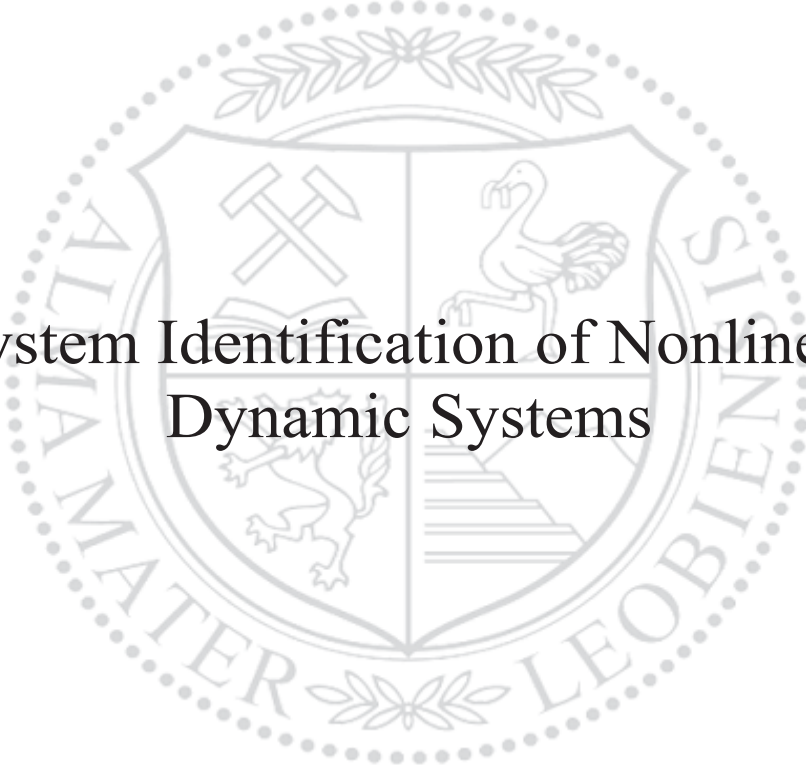




Chair of Automation

Master's Thesis



System Identification of Nonlinear
Dynamic Systems

Stefan Oliver Fladischer, BSc

September 2020



MONTANUNIVERSITÄT LEOBEN

www.unileoben.ac.at

EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich die Richtlinien des Senats der Montanuniversität Leoben zu "Gute wissenschaftliche Praxis" gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 16.09.2020

Fladischer Stefan

Unterschrift Verfasser/in
Stefan Oliver, Fladischer

Acknowledgements

I want to thank Paul O’Leary, the head of the Chair of Automation at the Montanuniversity Leoben, for enabling this work. I’m thankful to my supervisor Matthew Harker for all his support and the helpful discussions. I would like to thank Petra Hirtenlehner for always being helpful with any organisatory matters, and all the other people working at the Chair of Automation for providing a welcoming and friendly working environment. I’m grateful to my parents who always supported me throughout my life, in the pursuit of my education and through this venture.

Abstract

System identification is the experimental modeling of a dynamic system whose parameters or underlying physical principles are not precisely known. In particular, measurement data in the form of input-output data sets can be used to estimate the parameters of a system model. The goal of this work is the application of numerical methods to realize the parametrization of a model such that it predicts the behaviour of a nonlinear dynamic system in an optimal way.

The basis for the applied system identification algorithm is the minimization of the output error of the model. A *goodness-of-fit* criterion, the sum of squared vertical distances between the measurement data points and the simulated model output at these points in time, is to be minimized.

As the model of a nonlinear dynamic system is described by nonlinear differential equations, a numerical solver for the solution of initial value problems in conjunction with a numerical optimization method for the solution of the ensuing nonlinear curve fitting problem are applied in the system identification procedure.

This system identification algorithm is applied to solve a set of example problems: a free falling object that is subject to drag due to air, a nonlinear mass and spring system and a nonlinear dynamic friction model, the LuGre model. Different numerical solutions methods for initial value problems as well as different numerical optimization techniques are applied in the solution of these system identification problems based on synthetic measurement data.

