theta_1 & -(\theta_2+\theta_3) & \theta_4 \\ 0 & \theta_3 & -\theta_4 \end{bmatrix}, \qquad B(\theta) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad C(\theta) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{1-2}$$

It has been shown in [4] that the following system structures are identifiable: diagonal, companion form, cascaded compartment structure with only system input and output taking place at the last compartment. Any identifiable system model under some similarity transformation is also identifiable.

1-3 Continuous time vs. Discrete time

Consider a parameterized continuous-time linear state-space model as follows

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) \\ y(t) &= C(\theta)x(kT) \end{aligned}, \tag{1-3}$$

where $u(t) \in \mathbb{R}^m, x(t) \in \mathbb{R}^n$ and $y(kT) \in \mathbb{R}^p$ are system input, state and output respectively; $\theta \in \mathbb{R}^l$ is the parameter vector; t and k represent continuous and discrete time respectively; T denotes the sampling period. Denote the discrete time State-Space matrices as the triplet $(A_d(\theta), B_d(\theta), C_d(\theta))$. Then the discrete time system, obtained by the sampling period T with the system input being piecewise constant between the sampling instants kT, will be related to the continuous time matrices $(A(\theta), B(\theta), C(\theta))$

$$A_d(\theta) = e^{A(\theta)T} \quad B_d(\theta) = \int_{\tau=0}^{T} e^{A(\theta)\tau} d\tau B(\theta) \quad C_d(\theta) = C(\theta) \tag{1-4}$$

The above equations make it clear that discrete time matrices are a non-linear function of their continuous-time counterparts. Thus making identification methods developed for discrete time systems not directly applicable to continuous time ones. So we need to come up with a work-around. Options include, using continuous time data to directly estimate continuous time matrices. We do this in Chapters 3 and 4 using the Frequency Response Function (FRF). An other option, given the discrete-time matrices is to use the inverse mapping of (1-3). However, the resulting matrices will still be black box. This means we need to find a mapping between the continuous-time black-box and grey-box model. We attempt this in Chapters 6 and 7.

Chapter 2

Solving Bilinear Equations

2-1 Introduction

As the title of this thesis suggests; we are interested in identifying parametric models. And although the nature of the problems in the following chapters are different, the way we solve them is uniform; namely Sequential Convex Relaxation (SCR). SCR provides a technique for solving optimization problems which are convex in their objective function but have bilinear constraints, making the overall optimization non-convex. The method relaxes the bilinear equation into the objective function, by first casting the bilinear equation as a rank constraint, then by using a heuristic for the rank constraint having the tendency to provide low rank solutions. The method was first treated in [11], and this chapter is mostly intended as a synopsis thereof. SCR has successfully been applied in phase retrieval [12] and in robust static output feedback [13]. We will focus attention how SCR is applicable in a system identification setting.

outline

The rest of the chapter is organized as follows. §2-2 introduces the general framework used to pose the identification problems in and reviews SCR. §2-3 introduces a Semidefinite Programming (SDP) method to solve the intermediate iterates stemming from SCR. §2-4 gives an Alternating Direction Method of Multipliers (ADMM) implementation to solve these iterates, efficient when there are only linear equalities besides the bilinear equality. §2-5 treats two illustrative examples of the SCR-technique applied to non-convex problems (not in identification setting). We see that the method is able to escape local minima and give a visual interpretation. Finally in §2-6 we summarize.

2-2 Bilinear Equations

In the most general way, the way we pose the identification problems is

$$\begin{aligned} \text{find} \quad x \in \mathbb{R}^n \quad & \text{such that} \\ \mathcal{A}(x)P\mathcal{B}(x) &= \mathcal{C}(x) \\ \mathcal{D}(x) &= 0 \\ \mathcal{E}(x) &\succeq 0 \end{aligned} \tag{2-1}$$

where $P \in \mathbb{R}^{k\times l}$ is a nonzero matrix. $\mathcal{A}(x) : \mathbb{R}^n \to \mathbb{R}^{j\times k}, \mathcal{B}(x) : \mathbb{R}^n \to \mathbb{R}^{l\times m}, \mathcal{C}(x) : \mathbb{R}^n \to \mathbb{R}^{j\times m}, \mathcal{D}(x) : \mathbb{R}^n \to \mathbb{R}^d$ are affine mappings of the parameter x, i.e.

$$\mathcal{A}(x) = A_0 + x_1A_1 + \cdots + x_nA_n \tag{2-2}$$

similarly for $\mathcal{B}(x), \mathcal{C}(x), \mathcal{D}(x)$. On the other hand $\mathcal{E}(x) : \mathbb{R}^n \to \mathbb{S}^e$,

$$\mathcal{E}(x) = E_0 + x_1E_1 + \cdots + x_nE_n \tag{2-3}$$

where $E_i \in \mathbb{S}^e$, linearly maps x to a symmetric matrix. The statement $\mathcal{A}(x)P\mathcal{B}(x) = \mathcal{C}(x)$ contains bilinear equations. $\mathcal{D}(x) = 0$ is a set of underdetermined linear equations, it handles explicit linear constraints on the decision variable. The inequality $\mathcal{E}(x) \succeq 0$ is called an Linear Matrix Inequality (LMI), and means that $\mathcal{E}(x)$ is positive semidefinite, i.e. $z^T\mathcal{E}(x)z \geq 0 \quad \forall z \in \mathbb{R}^e$, and is a convex constraint [14].

In general a bilinear matrix equality is NP-hard. To see this note that the bilinear constraint $x_1x_1 = x_1$ is equivalent to the constraint $x_1 \in \{0, 1\}$, hence any 0-1 integer program (in which all variables have to be either 0 or 1) can be formulated as bilinear constraint. Since 0-1 integer programming is hard in general, solving problems with bilinear constraints are also NP-hard. In the optimization setting, a problem as (2-1) is referred to as a *feasibility problem* and can be written as

$$\begin{aligned} \text{find } x = \min \quad & 0 \\ \text{s.t.} \quad & \mathcal{A}(x)P\mathcal{B}(x) = \mathcal{C}(x) \\ & \mathcal{D}(x) = 0 \\ & \mathcal{E}(x) \succeq 0 \end{aligned} \tag{2-4}$$

The feasibility problem is thus to determine whether the constraints are consistent and if so, find a point that satisfies them. There are two problems with the first constraint in (2-4). First of all there is the bilinearly appearing decision variable x due the product $\mathcal{A}(x)P\mathcal{B}(x)$. The other problem is the equality constraint, which cannot just be relaxed; otherwise the solution to a relaxed problem is not a solution to the original problem. However it can be shown that the bilinear constraint is equivalent to a rank constraint[11].

Lemma 1. *Rank equivalence [11] The constraint $APB = C$ is equivalent to the rank constraint*

$$\text{Rank}(M(A, P, B, C, Q, R)) = \text{Rank}(P) \tag{2-5}$$

where $M(\cdot)$ is defined as

$$M(A, P, B, C, Q, R) \equiv \begin{bmatrix} C + APR + QPB + QPR & (A + Q)P \\ P(B + R) & P \end{bmatrix} \tag{2-6}$$

for any matrices Q, R of appropriate size.

Lemma 1 lets us rewrite the bilinear equality $\mathcal{A}(x)P\mathcal{B}(x) = \mathcal{C}(x)$, into a rank constraint. In this rank constraint, the variable x no longer appears bilinearly, but affinely. However our new optimization problem subject to some rank constraint is still a non-convex problem. A well known heuristic when minimizing the rank of a matrix is to use the nuclear norm [15]. The relaxation of (2-4) leads to

$$\begin{aligned} \min_{x} \quad & \|M(\mathcal{A}(x), P, \mathcal{B}(x), \mathcal{C}(x), Q, R)\|_\star \\ \text{s.t.} \quad & \mathcal{D}(x) = 0 \\ & \mathcal{E}(x) \succeq 0 \end{aligned} \tag{2-7}$$

where $\|\cdot\|_\star$ denotes the nuclear norm. The above minimization might not produce a desired solution right away, i.e. (2-5) might not hold. However the freedom to choose the values of Q, R, lead the authors in [11] to propose the following iterative scheme

Algorithm 2.1 *Sequential Convex Relaxation*

```
given some initial matrices $Q \in \mathbb{R}^{j\times k}, R \in \mathbb{R}^{l\times m}$
repeat
    1. Update $x$ using
       $\arg\min_x \left\| \begin{bmatrix} \mathcal{C}(x) + \mathcal{A}(x)PR + QP\mathcal{B}(x) + QPR & (\mathcal{A}(x)+Q)P \\ P(\mathcal{B}(x)+R) & P \end{bmatrix} \right\|_\star$
       s.t. $\mathcal{D}(x) = 0$
            $\mathcal{E}(x) \succeq 0$
    2. Update $Q := -\mathcal{A}(x)$ and $R := -\mathcal{B}(x)$
until maximum number of iterations reached
```

It is noted that the use of the nuclear norm as a means to minimize the rank of the matrix $M(\cdot)$ in (2-7) is merely a choice. Other possible choices of heuristic include the log-det heuristic [16], and the truncated-nuclear norm [17].

Complex Case

In this section it was assumed for explanatory reasons that the decision variable x and all other matrices are real valued. In later chapters we shall come across complex valued feasibility problems. By noting that any complex matrix $Z = X + jY$ where $X, Y \in \mathbb{R}^{n\times m}$ and $j = \sqrt{-1}$, has a real representation[18],

$$\begin{bmatrix} X & -Y \\ Y & X \end{bmatrix} \in \mathbb{R}^{2n\times 2m}, \tag{2-8}$$

the above handled theory is equally applicable.

2-3 SDP

This section treats the optimization of (2-7) via means of SDP. For sake of brevity denote $M(x) : \mathbb{R}^n \to R^{p\times q}$, $M(x) = M(\mathcal{A}(x), P, \mathcal{B}(x), \mathcal{C}(x), Q, R)$. It is well known that the nuclear norm approximation problem (2-7) can be cast into the SDP-problem[15],

$$\begin{aligned} \min \quad & \frac{1}{2}(Y+Z) \\ \text{s.t.} \quad & \begin{bmatrix} Y & M^T(x) \\ M(x) & Z \end{bmatrix} \succeq 0 \\ & \mathcal{D}(x) = 0 \\ & \mathcal{E}(x) \succeq 0 \end{aligned} \tag{2-9}$$

with decision variables $x \in \mathbb{R}^n, Y \in \mathbb{S}^p, Z \in \mathbb{S}^q$. (2-9) can be solved by interior point methods available in general purpose software for SDP's [19, 20, 21]. Solving nuclear norm through SDP can be computationally intensive. If we for simplicity assume that the dimensions p and q grow as fast as the number of variables n i.e. $q = O(n)$, $p = O(n)$ then the complexity per iteration grow as least as fast as $O(n^6)$ [22]. In [22] the authors address this issue and manage to bring down the complexity to $O(n^4)$. Concerning the problems solved throughout this thesis; this implementation was not used. Advantages to SDP's are that they require only a small amount of iterations 10-15 to reach the optimal point which only grows slightly with problem size. SDP's also have quadratic convergence near the optimal point making them very precise for little extra cost [2, 14]. Furthermore by reverting to an SDP solution it is straightforward to implement additional (generalized)-inequality constraints in the optimization problem.

2-4 ADMM

In the case that there are no (generalized)-inequalities in (2-7), i.e. $\mathcal{E}(x)$ is not present, there exists an efficient Alternating Direction Method of Multipliers (ADMM) implementation to solve the given optimization problem. First note that the minimization problem (2-7) can be reformulated as

$$\begin{aligned} \min_{x,Y} \quad & \|Y\|_\star \\ \text{s.t.} \quad & Y = M(\mathcal{A}(x), P, \mathcal{B}(x), \mathcal{C}(x), Q, R) \\ & \mathcal{D}(x) = 0 \end{aligned} \tag{2-10}$$

Applying the ADMM optimization technique [23] to the constrained optimization problem (2-10), we obtain the steps outlined in Algorithm 2.2

Algorithm 2.2 *ADMM implementation for SCR*

```
given matrices $Q, R$
initialize $Y := M(Q, R, P, Q, R)$ and $\Lambda := 0$
```

```
repeat
    1. Update $x$ using
       $x := \arg\min_{x} \left\| Y - M(\mathcal{A}(x), P, \mathcal{B}(x), \mathcal{C}(x), Q, R) + \frac{1}{\rho}\Lambda \right\|_F^2$
       s.t. $\mathcal{D}(x) = 0$
    2. Update $Y$
       $Y := \arg\min_{Y} \|Y\|_\star + \frac{\rho}{2} \left\| Y - M(\mathcal{A}(x), P, \mathcal{B}(x), \mathcal{C}(x), Q, R) + \frac{1}{\rho}\Lambda \right\|_F^2$
    3. Update the dual variable
       $\Lambda := \Lambda - \rho\Big(Y - M(\mathcal{A}(x), P, \mathcal{B}(x), \mathcal{C}(x), Q, R)\Big)$
until stopping criterion is satisfied
    $\|x^{i+1} - x^i\| / \|x^i\| \leq \epsilon$
```

The main advantage of using this ADMM formulation in the identification problems is that both updates (1) and (2) have analytic solutions. In (1), the matrix valued function $M(\cdot)$ is parameterized affinely in x, hence it can always be recast as a Least Squares (LS)-problem (constrained or unconstrained depending on the existence of $\mathcal{D}(x)$). Furthermore, (2) is of the form

$$\arg\min_{Y} \|Y\|_\star + \lambda \|Y - C\|_F^2 \tag{2-11}$$

and has the analytic optimal solution, commonly referred to as 'singular value threshholding' [24],

$$\hat{Y} = U_C \mathcal{T}(\Sigma) V_C^T \tag{2-12}$$

where $\mathcal{T}(\Sigma) = \max\{\sigma_i - \frac{1}{2\lambda}, 0\}$ and U_C and V_C denote the left and right singular vectors of C respectively. A proof is discussed in Appendix A-2.

Note that an ADMM implementation is equally applicable given that we do have generalized inequalities resembled in $\mathcal{E}(x) \succeq 0$. The difference being that step 2. in Algorithm 2.2 would have this LMI, $\mathcal{E}(x) \succeq 0$ as an additional constraint. However in such an instance the ADMM approach would no longer be efficient as this exact step is an SDP, requiring an interior point method to solve.

2-5 Examples

In this section we give two examples of how SCR can be be applied to solving non-convex functions. In the first example we show that SCR has the ability to escape local minima, in the second example we review SCR applied to the Rosenbrock function, which is an industry standard for testing optimization algorithms. It is shown that the Rosenbrock function can be written as an optimization problem with a convex objective function subject to bilinear constraints. Furthermore we will use this example to give an illustrative interpretation of convex relaxation.

2-5-1 Escaping local minima

The authors in [25] show that their identification method using a Difference of Convex Programming (DCP) technique has the ability to climb out of local minima. We show that the SCR technique, which we employ has the same ability. For the purpose of illustrating this, we use the same example as in [25]. Suppose we have to deal with the following optimization problem

$$\begin{aligned} \min_{\alpha,\beta} \quad & \alpha^2 + 0.1\beta^2 \\ \text{s.t} \quad & \alpha + 1 = (\beta - 1)^2 \end{aligned} \tag{2-13}$$

where $\alpha, \beta \in \mathbb{R}$ are scalar variables. It can be found that the objective function above has only one local minimum at $\alpha = 0, \beta = 0$, which is also the global minimum of (2-13). However, the constrained optimization problem has two local minima, as can be seen from Figure 2-1. By substituting the constraint $\alpha = (\beta - 1)^2 - 1$ into the objective function, we have that

$$\min_{\beta} [(\beta - 1)^2 - 1]^2 + 0.1\beta^2 \tag{2-14}$$

The above objective function is 4th order polynomial which has local minima at $\beta = 0$ and $\beta = 1.947$. The optimization problems (2-13) and (2-14) are equivalent. However minimizing (2-14) may end up in $\beta = 0.1947$ while minimizing (2-13) by relaxing the quadratic constraint leads to the minimum. Using Lemma 1 we can write the quadratically constrained problem (2-13) into the rank constrained problem

$$\begin{aligned} \min_{\alpha,\beta} \quad & \alpha^2 + 0.1\beta^2 \\ \text{s.t.} \quad & \text{Rank} \begin{bmatrix} (\alpha + 1) - 2(\beta - 1)x + x^2 & (\beta - 1) - x \\ (\beta - 1) - x & 1 \end{bmatrix} = 1 \end{aligned} \tag{2-15}$$

for any $x \in \mathbb{R}$. Using the SCR method with a regularization on the nuclear norm relaxation of the rank constraint of $\lambda = 0.1$, and using initial estimates $\alpha = 1$, $\beta = 2$, i.e. the initial x becomes $x = \beta - 1 = 1$, we iteratively solve the problem. Note that the initial point is quite close to a local minimum such that gradient-type optimization schemes such as Gauss-Newton method will get stuck in the local minimum. The SCR method however is able to escape the local minimum by relaxing the associated rank constraint. It should however be noted that this does not imply that SCR is able to find the global minimum of any given function.

2-5-2 Rosenbrock Function

In mathematical optimization, the Rosenbrock function is often used as a performance test problem for optimization algorithms. Because of its shape it is often referred to as the *banana function.* The function is defined by

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2, \tag{2-16}$$

and is non-convex. The optimum lies within a a long, narrow parabolic shaped flat valley, and has a global minimum at $(1, 1)$ where $f(x, y) = 0$. Finding the valley is trivial, however to converge to the global minimum is hard. We will see that SCR can be used to minimize the Rosenbrock function.

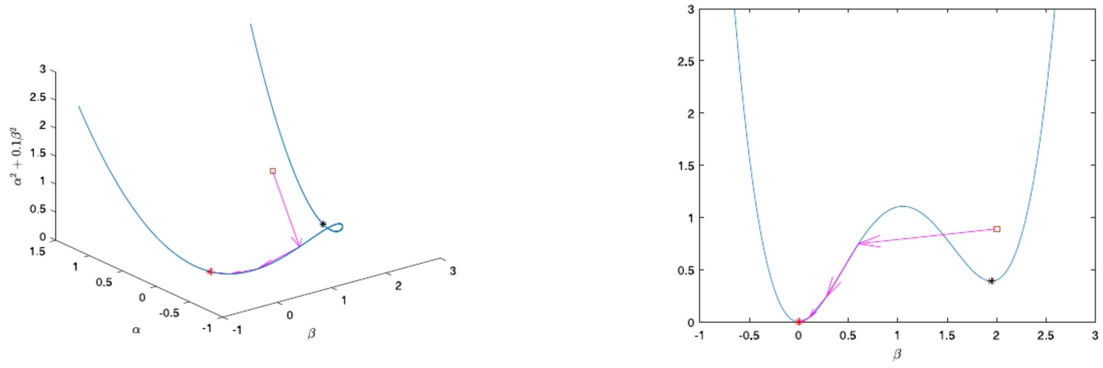


Figure 2-1: Iterative estimation of α, β

First of all note that $f(x, y)$ is a fourth order polynomial. Using the substitution $a = y - x^2$ we can write the minimization as a quadratic objective function subject to a quadratic constraint, i.e.

$$\begin{aligned} \min_{x,y,a} \quad & 100a^2 + (1-x)^2 \\ \text{s.t.} \quad & x^2 = y - a \end{aligned} \tag{2-17}$$

which is equivalent to the rank constrained function

$$\begin{aligned} \min_{x,y,a} \quad & 100a^2 + (1-x)^2 \\ \text{s.t.} \quad & \text{Rank} \begin{bmatrix} y - a + 2xr + r^2 & x + r \\ x + r & 1 \end{bmatrix} = 1 \end{aligned} \tag{2-18}$$

for any $r \in \mathbb{R}$. For ease of reference we give the procedure to solve the above function using the theory outlined in §2-2.

Algorithm 2.3 *Rosenbrock minimization with SCR*

```
given initial $r \in \mathbb{R}$, regularization $\lambda \in \mathbb{R}_{++}$
repeat
    1. Update $x, y, a$ using
       $\arg\min_{x,y,a} \; 100a^2 + (1-x)^2 + \lambda \left\| \begin{bmatrix} y - a + 2xr + r^2 & x + r \\ x + r & 1 \end{bmatrix} \right\|_\star$
    2. Update $r := -x$
until stopping criterion is satisfied
```

We will analyze the effects of the regularization parameter λ, by solving the Rosenbrock function for $\lambda \in \{0.1, 1, 10\}$, as a staring point we choose $(-1.9, 2)$ which lies at the other side of the valley. The results are depicted in Figure 2-2. We see that for a high regularization term the iterates follow a path similar to a path generated by a gradient based algorithm.

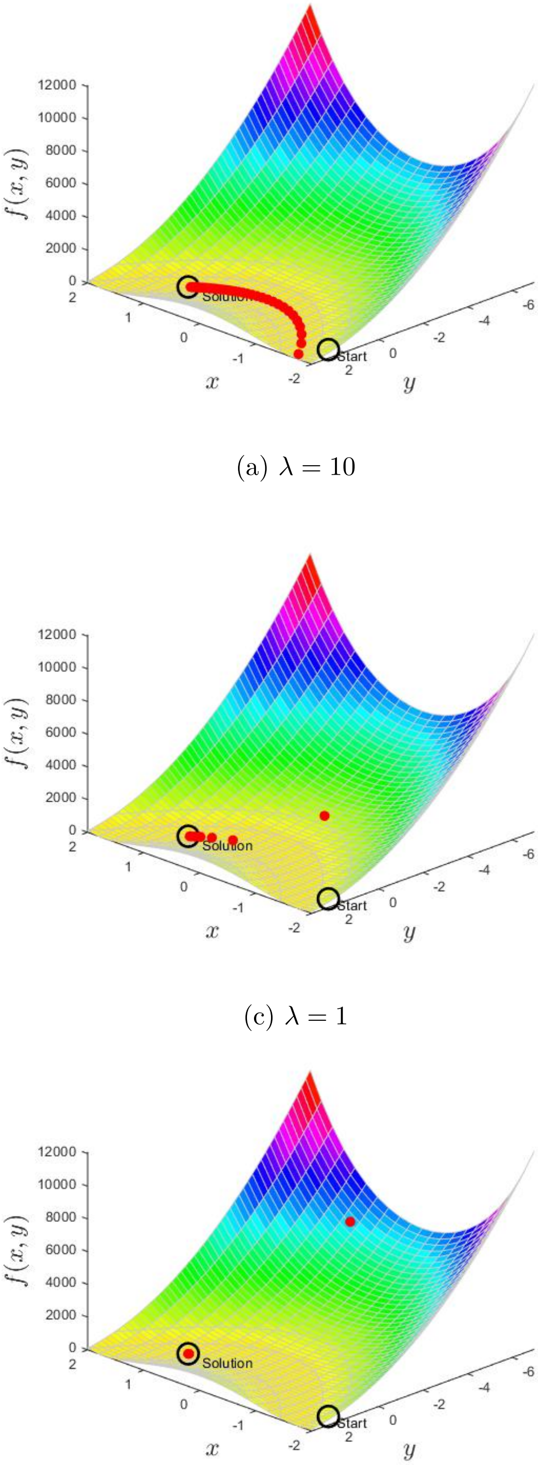
This not a entirely surprise as a high regularization term λ means that we would very much like to enforce the constraint $a = y - x^2$. Enforcing this constraint means we are solving the original problem. On the other hand when we have a low regularization term $\lambda = 0.1$, the first step ends up in $(x^1, y^1) = (0.9000, -6.0300)$ having objective value of $f(x, y) = 5405$. This might not seem as an improvement from our initial starting point, but one should remember we are no longer solving the minimization of $f(x, y)$ but the convex relaxation hereof. Denote $g_i(x, y, a)$ as the i'th relaxation of $f(x, y)$, of which the first two are depicted in Figure 2-3. From this figure it is clear that the point $(0.9000, -6.0300)$ does in fact minimize the relaxed objective function. The second iterate has the minimum in $(0.999, 0.998)$, which is very close to the global optimum $(1, 1)$.