The influence of gaussian measurement noise on the identified parameters as well as the feasibility of utilizing multiple measurement data sets in order to eliminate this disturbance induced variation is investigated. Furthermore, the combination of measurement data sets corresponding to different excitation levels of the object of interest is explored – a procedure that is of special importance in the system identification of nonlinear systems in order to accurately identify all the model parameters.

Kurzfassung

Systemidentifikation ist die experimentelle Modellierung eines dynamischen Systems, dessen Parameter oder grundlegende physikalische Prinzipien nicht genau bekannt sind. Im Speziellen können Messdaten in der Form von Input-Output-Datensätzen dazu verwendet werden die Parameter eines Systemmodells abzuschätzen. Das Ziel dieser Arbeit ist die Anwendung numerischer Methoden, um die Parametrisierung eines Modells zu realisieren, sodass das Verhalten eines nichtlinearen dynamischen Systems in einer optimalen Weise vorausberechnet wird.

Die Basis des angewendeten Systemidentifikationsalgorithmus ist die Minimierung des Outputfehlers des Modells. Ein Anpassungsgütekriterium, die Summe der quadrierten vertikalen Abstände zwischen den Messwerten und dem simulierten Modelloutput zu diesen Zeitpunkten, soll minimiert werden.

Da das Modell eines nichtlinearen dynamischen Systems durch nichtlineare Differentialgleichungen beschrieben wird, wird ein numerisches Lösungsverfahren für die Lösung von Initialwertaufgaben in Verbindung mit einem numerischen Optimierungsverfahren für die Lösung des hervorgehenden nichtlinearen Kurvenanpassungsproblems für die Systemidentifikationsprozedur angewendet.

Dieser Systemidentifikationsalgorithmus wird zur Lösung einiger Beispielprobleme angewendet: Ein Objekt im freien Fall unter Einfluss von Luftwiderstand, ein nichtlineares Feder-Masse-System und ein nichtlineares dynamisches Reibungsmodell, das LuGre-Modell.

Verschiedene numerische Lösungsverfahren für Initialwertaufgaben sowie verschiedene numerische Optimierungsverfahren werden zur Lösung dieser Systemidentifikationsprobleme auf Basis von synthetisierten Messdaten angewendet.

Der Einfluss von normalverteilten Messtörungen auf die identifizierten Parameter sowie die Möglichkeit der Nutzung mehrere Messdatensätze zur Eliminierung dieser störungsinduzierten Variation wird untersucht. Des Weiteren wird die Kombination von Messdatensätzen, die zu unterschiedlichen Anregungslevels des zu untersuchenden Objekts gehören, untersucht – eine Prozedur die von spezieller Bedeutung für die Systemidentifikation von nichtlinearen dynamischen Systemen ist, um alle Modellparameter genau zu identifizieren.

Contents

1	Introduction	1
1.1	Problem Formulation	1
1.2	Systems and Models	2
1.3	System Identification and its Place in Control Theory	3
1.4	Why are Linear Models Preferred over Nonlinear Models?	4
2	System Identification Theory	7
2.1	System Identification as a Part of System Modeling	7
2.2	The System Identification Procedure	8
2.3	Building a Design Model	9
2.3.1	Black-Box Models	9
2.3.2	White- and Grey-Box Models	10
2.4	Experimental Design	11
2.5	The <i>Goodness-Of-Fit</i> Measure	11
2.5.1	Vertical Distance Least Squares Fit of the Model Output	11
2.5.2	Utilization of Multiple Data Sets	12
2.5.3	Weighting of the Model Output	13
2.5.4	Model Uncertainty	13
2.6	Verification	14
3	Numerical Solution Methods	16
3.1	Numerical Solution to the IVP	16
3.1.1	Euler-Integration	16
3.1.2	Higher Order Numerical Integration Methods	17
3.2	Numerical Optimization	18
3.2.1	Topology of the Objective Function	19
3.2.2	Gradient Based Numerical Optimization	26
3.2.3	Numerical Optimization Without Gradient Information	31
3.2.4	Improving the Performance of the Numerical Optimization	32
3.3	Implementation of a System Identification Algorithm in MATLAB®	35