2-6 Summary

In this chapter we have reviewed Sequential Convex Relaxation [11], which provides a technique for solving optimization problems which are convex in the objective function and have bilinear constraints. These bilinear constraints make the overall optimization problem non-convex. Using the fact that a bilinear matrix equality can be written as a rank constraint, and using the nuclear norm's property of producing low-rank solutions, the non-convex problem can be turned into a convex one. However the nuclear norm doesn't always provide optimal solutions, by using an iterative technique we can improve upon the results.

Furthermore, we have reviewed the concept of a *feasibility problem.* A feasibility problem is an optimization problem where the only objective is to check if the constraints are consistent, and if they are to find a parameter vector satisfying these constraints. All of the identification problems treated in the next chapters will be written as such a problem.

Finally we have treated two examples which illustrate the flexibility of SCR. In the first example we showed that the method has the capability of escaping local minima, in the second we gave an intuitive interpretation of the concept of *convex relaxation.*

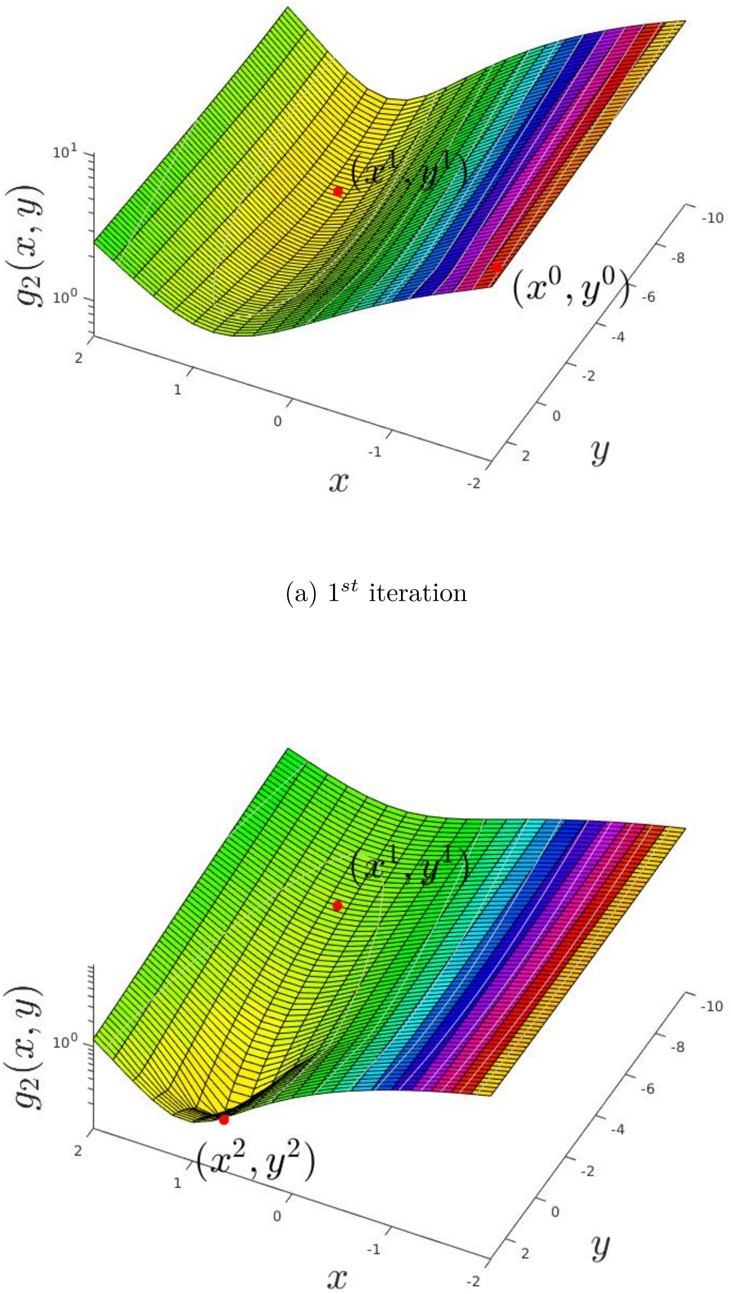


(a) $\lambda = 10$

(c) $\lambda = 1$

(b) $\lambda = 0.1$

Figure 2-2: Solving the Rosenbrock function using SCR for different regularization values



(a) 1^{st} iteration

(b) 2^{nd} iteration

Figure 2-3: Slice of the first and second convex relaxation of the Rosenbrock function ($a = 0$), per iteration showing starting point and minimum. Note that the second iterate seems to be non-convex, however this due to the logarithmic scale, chosen to emphasize change in z-direction.

Chapter 3

State Space via Frequency Response

3-1 Introduction

In this chapter we give a new method to identify continuous-time structured state space model given the Frequency Response Function (FRF). A great feature of frequency domain data is that it can handle continuous-time measurements and thus can directly be used for the identification of continuous-time systems. The FRF can be obtained for any Linear Time-Invariant (LTI) system and special instruments such as spectrum analyzers can be used for this purpose. Another option is to obtain the FRF from the discrete Input Output (IO)-data. To do so, the reader is referred to [26], which dedicates an entire chapter to the estimation of FRF-data given finite length data sequences. Also [27] contains an interesting discussion on continuous-time system identification using frequency data.

Assuming that the parameters are distributed among the (A, B, C) matrices (See §3-2), the transfer function non-linearly depends on the parameters. Furthermore, retrieving the parameters directly from the FRF, turns out to be a highly non-convex problem. The method we propose uses a certain substitution for the inverse term in the transfer function so that we can describe the transfer function as two distinct bilinear equations. Having the transfer function in bilinear equality form, we can apply Sequential Convex Relaxation (SCR) to solve the optimization problem. However, rewriting the transfer function as two bilinear equations comes at the expense of introducing new variables in the optimization problem. In the numerical example in §3-6 and §3-7, we asses if this leads to an easier optimization problem compared to the direct estimation of the parameter from the FRF with Prediction Error Method (PEM).

Outline

This chapter is organized as follows: §3-2 reviews the theory regarding the state space model and its associated transfer function. In §3-3 we introduce the problem formulation. §3-4 treats the literature. In §3-5 we develop an algorithm to estimate a parameterized State Space (SS)-model given the FRF. In §3-6 the method is validated on a numerical example and in §3-7 is compared to the prediction errror method. Finally in §3-8 we conclude.

3-2 State Space Model

The state space model for an LTI model consists of a set of n coupled first-order Ordinary Differential Equation (ODE), defined in the matrix A and B, and in a set of output equations expressed in terms of the matrices C and D

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{3-1}$$

where $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{p \times n}, D \in \mathbb{R}^{p \times m}$, and where u is the input of the system, y the output and x the state. The number of components of the state vector x is called the order of the model. The matrices A and B are properties of the system and are determined by the system structure and elements. The matrices C and D are determined by the particular choice of output variables. In cases there is no direct feedthrough from the input $u(t)$ to the output $y(t)$, D is the zero matrix.

Consider (3-1), with no direct feedthrough, as we will do throughout the rest of the thesis. Applying the Fourier transform with zero initial condition leads to

$$\begin{aligned} j\omega X(\omega) &= AX(\omega) + BU(\omega) \\ Y(\omega) &= CX(\omega) \end{aligned} \tag{3-2}$$

As we are interested in the ratio of $Y(\omega), U(\omega)$, we solve for the state equations for $X(\omega)$.

$$\begin{aligned} j\omega X(\omega) - AX(\omega) &= BU(\omega) \\ (j\omega I - A)X(\omega) &= BU(\omega) \\ X(\omega) &= (Ij\omega - A)^{-1}BU(\omega) \end{aligned} \tag{3-3}$$

We can now put the expression for $X(\omega)$ in the output equation

$$Y(\omega) = C(Ij\omega - A)^{-1}BU(\omega) \tag{3-4}$$

and can solve for the transfer function

$$H(\omega) = \frac{Y(\omega)}{U(\omega)} = C(Ij\omega - A)^{-1}B \tag{3-5}$$

The transfer function of a given system is unique, note however that there are many State Space realizations that will result in the same transfer function. Clearly $H(\omega)$ in (3-5) must have dimensionality $p \times q$, so for every input there are p transfer functions, one for each output.

3-3 Problem formulation

Consider a parameterized continuous-time linear state space model as follows

$$\begin{aligned} \dot{x}(t) &= A(\theta)x(t) + B(\theta)u(t) \\ y(t) &= C(\theta)x(t) \end{aligned} \tag{3-6}$$

where $u(t) \in \mathbb{R}^m, x(t) \in \mathbb{R}^n, y(t) \in \mathbb{R}^p$; $\theta \in \mathbb{R}^l$. We assume that the structured system matrices are affine with respect to θ, i.e. they have the same structure as given in Eq. (**??**).

problem of interest Given a set of measured frequency function matrices

$$H(\omega_i, \theta) = C(\theta)(Ij\omega_i - A(\theta))^{-1}B(\theta) \qquad \text{for} \qquad i = 1 \dots s \tag{3-7}$$

that are generated from Eq. (3-6) for a certain value θ^*, the concerned grey-box identification problem is to estimate the parameter vector θ^* from the measurements.

To address the concerned identification problem, the following assumptions are made

1. The system model in Eq. (3-6) is minimal and stable, and the system order is known.
2. The system model in Eq. (3-6) is identifiable
3. The generated FRF for the system model Eq. (3-6) are noise-free.

3-4 Overview

According to [27] the prediction error approaches such as PEM implement variants of the minimization problem

$$\hat{\theta}_N = \arg\min_{\theta} \sum_{i=1}^{N} |Y(\omega_i) - H(\omega_i, \theta)U(\omega_i)|^2 \tag{3-8}$$

where $Y(\omega)$ and $U(\omega)$ are defined as in (3-5). The above optimization problem forms a natural least squares estimate, but could also be weighted to represent varying reliability and relevance of the measurements at different frequencies [27]. A whole different class of identification schemes to estimate parameters in continuous-time state space representations, consist of first estimating a black-box state space model before solving a small scale optimization problem to transform the black-box model so that it fits the grey-box representation. Given that we develop our own such method in Chapter 6, the reader is referred to §6-4 for literature regarding such methods. In this chapter we will be focusing on the identification of continuous-time state space systems directly from the frequency data.

3-5 Identification Method

Consider a single measurement in Eq. (3-7),by introducing the notation

$$\Gamma_i = (Ij\omega_i - A(\theta))^{-1}B(\theta) \tag{3-9}$$

where $\Gamma_i \in \mathbb{C}^{n\times m}$. For sake of clarity denote $H_i = H(\omega_i)$. Using (3-9) it is possible to write a frequency measurement (3-7) as the pair of bilinear equations

$$j\omega\Gamma_i - B(\theta) = A(\theta)\Gamma_i \tag{3-10a}$$
$$H_i = C(\theta)\Gamma_i \tag{3-10b}$$

In a similar fashion we can apply the substitution,

$$\Psi_i = C(\theta)(Ij\omega_i - A(\theta))^{-1} \tag{3-11}$$

letting us write the transfer function (3-5) as the two bilinear equations

$$j\omega_i\Psi_i - C(\theta) = \Psi_i A(\theta) \tag{3-12a}$$
$$H_i = \Psi_i B(\theta) \tag{3-12b}$$

Equations (3-9) and (3-12) are equivalent in the following sense,

$$\text{given } \theta, \Gamma_i \text{ such that } \begin{pmatrix} j\omega\Gamma_i - B(\theta) & = A(\theta)\Gamma_i \\ H_i & = C(\theta)\Gamma_i \end{pmatrix} \text{ then } H_i = C(\theta)(Ij\omega_i - A(\theta))^{-1}B(\theta) \tag{3-13}$$

$$\text{given } \theta, \Psi_i \text{ such that } \begin{pmatrix} j\omega_i\Psi_i - C(\theta) & = \Psi_i A(\theta) \\ H_i & = \Psi_i B(\theta) \end{pmatrix} \text{ then } H_i = C(\theta)(Ij\omega_i - A(\theta))^{-1}B(\theta) \tag{3-14}$$

Meaning that we only need to find the doublet $(\theta, \Gamma_1, \dots, \Gamma_s)$ *or* the doublet $(\theta, \Psi_1, \dots, \Psi_s)$ in order to meet the problem statement given in §3-3. However for reasons discussed in the **note** at the end of this section we opt to use both equations (3-10,3-12) simultaneously in the feasibility problem, resulting in

$$\begin{array}{ll} \text{find} & \theta \in \mathbb{R}^l, \Gamma_i \in \mathbb{C}^{n\times m}, \Psi_i \in \mathbb{C}^{p\times n} \quad \text{for} \quad i = 1, \dots, s \\ \text{subject to} & \\ & \left.\begin{array}{rl} j\omega_i\Gamma_i - B(\theta) & = A(\theta)\Gamma_i \\ H_i & = C(\theta)\Gamma_i \\ j\omega_i\Psi_i - C(\theta) & = \Psi_i A(\theta) \\ H_i & = \Psi_i B(\theta) \end{array}\right\} \quad \text{for} \quad i = 1, \dots, s \end{array} \tag{3-15}$$

which contains $4 \times s$ bilinear equations in the constraint set. Using the theory outlined in §2-2 it is possible to lift each bilinear constraint into the objective function, assigning to each individual relaxation a weighting parameter, possibly reflecting confidence in the measured FRF at the given frequency, see Algorithm 3.1. Furthermore as (3-15) contains no inequalities, it is efficient to use the Alternating Direction Method of Multipliers (ADMM) implementation as discussed in §2-4 to solve the relaxed problem. If the weighting parameters are to be chosen uniform, we can also write the feasibility problem as

$$\begin{aligned}
&\text{find} \quad && \theta \in \mathbb{R}^l, \Gamma_i \in \mathbb{C}^{n\times m}, \Psi_i \in \mathbb{C}^{p\times n} \quad \text{for} \quad i = 1, \ldots, s \\
&\text{subject to} && \\
& && \begin{bmatrix} j\omega_1\Gamma_1 - B(\theta) & \ldots & j\omega_s\Gamma_s - B(\theta) \\ H_1 & \ldots & H_s \end{bmatrix} = \begin{bmatrix} A(\theta) \\ C(\theta) \end{bmatrix} \begin{bmatrix} \Gamma_1 & \ldots & \Gamma_s \end{bmatrix} \\
& && \begin{bmatrix} j\omega_1\Psi_i - C(\theta) & H_1 \\ \vdots & \vdots \\ j\omega_s\Psi_s - C(\theta) & H_s \end{bmatrix} = \begin{bmatrix} \Psi_1 \\ \vdots \\ \Psi_s \end{bmatrix} \begin{bmatrix} A(\theta) & B(\theta) \end{bmatrix}
\end{aligned} \tag{3-16}$$

The feasibility problem (3-16) contains 2 bilinear matrix equations compared to the $4 \times s$ equations in Eq. (3-15). Although the bilinear equations are now substantially larger compared to (3-15), (3-16) can still be be more efficiently solved by the methods discussed in Chapter 2.

Note The simultaneous use of (3-10) and (3-12) in the feasibility problem (3-15) might seem redundant, especially considering it leads to a doubling of the number of bilinear constraints in (3-15) or (3-16). However it should be noted that during the bilinear optimization procedure (SCR) all of the constraints in (3-15) are relaxed. Using both constraint pairs, we have that the matrices $B(\theta), C(\theta)$ are simultaneously sitting in equations which relate them directly to the transfer function $H(\omega)$ *and* to the $A(\theta)$ matrix. It has been found that this is beneficial w.r.t. being able to converge to the global optimum, and makes it less necessary to weight individual relaxations.

3-6 Simulation

The following example is extracted from [28], which aims to identify the physical parameters governing a printer belt drive system. The associated system matrices are parameterized as follows:

$$A(\theta) = \begin{bmatrix} 0 & -1 & 0.15 \\ 0.2 & 0 & 0 \\ \theta_1 & \theta_2 & \theta_3 \end{bmatrix} \quad B(\theta) = \begin{bmatrix} 0 \\ 0 \\ \theta_4 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}. \tag{3-17}$$

The true parameter vector is set to be

$$\{\theta_1, \theta_2, \theta_3, \theta_4\} = \{-1, -0.5, -0.2, -0.15\}. \tag{3-18}$$

The bode plot of the system is depicted in Figure 3-1 - from which we see that the system has a frequency spike at roughly 0.6 Hz. Because these spikes contain information, it is chosen to use this spike for the identification of the system. We sample the spike in the frequency range $0.4 - 0.6$ Hz in a logspaced manner. The resulting 10 data points - also visible in Figure 3-1 - are used for the identification method as given in Algorithm 3.1. Concerning the initialization of the matrices: all matrices are initialized as zero matrices.

In Figure 3-2 a convergence plot is given showing the difference of the objective function with respect to the optimal value, per iteration. It not only shows the value of the overall

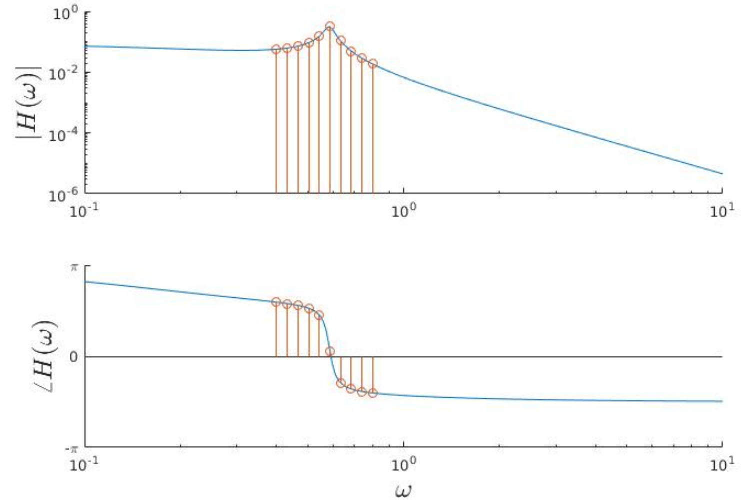


Figure 3-1: Bode plot of the driver belt (3-17) showing the used frequency samples for Identification.

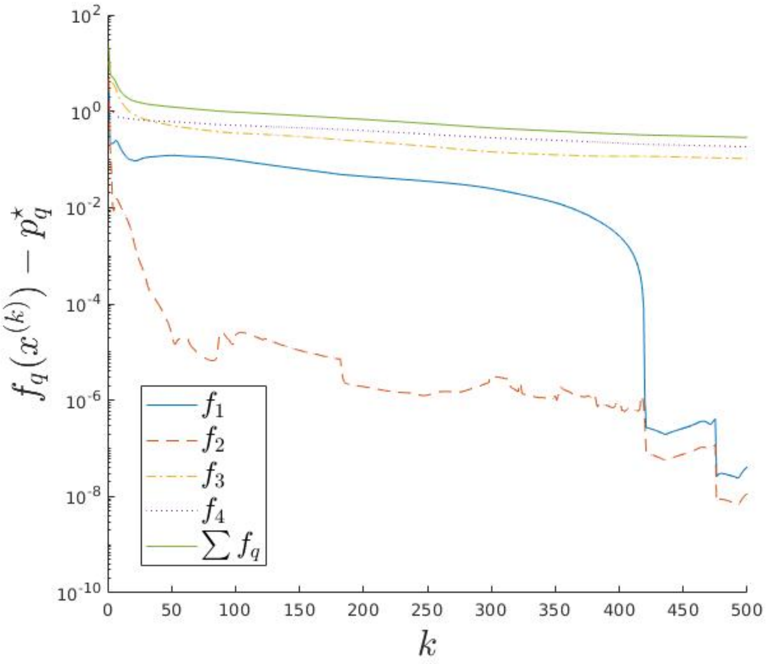


Figure 3-2: Error versus iteration k.

objective function, but also four separate functions $f_q(x)$ which together form the overall objective function. Each separate $f_q(x)$ is a measure for the equality violation of the q'th constraint in (3-15), it is the summation of the relaxations of the q'th constraint in (3-15), e.g.

$$f_1(x) = \sum_{i=1}^{s} \left\| \begin{bmatrix} \Gamma_i j\omega_i - B(\theta) + A(\theta)\mathbb{G}_i + \mathbb{A}\Gamma_i + \mathbb{A}\mathbb{G}_i & A(\theta) + \mathbb{A} \\ \Gamma_i + \mathbb{G}_i & I \end{bmatrix} \right\|_\star , \tag{3-19}$$

where x contains all decision variables. Similarly for $f_2(x), f_3(x), f_4(x)$ - also see Algorithm 3.1. By looking at the convergence plot in (3-2), we see that the total objective error has not yet been reduced to zero by the 500'th iteration. On the other hand, we do see that $f_1(x), f_2(x)$ have converged to practically zero around the 425'th iteration, meaning that the first two constraints in (3-15) hold for all i. By (3-13), we then conclude we should have solved the problem. And this is indeed the case, the estimated parameter vector equals:

$$\hat{\theta} = \begin{bmatrix} -1.000 & -0.500 & -0.200 & -0.15 \end{bmatrix}^T , \tag{3-20}$$

with an estimation error equalling:

$$\|\theta - \hat{\theta}\|_2 = 3.4987 \cdot 10^{-7}. \tag{3-21}$$

Thus we conclude we were able to successfully identify the system using only the 10 frequency data points as shown in Figure 3-1.

3-7 Comparison

In the overview we saw that an other method which is able to directly estimate grey-box models using FRF-data is PEM. In this section we will be assessing if PEM has the ability to extract the unknown parameter vector in (3-17). To do so we make use of the commands provided by the System Identification Toolbox [29] in Matlab. Initially, the system of (3-17) is put into a identifiable grey-box object with `idgrey`. Using the FRF-data in an `iddata`-object we are then ready to estimate the model using the `greyest` command. It turns out that PEM finds it a hard problem to solve. At first, the initialization of the algorithm θ_0 is set to be zeros. This leads to a fit of the estimation data of $-8,5\%$ and an estimated parameter vector $\hat{\theta} = \begin{bmatrix} 0.0010 & 0.0005 & 0.1575 & -0.0013 \end{bmatrix}^T$. Hereafter we try a series of random initialization of θ_0, using a Gaussian distribution. We try this 10 times, and then pick the solution which leads to the highest estimation data fit. In this case, this leads to a fit of 68.5% and an associated estimated parameter vector $\hat{\theta} = \begin{bmatrix} -0.8846 & 1.6125 & 0.2406 & -0.2042 \end{bmatrix}^T$. Even when we pick the initial estimate relatively close to the parameter vector, e.g. picking $\theta_0 = \begin{bmatrix} -0.5 & -0.5 & -0.2 & -0.1 \end{bmatrix}^T$, PEM is not able to converge to the global optimum. We presume this to be a consequence of the highly non-convex optimization problem which PEM tries to solve. The conclusion is that our proposed algorithm outperforms PEM for this problem instance.

3-8 Summary

In this chapter we have given a novel method to identify a parameterized state space model using the Frequency Response Function. By introducing two substitutions for certain parts of the transfer function we were able to write the transfer function as a set of bilinear equations. Finding the parameter vector, then turned out to be solving a feasibility problem subject to some bilinear constraints. Utilizing the fact that such a problem can be solved using SCR we were able to come up with an algorithm. The algorithm has been tested on physical example. We saw that the proposed algorithm was able to extract the unknown parameter vector, whereas the Prediction Error Method failed.

Algorithm 3.1 *Estimate parameters in State Space model given FRF-data*

given frequency samples $H(\omega_i) \in \mathbb{C}^{p \times m}$ for $i = 1, \ldots, s$

initialize $\mathbb{A}, \mathbb{B}, \mathbb{C}$ and $\mathbb{G}_i, \mathbb{P}_i$ for $i = 1, \ldots, s$

do

1. *Update* the decision variables $\theta, \Gamma_1, \ldots, \Gamma_s, \Psi_1, \ldots, \Psi_s$, using

$$\begin{aligned}\min_{\theta,\Gamma,\Psi} \sum_{i=1}^{s} \Bigg(\lambda_i & \left\| \begin{bmatrix} \Gamma_i j\omega_i - B(\theta) + A(\theta)\mathbb{G}_i + \mathbb{A}\Gamma_i + \mathbb{A}\mathbb{G}_i & A(\theta) + \mathbb{A} \\ \Gamma_i + \mathbb{G}_i & I \end{bmatrix} \right\|_\star + \ldots \\ \gamma_i & \left\| \begin{bmatrix} H_i + C(\theta)\mathbb{G}_i + \mathbb{C}\Gamma_i + \mathbb{C}\mathbb{G}_i & C(\theta) + \mathbb{C} \\ \Gamma_i + \mathbb{G}_i & I \end{bmatrix} \right\|_\star + \ldots \\ \delta_i & \left\| \begin{bmatrix} \Psi j\omega_i - C(\theta) + \mathbb{P}_i A(\theta) + \Psi_i\mathbb{A} + \mathbb{P}_i\mathbb{A} & \Psi_i + \mathbb{P}_i \\ A(\theta) + \mathbb{A} & I \end{bmatrix} \right\|_\star + \ldots \\ \rho_i & \left\| \begin{bmatrix} H_i + \mathbb{P}_i B(\theta) + \Psi_i\mathbb{B} + \mathbb{P}_i\mathbb{B} & \Psi_i + \mathbb{P}_i \\ B(\theta) + \mathbb{B} & I \end{bmatrix} \right\|_\star \Bigg),\end{aligned}$$

2. *Update*

$$\begin{aligned} &\mathbb{A} := -A(\theta) \quad \mathbb{B} := -B(\theta) \quad \mathbb{C} := -C(\theta) \\ &\mathbb{G}_i := -\Gamma_i, \quad \mathbb{P}_i := -\Psi_i \quad \text{for } i = 1, \ldots, s \end{aligned}$$

until stopping criterion satisfied

Chapter 4

Matrix Differential Equation via Frequency Response

4-1 Introduction

In the preceding chapter we have developed a technique for estimating parameters in structured state-space models using frequency domain data. In this chapter we will extend this theory and apply it to Matrix Differential Equation (MDE)'s. MDE's form a way of representing a differential equation. The second-order variant of the MDE is an often studied object in the field of structural identification. In this case the matrices represent physical quantities such as the mass, damping and stiffness of the given system. In addition, due to first order principals the associated matrices are positive definite, this make them an interesting object of study.

Outline

This chapter is organized as follows: §4-2 treats the MDE and the 2nd order instance we will be focusing on this chapter. §4-3 formulates the problem statement. §4-4 treats the concerning literature. In §4-5 a new identification algorithm is proposed to deal with the problem statement, and treats two special cases. §4-6 the proposed method is applied to a physical example. §4-7 gives some possibilities for future work. Finally in §4-8 we conclude.

4-2 Matrix Differential Equation

The state space representation of a system replaces a q'th order differential equation with q first-order Ordinary Differential Equation (ODE)'s. A different representation is to leave q'th order differential equations intact and collect them into a MDE

$$\begin{aligned} A_q x^{(q)}(t) + A_{q-1}x^{(q-1)}(t) + \cdots + A_0 x(t) &= Bu(t) \\ y(t) &= C_q x^{(q)}(t) + C_{q-1}x^{(q-1)}(t) + \cdots + C_0 x(t) \end{aligned} \tag{4-1}$$

where $x(t) \in \mathbb{R}^n, u(t) \in \mathbb{R}^m, y(t) \in \mathbb{R}^p$.

Given that we have 2nd order differential equation and only measure displacement, (4-1) reads as,

$$\begin{aligned} &M\ddot{x}(t) + V\dot{x}(t) + Kx(t) = Bf(t) \\ &y(t) = Cx(t) \end{aligned} \tag{4-2}$$

where $B \in \mathbb{R}^{n\times m}, C \in \mathbb{R}^{p\times n}$ and due to the principal of reciprocity $M, V, K \in \mathbb{S}^n_{++}$. Commonly M is referred to as the mass matrix, V to the damping matrix and K the stiffness matrix. For acceleration measurements the output equation in (4-2) becomes $y(t) = C\ddot{x}(t)$

The dynamical equations of motion are related to the state space representation in the following way[30],

$$\begin{aligned} \begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}V \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -M^{-1}B \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \end{aligned} \tag{4-3}$$

We remark that when parameterizing physical systems, the dynamical equations lend themselves far better to construct affine parameterizations w.r.t. the parameter vector θ, because in (4-3), the matrix M is inverted.

4-3 Problem formulation

Consider a parameterized continuous-time linear state space model as follows,

$$\begin{aligned} &M(\theta)\ddot{x}(t) + V(\theta)\dot{x}(t) + K(\theta)x(t) = Bu(t) \\ &y(t) = Cx(t) \end{aligned} \tag{4-4}$$

where $u(t) \in \mathbb{R}^m, x(t) \in \mathbb{R}^n, y(t) \in \mathbb{R}^p$; $\theta \in \mathbb{R}^l$. We assume that the structured system matrices are affine with respect to θ. We will assume that $M(\theta), V(\theta), K(\theta)$ are positive definite matrices due to reciprocity, and are fully or partially unknown. Furthermore we will assume that B, C are known a priori which is common in literature.

problem of interest Given a set of measured frequency function matrices

$$H(\omega_i, \theta) = C(-\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta))^{-1}B \qquad \text{for} \qquad i = 1 \ldots s \tag{4-5}$$

that are generated from Eq. (4-4) for a certain value θ^*, the concerned grey-box identification problem is to estimate the parameter vector θ^* from the measurements.

To address the concerned identification problem, the following assumptions are made

1. The system model in Eq. (3-6) is minimal and stable, and the system order is known.
2. The system model in Eq. (3-6) is identifiable
3. The generated Frequency Response Function (FRF) for the system model Eq. (3-6) are noise-free.

4-4 Overview

Models fitting the description (4-2) are often used in Finite Element Modeling, Structural Analysis and Modal Analysis. These fields largely use black-box descriptions of the matrices M, V, K and are mostly interested in finding the resonant frequencies and the associated mode shapes [31, 32, 33].

A Maximum Likelihood algorithm is used in [34] for Bayesian finite element updating. They test the method on the parameterized model of the form (4-4).

4-5 Identification Method

Consider a single measurement in Eq. (4-5), by introducing the substitution

$$\Gamma(\omega_i, \theta) = (-\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta))^{-1} B \tag{4-6}$$

We can write a single measurement as

$$\begin{aligned} B &= \begin{bmatrix} M(\theta) & V(\theta) & K(\theta) \end{bmatrix} \begin{bmatrix} -\omega_i^2 \Gamma(\omega) \\ -j\omega_i \Gamma(\omega_i) \\ \Gamma(\omega_i) \end{bmatrix} \\ H(\omega_i) &= C\Gamma(\omega_i) \end{aligned} \tag{4-7}$$

Where $M(\theta), V(\theta), C, K(\theta)$ are (partially-)parameterized matrices, $\Gamma(\omega)$ is fully parameterized, and $G(\omega), \omega, C, B$ are fully known to us. Now supposing we have s frequency samples we can write the problem directly into *optimization form*

$$
\begin{aligned}
&\text{find} \quad \theta, \Gamma \\
&\text{subject to} \\
&B = \overbrace{\begin{bmatrix} M(\theta) & V(\theta) & K(\theta) \end{bmatrix}}^{\mathcal{A}(x)} \overbrace{\begin{bmatrix} -\omega_i^2 I \\ -j\omega_i I \\ I \end{bmatrix}}^{P_i} \overbrace{\Gamma(\omega_i)}^{\mathcal{B}_i(x)} \qquad \text{for} \quad i = 1, \dots, s \\
&\underbrace{\begin{bmatrix} H(\omega_2) & \dots & H(\omega_s) \end{bmatrix} - C \begin{bmatrix} \Gamma(\omega_1) & \dots & \Gamma(\omega_s) \end{bmatrix}}_{\mathcal{D}(x)} = 0 \\
&M(\theta) \succeq 0 \\
&V(\theta) \succeq 0 \\
&K(\theta) \succeq 0
\end{aligned}
\tag{4-8}
$$

Because we have s bilinear equations in (4-8) it is needed to relax every bilinear equation individually into the objective function. This leads to the minimization over the summation of s nuclear norms. Due to the individual relaxations, it is possible to assign a weight λ_i to every term, possibly reflecting the confidence in frequency measurement $H(\omega_i)$. Furthermore (4-8) contains generalized matrix inequalities, hence the Semidefinite Programming (SDP) formulation §2-3 can be applied to solve the optimization problem.

Velocity and/or acceleration measurements

It should be noted that in the above we have considered position measurements only, this is for exposition reasons. In the case we have velocity measurements, or accelerations measurements, the generic description is

$$C = -\omega^2 C_a + j\omega C_v + C_p \tag{4-9}$$

where C_p, C_v, C_a denote the output matrices of the position, velocity and acceleration respectively.

4-5-1 Special Cases

We identify two cases where the the solution can be obtained in a linear manner. The first is the case when we have as many sensors as there as second order modes. The other being that the number of second order modes equals the number of actuators. We briefly review both instances here.

Full set of Sensors In case the matrix C is known a priori and is of full rank the bilinear problem reduces to a linear problem. Because C is invertable we can solve the following,

$$\arg\min_{\theta \in \mathbb{R}} \sum_{i=1}^{s} \left\| (-\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta)) C^{-1} H(\omega_i) - B \right\|_F^2 \tag{4-10}$$

which can be written as a standard Least Squares (LS).

Full set of Actuators In the event that we have as many actuators as there are second order modes we have that B is square and invertible. We can then write

$$\arg\min_{\theta\in\mathbb{R}} \sum_{i=1}^{s} \left\| H(\omega_i)B^{-1}(-\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta)) - C \right\|_F^2 \tag{4-11}$$

The above problem is a summation of Frobenius norms linear in the argument θ, thus equivalent to a linear LS.

Note that the above two LS are only stricly valid in the noiseless case, when equality holds. In the case of noise; and defining the modeling miss-match error as

$$\epsilon = H(\omega) - C(-\omega^2 M(\theta) + \omega V(\theta) + K(\theta))B, \tag{4-12}$$

we are in effect minimizing

$$(-\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta))C^{-1}\epsilon \tag{4-13}$$

and

$$\epsilon B^{-1}(-\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta)) \tag{4-14}$$

in (4-11) and (4-10) respectively.

4-6 Simulations

In this section we give two numeric examples validating the methodology. We will consider the special case where B is full rank as well as the general case. We omit a validation of C being full rank, because the case that B is full rank, is almost alike, (4-10) (4-11).

4-6-1 Full set of Actuators Truss Structure

In the first numerical example we consider the two-dimensional truss structure as depicted in Figure 4-1. We use it to validate the LS-formulation of (4-11), possible when B is full rank. The example has previously been studied in [1]. The system has a total number of eight nodes of which four are fully restrained, and hence the total number of active Degree Of Freedom (DOF)'s is 8 (one vertical and one horizontal per each node). The mass, damping and stiffness matrices are presented in (4-15). The second-order matrices contain the coefficients only for the unrestrained DOF's and are ordered such that the displacement vector can be written as $x(t) = \left[u_1(t)v_1(t)\dots u_4(t)v_4(t)\right]$, where $u_i(t), v_i(t)$ are the horizontal and vertical DOF of node i respectively.

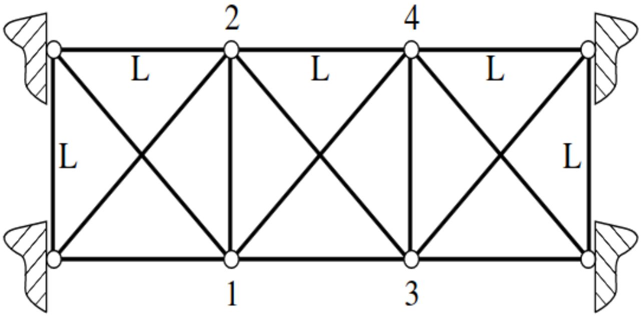


Figure 4-1: [1] Truss structure with eight unrestrained DOF's (one horizontal and one vertical for each of the nodes denoted by 1,2,3,4)

$$
\begin{aligned}
M &= \begin{bmatrix}
100.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 100.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 100.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 100.0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 100.0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 100.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 100.0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 100.0
\end{bmatrix} \\
V &= \begin{bmatrix}
136.0 & 0 & 0 & 0 & -50.0 & 0 & -17.7 & -17.7 \\
0 & 86.4 & 0 & -50.0 & 0 & 0 & -17.7 & -17.7 \\
0 & 0 & 136.0 & 0 & -17.7 & 17.7 & -50.0 & 0 \\
0 & -50.0 & 0 & 86.4 & 17.7 & -17.7 & 0 & 0 \\
-50.0 & 0 & -17.7 & 17.7 & 136.0 & 0 & 0 & -50.0 \\
0 & 0 & 17.7 & -17.7 & 0 & 86.4 & 0 & 0 \\
-17.7 & -17.7 & -50.0 & 0 & 0 & 0 & 136.0 & 0 \\
-17.7 & -17.7 & 0 & 0 & -50.0 & 0 & 0 & 86.4
\end{bmatrix} \\
K &= \begin{bmatrix}
27071.0 & 0 & 0 & 0 & -10000.0 & 0 & -3535.5 & -3535.5 \\
0 & 17071.0 & 0 & -10000.0 & 0 & 0 & -3535.5 & -3535.5 \\
0 & 0 & 27071.0 & 0 & -3535.5 & 3535.5 & -10000.0 & 0 \\
0 & -10000.0 & 0 & 17071.0 & 3535.5 & -3535.5 & 0 & 0 \\
-10000.0 & 0 & -3535.5 & 3535.5 & 27071.0 & 0 & 0 & 0 \\
0 & 0 & 3535.5 & -3535.5 & 0 & 17071.0 & 0 & -10000.0 \\
-3535.5 & -3535.5 & -10000.0 & 0 & 0 & 0 & 27071.0 & 0 \\
-3535.5 & -3535.5 & 0 & 0 & 0 & -10000.0 & 0 & 17071.0
\end{bmatrix}
\end{aligned}
\tag{4-15}
$$

We will assume no prior knowledge of the internal couplings of the modes and we will use full symmetric parameterizations of $M(\theta), V(\theta), K(\theta)$, i.e.

$$
M = \begin{bmatrix}
\theta_1 & \theta_9 & \theta_{16} & \theta_{22} & \theta_{27} & \theta_{31} & \theta_{34} & \theta_{36} \\
\theta_9 & \theta_2 & \theta_{10} & \theta_{17} & \theta_{23} & \theta_{28} & \theta_{32} & \theta_{35} \\
\theta_{16} & \theta_{10} & \theta_3 & \theta_{11} & \theta_{18} & \theta_{24} & \theta_{29} & \theta_{31} \\
\theta_{22} & \theta_{17} & \theta_{11} & \theta_4 & \theta_{12} & \theta_{19} & \theta_{25} & \theta_{28} \\
\theta_{27} & \theta_{23} & \theta_{18} & \theta_{12} & \theta_5 & \theta_{13} & \theta_{20} & \theta_{24} \\
\theta_{31} & \theta_{28} & \theta_{24} & \theta_{19} & \theta_{13} & \theta_6 & \theta_{14} & \theta_{21} \\
\theta_{34} & \theta_{32} & \theta_{29} & \theta_{25} & \theta_{20} & \theta_{14} & \theta_7 & \theta_{15} \\
\theta_{36} & \theta_{35} & \theta_{31} & \theta_{28} & \theta_{24} & \theta_{21} & \theta_{15} & \theta_8
\end{bmatrix}
\tag{4-16}
$$

similarly for $V(\theta), K(\theta)$. The frequency response of the truss structure is simulated by casting M, V, K matrices from (4-15) into State Space (SS) form, see (4-3), and obtaining the frequency response from there[1]. By inspection of the bode plot of the system we see that most information is contained in the $5\text{rads}^{-1} - 25\text{rads}^{-1}$ frequency band. 20 samples $H(\omega_i)$ are taken from this frequency band in a log-spaced manner. These samples are used for the identification algorithm from (4-11). The resulting matrices are given in Eq. (4-17).

[1] MATLAB command `freqresp`

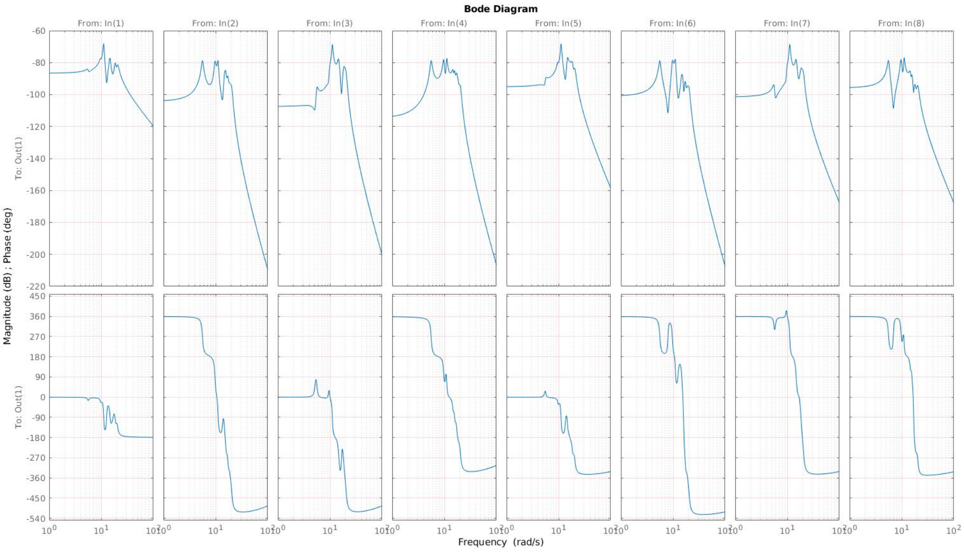


Figure 4-2: Bode plot of the 8 DOF Truss structure from Figure 4-15

$$
\begin{aligned}
M(\theta^*) &= \begin{bmatrix}
100.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 100.0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 100.0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 100.0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 100.0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 100.0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 100.0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 100.0
\end{bmatrix} \\
V(\theta^*) &= \begin{bmatrix}
136.0 & 0 & 0 & 0 & -50.0 & 0 & -17.7 & -17.7 \\
0 & 86.4 & 0 & -50.0 & 0 & 0 & -17.7 & -17.7 \\
0 & 0 & 136.0 & 0 & -17.7 & 17.7 & -50.0 & 0 \\
0 & -50.0 & 0 & 86.4 & 17.7 & -17.7 & 0 & 0 \\
-50.0 & 0 & -17.7 & 17.7 & 136.0 & 0 & 0 & -50.0 \\
0 & 0 & 17.7 & -17.7 & 0 & 86.4 & 0 & 0 \\
-17.7 & -17.7 & -50.0 & 0 & 0 & 0 & 136.0 & 0 \\
-17.7 & -17.7 & 0 & 0 & -50.0 & 0 & 0 & 86.4
\end{bmatrix} \\
K(\theta^*) &= \begin{bmatrix}
27071.0 & 0 & 0 & 0 & -10000.0 & 0 & -3535.5 & -3535.5 \\
0 & 17071.0 & 0 & -10000.0 & 0 & 0 & -3535.5 & -3535.5 \\
0 & 0 & 27071.0 & 0 & -3535.5 & 3535.5 & -10000.0 & 0 \\
0 & -10000.0 & 0 & 17071.0 & 3535.5 & -3535.5 & 0 & 0 \\
-10000.0 & 0 & -3535.5 & 3535.5 & 27071.0 & 0 & 0 & 0 \\
0 & 0 & 3535.5 & -3535.5 & 0 & 17071.0 & 0 & -10000.0 \\
-3535.5 & -3535.5 & -10000.0 & 0 & 0 & 0 & 27071.0 & 0 \\
-3535.5 & -3535.5 & 0 & 0 & 0 & -10000.0 & 0 & 17071.0
\end{bmatrix}
\end{aligned}
\tag{4-17}
$$

The normalized differences between actual system matrices and estimated matrices are

$$
\begin{aligned}
\frac{\|M - M(\theta^*)\|_F}{\|M\|_F} &= 3.7964 \cdot 10^{-9} \\
\frac{\|V - V(\theta^*)\|_F}{\|V\|_F} &= 4.7592 \cdot 10^{-9} \\
\frac{\|K - K(\theta^*)\|_F}{\|K\|_F} &= 4.8011 \cdot 10^{-9}
\end{aligned}
\tag{4-18}
$$

And the conclusion is that the system matrices have successfully been identified.

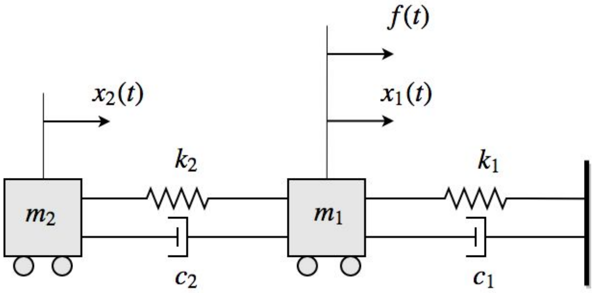


Figure 4-3: 2DOF-Mass Damper system

4-6-2 General Case Mass Damper

The next example is used to validate the general case, thus: B and C are rank deficient. We review a classical text book example, namely the 2DOF mass damper system as given in Figure 4-3. We will assume that the mass m_1 is excited by the force $f(t)$, and that the only output is the position of m_1. Note that in this specific instance we have $p + m = n$ actuators and sensors and hence, even the most permissible of formulations [1] is not applicable. However we will see that due to additional imposed structure on $M(\theta), V(\theta), K(\theta)$ that we can still obtain the actual physical constants. The dynamical equations of motion read as

$$\begin{aligned}\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \ddot{x}(t) \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \dot{x}(t) \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} x(t) &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} f(t) \\ y(t) &= \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)\end{aligned} \tag{4-19}$$

The physical parameters for this problem are chosen to be

$$\{m_1, m_2, c_1, c_2, k_1, k_2\} = \{1, 1, 0.1, 0.3, 2, 1\} \tag{4-20}$$

By eyeballing the bode plot in Figure 4-4, it is observable that most information lies within the $0.5 - 2\,\text{rads}^{-1}$ frequency band. As input data for the algorithm, 30 logspaced samples are taken from this frequency band.

From the equations of motion, the following parameterization is derived

$$M(\theta) = \begin{bmatrix} \theta_1 & 0 \\ 0 & \theta_2 \end{bmatrix} \quad V(\theta) = \begin{bmatrix} \theta_3 + \theta_4 & -\theta_4 \\ -\theta_4 & \theta_4 \end{bmatrix} \quad K(\theta) = \begin{bmatrix} \theta_5 + \theta_6 & -\theta_6 \\ -\theta_6 & \theta_6 \end{bmatrix} \tag{4-21}$$

A uniform regularization parameter over all frequencies $\lambda_i = 1\,\text{for}\, i = 1, \dots, s$ is chosen. At first the algorithm is initialized with a variable vector of all zeros, however this leads to a local minimum where $\hat{m}_1, \hat{k}_1, \hat{v}_1 = 0$. Hereafter the algorithm is initialized with a variable vector where the entries are taken from the normal distribution, we get the solution

$$M(\theta) = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} \quad V(\theta) = \begin{bmatrix} 0.40 & -0.10 \\ -0.10 & 0.10 \end{bmatrix} \quad K(\theta) = \begin{bmatrix} 3.00 & -1.00 \\ -1.00 & 1.00 \end{bmatrix} \tag{4-22}$$

Which is consistent with the true parameter vector. The convergence plot of the this problem instance is given in Figure 4-5.