4	Case Studies	38
4.1	Free Fall with Drag due to Air	38
4.1.1	The <i>Riccati</i> Equation	38
4.1.2	System Modeling	39
4.1.3	Experimental Design	40
4.1.4	System Identification Based on One Data Set	42
4.1.5	System Identification Based on Two Data Sets	45
4.1.6	Statistical Analysis	49
4.1.7	System Identification Based on More Than Two Data Sets	52
4.1.8	System Identification of the Model Parameter and the Initial Con- ditions of the Experiment	53
4.2	The <i>Duffing</i> Equation and the Nonlinear Mass and Spring System	56
4.2.1	Simulation Model	57
4.2.2	System Identification Based on Two Data Sets <i>A</i> and <i>B</i>	58
4.2.3	Statistical Analysis – Monte-Carlo Simulation	61
4.2.4	Multiple Starting Point Procedure	65
4.3	Dynamic Friction Models – Stribeck Effect	68
4.3.1	Modeling of Friction	68
4.3.2	The <i>LuGre</i> Friction Model	69
4.3.3	The Simulation Model	70
4.3.4	System Identification of the LuGre Friction Model	78
5	Conclusion	85

List of Figures

1.1	Block diagram model of a system with an input $u(t)$ and an output $y(t)$. . .	3
3.1	Fitting a polynomial to noisy data.	21
3.2	Contour plot of the objective function for a linear curve fitting problem in two parameters ψ_1 and ψ_2	22
3.3	The objective function for the nonlinear curve fitting problem in one parameter for $y(x) = \sin(\psi x)$	23
3.4	The objective function for the nonlinear curve fitting problem in two parameters for $y(x) = \psi_1 \sin(\psi_2 x)$	24
3.5	The objective function for the nonlinear curve fitting problem in one parameter for $y(x) = e^{\psi x}$	25
3.6	The objective function for the nonlinear curve fitting problem in two parameters for $y(x) = \psi_1 e^{\psi_2 x}$	26
4.1	Local minima of the objective function for the free fall experiment A. . .	43
4.2	Position data and estimate of the object in free fall with drag due to air resistance.	44
4.3	Velocity estimate of the object in free fall with drag due to air resistance. .	45
4.4	Local minima of the objective functions for the free fall experiment A and B.	48
4.5	Estimated position trajectories for the free fall experiment based in the data sets A and B.	48
4.6	Estimated velocity trajectories for the free fall experiment based in the data sets A and B.	49
4.7	Cumulative distribution function of the results of the Monte-Carlo simulation for the free fall experiment based on one and two data sets A and B.	51
4.8	Histogram of the results of the Monte-Carlo simulation for the free fall experiment based on one data set A.	51
4.9	Histogram of the results of the Monte-Carlo simulation for the free fall experiment based on two data sets A and B.	52