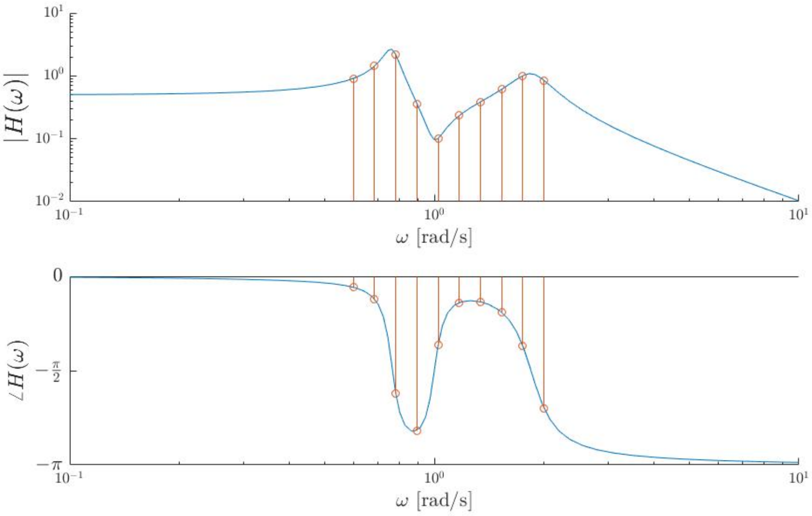


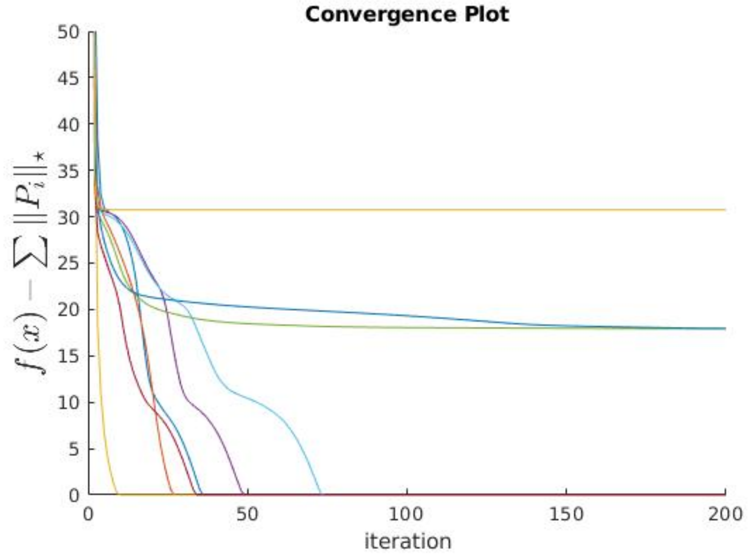
Figure 4-4: Bode plot of the 2DOF mass-damper system Figure 4-3. The stemplots represent the 10 data points used for identification. Note that although the data points seem linearly spaced they are in fact log-spaced between 0.6 and 2 rad/s.

4-7 Future Work

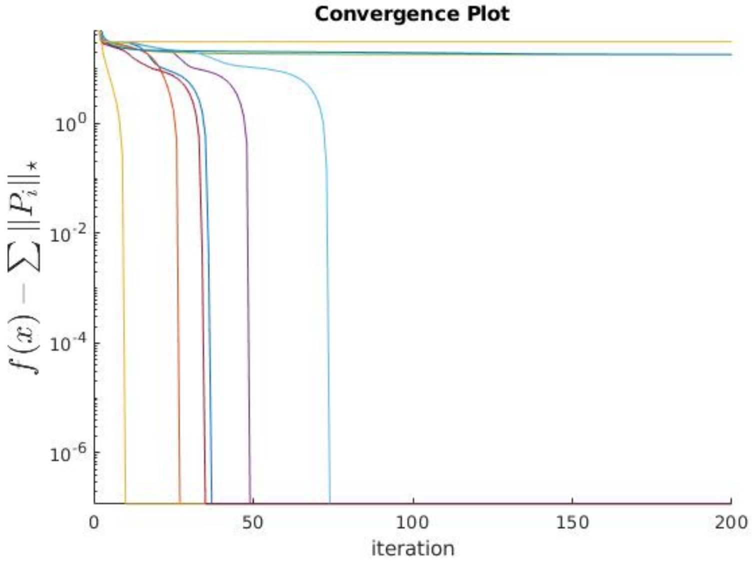
In Chapter 3 we applied a pair of substitutions to the transfer function of the state-space model. It was shown that is helpful during the optimization procedure, because we then have the same matrices, distributed among multiple equations, in different contexts. This leads to a better balanced optimization problem. It is presumed that applying such a double substitution for the 2nd order MDE's transfer function could also be beneficial, leading to a improved solvable optimization problem. However, we have not yet studied its potential.

4-8 Summary

In this chapter we have discussed how we can directly identify parameters in a q'th order MDE's using frequency domain data. We have focused our attention on 2nd order models which are widely used in structural identification. We have seen that given a full rank input or output matrix the problem reduces to a linear LS. However in the general case the problem is harder, but can be posed as a bilinear feasibility problem, meaning it is solvable with Sequential Convex Relaxation (SCR). We have tested the method on a physical example. It was shown that we were able to identify a model, which other identification methods cannot. We did this by exploiting additional known structure of the model. The proposed methodology can handle all types of measurements (i.e. displacements, velocities,and/or accelerations).



(a) linear scale



(b) logarithmic scale

Figure 4-5: Convergence plot of the mass-damper problem. On the $x - axis$ the iteration number, on the $y - axis$ the difference of $f(x)$ w.r.t. the optimal value. Every traced line shows the the convergence for a certain initialization of x where each entry in x is take from a normal distribution with zero mean and unit variance. Out of the 10 initializations, the algorithm manages to converge to the optimal point 6 times within 200 iterations, the 2 traces around 20 are still converging and eventually do get to the global optimum after 258, two traces near 31 do not converge and are stuck, for this case $\hat{m}_1, \hat{k}_1, \hat{v}_1 = 0$. Also by looking at log-scaled plot: note the super-linear convergence near the optimal point.

Chapter 5

Estimating 2nd Order Models via Power Spectral Density

5-1 Introduction

Classically one applies a measurable input to a system under investigation and measures its output. These measurements can be used in a variety of different identification methods, which lead to the experimental model. However there are cases when one does not have the possibility to apply an artificial measurable force, one then has to rely on ambient excitation sources. It is however virtually impossible to measure this ambient excitation. In these cases one has to rely solely on the output measurements and the statistical knowledge of the input. The statistical average of a certain signal or sort of signal (noise), as analyzed in terms of its frequency content, is called its spectrum. We will use this spectrum, input- and output- to identify second order models. [35] provides a thorough reference, how to obtain the power spectrum given discrete data. For a compact reference the reader is referred to [26]. Furthermore, the focus lies on processes which are a function of time, but one can similarly apply the technique to spatial problems where the power spectral density is given in terms of spatial frequency.

Outline

This chapter is organized as follows: §5-2 reviews some import properties of the Power Spectral Density (PSD). §5-3 gives the problem statement. §5-4 treats the literature regarding the problem at hand. In §5-5 an identification algorithm is proposed to address the problem statement. In §5-6 a numerical example is looked upon. In §5-7 we solve the same problem using Non-Linear Least Squares and make a comparison. §5-8 presents some future work. Finally in §5-9 we conclude.

5-2 Spectral Density

Energy spectral density describes how the energy of a signal is distributed with frequency The energy spectral density function $\Phi_u(\omega)$ of a finite energy signal continuous-time signal $x(t)$ is by

$$\Phi_x(\omega) = |X(\omega)|^2 \tag{5-1}$$

Property 5.1 (Real-valued)**.** *The spectrum of a real valued process is real and an even function of frequency*

$$\Phi_x(-\omega) = \Phi_x(\omega) \tag{5-2}$$

White noise is a random signal having equal intensity at different frequencies, giving it a constant PSD

5-3 Problem formulation

problem of interest Given a system with m inputs, p outputs, state dimension n, and given spectral densities $\Phi^y(\omega_i) \in \mathbb{C}^{p\times p}, \Phi^u(\omega_i) \in \mathbb{C}^{m\times m}$ for $i = 1..s$, which have the following relationship

$$\Phi^{yy}(\omega_i) = H(\omega_i,\theta)\Phi^{uu}(\omega_i)H^H(\omega_i,\theta) \qquad \text{for} \quad i = 1,\dots,s \tag{5-3}$$

where

$$H(\omega_i,\theta) = C\big(-\omega_i^2 M(\theta) + j\omega V(\theta) + K(\theta)\big)^{-1}B \quad \text{for} \quad i = 1\dots s \tag{5-4}$$

are generated from the parameterized model (4-4) for a certain value of $\theta^\star$. The problem of interest is to find the parameter vector $\theta^\star$. For the rest of the chapter we will assume we have a full set of actuators, i.e. $p = n$, which is not uncommon in literature.

5-4 Overview

In [36] the authors also review the question of estimating a 2nd order model whilst not being able to measure the inputs. Using the same equation as (4-3), and given an immeasurable input, the second order system is said to be equivalent to the purely stochastic system

$$\begin{aligned} x_{k+1} &= Ax_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{5-5}$$

where w_k, v_k are zero mean identically distributed normal vectors. Using the outputs $\{y_1,\dots,y_N\}$ the maximum likelihood of the given output is maximized using the innovation form representation in an iterative scheme. After having obtained A, C the matrices M, K are extracted in modal form, i.e. M, K diagonal. The authors claim that the input should be white-noise inorder to be succesful.

5-5 Identification Method

Given that C is full rank, (5-3) can be written as,

$$X(\theta,\omega_i)C^{-1}\Phi^{yy}(\omega_i)C^{-H}X(\theta,\omega_i)^H = B\Phi^{uu}(\omega_i)B^H \qquad \text{for} \quad i = 1\ldots s \tag{5-6}$$

where, for readability we have written

$$X(\theta,\omega_i) = -\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta) \qquad \text{for} \quad i = 1,\ldots,s \tag{5-7}$$

Notice that (5-6) is a quadratic matrix equality constraint, however as we have seen in §2-2, $x(x-1) = 0$ is also a quadratic equality constraint but still NP-hard. Using the notation $P_i = C^{-1}\Phi^{yy}(\omega_i)C^{-H}$ and $Q_i = B\Phi^{uu}(\omega_i)B^H$, the feasibility problem we would like to solve is the following,

$$\begin{aligned} &\text{find} \quad \theta \\ &\text{such that} \\ &\qquad X(\theta,\omega_i)P_iX(\theta,\omega_i)^H = Q_i \qquad \text{for} \quad i = 1\ldots s \\ &\qquad M(\theta) \succeq 0 \\ &\qquad V(\theta) \succeq 0 \\ &\qquad K(\theta) \succeq 0 \end{aligned} \tag{5-8}$$

which, using Sequential Convex Relaxation (SCR), can be solved by the procedure outlined in Algorihtm 5.1.

Algorithm 5.1 *Estimate 2nd order system from Spectra*

```
given initial estimate $\theta \in \mathbb{R}^l$, spectra $\Phi^{yy}(\omega_i)$, $\Phi^{uu}(\omega_i)$ for $i = 1..s$
repeat
  1. Update $Y_i := -X(\theta,\omega_i)$ for $i = 1,\ldots,s$
  2. Update $\theta$ using
     $\arg\min_{\theta} \sum_i^s \left\| \begin{bmatrix} Q_i + Y_iP_iX_i^H + X_iP_iY_i^H + Y_iP_iY_i^H & (X_i+Y_i)P_i \\ P_i(X_i+Y_i)^H & P_i \end{bmatrix} \right\|_\star$
       subject to
           $M(\theta) \succeq 0$
           $V(\theta) \succeq 0$
           $K(\theta) \succeq 0$
     where
           $X_i = -\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta)$
           $P_i = C^{-1}\Phi^{yy}(\omega_i)C^{-H}$
           $Q_i = B\Phi^{uu}(\omega_i)B^H$
until stopping criterion is satisfied
```

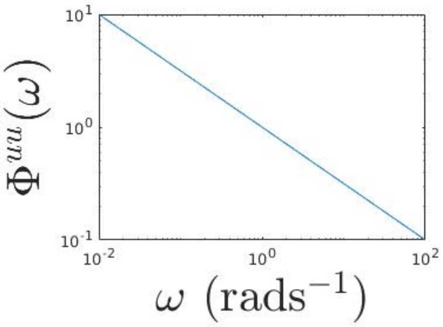


Figure 5-1: Pink-noise, input power spectrum.

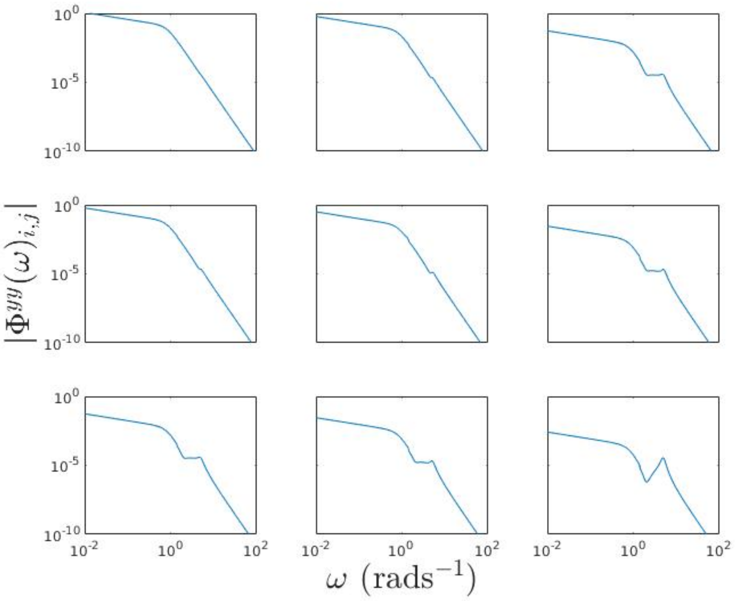


Figure 5-2: Figural representation of magnitude of the output power spectrum $\Phi^{yy}(\omega)$. The i, j'th subplot depicts the absolute value of $\Phi^{yy}_{i,j}(\omega)$ for $0.01 \leq \omega \leq 100$.

5-6 Simulations

In this section we consider a numerical example, The system under investigation is depicted in Figure 5-4. The mass, damping, stiffness matrices are chosen as the following.

$$M = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 2 & -0.5 \\ 0 & -0.5 & 1 \end{bmatrix} \quad V = \begin{bmatrix} 5 & -0.1 & -0.1 \\ -0.1 & 0.2 & -0.1 \\ -0.1 & -0.1 & 1.1 \end{bmatrix} \quad K = \begin{bmatrix} 4 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 22 \end{bmatrix} \tag{5-9}$$

We will assume that all Degree Of Freedom (DOF)'s are directly measured, and that the input is applied only in the x-direction, i.e. the input and output matrices read as

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad F = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{5-10}$$

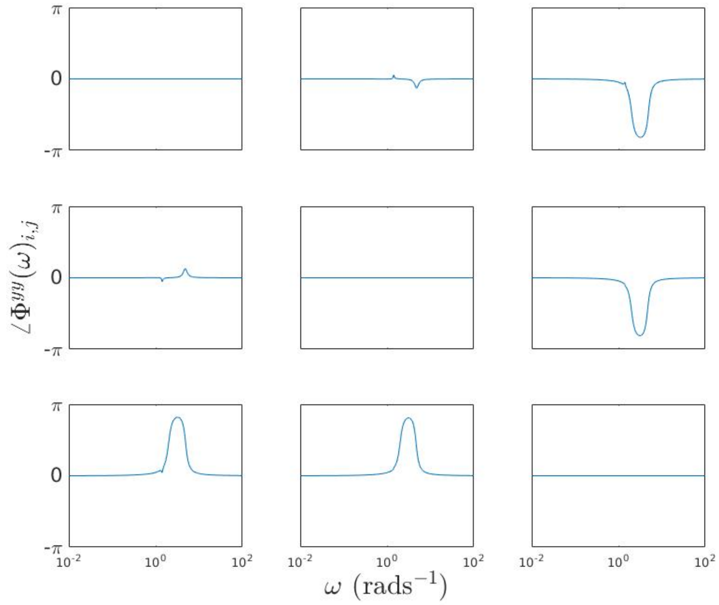


Figure 5-3: Figural representation of the phase angle of the output power spectrum $\Phi^{yy}(\omega)$. The i,j'th subplot depicts the phase of $\Phi^{yy}_{i,j}(\omega)$ for $0.01 \leq \omega \leq 100$.

To show that the method works equally well for white as coloured noise, a pink noise input spectrum is artificially generated. Pink noise or $1/f$ noise is a signal or process with a frequency spectrum such that the PSD is inversely proportional to the frequency of the signal. They occur widely in nature and are a source of considerable interest in many fields. We create the pink noise by using 170 logspaced frequency points between $\omega = 0.1$ and $\omega = 100$ rads^{-1} and applying

$$\Phi^{uu}(\omega_i) = \omega_i^{-0.5} \tag{5-11}$$

The resulting spectrum is shown in Figure 5-1. Using the quintuple of matrices (M, V, K, F, H) and the input spectrum $\Phi^{uu}(\omega_i)$ for $i = 1, \ldots, 170$, we generate the output spectrum $\Phi^{yy}(\omega_i)$ using (5-3). The magnitude and phase of the output spectrum as a function of frequency are shown in Figure 5-2 and Figure 5-3 respectively.

At first the algorithm is initialized such that each Y_i in Algorithm 5.1 is equal to identity, however this leads to a non-global minimum, hereafter it is found that fast convergence and a high chance of converging to the global optimum are obtained by initializing each Y_i as a high valued random positive definite matrix, this is done by randomly generating a matrix Z_i where each entry is taken from a normal distribution with zero mean and a variance of 10^2, i.e. $z_{ij} \sim (0, 100)$. The positive definite matrix is then created by $Y_i = Z_i Z_i^H$. The converge plot using these initial conditions is given in Figure 5-5. Given that the algorithm has converged to the global minimum, a certain realization of the given difference in estimated

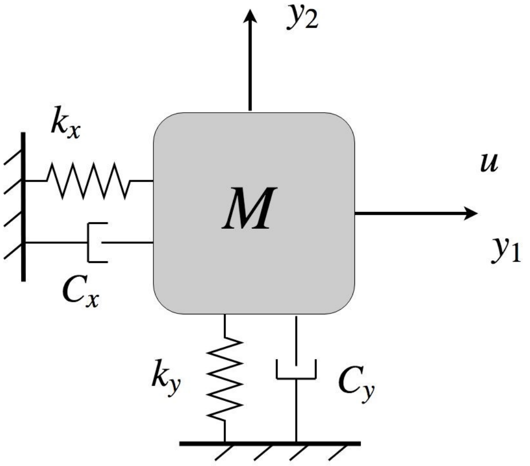


Figure 5-4: Schematic depiction of a spring-mass-damper system in two dimensions. The problem discussed in (7-17) has 3 dimensions, so one should imagine an additional set of spring and damper C_z, k_z lateral to the paper. The input u are applied to the first mode, the outputs are measured in the x, y and z direction.

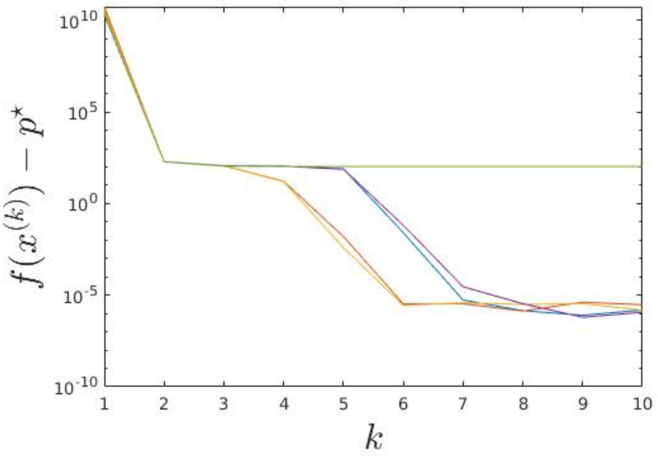


Figure 5-5: Error versus iteration k for 5 initializations. 4 out of 5 times maximum accuracy is attained within 7 iterations.

mass, damping, stiffness matrices is given in (5-12).

$$
\begin{aligned}
M - M(\hat{\theta}) &= 10^{-5}\begin{bmatrix} 0.388 & -0.7629 & 0.1737 \\ -0.7629 & 1.507 & -0.3429 \\ 0.1737 & -0.3429 & 0.0778 \end{bmatrix} \\
V - V(\hat{\theta}) &= 10^{-6}\begin{bmatrix} 0.4446 & -0.7931 & 0.1828 \\ -0.7931 & 1.471 & -0.3386 \\ 0.1828 & -0.3386 & 0.0759 \end{bmatrix} \\
K - K(\hat{\theta}) &= 10^{-5}\begin{bmatrix} 0.7829 & -1.524 & 0.3476 \\ -1.524 & 2.971 & -0.6772 \\ 0.3476 & -0.6772 & 0.1493 \end{bmatrix}
\end{aligned}
\tag{5-12}
$$

It is shown that the error is in magnitude of 10^{-5}, thus it is concluded that the system matrices have accurately been retrieved.

5-7 Comparison

An alternative approach to identify the parameter vector as stated in the problem statement §5-3, is to pose the problem as a Non-Linear Least Squares (NLLS) problem. The problem would then look like

$$
\min_{\theta \in \mathbb{R}^l} \sum_{i=1}^{s} \|\Phi^{yy}(\omega_i) - CX_i^{-1}H\Phi^{uu}(\omega_i)H^H X_i^{-H}C^H\|_F^2 \tag{5-13}
$$

where $X_i = -\omega_i^2 M(\theta) + j\omega_i V(\theta) + K(\theta)$, or variants hereof (using the invertibility of C, X). The optimization problem (5-13) can be solved using a variety of functions in MATLAB's optimization toolbox[37]. The function `lsqnonlin` exclusively handles NLLS problems, `fminunc` can be used for unconstrained optimization problems, whereas `fmincon` is also applicable for constrained optimization. Although, `fmincon` can deal with equalities and inequalities it cannot explicitly deal with generalized inequality constraints in the form of Linear Matrix Inequality (LMI)'s as in (5-8). We show in the next paragraph that the inclusion of the three LMI's in (5-8) makes all the difference in successfully retrieving the parameter vector.

Unconstrained Solver

In this section we apply MATLAB's non-linear unconstrained solver, `fmincon` to the NLLS formulation from (5-13). 20 attempts were made, at each attempt intializing the parameter vector θ_0 with entries from the normal distribution. From these 20 initialization none of the attempts converged to the global optimum. Below the matrices are shown, extracted from the attempt with the lowest final objective value, $f(x^\star) = 0.064485$.

$$\begin{aligned} M(\hat{\theta}) &= \begin{bmatrix} 6.42 & -3.554 & 0.05659 \\ -3.554 & 6.476 & -0.2365 \\ 0.05659 & -0.2365 & 2.385 \end{bmatrix} \\ V(\hat{\theta}) &= \begin{bmatrix} 5.676 & -1.177 & -1.42 \\ -1.177 & 1.912 & 1.635 \\ -1.42 & 1.635 & 11.06 \end{bmatrix} \\ K(\hat{\theta}) &= \begin{bmatrix} 4.508 & -1.797 & -12.91 \\ -1.797 & 1.205 & 24.49 \\ -12.91 & 24.49 & 1.741 \end{bmatrix} \end{aligned} \tag{5-14}$$

The magnitudes of the entries seem to be of the same order as the ones of the original system (5-9), but that is all we can say, most estimates are way of their actual values.

Sequential Convex Relaxation without Linear Matrix Inequatities

To emphasize the fact that the triplet of constraints $M(\theta) \succeq 0, V(\theta) \succeq 0, K(\theta) \succeq 0$, really do make a difference, and the mismatched models in (5-14) are not for instance a result of the inverse in (5-13), it is possible to solve the feasibility problem (5-8) without the three LMI's. After again multiple (20) initializations with the same distribution of values we used for the initialization of the original algorithm, the one with the lowest objective value is singled out. The resulting matrices are:

$$\begin{aligned} M(\hat{\theta}) &= \begin{bmatrix} 306.6 & -611.4 & 137.1 \\ -611.4 & 1207.0 & -275.2 \\ 137.1 & -275.2 & 70.62 \end{bmatrix} \\ V(\hat{\theta}) &= \begin{bmatrix} 34.36 & -63.96 & 13.33 \\ -63.96 & 118.8 & -27.28 \\ 13.33 & -27.28 & 15.85 \end{bmatrix} \\ K(\hat{\theta}) &= \begin{bmatrix} 630.3 & -1223.0 & 274.2 \\ -1223.0 & 2384.0 & -552.6 \\ 274.2 & -552.6 & 319.8 \end{bmatrix} \end{aligned} \tag{5-15}$$

which by no means resemble the actual system matrices as given in (5-9). Thus we conclude that the flexibility of SCR, allowing us to incorporate additional LMI constraints, is decisive in estimating the actual parameter vector θ.

5-8 Future Work

It was assumed in this chapter that the output matrix H is of full rank. This assumption lead us to a bilinear optimization problem with the number of decision variables exactly equal to the number of unknowns in the matrices $M(\theta), V(\theta), K(\theta)$. In chapter 4, we reviewed the concept of input-output equivalence, we saw that given a full rank input- or output matrix there exists a Least Squares (LS) solution to retrieve the unknown parameter vector θ. It would be interesting to think if there exists such equivalence in the spectral identification setting, i.e. given the fact that B is of full rank: is it possible to pose the identification optimization problem in such a way that we do not introduce additional decision variables? And thus leading to an equally hard bilinear optimization problem as is the case with a full rank C matrix.

Consider for example again the truss-structured from Figure 4-1, It would be likely that the wind (possibly modeled as white noise) affects all 8 nodes. With the algorithm described in this chapter we would need as many sensors as nodes to identify the bridge, but maybe a single sensor could be enough.

5-9 Summary

In this chapter an algorithm was presented that takes the Power Spectral Density of the input and the output, and extracts the mass, stiffness and damping matrices. Given that we have a full set of sensors, it turns out that the problem can be posed as bilinear feasibility problem, with the number of decision variables equal to the number of unknown parameters. The bilinear feasibility problem is infact a quadratic feasibility problem, but as such, is still non-convex. Using the SCR algorithm the non-convex problem can be solved in an iterative way. Furthermore the flexibility of SCR allows us to incorporate additional knowledge of the structure into the constraints; due to reciprocity the parameterized physical matrices are positive definite, which in the form of LMI's can be incorporated into the constraints. The incorporation of these LMI's make the problem less non-convex, as the search space is now contained within the cones associated with the generalized inequalities. The method has been validated on a physical example.

Chapter 6

State Space via Similarity Transformation

6-1 Introduction

In this chapter an alternative approach to the grey-box identification problem is proposed. The method consists of first estimating an unstructured, black-box model using e.g. Subspace Identification (SI) methods, followed by the recovery of the physical parameters embedded in the concerning structured model. The classical SI methods such as MOESP [38] and N4SID [26], do not consistently estimate the system matrices, but converge to models that give the true matrices up to a similarity transformation. The second step turns out to be a small-scale non-convex optimization problem. The proposed algorithm framework introduces additional variables to represent the inverse of the similarity transformation. A bilinear constraint is employed to enforce the relationship between transformation matrix and its inverse. The algorithm can be be applied to the realization of non-identifiable grey-box models.

Outline

The chapter is organized as follows: §6-2 reviews the similarity transformation. §6-3 formulates the problem statement. §6-4 reviews the literature concerning the grey-box identification problem via the similarity transformation. In §6-5 we provide an alternative way to solve the identification problem. §6-6 demonstrates the performance of the proposed identification method applied to a physical example. In §6-7 we compare the algorithm with other methods followed by some conclusions in §6-8.

6-2 Similarity Transformation

Consider again the State Space (SS) model

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{6-1}$$

It is possible to define a new set of independent variables (i.e., T is invertible) by

$$z(t) = Tx(t) \tag{6-2}$$

This set of new variables can be used as state variables. Start by solving for $x(t) = T^{-1}z(t)$, we can then write the SS equation as

$$\begin{aligned} \dot{z(t)} &= TAT^{-1}z(t) + TBu(t) \\ y(t) &= CT^{-1}z(t) \end{aligned} \tag{6-3}$$

Which is recognized as the SS representation

$$\begin{aligned} \dot{z(t)} &= \mathbb{A}z(t) + \mathbb{B}u(t) \\ y(t) &= \mathbb{C}z(t) \end{aligned} \tag{6-4}$$

where

$$\mathbb{A} = TAT^{-1}, \quad \mathbb{B} = TB, \quad \mathbb{C} = CT^{-1} \tag{6-5}$$

The original state space equation and the transformed system describe exactly the same system, they have the same input to output mapping, only different state-variable coordinates. This shows that the SS representation is non-unique, however the transfer function is unique,

$$\begin{aligned} \mathbb{H}(s) &= \mathbb{C}(sI - \mathbb{A})^{-1}\mathbb{B} \\ &= CT^{-1}(sI - TAT^{-1})^{-1}TB \\ &= C(T^{-1}(sI - TAT^{-1})T)^{-1}B \\ &= C(sI - A)^{-1}B \qquad = H(s) \end{aligned} \tag{6-6}$$

6-3 Problem Formulation

Consider a parameterized state space model as given in Eq. (3-6), the problem of interest is then given as

problem of interest Given the quadruple of matrices $(\mathbb{A}, \mathbb{B}, \mathbb{C}, D_T)$ in an unknown sate basis , determine the parameter vector θ satisfying;

$$\mathbb{C}(sI - \mathbb{A})^{-1}\mathbb{B} = C(\theta)(sI - A(\theta))^{-1}B(\theta) \tag{6-7}$$

6-4 Overview

There are multiple routes to solving the algebraic problem in Eq. (7-3). One is to simultaneously estimate the similarity transformation T and the parameter vector θ. Other routes involve directly solving Eq. (7-3) at the expense of non-convexity. Furthermore, there is a method where we use the transposed matrices to simulate a system response, which in turn we use to estimate the true system matrices.

6-4-1 Using the Similarity Transformation

Eq. (7-3) means that there exists a similarity transformation T such that

$$T\mathbb{A} = A(\theta)T \tag{6-8a}$$

$$T\mathbb{B} = B(\theta) \tag{6-8b}$$

$$\mathbb{C} = C(\theta)T \tag{6-8c}$$

From that it is possible to formulate the minimization problem

$$\min_{T,\theta} \|T\mathbb{A} - A(\theta)T\|_F^2 + \|T\mathbb{B} - B(\theta)\|_F^2 + \|\mathbb{C} - C(\theta)T\|_F^2 \tag{6-9}$$

The above optimization problem is bilinear in θ and T. To solve this problem various methods have been proposed. In [39] an alternating minimization method was presented, a gradient descent method was given in [26] and a null-space-based optimization method was developed in [40]. Even if the above methods can find a global optimum $(T^\star, \theta^\star)$, It might not be meaningful for the identification purpose stated in Eq. (7-3). The reason is that the optimal solution $T^\star$ might be singular and the obtained transfer function $C^\star(sI - A^\star)^{-1}B^\star$ might not be equal to $\mathbb{C}(sI - \mathbb{A})^{-1}\mathbb{B}$. In fact equations (7-3) and (6-8) are equivalent if and only if T is non-singular [41]. To deal with this problem, a condition-number constraint on T was considered in [28], which turns out to be a non-smooth and highly non-convex optimization problem.

6-4-2 Minimize the model-matching Criterion

Another route is to minimize the model-matching criterion $\|G(s,\theta) - G(s,\theta)\|$ using either H_2 norm or H_∞ norm, as suggested in [42]. The H_∞ norm based matching has been investigated in [43] and [44]. Compared with the minimization Eq. (6-9), the H_∞ method reduces the number of variables, but at the price of a semi-infinite and non-smooth program.

6-4-3 Using the Hankel Matrix of Impulse Response

Additionally in [25] a method is proposed which takes a slight detour; a block Hankel matrix of the system impulse response is created. By exploiting the low-rank property of this matrix with its shift structure, the problem is formulated as a structured low-rank matrix factorization problem. Using a difference of convex programming technique, this is numerically solved. Because this method doesn't make use of a similarity transformation matrix, the method circumvents issues concerning the singularity of the transformation matrix.

6-5 Identification Method

In this section we propose a new method at solving Eq. (7-3). The method also aims estimating the transformation matrix T, and the parameter vector θ. By introducing a substitution for the inverse of T, we can pose the problem in a bilinear fashion, which is equivalent to the original problem statement, in the sense that if we solve the bilinear problem, we have a non-singular T and thus solve Eq. (7-3).

6-5-1 Construction of Bilinear form

Consider the equations from Eq. (6-8), by pulling T to the other side in (6-8a) and (6-8c), then applying the substitution $S = T^{-1}$ we can write

$$A(\theta) = T\mathbb{A}S \tag{6-10a}$$
$$B(\theta) = T\mathbb{B} \tag{6-10b}$$
$$C(\theta) = \mathbb{C}S \tag{6-10c}$$
$$I = ST \tag{6-10d}$$

where the last equation is needed to describe the relationship between S and T. In the above equation set we have two linear equations namely (6-10b) and (6-10c) and two bilinear ones (6-10a) and (6-10d). Note that the equation (6-10d) implies that the matrix pair S, T are invertible by Definition 6.1, making the set of equations (6-10) equivalent to the problem statement (7-3).

Definition 6.1. *[18] A square matrix $A \in \mathbb{R}^{n\times n}$ is said to be invertible if and only if there exists another square matrix $B \in \mathbb{R}^{n\times n}$ such that*

$$AB = BA = I_n \tag{6-11}$$

The proposed identification scheme uses a relaxation of the bilinear constraints in an indirect manner. It is therefore proposed to not only use the constraint $I = ST$ but also its counterpart

$$I = TS \tag{6-12}$$

Note that it would also have been possible to directly deal with the equation set from (6-8). However as stated earlier, the invertibility of the similarity transformation matrix is not guaranteed.

6-5-2 Standard form

Casting the identifcation problem into the standard optimization form leads to

$$\begin{aligned} \text{find} \quad & \theta, T, S \\ \text{subject to} \quad & \\ T\mathbb{A}S &= A(\theta) \\ TS &= I \\ ST &= I \\ T\mathbb{B} - B(\theta) &= 0 \\ \mathbb{C}S - C(\theta) &= 0 \end{aligned} \tag{6-13}$$

Note that the above problem has three bilinearities contained in the constraints, however we can still apply the relaxation technique as discussed in Chapter 2. By relaxing each bilinear constraint individually into the objective function with the relaxation parameter $\lambda \in \mathbb{R}$ chosen to be equal for the relaxation of $ST = I$ and $TS = I$ we get

$$\begin{aligned} \min_{\theta,T,S} & \left\| \begin{bmatrix} A(\theta) - T\mathbb{A}\bar{S} - \bar{T}\mathbb{A}S + \bar{T}\mathbb{A}\bar{S} & (T - \bar{T})\mathbb{A} \\ \mathbb{A}(S - \bar{S}) & \mathbb{A} \end{bmatrix} \right\|_\star + \\ & \lambda \left\| \begin{bmatrix} I - S\bar{T} - \bar{S}T + \bar{S}\bar{T} & S - \bar{S} \\ T - \bar{T} & I \end{bmatrix} \right\|_\star + \lambda \left\| \begin{bmatrix} I - T\bar{S} - \bar{T}S + \bar{T}\bar{S} & T - \bar{T} \\ S - \bar{S} & I \end{bmatrix} \right\|_\star \\ & \text{subject to} \qquad B(\theta) = T\mathbb{B}, C(\theta) = \mathbb{C}S \end{aligned} \tag{6-14}$$

The summary of the identification procedure is the following

Algorithm 6.1 *parameteric State Space Identification using the Similarity Transformation matrix*

given IO data $u(k), y(k)$ for $k = 1 \dots N$

do

1. Estimate discrete time black box state space model using subspace technique
2. Apply zero order hold to the discrete time model to obtain continuous time model
3. Run the Sequential Convex Relaxation (SCR) algorithm

6-6 Simulation of an E-Nose model

This simulation example takes the same model as the one treated in [40]. Consider the three layer compartment system shown in Figure 6-1. where $s_i(t)$ and k_{ij} are respectively, the concentration of a chemical compound at time t (in mg/l) and the kinetic rate constant in units

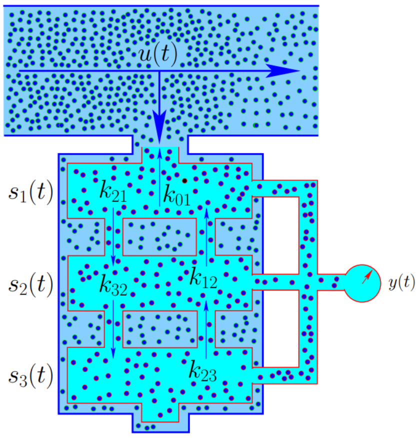


Figure 6-1: Three compartment e-nose sensor model

of time^{-1}, associated with the mass leaving the j'th compartment and arriving at the i'th compartment. The light blue region denotes the delineation of the three imaginary compartments representing a simplified model of a physical system. The accuracy will be governed by the number of compartments representing the system. Notice that each compartment allows interaction allows interaction from their upstream and downstream neighbouring compartments. The input $u(t)$ is a gas released at $t = 0$ and allowed to disperse through the sensor chamber. The conservation of mass principle states that the rate of change of mass accumulation within the $i'th$ compartments is equal to the mass flow rate-in minus the mass flow rate-out of the compartment at time t. Thus we have three state equations of the form

$$\dot{s}_1(t) = u(t) - (k_{01} + k_{21})s_1(t) + k_{12}s_2t \tag{6-15a}$$

$$\dot{s}_2(t) = k_{21}s_1(t) - (k_{12} + k_{32})s_2(t) + k_{23}s_3(t) \tag{6-15b}$$

$$\dot{s}_3(t) = k_{32}s_2(t) - k_{23}s_3(t) \tag{6-15c}$$

and with the system being initially at rest, the initial conditions are $s_i(0) = 0$, for $i = 1, 2, 3$. The output of the system is the sum of the concentrations in each compartment. i.e. $y(t) = f(s_1(t) + s_2(t) + s_3(t))$ where f is a proportion of the total concentration of the gas in the system. The values of the parameters used in this study are the same as in [40],

$$\{k_{01}, k_{21}, k_{12}, k_{32}, k_{23}, f\} = \{0.1, 0.3, 0.2, 0.4, 0.25, 0.15\} \tag{6-16}$$

The concerned actual state space matrices then read as

$$A = \begin{bmatrix} -0.4 & 0.2 & 0 \\ 0.3 & -0.6 & 0.25 \\ 0 & 0.4 & -0.25 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 0.15 & 0.15 & 0.15 \end{bmatrix} \tag{6-17}$$

6-6-1 n4sid

The first step of the procedure consists of estimating a fully parameterized state-space model $(\mathbb{A}, \mathbb{B}, \mathbb{C})$. In order to be realistic with respect to standard use of such a system, the e-nose

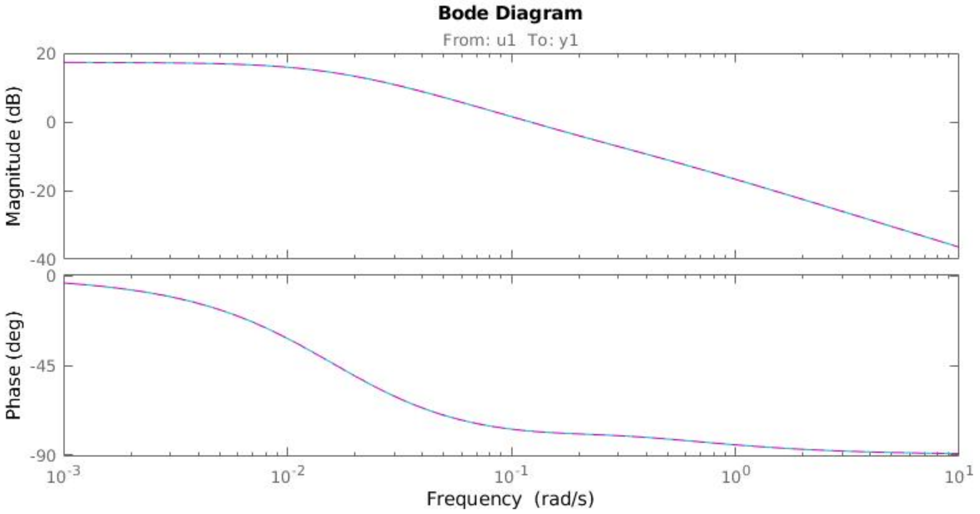


Figure 6-2: Bode plot of the real system (A, B, C) and the initial model $(\mathbb{A}, \mathbb{B}, \mathbb{C})$

sensor model is excited with a Gaussian pulse input signal. Second the N4SID algorithm[1] [26] is applied to the acquired data set. Because this subspace-based method leads to a discrete-time model, its continuous counterpart is computed using the zero-order hold technique[2] [45], leading to matrices

$$\mathbb{A} = \begin{bmatrix} -0.6773 & 0.7994 & -0.1172 \\ 0.2979 & -0.4156 & -0.2437 \\ -0.6528 & 0.7859 & -0.1571 \end{bmatrix} \qquad \mathbb{B} = \begin{bmatrix} 0.1233 \\ -0.1025 \\ 0.0645 \end{bmatrix} \qquad \mathbb{C} = \begin{bmatrix} 2.4717 & -0.0417 & -2.4254 \end{bmatrix} \tag{6-18}$$

The initial model is validated by comparing the frequency response with the actual system Figure 6-2.

6-6-2 Identifiable model

The state and output equations can be written as

$$A(\theta) = \begin{bmatrix} -\theta_1 - \theta_2 & \theta_3 & 0 \\ \theta_2 & -\theta_3 - \theta_4 & \theta_5 \\ 0 & \theta_4 & -\theta_5 \end{bmatrix} \qquad B(\theta) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad C(\theta) = \begin{bmatrix} \theta_6 & \theta_6 & \theta_6 \end{bmatrix} \tag{6-19}$$

where the unknown parameters gathered into the vector θ are related the physical parameters governing the behaviour of the system as follows

$$\theta_1 = k_{01} \quad \theta_2 = k_{21} \quad \theta_3 = k_{12} \quad \theta_4 = k_{32} \quad \theta_5 = k_{23} \quad \theta_6 = f \tag{6-20}$$

As far as the identifiability of the parameterization in (6-19) goes, it is possible to show that the structure is identifiable. This can be shown by calculating the transfer function of the

[1] The MATLAB function `n4sid` is used to get the initial discrete time black-box model

[2] The MATLABfunction `d2c` is used for this.

system

$$H(s) = \frac{\alpha_1 s^2 + \alpha_2 s + \alpha_3}{s^3 + \alpha_4 s^2 + \alpha_5 s + \alpha_6 s} \tag{6-21}$$

Where α_i for $i = 1 \ldots 6$ are well-defined combinations of the physical parameters $\{k_{01}, k_{21}, k_{12}, k_{32}, k_{23}, f\}$ which by inspection can be uniquely extracted from the α_i's.

The proposed algorithm is applied to restructure the initial black-box model $(\mathbb{A}, \mathbb{B}, \mathbb{C})$ into the grey-box structure $(A(\theta), B(\theta), C(\theta))$. At first the matrices S, T are initialized as zero matrices, however the method fails to converge to the global minimum for this initialization. On the other hand initializing the optimization algorithm with S and T with entries drawn from a random distribution lets us converge to the global minimum after a few trials. Roughly 3 out of every 10 of these initialization lead to the global minimum.

6-6-3 Non-Identifiable model

As it was shown in the foregoing, it can be really time consuming and difficult to to find the unique optimal point. The authors in [28] note that their estimation method requires at least 2 hours of computation on a recent computer in order to converge the local minimum θ^*, T^*. The authors explain the huge computational load by the existence of long and tight valleys in the involved cost function. Therefore the authors in [28] and [40], instead of focusing on the unique point suggest to 1) estimate a manifold containing the unique point; 2) extract the identifiable model among all the models included in the afore-estimated manifold by resorting to an analytic procedure. In the subsequent paragraph this two step procedure is studied.

Reparameterize the state-space model as

$$\mathcal{A}(\hat{\theta}) = \begin{bmatrix} \hat{\theta}_1 & \hat{\theta}_2 & 0 \\ \hat{\theta}_3 & \hat{\theta}_4 & \hat{\theta}_5 \\ 0 & \hat{\theta}_6 & -\hat{\theta}_5 \end{bmatrix} \qquad \mathcal{B}(\hat{\theta}) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \mathcal{C}(\theta) = \begin{bmatrix} \hat{\theta}_7 & \hat{\theta}_7 & \hat{\theta}_7 \end{bmatrix} \tag{6-22}$$

The unknown parameters θ_i are now related to the physical parameters as such

$$\begin{aligned} &\hat{\theta}_1 = -(k_{01} + k_{21}) \quad \hat{\theta}_2 = k_{12} \quad \hat{\theta}_3 = k_{21} \\ &\hat{\theta}_4 = -(k_{12} + k_{32}) \quad \hat{\theta}_5 = k_{23} \quad \hat{\theta}_6 = k_{32} \quad \hat{\theta}_7 = f \end{aligned} \tag{6-23}$$

Regarding the identifiability of the model: there are 7 parameters, a quantity larger than $n(p+m) = 6$. Having less parameters than $n(p+m)$ is a necessary condition; therefore the structure involving the parameters is not identifiable [10]. As opposed to the identifiable case it now takes 30 seconds to find the estimate:

$$\mathbb{A}(\hat{\theta}) = \begin{bmatrix} -0.3732 & 0.2024 & 0 \\ 0.2732 & -0.5753 & 0.3015 \\ 0 & 0.3827 & -0.3015 \end{bmatrix} \quad \mathbb{B}(\hat{\theta}) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \mathbb{C}(\hat{\theta}) = \begin{bmatrix} 0.1500 & 0.1500 & 0.1500 \end{bmatrix} \tag{6-24}$$

and the similarity transformation matrices

$$\mathcal{T} = \begin{bmatrix} 7.2953 & -5.6330 & -7.2749 \\ 4.2219 & 1.7436 & -5.2115 \\ 4.9706 & 3.6115 & -3,6827 \end{bmatrix} \qquad \mathcal{S} = \begin{bmatrix} 0.1223 & -0.4637 & 0.4146 \\ -0.1025 & 0.0917 & 0.0728 \\ 0.0645 & -0.5360 & 0.3594 \end{bmatrix} \tag{6-25}$$

which is indeed a similarity transformation pair, as can be validated by computing $\mathbb{TS}$. The matrix triplet $(\mathbb{A}(\hat{\theta}), \mathbb{B}(\hat{\theta}), \mathbb{C}(\hat{\theta}))$ can be validated by its frequency response, which is identical to the responses shown in Figure 6-2. Because of the identifiability problem stressed earlier, the values of the estimated parameters are not equal to those of the system. We can bypass this problem by determining a unique similarity transformation Q such that $Q\mathbb{A}(\hat{\theta})Q^{-1}$ satisfies the structure of $A(\theta)$ and the constraint $\hat{\theta}_2 + \hat{\theta}_4 + \hat{\theta}_6 = 0$ is verified, as the system does. Note that right from the start, such an analytic approach is possible. We would then apply the analytic procedure to the unstructured matrix triple $(\mathbb{A}, \mathbb{B}, \mathbb{C})$. This is however not straightforward and further studies are needed to find the analytic solution efficiently. By first the estimating the manifold $\hat{\theta}$ there is a lot more structure to be exploited in the symbolic computation. Furthermore, note that such a procedure is problem specific and requires insight from the user. From the equation $B(\theta) = Q\mathbb{B}(\hat{\theta})$ we know that the structure of Q must be of the form

$$Q = \begin{bmatrix} 1 & q_4 & q_7 \\ 0 & q_5 & q_8 \\ 0 & q_6 & q_9 \end{bmatrix} \tag{6-26}$$

From $QC(\theta) = \mathbb{C}(\hat{\theta})$ we have that each column of Q should sum up to one. Using symbolic computational software, in our case MAPLE, we then perform a symbolic computation of Q, leading to[3]

$$Q = \begin{bmatrix} 1 & -0.09800000000 & 0.0 \\ 0 & 1.098000000 & -0.147747119 \\ 0 & 0.0 & 1.147747119 \end{bmatrix} \tag{6-27}$$

and

$$Q\mathbb{A}(\theta)Q^{-1} = \begin{bmatrix} -0.3999735999999999985 & 0.1999835948998817846 & 0.0 \\ 0.2999736000000000007 & -0.6000225953017306 32 & 0.2500038047037488 89 \\ 0.0 & 0.4000390004019125 37 & -0.2500038046982696 05 \end{bmatrix} \tag{6-28}$$

which is fairly precise if we compare it to the original system (6-17).

6-7 Comparison study

In this section we compare the workings of the proposed method. For comparison purposes, the Alternating Least Squares (ALS)-method [39], the Null-Space based method (NS) [40, 28], and a Gradient Based (GB)-method [28, 46] are simulated alongside the own SCR-method. A detailed derivation of the gradients used in NS are given in [47]. The gradients used for the GB method are derived in Appendix C-2. We will review two cases: 1) the user has no initial information on the parameter vector, hence all the methods will be initialized as stated in their specific papers 2) there is prior information available on the parameter vector. To simulate this: we will corrupt initial guess with varying degrees of noise, to review how well the methods are still able to converge.

[3] Special thanks to Guillaume Mercère (Université de Poitiers) and Olivier Prot (Université de Limoges) for providing the MAPLE code

6-7-1 No a priori knowledge

The first simulation study is to evaluate the success rate of the algorithms vs. the number of unknowns present in the system. We do this by generating a stable Single Input Single Output (SISO) Continuous Time (CT) model[4] of order 5 with entries of the system being

$$\begin{aligned} A^* &= \begin{bmatrix} -3.486 & 3.635 & 3.689 & -3.899 & -1.933 \\ -2.634 & -1.386 & 4.419 & 4.286 & -0.859 \\ -0.874 & -5.314 & -2.373 & 5.46 & 5.883 \\ 1.213 & -0.935 & -6.277 & -3.904 & 5.637 \\ 6.049 & 1.803 & -4.571 & -3.182 & -3.719 \end{bmatrix} \\ B^* &= \begin{bmatrix} -1.636 & 0.173 & 0.828 & 0.217 & -1.909 \end{bmatrix}^T \\ C^\star &= \begin{bmatrix} 0.205 & 1.193 & -0.802 & -1.266 & -0.149 \end{bmatrix} \end{aligned} \tag{6-29}$$

The input data for the algorithms will be the similarity tranformed triplet $(\mathbb{A}, \mathbb{B}, \mathbb{C})$, which have been obtained by balancing[5] the system (6-29). Note that this balancing operation could also have been done after a subspace scheme followed by ZOH, which would have led to the same model. The transformation is merely applied straight away to ensure we have a hundred percent valid transformed initial model.

$$\begin{aligned} \mathbb{A} &= \begin{bmatrix} -0.007884 & 0.9789 & 0.1331 & 0.01995 & 0.05165 \\ -0.9789 & -7.531 & -7.473 & -0.6011 & -1.801 \\ 0.1331 & 7.473 & -5.651 & -8.332 & -6.007 \\ -0.01995 & -0.6011 & 8.332 & -0.09533 & -0.3607 \\ -0.05165 & -1.801 & 6.007 & -0.3607 & -1.585 \end{bmatrix} \\ \mathbb{B} &= \begin{bmatrix} -0.08139 & -1.966 & 0.774 & -0.09207 & -0.2538 \end{bmatrix}^T \\ \mathbb{C} &= \begin{bmatrix} -0.08139 & 1.966 & 0.774 & 0.09207 & 0.2538 \end{bmatrix} \end{aligned} \tag{6-30}$$

The identification of a fixed number of randomly generated entries of the actual system matrices (6-29) will be performed by four algorithms whose details are given below.

- SCR is setup using a maximum number of outer-iterations of 500 nuclear norm iterations and 500 truncated nuclear norm iterations, the number of inner-iterations per outer-iteration is set to 30. The regularization parameter λ is fixed to be 10.
- The ALS-method is setup with a stopping criterion of

$$\frac{\|z^{i+1} - z^i\|}{\|z^i\|} \leq 10^{-6} \tag{6-31}$$

  where z is a vector collecting all the variables and i is the iteration number.
- The NS-method is solved using the MATLAB command `fminunc`, using the analytic gradients as given in [47], the default step tolerance is used of 10^{-6}, together with the default optimallity tolerance 10^{-6}.

[4] MATLABcommand `rss`

[5] MATLABcommand `balreal`

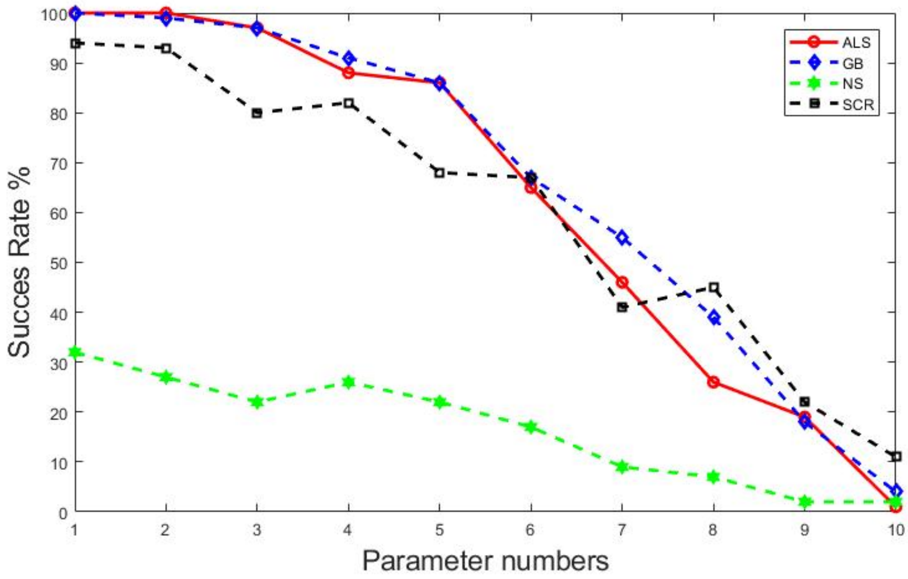


Figure 6-3: Performance comparison of different identifications when there is no prior knowledge

- The GB-method is solved using the MATLABcommand `fminunc` using the analytic gradients as discussed in Appendix C-2. The default step tolerance is used of 10^{-6}, together with the default optimallity tolerance 10^{-6}.

To evaluate the identification performance, the normalized estimation error criterion will be used,

$$\text{NEE} = \frac{\|\theta^{\star} - \theta_{est}\|}{\|\theta^{\star}\|} \tag{6-32}$$

where $\theta^{\star}$ is the true system parameter vector. The number of parameters to be identified run from 1 to 10. Note again that 10 is maximum number of parameters in a system of $n(p+m)$. For each fixed number of parameters 100 experiments are carried out by randomly selecting the matrix entries to identified, and the criterion success rate defined as

$$\sum_{r=1}^{100} \mathbf{1}(\text{NEE}^r \leq 10^{-3})\% \tag{6-33}$$

where $\mathbf{1}(\cdot)$ is an indicator function. Figure 6-3 shows the performance of the four tested methods against the number of system parameters. We notice that:

- SCR is out-performed by ALS and GB for an unknown number of parameters up to 7. From 8 onward SCR performs better. Note that for the an unknown number of parameters of 10, so the maximum number of unknown parameters for this system, SCR performs the other methods with at least a factor two. A possible reason for this is, that the other methods estimate near singular transformation matrices, while SCR still manages to estimate a non-singular T.
- GB and ALS roughly follow the same curve. This is not entirely surprising as they both minimize the same objective function, the difference being ALS does the updates

sequentially and GB simultaneously. The figure shows that for (at least this problem instance) the simultaneous update performs better.

- NS has generally poor performance for this problem instance.

6-7-2 A priori knowledge

In this subsection we test the workings of the proposed algorithm given there exists a priori information on the parameter to be estimated. We will assume that the user has enough affinity with the system that he or she can make an initial guess of the parameter vector. To simulate this we let the initial guess $\hat{\theta}_0$ be a realization of a stochastic process with the true theta as mean and σ_i^2 variance, i.e.

$$\theta_0 = \theta^\star + \epsilon_i \tag{6-34}$$

where $\epsilon_i \in \mathbb{R}^{\mathbf{dim}(\theta)}$ is a stochastic variable with the statistical properties

$$\epsilon \sim \left(0, \sigma_i^2 I_{\mathbf{dim}(\theta)}\right) \tag{6-35}$$

The experiment is then conducted with a multitude of variances σ_i^2 to represent different qualities of the initial guess. The goal of this experiment is to see how the error of the initial guess affects the error of the final estimate.

$$\sigma^2 \in \{0.01, \quad 0.1, \quad 1, \quad 10, \quad 100\} \tag{6-36}$$

Driver Belt For the simulation example we use the driver belt example from [48]. The parameterized state matrices are

$$A = \begin{bmatrix} 0 & 1 & -1 \\ 1 & 0 & 0 \\ \theta_1 & \theta_2 & \theta_2 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \theta_4 \end{bmatrix} \qquad C = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{6-37}$$

As the true parameter vector θ we pick all θ_i's equal

$$\{\theta_1^*, \theta_2^*, \theta_2^*, \theta_3^*\} = \{-10, -10, -10, -10\} \tag{6-38}$$

in order to let the offsets induced by $\sigma^2\epsilon$ have the same effect on all θ_i's. For each variance level σ_i^2 we run the simulation 10 times. The results are depicted in Figure 6-4. We observe the following:

- The errors in ALS appear to be mostly in a narrow band just around 10^{-1} , this could be due to the slow convergence properties of the method in combination with the maximum number of iterations of 10000. Note however all points in this band provide good enough estimates to find the exact minimum, with for instance the GB-method. For higher errors the method fails more often, but still manages to find to optimum in roughly 50% of the cases.

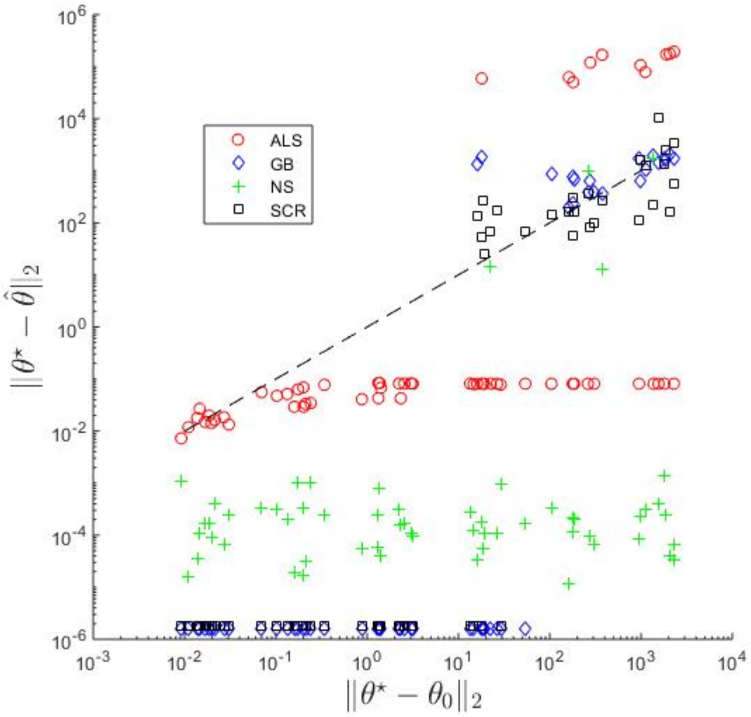


Figure 6-4: Plot depicting the error of the initial guess vs. the error of the estimated θ measured in the 2-norm. The dotted line resembles the equality of the initial error and final error.

- We see that the NS-method as opposed to the other problem instance is very effective. Over almost the entire band of initial errors, the method is able to find the global minimum. Only for large errors (>100) the method fails a few times.
- GB and SCR have similar profiles: they are able to find the minimum for initial errors up to $\sigma_i = 40$, thereafter the initial offset is too large to be able to converge to the global minimum.

6-8 Summary

In this chapter we have proposed a new grey-box identification method that takes a fully parameterized realization of an LTI state-space model and extracts the unknown physical parameters of a same dimensional physical state-space model. The algorithm initially carries out a similarity transformation between the fully parameterized model and the physical model. The resulting bilinear equation is then turned into a rank constraint. This rank constraint is in turn relaxed with help of the nuclear norm. Because the nuclear norm has a tendency to produce sub-optimal results an iterative scheme is used to sequentially come closer to the optimal point. Furthermore, by applying a substitution for the inverse of the similarity transformation and constraining their multiplication to Identity, we are able to invoke solutions which have non-singular transformation matrices.

Chapter 7

Differential Matrix Equation via Similarity Transformation

7-1 Introduction

It is well known that, given a set of second-order differential equations, one can easily construct a corresponding set of first-order equations in quite a straightforward fashion. The inverse problem on the other hand is much more complex. In this chapter we extend the similarity transformation approach developed in the previous chapter to Matrix Differential Equation (MDE)'s. The problem we asses is the following: given a black-box state space representation identified from a 2nd order model, and given the parameterization of the mass-,damper-,stiffness-matrices, can we extract the true parameter vector?

Outline

The chapter is organized as follows: §7-2 briefly reviews the MDE and especially its connection to State Space (SS)-models. §7-3 formulates the problem statement. §7-4 reviews the literature concerning the identification of MDE's. In §7-5 we provide a new method to estimate parameters in a (partly-)parameterized MDE. §7-6 demonstrates the performance of the proposed identification method applied to some physical examples. In §7-7 we describe some possible future work, followed by some conclusions in §7-8.

7-2 Preliminaries

We will assume that the input matrix F and the output matrix H are known. The internal dynamics presented by the coefficients in the mass matrix M, damping matrix V, stiffness matrix K, may have 0 or more unknown values to us, leading to

$$\begin{aligned} M(\theta)\ddot{x}(t) + V(\theta)\dot{x}(t) + K(\theta)x(t) &= Fu(t) \\ y(t) &= Hx(t) \end{aligned} \tag{7-1}$$

where $M(\theta), V(\theta), K(\theta) \in \mathbb{S}^n_{++}$. By defining $q(t) = \begin{bmatrix} x(t)^T & \dot{x}(t)^T \end{bmatrix}^T$, we can write (7-1) into state space form

$$\begin{aligned} \dot{q}(t) &= \underbrace{\begin{bmatrix} 0 & I \\ -M(\theta)^{-1}K(\theta) & -M(\theta)^{-1}V(\theta) \end{bmatrix}}_{A} q(t) + \underbrace{\begin{bmatrix} 0 \\ M(\theta)^{-1}F \end{bmatrix}}_{B} u(t) \\ y(t) &= \underbrace{\begin{bmatrix} H & 0 \end{bmatrix}}_{C} q(t) \end{aligned} \tag{7-2}$$

7-3 Problem Formulation

Consider a parameterized state space model as given in Eq. (3-6), the problem of interest is then given as

problem of interest Given the triple of matrices (A_T, B_T, C_T) in an unknown sate basis, determine the parameter vector θ satisfying;

$$C_T(sI - A_T)^{-1}B_T = H(s^2M(\theta) + sV(\theta) + K(\theta))^{-1}F \tag{7-3}$$

7-4 Overview

In [49] a continuous time subspace technique is proposed to estimate the M, V, K matrices. The proposed method involves a two step procedure, initially a recursive scheme based on N4SID is employed to find a structured state space model, namely one of the form (4-3). Hereafter a convex optimization is performed to extract the matrix triplet (M, V, K). The convex optimization imposes positive definiteness on these matrices. To this authors knowledge, no further structure is imposed. Also the authors assume full information of the state vector.

The most restrictive formulation is given [50], where the requirement is having as many sensors and actuators as modes. This was improved upon in [51], where the requirement was loosened to having only the number of sensors equal to the number of modes. This in turn was further generalized in [52], requiring that the number of sensors be equal to the number of second order modes. The most permissive formulation is given in [1], The minimum requirement for the methodology is that all Degree Of Freedom (DOF)'s should contain either an actuator or a sensor, with at least one co-located actuator-sensor pair.

The authors [53] avoid having to estimate a similarity transformation by estimating the continuous time parameters directly from a discrete input-output sequence, using an Orthogonal complement approach. The method requires the most restrictive, sensor actuator setup, namely a full set of actuators and sensors. In the same paper a similarity transformation approach is discussed.

7-5 Identification Method

The identification is a two step procedure: 1) the determination of a first order model of the system in continuous time, and 2) the transformation of such an identified model into a second-order model. From general input-output data, it is possible to use subspace techniques [26] to acquire a discrete state space realization. Using the zero-order hold technique [45] it is then possible to transform his description into continuous time, leading to a black-box model of the form

$$\begin{aligned} \dot{z}(t) &= \mathbb{A}z(t) + \mathbb{B}u(t) \\ y(t) &= \mathbb{C}z(t) \end{aligned} \tag{7-4}$$

where now $\mathbb{A} \in \mathbb{R}^{2n\times 2n}, \mathbb{B} \in \mathbb{R}^{2n\times m}, \mathbb{C} \in \mathbb{R}^{p\times 2n}$. If the first-order model of (7-4) was identified using data actually originating from the second-order model of (7-1), the models represented by (7-2) and (7-4) are different models of the same system. As shown in the previous chapter, two equidimensional models of the same system are related by a non-singular transformation matrix T, such that

$$\begin{aligned} T\mathbb{A} &= AT \\ T\mathbb{B} &= B \\ \mathbb{C} &= CT \end{aligned} \tag{7-5}$$

For the identification of the M, V, K matrices we make a distinction between different cases; the special instance where the input matrix *or* output matrix is full rank, and the general cases where this is not the case. We will see that for the special cases we can solve both problems by solving a set of linear equations.

7-5-1 General Case

Using the special structure of A, B, C, multiplying by the right with T^{-1} and substituting this inverse with S as explained in §6-5-1, we can rewrite (7-5) as

$$ST = I \tag{7-6a}$$

$$TS = I \tag{7-6b}$$

$$T\mathbb{A}S = \begin{bmatrix} 0 & I \\ -M(\theta)^{-1}K(\theta) & -M(\theta)^{-1}V(\theta) \end{bmatrix} \tag{7-6c}$$

$$T\mathbb{B} = \begin{bmatrix} 0 \\ M(\theta)^{-1}F \end{bmatrix} \tag{7-6d}$$

$$\mathbb{C}S = \begin{bmatrix} H & 0 \end{bmatrix} \tag{7-6e}$$

In Equations (7-6c) and (7-6d) there still exists the inverse of the parameterized mass matrix, $M(\theta)^{-1}$. This makes that the Equations (7-6c) and (7-6d) are not bilinear. However, instead of using the parameterization of the mass matrix in the optimization procedure, we use the linear parameterization of the inverse of the mass matrix, i.e.

$$M^{-1}(\tau) \quad \text{instead of} \quad M(\theta)^{-1} \tag{7-7}$$

Because $M(\theta)$ is constrained to be positive semidefinite, so is the parameterized inverse $M^{-1}(\tau)$. If $M(\theta)$ has diagonal structure, then we can parameterize $M(\tau)^{-1}$ as a diagonal matrix. If $M(\theta)$ is fully parameterized, so will $M(\tau)^{-1}$, however if $M(\theta)$ contains zeros in some locations i, j it will not be the case in general that $M(\tau)^{-1}$ will contain a zero at i, j or even a zero in general. In that case it is needed to use a full parameterization of $M(\tau)^{-1}$. Using this new parameterization (7-6c) reads as

$$T\mathbb{A}S = \begin{bmatrix} 0 & I \\ -M^{-1}(\tau)K(\tau) & -M^{-1}(\tau)V(\tau) \end{bmatrix} \tag{7-8}$$

which still contains bilinearities on both sides of the equation, however it is possible to rewrite (7-8) to a single bilinearity, namely

$$\begin{bmatrix} T & 0 \\ & M^{-1}(\tau) \end{bmatrix} \begin{bmatrix} \mathbb{A} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} & S \\ K(\tau) & V(\tau) \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \tag{7-9}$$

The final identification/feasibility problem reads as

$$\begin{aligned}
\text{find} \quad & \tau, T, S \\
\text{subject to} \quad & \\
\begin{bmatrix} T & 0 \\ & M^{-1}(\tau) \end{bmatrix} \begin{bmatrix} \mathbb{A} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} & S \\ K(\tau) & V(\tau) \end{bmatrix} &= \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \\
ST &= I \\
TS &= I \\
T\mathbb{B} &= \begin{bmatrix} 0 \\ M^{-1}(\tau)F \end{bmatrix} \\
\mathbb{C}S &= \begin{bmatrix} H & 0 \end{bmatrix} \\
M^{-1}(\tau) &\succeq 0 \\
V(\tau) &\succeq 0 \\
K(\tau) &\succeq 0
\end{aligned} \tag{7-10}$$

Once we obtain the solution τ, we determine $M(\theta)$ by computing the inverse of $M^{-1}(\tau)$.

7-5-2 Special Cases

We identify two special cases: the case when the input matrix or the output matrix is of full rank. By expanding the equations $T\mathbb{A} = A(\theta)T, \quad T\mathbb{B} = B, \quad \mathbb{C} = CT$, a lot of additional structure can be perceived. By partitioning the matrices as

$$T = \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix} \quad \mathbb{A} = \begin{bmatrix} \mathbb{A}_1 & \mathbb{A}_2 \\ \mathbb{A}_3 & \mathbb{A}_4 \end{bmatrix} \quad \mathbb{B} = \begin{bmatrix} \mathbb{B}_1 \\ \mathbb{B}_2 \end{bmatrix} \quad \mathbb{C} = \begin{bmatrix} \mathbb{C}_1 & \mathbb{C}_2 \end{bmatrix} \quad , \tag{7-11}$$

the expanded equations read as

$$T_1\mathbb{A}_1 + T_2\mathbb{A}_3 = T_3 \tag{7-12a}$$

$$T_1\mathbb{A}_2 + T_2\mathbb{A}_4 = T_4 \tag{7-12b}$$

$$T_3\mathbb{A}_1 + T_4\mathbb{A}_3 = -M(\theta)^{-1}K(\theta)T_1 - M(\theta)^{-1}V(\theta)T_3 \tag{7-12c}$$

$$T_3\mathbb{A}_2 + T_4\mathbb{A}_4 = -M(\theta)^{-1}K(\theta)T_2 - M(\theta)^{-1}V(\theta)T_4 \tag{7-12d}$$

$$T_1\mathbb{B}_1 + T_2\mathbb{B}_2 = 0 \tag{7-12e}$$

$$T_3\mathbb{B}_1 + T_4\mathbb{B}_2 = M(\theta)^{-1}F \tag{7-12f}$$

$$\mathbb{C}_1 = HT_1 \tag{7-12g}$$

$$\mathbb{C}_2 = HT_2 \tag{7-12h}$$

Full set of sensors

In the case that we have as many independent sensors as there are states within the system, we have that matrices $H, \mathbb{C}_1, \mathbb{C}_2$ are invertible in (7-12), and the solution follows rather trivially. We can determine T_1 and T_2 from (7-12g) and (7-12h). T_3 and T_4 then follow from (7-12a) and (7-12b). Having obtained the similarity transformation matrix, the mass matrix now follows from (7-12f), $V(\theta), K(\theta)$ can now be extracted from a Least Squares (LS) solution applied to (7-12c) and (7-12d).

Full set of actuators

In the event that there exist as many actuators as there are states, the solution follows somewhat less trivial, but can still be obtained by solving sets of (over-)determined linear equations. The basis of the solution depends on writing all sub-matrices of the transformation matrix as a function of T_1. We have that in the full set of actuator case, that $F, \mathbb{B}_1, \mathbb{B}_2$ are invertible. From (7-12a,7-12b,7-12f) we can write

$$\begin{aligned} T_2 &= -T_1\mathbb{B}_1\mathbb{B}_2^{-1} \\ T_3 &= T_1(\mathbb{A}_1 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_3) \\ T_4 &= T_1(\mathbb{A}_2 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_4) \end{aligned} \tag{7-13}$$

Using the expression for T_3, T_4 and (7-12f) we can express the inverse of the mass matrix as

$$M(\theta)^{-1} = T_1\left((\mathbb{A}_1 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_3)\mathbb{B}_1 + (\mathbb{A}_2 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_4)\mathbb{B}_2\right)F^{-1} = T_1Q \tag{7-14}$$

Combining all expressions in (7-12c,7-12d) leads till

$$\begin{aligned} \cancel{T_1}(\mathbb{A}_1 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_3)\mathbb{A}_1 + \cancel{T_1}(\mathbb{A}_2 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_4)\mathbb{A}_3 &= -\cancel{T_1}QK(\theta)T_1 - \cancel{T_1}QV(\theta)T_1(\mathbb{A}_1 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_3) \\ \cancel{T_1}(\mathbb{A}_1 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_3)\mathbb{A}_2 + \cancel{T_1}(\mathbb{A}_2 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_4)\mathbb{A}_4 &= \cancel{T_1}QK(\theta)T_1\mathbb{B}_1\mathbb{B}_2^{-1} - \cancel{T_1}QV(\theta)T_1(\mathbb{A}_2 - \mathbb{B}_1\mathbb{B}_2^{-1}\mathbb{A}_4) \end{aligned} \tag{7-15}$$

If we now apply the substitution $X = K(\theta)T_1$ and $Y = V(\theta)T_1$, we can uniquely determine the $2n^2$ unknowns in X and Y from the $2n^2$ linear equations in Eq. (7-15). Having obtained X and Y, we would now like to obtain $V(\theta)$,$K(\theta), T_1$ which can be done by solving the following set of equations

$$XT_1^{-1} = K(\theta) \tag{7-16a}$$

$$YT_1^{-1} = V(\theta) \tag{7-16b}$$

$$\mathbb{C}_1T_1^{-1} = H \tag{7-16c}$$

Note that we would like to solve for n^2 variables in T_1^{-1} together with $n(n+1)/2$ variables in $V(\theta)$ and $n(n+1)/2$ in $K(\theta)$ making a total of $2n^2 + n$ variables. If the number of outputs $p = 1$, then (7-16) has $2n^2 + n$ equations, leading to a unique solution. If $p > 1$ then (7-16) is easily transformed into a LS. $M(\theta)$ hereafter follows from (7-14).

Recap

The entire optimization procedure is summarized in the following table:

Algorithm 7.1 *Estimate 2nd order Matrix differential equation through Similarity Transformation*

```
given IO data $u(k), y(k)$ for $i = 1 \dots N$
do
    1. Estimate discrete time state space model using subspace technique
    2. Convert model to continuous time using Zero Order Hold (ZOH)
    3. Estimate the Mass, Damping, Stiffness matrix using optimization (7-10).
```

7-6 Simulation

In order to verify the proposed method we simulate a few examples, mostly taken from other papers. All results were obtained by using MOSEK 8.1 [19] in conjunction with Matlab and YALMIP [54].

7-6-1 Mass Damper

The first simulation example is a two-dimensional spring-mass-damper system as depicted in Figure 7-1, the example is much like the one treated in [49]. Such a setup leads to a non-diagonal mass-matrix in the MDE. The physical parameters are chosen as follows

$$M = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 2 & -0.5 \\ 0 & -0.5 & 1 \end{bmatrix} \quad V = \begin{bmatrix} 5 & -0.1 & -0.1 \\ -0.1 & 0.2 & -0.1 \\ -0.1 & -0.1 & 1.1 \end{bmatrix} \quad K = \begin{bmatrix} 4 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 22 \end{bmatrix}$$
$$H = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad F = \begin{bmatrix} 1 & 1 \\ -1 & 1 \\ 0 & 0 \end{bmatrix} \tag{7-17}$$

Leading to a frequency response as given in Figure 7-2. The method proposed in [49], uses a full rank H matrix, Identity to precise, and furthermore the success of the their method depends on the diagonal structure of the input matrix F. Hence we opt to test the algorithm on a non-diagonal input matrix, and rank-deficient output matrix. The system is simulated with a continuous time state space model with parameter matrices

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -0.6452 & 0.0645 & -1.1613 & -0.4323 & 0.0032 & -0.0355 \\ 0.7742 & -1.6774 & -5.8065 & -0.1613 & -0.0839 & -0.2774 \\ 0.3871 & 1.1613 & -24.9032 & 0.0194 & 0.0581 & -1.2387 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.0968 & 0.3548 \\ -0.5161 & 0.7742 \\ -0.2581 & 0.3871 \end{bmatrix}$$
$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{7-18}$$

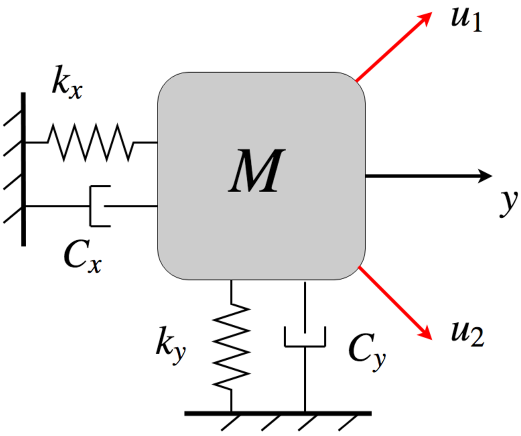


Figure 7-1: Schematic depiction of a spring-mass-damper system in two dimensions. The problem discussed in (7-17) has 3 dimensions, so one should imagine an additional set of spring and damper C_z, k_z lateral to the paper. The inputs u_1, u_2 are applied diagonally to the mass in the xy-plane, the outputs are measured in the y and z direction.

The sampling time used in the simulation equals $t_s = 0.001$ and $N = 5000$ data points were used. The $\mathbb{A}, \mathbb{B}, \mathbb{C}$ are estimated using N4SID[26] followed by ZOH, and lead to

$$\mathbb{A} = \begin{bmatrix} -2.0055 & 0.5081 & 0.8634 & 3.2443 & -0.8011 & -2.1785 \\ -0.2879 & -1.1741 & -1.0625 & -1.5843 & 1.3832 & 0.5346 \\ -1.1301 & 0.65188 & 0.2561 & 0.6697 & -0.55112 & -1.5559 \\ -3.2937 & 1.2658 & -0.2243 & 0.4310 & 0.8212 & 1.7693 \\ 0.8924 & -1.2032 & 0.33622 & -1.0434 & 0.0521 & 0.0940 \\ 2.2319 & -0.4929 & 1.4974 & -1.7734 & -0.0800 & 0.0080 \end{bmatrix} \quad \mathbb{B} = \begin{bmatrix} -0.2150 & 0.6992 \\ 0.4815 & 0.2971 \\ 0.0740 & -0.2028 \\ 0.1094 & -0.2258 \\ -0.0036 & 0.1005 \\ -0.0254 & 0.0105 \end{bmatrix}$$
$$\mathbb{C} = \begin{bmatrix} -0.3118 & 0.0339 & -0.5499 & -0.6235 & -0.3418 & -0.9562 \\ -0.0527 & 0.0077 & -0.0682 & -0.0844 & 0.0136 & 0.0283 \end{bmatrix} \quad \mathbb{D} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{7-19}$$

The $M^{-1}(\tau), V(\tau), K(\tau)$ matrices are parameterized as symmetric matrices, i.e.

$$M^{-1}(\tau) = \begin{bmatrix} \tau_1 & \tau_4 & \tau_6 \\ \tau_4 & \tau_2 & \tau_5 \\ \tau_6 & \tau_5 & \tau_3 \end{bmatrix} \quad V(\tau) = \begin{bmatrix} \tau_7 & \tau_{10} & \tau_{12} \\ \tau_{10} & \tau_8 & \tau_{11} \\ \tau_{12} & \tau_{11} & \tau_9 \end{bmatrix} \quad K(\tau) = \begin{bmatrix} \tau_{13} & \tau_{16} & \tau_{18} \\ \tau_{16} & \tau_{14} & \tau_{17} \\ \tau_{18} & \tau_{17} & \tau_{15} \end{bmatrix} \tag{7-20}$$

together with the 36 variables in T and 36 variables in S, makes a total of 87 decision variables. After running the proposed algorithm with the regularization parameter $\lambda = 10$, we get

$$M^{-1}(\tau) = \begin{bmatrix} 0.2273 & 0.1288 & 0.0645 \\ 0.1288 & 0.6447 & 0.3255 \\ 0.0645 & 0.3255 & 1.1612 \end{bmatrix} \quad V(\tau) = \begin{bmatrix} 4.9664 & -0.0984 & -0.0996 \\ -0.0984 & 0.1998 & -0.1002 \\ -0.0996 & -0.1002 & 1.0992 \end{bmatrix}$$
$$K(\tau) = \begin{bmatrix} 3.9614 & -1.9816 & -0.0000 \\ -1.9816 & 3.9989 & -2.008 \\ -0.000 & -2.008 & 22.0007 \end{bmatrix} \tag{7-21}$$

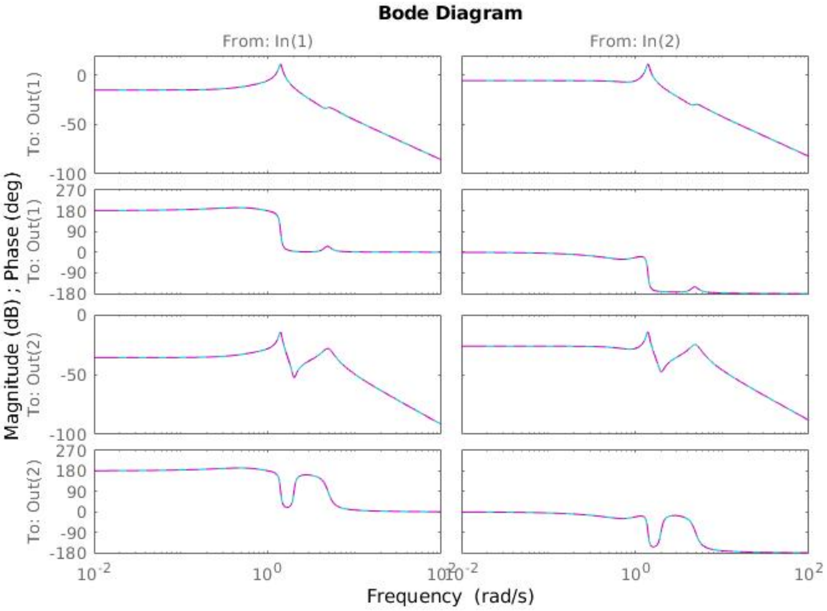


Figure 7-2: Bode diagram of the three-dimensional spring-mass-damper system. In blue the original system, in purple the final estimate using using the estimated M, V, K matrices.

We have that $V(\theta) = V(\tau), K(\theta) = K(\tau)$ and that the estimate of the mass matrix is obtained by inverting $M^{-1}(\tau)$, leading to

$$M(\theta) = \left(M^{-1}(\tau)\right)^{-1} = \begin{bmatrix} 4.9612 & -0.9912 & -0.002 \\ -0.9912 & 1.9994 & -0.5003 \\ -0.002 & -0.5003 & 1.001 \end{bmatrix} \tag{7-22}$$

By comparing (7-21) and (7-22) with (7-17), we see there exists a slight discrepancy between the actual physical parameters and the estimated physical parameters. This discrepancy is probably not due to numerical issues as the error is much higher than machine precision. The problem, probably lies in the subspace estimation step using N4SID. If we compare the original model (A, B, C) with the estimated $(\mathbb{A}, \mathbb{B}, \mathbb{C})$ triplet, and compare the frequency response of both models it can be observed that the maximum absolute difference in magnitude of both models also lies around 10^{-2}. Furthermore a convergence plot is given in Figure 7-3.

7-7 Future Work

In this section we give some Ideas to further improve the methodology discussed in this chapter.

7-7-1 Structure

In §7-5-2 we derived alternative solutions for special cases: if the input or output matrix is full rank there exists a linear solution. We came up with this solution by expanding out

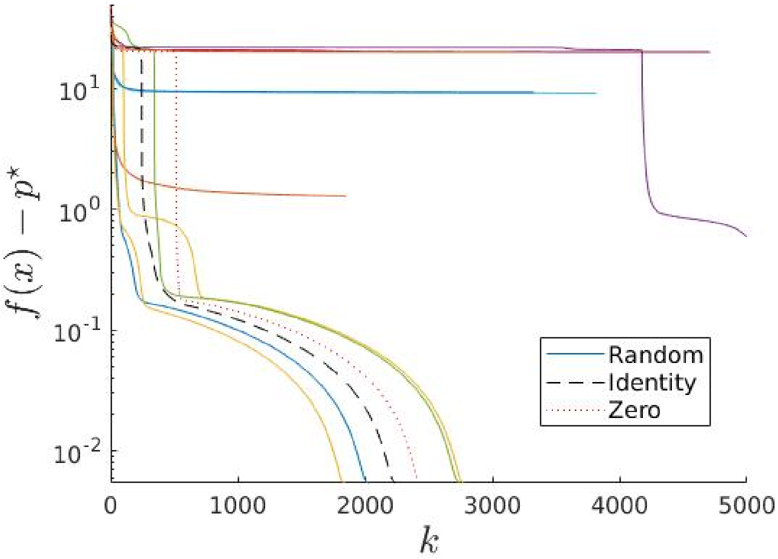


Figure 7-3: Error versus iteration k for different initial conditions, all continuous lines depict initializations where the entries in x are taken from the normal distribution, the dashed line is the case where S, T are initialized as the Identity matrix. The dotted line: S, T as the zero matrix. $p^\star$ is equal to $\|\mathbb{A}\|_\star + \lambda 6n$ and is obtained when bilinear equality holds in the relaxation of (7-10). Note that the error, given convergence is still around 10^{-2}, which is rather large. Also note the trace which after 4000 iterations still manages to converge.

the equations in $T\mathbb{A} = A(\theta)T, \quad T\mathbb{B} = B, \quad \mathbb{C} = CT$. By doing so, we saw there exist a lot of linear dependencies in the equation, also within the transformation matrix. Equations (7-12a) and (7-12b) reflect these internal dependencies. However, concerning the general case: we have not used these inter-dependencies, although they could in effect be used to halve the number of decision variables within T. Maybe one could use the equations (7-12) as a starting point to create a smaller scale optimization problem in the number of decision variables. Alternatively one could, by using the approach discussed in [55] straightforwardly transform the triplet $(\mathbb{A}, \mathbb{B}, \mathbb{C})$ into a triplet $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ with structure

$$\mathcal{A} = \begin{bmatrix} 0 & I \\ \mathcal{A}_3 & \mathcal{A}_4 \end{bmatrix} \qquad \mathcal{B} = \begin{bmatrix} 0 \\ \mathcal{B}_2 \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} \mathcal{C}_1 & 0 \end{bmatrix} \tag{7-23}$$

where $\mathcal{A}_2, \mathcal{A}_3 \in \mathbb{R}^{n \times n}, \mathcal{C}_1 \in \mathbb{R}^{p \times n}, \mathcal{B}_2 \in \mathbb{R}^{n \times m}$ are fully parameterized unstructured matrices. And continue from there. This would mean (slightly abusing notation) that $\mathbb{B}_1, \mathbb{C}_2, \mathbb{A}_1$ equal the zero matrix in (7-12). The combination of equations (7-12e) with (7-12h) then provide $p \times n + n \times m$ equations to solve for $n \times n$ unknowns in T_2. Assuming that $p + m \geq n$, this would be enough to uniquely determine T_2. These reductions combined make it possible to have n^2 variables in T instead of $(2n)^2$, meaning a quadratic reduction in the number of decision variables within the transformation matrix.

7-7-2 Linear Mass Matrix parameterization

During this chapter we have used the linear parameterization of the inverse mass matrix $M^{-1}(\tau)$. This is not a problem when we don't have any prior information on the entries in M. However, if we do have prior knowledge, this parameterization makes it hard to incorporate this knowledge into this matrix, unless the mass matrix is known to be diagonal.

7-7-3 Measurements

The methodology discussed in this chapter focused on position measurements exclusively, but is easily extended to allow for velocity measurements. This is done by substituting the the output matrix in (7-2) by

$$C = \begin{bmatrix} 0 & H \end{bmatrix} \tag{7-24}$$

However, future work needs to be done to allow for acceleration measurements. Possibly inspiration can be found in [1],[51], which contain methods that can also deal with acceleration and mixed type of measurements.

7-8 Summary

In this chapter we have presented an algorithm for identifying the physical parameters in physical structures. This problem has adequately been solved with non-iterative schemes for cases where there exists at least one co-located sensor-actuator pair and the number of sensors and actuators complies $p + m > n$. On the other hand, the methodology we propose allows to incorporate prior information into the matrices. By exploiting structure of the physical system at hand or using known a piori constants in the matrices M, V, K we can determine the unknown values in these matrices using less sensors and actuators. Furthermore, in the chapter we have given two special cases, for which we do not need to apply bilinear techniques. When we have as many sensors or as many actuators as there are states, the solution can be obtained by solving a number of over-determined linear programs.

Given the general case, so no full set of actuators or sensors, the method uses the fact that the identification problem can be cast into a bilinear feasibility problem. This bilinear problem can be turned into a rank constraint and relaxed using the Sequential Convex Relaxation (SCR) technique. It was presumed that this convex relaxation technique has better characteristics being able to converge to the global optimum compared to other techniques, such as Newton-based methods. However, further tests are needed to validate this for this problem instance.

Chapter 8

Conclusion and Discussion

In this chapter we asses if we have reached the goals we set ourselves, and to give possible further research directions.

Goal

In the preface of this thesis we set ourselves the goal,

> *To develop new algorithms for the purpose of identifying continuous-time systems.*

In order to do so, we looked at the applicability of Sequential Convex Relaxation (SCR) in the system identification setting, which had not previously been done. We explored different approaches to extract the unknown parameter of parameterized models; one, where we directly used Frequency Response Function (FRF)-data; and one where we used a black-box representation as input data. We did so for state space models and 2nd order models alike. It is generally the case in grey-box estimation that the associated optimization is non-convex. This means that local-optimality does not imply global optimality. The quest in the identification community is, despite this, to find algorithms which do have a high probability of finding the global optimal solution. It was - at an early stage - supposed that SCR could be an effective means to meet this end. In §2-5, we saw that the method has the ability to escape local minima and this gave the first incentive (although this property is not guaranteed). In the previous chapters we have compared our own proposed algorithms with other (state-of-the-art) methods, to validate this initial expectation.

Results

In Chapter 3 we saw that the problem of identifying a state space model from the FRF - is indeed better solvable by transforming the transfer function into a set of bilinear equations and then solving them with SCR. This compared to identifying the model with the Prediction Error Method (PEM).

Chapter 4 we investigated the problem of identifying a parametric 2nd order model using the FRF. We gave an example where we were able to extract the parameter of 2 Degree Of Freedom (DOF) mass damper system. Even the most permissible of existing formations was not able to do such a task, because there was no co-located actuator-sensor pair. We saw that the use of the additional imposed physical structure of system made the difference in successfully extracting the parameter.

In Chapter 5 we introduced a novel method, which is applicable for estimating 2nd order models from the Power Spectral Density (PSD). In this chapter we assumed a full rank C matrix, which is not uncommon in literature for estimating the same system using FRF-data. Using the principles of reciprocity, .i.e. positive definiteness of the physical matrices lead us to an optimization problem with bilinear matrix equalities, and Linear Matrix Inequality (LMI)'s, which can be solved using SCR. Because others have not tempted to address such an identification scheme we could not make a comparison study. However, by exclusively looking at the outcomes of the method applied to a physical example; the results seem promising.

Chapter 6 we for the first time looked at a different identification strategy. This is to say, by estimating a continuous-time black-box representation prior to converting this representation into form such that it fits the grey-box model. This is a much studied problem in literature, so it was interesting to see how the method holds up compared to others.

Chapter 7 again looked into Matrix Differential Equation (MDE)'s. Now we looked into the problem if we could identify such a system

Conclusion

Even though we gave different approaches to tackle the same problem: through the FRF, and through a black-box estimation, it is hard to state that one approach is exclusively better than the other. For example the physical example we gave in Chapter 3, we could identify through the FRF, but not through a similarity transformation approach as discussed in Chapter 6. A possible reason here for is that the actual similarity transformation matrix between the black-box model and grey-box model had a very high condition number.

On the other hand using the similarity transformation approach discussed in Chapter 7 we were able to identify 2nd order models, some of which we could not have identified through the FRF-method from Chapter 4. The FRF - method would not identify the smaller modes of the model and only fit the high peaks. In the similarity transformation approach such problems were not encountered.

Recommendations

In this thesis we assumed a noiseless setting. We could do so because the input data used for the concerning identification schemes was already pre-processed, i.e. the FRF can be obtained almost perfectly, given that the original time series is long enough. The same holds for the black-box models used as input data in Chapter 6 and 7. Done right, these can be obtained from the raw input-output data, even in high noise cases. However, it could be investigated how the methods hold up, if these are not exactly obtained.

Furthermore it is possible to improve on the underlying algorithms we use to solve the iterates in the SCR scheme. For the case there are no inequality constraints, we use Alternating Direction Method of Multipliers (ADMM). However, ADMM is known to be slow, having only linear convergence properties. Many alternative ways exist to solve the nuclear norm minimization problem. If the proposed algorithms were to be applied to larger systems, it could be worthwhile to explore such methods. Also the Semidefinite Programming (SDP) solution used to solve the SCR iterates when dealing with inequality constraints could be improved. We saw in chapter 2, that such problems scale poorly (n^6). It could be an option to look into the implementation of [22] to get better scaling.

Finally, the MDE formulation we gave in Chapter 4 allows us to retrieve parameters from a q'th order MDE. We confined ourselves to 2nd order ones, as most physical processes are not higher than 2nd order. However, there are processes in the world, which are governed by 3rd order differential equations. It would be very interesting to try to identify such a process using the method from Chapter 4.

Appendix A

Mathematical background

In this appendix we give a brief overview of some used Identities used. The treatment is by no means complete.

A-1 Vectorization

Vectorization is often used in combination with the Kronecker product to express matrix multiplication as a linear transformation on matrices. In particular, let $A \in \mathbb{R}^{k \times l}, B \in \mathbb{R}^{l \times m}, C \in \mathbb{R}^{m \times n}$ then

$$\overrightarrow{ABC} = (C^T \otimes A)\overrightarrow{B} \tag{A-1}$$

A-2 Singular Value Thresholding

The Singular Value Thresholding (SVT) was first presented in [24]. Here we discuss the SVT algorithm, it almost entirely an extract from [12]. Consider the minimization of the form

$$\arg\min_X \|X\|_\star + \lambda \|X - Y\|_F^2 \tag{A-2}$$

Let $Y = U_Y \Sigma_Y V_Y^T$ be the singular value decomposition of $Y \in \mathbb{C}^{n \times m}$.

Lemma 2. *The solution $\hat{X}$ to* (A-2) *has singular vectors U_Y and V_Y*

Proof. Let $X = U_X \Sigma V_X^T$ be a singular value decomposition of X. Then

$$\|X\|_\star + \lambda \|X - Y\|_F^2 = \text{trace}(\Sigma_X) + \lambda(\langle X, X\rangle + \langle C, C\rangle - 2\langle X, C\rangle)$$

Using Von Neumann's trace inequality we get

$$\min_X \text{trace}(\Sigma_X) + \lambda(\langle X, X\rangle + \langle Y, Y\rangle - 2\langle X, C\rangle)$$
$$\geq \min_X \text{trace}(\Sigma_X) + \lambda(\langle X, X\rangle + \langle Y, Y\rangle - 2\langle \Sigma_X, \Sigma_Y\rangle)$$

With equality holding true when C and X are simultaneously unitarily diagonizable. The optimal solution to X to (A-2) therefor has the same singular vectors as Y, i.e. $U_X = U_Y, V_X = V_Y$. □

i Denote the singular values of Y in descending order as $\sigma_{Y,1}, \ldots, \sigma_{Y,2n}$, and those of X similarly, Thanks to Lemma 2, A-2 can be simplified to

$$\arg\min_{\sigma_{X,i}} \sum_{i=1}^{n} (\sigma_{X,i} + \lambda \left(\sigma_{X,i} - \sigma_{C,i})^2\right) \tag{A-3}$$

This problem is completely decoupled in $\sigma_{X,i}$ and the optimal solution to A-2 is computed with

$$\sigma_{X,i} = \max\{0, \sigma_{Y,i} - \frac{1}{2\lambda}\}, \qquad i = 1, \ldots, n$$

Appendix B

Least Square Updates

This appendix is concerned with the Least Squares (LS) updates in the Alternating Direction Method of Multipliers (ADMM) algorithm as desribed in §2-4,

B-1 From Frobenius-norm to 2-norm

The x updates in the ADMM implementation are always of them form

$$\begin{aligned}\arg\min_x &\left\| \begin{bmatrix} W_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{C}(x)+\mathcal{A}(x)PR+QP\mathcal{B}(x)+QPR & (\mathcal{A}(x)+Q)P \\ P(\mathcal{B}(x)+R) & P \end{bmatrix} \begin{bmatrix} W_2 & 0 \\ 0 & I\end{bmatrix} - \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22}\end{bmatrix}\right\|_F \\ \text{s.t} \quad & \mathcal{D}(x) = 0\end{aligned} \tag{B-1}$$

where $P, Q, R, W_1, W_2, M_{ij}$ are of appropriate size and $\mathcal{A}(x) : \mathbb{R}^n \to \mathbb{R}^{j\times k}, \mathcal{B}(x) : \mathbb{R}^n \to \mathbb{R}^{l\times m}, \mathcal{C}(x) : \mathbb{R}^n \to \mathbb{R}^{j\times m}, \mathcal{D}(x) : \mathbb{R}^n \to \mathbb{R}^d$ are affine maps, i.e. of the form

$$\mathcal{A}(x) = A_0 + x_1 A_1 + \dots x_n A_n \tag{B-2}$$

similarly for $\mathcal{B}(x), \mathcal{C}(x), \mathcal{D}(x)$. Applying the vec operator on $\mathcal{A}(x)$ leads to

$$\overrightarrow{\mathcal{A}(x)} = Ax + a \tag{B-3}$$

where $x = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^T, A = \begin{bmatrix} \vec{A_1} & \dots & \vec{A_n} \end{bmatrix}, a = \vec{A_0}$. Likewise

$$\begin{aligned}\overrightarrow{\mathcal{B}(x)} &= Bx + b \\ \overrightarrow{\mathcal{C}(x)} &= Cx + c \\ \overrightarrow{\mathcal{D}(x)} &= Dx + d\end{aligned} \tag{B-4}$$

Denote the vectorized versions of M_{ij} in (B-1) by m_{ij}. Using the vectorization formula (A-1), we can define the vectorized residuals in the cost function as

$$\begin{aligned}
\epsilon_{11} &= (W_2^T \otimes W_1)(Cx + c) + ((PRW_2)^T \otimes W_1)(Ax + a) + (W_2^T \otimes W_1QP)(Bx + b) + \overrightarrow{W_1QPRW_2} - m_{11} \\
\epsilon_{12} &= (P^T \otimes W_1)(Ax + a) + \overrightarrow{W_1QP} - m_{12} \\
\epsilon_{21} &= (W_2^T \otimes P)(Bx + b) + \overrightarrow{PRW_2} - m_{21}
\end{aligned} \tag{B-5}$$

The minimization of the Frobenius norm in (B-1) is then equivalent to the minimization of the 2-norm in

$$\begin{aligned}
\arg\min_{x} \quad & \|Fx - h\|_2^2 \\
\text{s.t} \quad & Dx = -d
\end{aligned} \tag{B-6}$$

where

$$F = \begin{bmatrix} ((PRW_2)^T \otimes W_1)A + (W_2^T \otimes W_1QP)B + (W_2^T \otimes W_1)C \\ (P^T \otimes W_1)A \\ (W_2^T \otimes P)B \end{bmatrix} \tag{B-7}$$

and

$$h = \begin{bmatrix} m_{11} - ((PRW_2)^T \otimes W_1)a - (W_2^T \otimes W_1QP)b - (W_2^T \otimes W_1)c - \overrightarrow{W_1QPRW_2} \\ m_{12} - (P^T \otimes W_1)a + \overrightarrow{W_1QP} \\ m_{21} - (W_2^T \otimes P)b + \overrightarrow{PRW_2} \end{bmatrix} \tag{B-8}$$

B-2 Constrained Least Squares

The constrained LS-problem in (B-6), has the optimal solution $\hat{x}$

$$\begin{bmatrix} 2F^TF & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ v \end{bmatrix} = \begin{bmatrix} 2F^Th \\ -d \end{bmatrix} \tag{B-9}$$

which can be derived from the KKT-conditions. The left hand side usually referred to as the KKT-matrix and is invertible if and only if D has independent rows and

$$\begin{bmatrix} F \\ D \end{bmatrix} \tag{B-10}$$

has independent columns [2]. Because the KKT matrix is symmetric, there exist efficient methods to solve the set of linear equations in (B-9). These methods include the $\mathbf{LDL}^T$ factorization of KKT matrix, or if F^TF is non-singular computing a Cholesky factorization of F^TF and using this factorization in a block solve manner, see [2] for more details. All of these methods have computational complexity of $O(n^3)$ to factor the KKT-matrix, and $O(n^2)$ complexity for each subsequent solve. If D is non-existent in (B-9), then the optimal solution $\hat{x}$ reduces to $\hat{x} = F^\dagger h$, where $F^\dagger$ is the pseudo inverse of F.

Appendix C

Other Methods

This appendix is devoted to two of the methods used in the comparison study conducted in §6-7. Both methods solve the cost function

$$F(\theta, T) = \|T\mathbb{A} - A(\theta)T\|_F^2 + \|T\mathbb{B} - B(\theta)\|_F^2 + \|\mathbb{C} - C(\theta)T\|_F^2 \tag{C-1}$$

The difference being that the Alternating Least Squares (ALS)-method solves (C-1) sequentially over T and θ, and that the Gradient Based (GB)-method solves then non-linear problem (C-1) using a quasi-Newton method, simultaneously updating T and θ. For the Null-Space based method (NS)-method the reader is referred to [40].

C-1 Alternating Least Squares

TheALS method as proposed in [39], tries to solve the optimization problem by sequentially minimizing the objective function with respect to the parameter vector θ and the similarity transform matrix T, i.e. sequentially solve

$$\min_T F(\theta, T) \tag{C-2}$$

$$\min_\theta F(\theta, T) \tag{C-3}$$

until some stopping criterion holds. Because θ, T individually appear linearly in the arguments of the frobenius norms in $F(\theta, T)$, we can write both above optimization as standard least square problems. Using the relationship $\|X\|_F^2 = \|\text{vec}(X)\|_2^2$, and

$$\text{vec}(A(\theta)) = K_A\theta + \kappa_A \tag{C-4}$$

$$\text{vec}(B(\theta)) = K_B\theta + \kappa_B \tag{C-5}$$

$$\text{vec}(C(\theta)) = K_C\theta + \kappa_C \tag{C-6}$$

where K_X and κ_X are constant matrices and vectors of appropriate size respectively, we can write

$$\min_T F(\theta, T) = \min_T \left\| \begin{bmatrix} (\mathbb{A}^T \otimes I_n) - (I_n \otimes A(\theta)) \\ \mathbb{B}^T \otimes I_n \\ I_p \otimes C(\theta) \end{bmatrix} \text{vec}(T) - \begin{bmatrix} 0 \\ \text{vec}(B(\theta)) \\ \text{vec}(\mathbb{C}) \end{bmatrix} \right\|_2^2 \tag{C-7}$$

and

$$\min_\theta F(\theta, T) = \min_\theta \left\| \begin{bmatrix} T\mathbb{A} \\ \mathbb{T}\mathbb{B} \\ \mathbb{C} \end{bmatrix} - \begin{bmatrix} (T^T \otimes I_n) & & \\ & I & \\ & & (T^T \otimes I_p) \end{bmatrix} \left(\begin{bmatrix} K_A \\ K_B \\ K_C \end{bmatrix} \theta + \begin{bmatrix} \kappa_A \\ \kappa_B \\ \kappa_C \end{bmatrix} \right) \right\|_2^2 \tag{C-8}$$

C-2 Gradient Based Method

The GB-method [28] aims at finding a local minimum for the cost function $F(\theta, T)$, using a quasi-Broyden-Fletcher-Goldfarb-Shannon (BFGS) method. In order to set up the method, the gradients of the cost function F with respect to θ, and T have to be computed. To compute these gradients, the following relationships are useful [56]

$$\|X\|_F^2 = \mathbf{tr}(X^T X) \tag{C-9}$$

$$\frac{\partial}{\partial X}\mathbf{tr}(AXB) = A^T B^T \qquad \frac{\partial}{\partial X}\mathbf{tr}(AX^T B) = BA \tag{C-10}$$

And the chainrule for matrix derivatives. Let $U = f(X)$, the goal is to find the derivative of $g(U)$ with respect to x_i. As derived in [56] this is equivalent to

$$\frac{\partial g(U)}{\partial x_i} = \mathbf{tr}\left[\left(\frac{\partial g(U)}{\partial U} \right)^T \frac{\partial U}{\partial x_i} \right] \tag{C-11}$$

The gradient of $F(\theta, T)$ with respect to T is found by applying (C-9) followed by (C-10), leading to

$$\begin{aligned} \nabla_T F_A(\theta, T) =& 2\left(T\mathbb{A}\mathbb{A}^T - A(\theta)T\mathbb{A}^T - A^T(\theta)T\mathbb{A} + A^T(\theta)A(\theta)T \right) \\ &+ 2\left(C^T(\theta)C(\theta)T - C^T(\theta)\mathbb{C} \right) + 2\left(T\mathbb{B}^T\mathbb{B} - B^T(\theta)\mathbb{B} \right) \end{aligned} \tag{C-12}$$

The gradient of $F(\theta, T)$ with respect to θ, can be obtained by initially regarding the partial derivative $\partial F(\theta, T)/\partial\theta_i$, namely

$$\frac{\partial F(\theta,T)}{\partial \theta_i} = \mathbf{tr}\left[\left(\frac{\partial F(\theta,T)}{\partial A(\theta)} \right)^T \frac{\partial A(\theta)}{\partial \theta_i} \right] + \mathbf{tr}\left[\left(\frac{\partial F(\theta,T)}{\partial B(\theta)} \right)^T \frac{\partial B(\theta)}{\partial \theta_i} \right] + \mathbf{tr}\left[\left(\frac{\partial F(\theta,T)}{\partial C(\theta)} \right)^T \frac{\partial C(\theta)}{\partial \theta_i} \right] \tag{C-13}$$

The partial derivatives of the parameterized matrices with repsect to θ_i are

$$\frac{\partial}{\partial \theta_i} A(\theta) = \frac{\partial}{\partial \theta_i} \left(A_0 + \theta_1 A_1 + \cdots + \theta_i A_i + \cdots + \theta_l A_l \right) = A_i \tag{C-14}$$

similarly for $\partial B(\theta)/\partial\theta_i$ and $\partial C(\theta)/\partial\theta_i$. Using the partial derivatives we are now ready to set up the gradient, resulting in

$$\nabla_\theta F(\theta, T) = -2 \begin{bmatrix} \mathbf{tr}(M_A^T A_1) \\ \vdots \\ \mathbf{tr}(M_A^T A_l) \end{bmatrix} - 2 \begin{bmatrix} \mathbf{tr}(M_B^T B_1) \\ \vdots \\ \mathbf{tr}(M_B^T B_l) \end{bmatrix} - 2 \begin{bmatrix} \mathbf{tr}(M_C^T C_1) \\ \vdots \\ \mathbf{tr}(M_C^T C_l) \end{bmatrix} \tag{C-15}$$

where

$$M_A = (T\mathbb{A} - A(\theta)T)T^T \tag{C-16}$$

$$M_B = T\mathbb{B} - B(\theta) \tag{C-17}$$

$$M_C = (\mathbb{C} - C(\theta)T)T^T \tag{C-18}$$

Bibliography

[1] M. De Angelis, H. Lus, R. Betti, and R. W. Longman, "Extracting physical parameters of mechanical models from identified state-space representations," *Journal of applied mechanics*, vol. 69, no. 5, pp. 617–625, 2002.

[2] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[3] S. Audoly, L. D'Angio, M. P. Saccomani, and C. Cobelli, "Global identifiability of linear compartmental models-a computer algebra algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 1, pp. 36–47, 1998.

[4] R. Bellman and K. J. Åström, "On structural identifiability," *Mathematical biosciences*, vol. 7, no. 3-4, pp. 329–339, 1970.

[5] C. Cobelli and G. Romanin-Jacur, "On the structural identifiability of biological compartmental systems in a general input-output configuration," *Mathematical Biosciences*, vol. 30, no. 1-2, pp. 139–151, 1976.

[6] C. Cobelli and J. J. Distefano 3rd, "Parameter and structural identifiability concepts and ambiguities: a critical review and analysis," *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, vol. 239, no. 1, pp. R7–R24, 1980.

[7] K. Godfrey and J. DiStefano III, "Identifiability of model parameter," *IFAC Proceedings Volumes*, vol. 18, no. 5, pp. 89–114, 1985.

[8] L. Ljung and T. Glad, "On global identifiability for arbitrary model parametrizations," *Automatica*, vol. 30, no. 2, pp. 265–276, 1994.

[9] E. Walter, *Identifiability of parametric models.* Elsevier, 2014.

[10] L. Ljung, "System identification-theory for the user, prentice hall, upper saddle river n.," *System identification: Theory for the user. 2nd ed. Prentice Hall, Upper Saddle River, NJ.*, 1999.

[11] R. Doelman and M. Verhaegen, "Sequential convex relaxation for convex optimization with bilinear matrix equalities," in *Control Conference (ECC), 2016 European*, pp. 1946–1951, IEEE, 2016.

[12] R. Doelman, H. T. Nguyen, and M. Verhaegen, "Solving large-scale general phase retrieval problems via a sequence of convex relaxations," *arXiv preprint arXiv:1803.02652*, 2018.

[13] R. Doelman and M. Verhaegen, "Sequential convex relaxation for robust static output feedback structured control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15518–15523, 2017.

[14] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[15] M. Fazel, H. Hindi, and S. P. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *American Control Conference, 2001. Proceedings of the 2001*, vol. 6, pp. 4734–4739, IEEE, 2001.

[16] M. Fazel, H. Hindi, and S. P. Boyd, "Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices," in *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 2156–2162, IEEE, 2003.

[17] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE transactions on pattern analysis and machine intelligence*, p. 1, 2012.

[18] R. A. Horn, R. A. Horn, and C. R. Johnson, *Matrix analysis.* Cambridge university press, 1990.

[19] M. ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 8.1.*, 2017.

[20] D. Peaucelle, D. Henrion, Y. Labit, and K. Taitz, *UserâĂŹs guide for sedumi interface 1.04*, 2002.

[21] K. Toh, R. Tütüncü, and M. Todd, "Sdpt3 version 3.02-a matlab software for semidefinite-quadratic-linear programming," *updated in December*, 2002.

[22] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 3, pp. 1235–1256, 2009.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[24] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[25] C. Yu, L. Ljung, and M. Verhaegen, "Identification of structured state-space models," *Automatica*, vol. 90, pp. 54–61, 2018.

[26] M. Verhaegen and V. Verdult, *Filtering and system identification: a least squares approach.* Cambridge university press, 2007.

[27] L. Ljung, "Frequency domain versus time domain methods in system identification–revisited," in *Control of Uncertain Systems: Modelling, Approximation, and Design*, pp. 277–291, Springer, 2006.

[28] G. Mercère, O. Prot, and J. A. Ramos, "Identification of parameterized gray-box state-space systems: From a black-box linear time-invariant representation to a structured one," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2873–2885, 2014.

[29] L. Ljung, *System identification toolbox*, 1988.

[30] P. Verboven, "Frequency-domain system identification for modal analysis," *Vrije Universiteit Brussel, Brussels*, 2002.

[31] M. Friswell and J. E. Mottershead, *Finite element model updating in structural dynamics*, vol. 38. Springer Science & Business Media, 2013.

[32] W. Heylen, S. Lammens, and P. Sas, "Modal analysis theory and testing. katholieke universiteit leuven, faculty of engineering, department of mechanical engineering, division of production engineering," *Machine Design and Automation*, 1998.

[33] B. Cauberghe, "Applied frequency-domain system identification in the field of experimental and operational modal analysis," *Praca doktorska, VUB, Brussel*, 2004.

[34] M. K. Vakilzadeh, V. Yaghoubi, A. T. Johansson, and T. Abrahamsson, "Manifold metropolis adjusted langevin algorithm for high-dimensional bayesian fe model updating in structural dynamics," in *Proceedings of the 9th international conference on structural dynamics, EURODYN*, 2014.

[35] P. Stoica, R. L. Moses, *et al.*, "Spectral analysis of signals," 2005.

[36] F. J. Cara Cañas, J. Carpio Huertas, J. Juan Ruiz, and E. Alarcón Álvarez, "Operational modal analysis using expectation maximization algorithm," 2011.

[37] M. O. Toolbox, "UserâĂŹs guide (2012) the math works," *Inc. www. mathworks. com*, 5.

[38] L. Ljung, *System identification: theory for the user.* Prentice-hall, 1987.

[39] L.-L. Xie and L. Ljung, "Estimate physical parameters by black box modeling," in *Proceedings of the 21st Chinese Control Conference*, pp. 673–677, 2002.

[40] O. Prot, G. Mercère, and J. Ramos, "A null-space-based technique for the estimation of linear-time invariant structured state-space representations," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 191–196, 2012.

[41] T. Kailath, *Linear systems*, vol. 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

[42] P. Parrilo and L. Ljung, *Initialization of physical parameter estimates.* Linköping University Electronic Press, 2003.

[43] M. Bergamasco and M. Lovera, "State space model identification: from unstructured to structured models with an hâĹđ approach," *IFAC Proceedings Volumes*, vol. 46, no. 2, pp. 202–207, 2013.

[44] D. Vizer, G. Mercère, O. Prot, and E. Laroche, “HâĹđ-norm-based optimization for the identification of gray-box lti state-space model parameters,” *Systems & Control Letters*, vol. 92, pp. 34–41, 2016.

[45] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital control of dynamic systems*, vol. 3. Addison-wesley Menlo Park, CA, 1998.

[46] O. Prot, G. Mercère, *et al.*, “Initialization of gradient-based optimization algorithms for the identification of structured state-space models,” in *Proceedings of the IFAC World Congress*, Milan Milan, Italy, 2011.

[47] G. Mercère, J. Ramos, and O. Prot, “Identification of parameterized gray-box state-space systems: from a black-box linear time-invariant representation to a structured one: detailed derivation of the gradients involved in the cost functions,” *ArXiv e-prints*, June 2014.

[48] R. C. Dorf and R. H. Bishop, *Modern control systems*. Pearson, 2011.

[49] B. Olofsson, O. Sornmo, A. Robertsson, and R. Johansson, “Continuous-time gray-box identification of mechanical systems using subspace-based identification methods,” in *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, pp. 327–333, IEEE, 2014.

[50] C.-D. Yang and F.-B. Yeh, “Identification, reduction, and refinement of model parameters by theeigensystem realization algorithm,” *Journal of Guidance, Control, and Dynamics*, vol. 13, no. 6, pp. 1051–1059, 1990.

[51] K. F. Alvin and K. Park, “Second-order structural identification procedure via state-space-based system identification,” *AIAA journal*, vol. 32, no. 2, pp. 397–406, 1994.

[52] D.-H. Tseng, R. W. Longman, and J.-N. Juang, “Identification of the structure of the damping matrix in second order mechanical systems,” *Spaceflight mechanics 1994*, pp. 167–190, 1994.

[53] J. Ramos, G. Mercere, and O. Prot, “Identifying second-order models of mechanical structures in physical coordinates: An orthogonal complement approach,” in *Control Conference (ECC), 2013 European*, pp. 3973–3978, IEEE, 2013.

[54] J. Lofberg, “Yalmip: A toolbox for modeling and optimization in matlab,” in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pp. 284–289, IEEE, 2004.

[55] M. Friswell, S. Garvey, and J. T. Penny, “Extracting second-order systems from state-space representations,” *AIAA journal*, vol. 37, no. 1, pp. 132–135, 1999.

[56] K. Petersen and M. Pedersen, “The matrix cookbook tech. univ. denmark, kongens lyngby,” tech. rep., Denmark, Tech. Rep., Ver, 2012.

Glossary

List of Acronyms

LTI	Linear Time-Invariant
SS	State Space
ODE	Ordinary Differential Equation
PSD	Power Spectral Density
SDP	Semidefinite Programming
FRF	Frequency Response Function
SISO	Single Input Single Output
SI	Subspace Identification
DOF	Degree Of Freedom
SVT	Singular Value Thresholding
LS	Least Squares
ZOH	Zero Order Hold
DCP	Difference of Convex Programming
SCR	Sequential Convex Relaxation
PEM	Prediction-Error Method
NS	Null-Space based method
ADMM	Alternating Direction Method of Multipliers
CT	Continuous Time
MDE	Matrix Differential Equation

BFGS	Broyden-Fletcher-Goldfarb-Shannon
ALS	Alternating Least Squares
GB	Gradient Based
IO	Input Output
LMI	Linear Matrix Inequality
PEM	Prediction Error Method
NLLS	Non-Linear Least Squares

Index

ADMM, 6

Identifiability, 1

Power Spectral Density, 31

Rosenbrock, 8

SDP, 6, 24
Similarity Transformation, 41
State Space, 14, 41