4.10	Expected standard deviation due to disturbance induced variation of the parameter ψ when combining multiple data sets. The results are based on a Monte-Carlo simulation.	53
4.11	Contour plot of the objective function $\varepsilon(\psi)$ evaluated for a two dimensional slice of the parameter space: $\psi_1 \in [0, 1]$, $\psi_2 \in [25, 35]$ and $\psi_3 = 0$. .	55
4.12	Contour plot of the objective function $\varepsilon(\psi)$ evaluated for a two dimensional slice of the parameter space: $\psi_1 = 0.5$, $\psi_2 \in [25, 35]$ and $\psi_3 \in [-5, 5]$. .	55
4.13	Spring characteristic for a linear, a hardening and a softening spring. . . .	57
4.14	System inputs $u_A(t)$ and $u_B(t)$ for the experiments A and B simulated on the mass and hardening spring system.	59
4.15	Comparison of the estimated system outputs of experiment A based on the results ψ_A , ψ_A and ψ_{AB} of the system identification.	60
4.16	Comparison of the estimated system outputs of experiment B based on the results ψ_A , ψ_A and ψ_{AB} of the system identification.	61
4.17	Comparison of the simulation output of experiment A corresponding to all data sets ψ_A , ψ_A and ψ_{AB} identified in the Monte-Carlo simulation. . .	62
4.18	Comparison of the simulation output of experiment B corresponding to all data sets ψ_A , ψ_A and ψ_{AB} identified in the Monte-Carlo simulation. . .	62
4.19	Comparison of the results of the Monte-Carlo simulation in the form of histograms for the identified parameter k based on the identification data generated from experiment A and B as well as both of them combined. . .	63
4.20	Histogram of the identified parameter k_{AB} after the removal of outliers. . .	64
4.21	Comparison of the system response to the verification test case with input $u_V(t)$. The system is parameterized according to the selected results $\psi_{A,s}$, $\psi_{B,s}$ and $\psi_{AB,s}$ as well as ψ_{exact}	65
4.22	Comparison of the objective function values $\varepsilon(\psi_{AB})$ for each identified parameter set ψ_{AB} based on a different initial guess that lead to a convergence of the numerical optimization.	67
4.23	Comparison of the simulation output of experiment A and B for all 682 identified parameter sets corresponding to $\varepsilon(\psi) < \varepsilon_{thresh}$ compared to the identification data.	68
4.24	Triangular signal $u(t)$ that is applied to the LuGre friction model in order to analyse its properties.	71
4.25	System response $x(t)$ to the triangular input signal $u(t)$ at smaller excitation levels.	73
4.26	System response $x(t)$ to the triangular input signal $u(t)$ at larger excitation levels.	73
4.27	$(x(t), F_f(t))$ characteristic for smaller excitation levels $u(t) < \mu_0 mg$	74
4.28	$(x(t), F_f(t))$ characteristic for higher excitation levels $u(t) > \mu_0 mg$	74

4.29	$(v(t), F_f(t))$ characteristic for smaller excitation levels $u(t) < \mu_0 mg$	75
4.30	$(v(t), F_f(t))$ characteristic for higher excitation levels $u(t) > \mu_0 mg$	75
4.31	Numerically approximated Jacobian for a triangular input signal $u(t)$ with $\hat{u} = 0.5\mu_0 mg$	77
4.32	Selected components of the numerically approximated Jacobian for a triangular input signal $u(t)$ with $\hat{u} = 0.5\mu_0 mg$	77
4.33	Numerically approximated Jacobian for a triangular input signal $u(t)$ with $\hat{u} = 1.5\mu_0 mg$	78
4.34	Triangular input signal $u_A(t)$ with $\hat{u} = 0.5\mu_0 mg$ that is applied to the Lu-Gre friction model on order to generate identification data.	79
4.35	Identification data set generated by adding noise to the sampled response of the exactly parameterized system to the input signal $u_A(t)$	80
4.36	Identification data set generated by adding noise to the sampled response of the exactly parametrized system to the input signal $u_B(t)$	81
4.37	Comparison of the system response to the input signal $u_A(t)$ for the exact and the identified parameter sets.	84
4.38	Comparison of the system response to the input signal $u_B(t)$ for the exact and the identified parameter sets.	84

List of Tables

4.1	Results of the system identification based on the data from the free fall experiment <i>A</i>	43
4.2	Results of the system identification based on the data from the free fall experiment <i>A</i> and <i>B</i>	47
4.3	Results of the Monte-Carlo simulation of the system identification of the free fall experiment based on one and two data sets.	50
4.4	Results of the Monte-Carlo simulation of the system identification of the parameter and the initial conditions of free fall experiment.	56
4.5	Results of the Monte-Carlo simulation for the nonlinear mass and spring system after the removal of outliers.	64

Chapter 1

Introduction

1.1 Problem Formulation

System Identification is the determination of the properties of a system based on observations made of the behaviour of that system. For controlled dynamic systems these observations relate to measurement time series corresponding to a specific applied input signal, i.e. input-output data sets. The goal of system identification is finding a parametrization for a simulation model such that this model produces the same input-output behaviour as presented by the given identification data. The main task of this work is the exploration and application of numerical solution methods with the goal of performing the system identification of nonlinear dynamic systems.

As a dynamic system is typically described by a model consisting of differential equations, the task of system identification is the determination of the parameters of that simulation model.

The first part of the solution process is the implementation of numerical solution methods for these differential equations. The simulation output of an arbitrarily parametrized model is the solution of an initial values problem. Under the correct parametrization this should be reflecting the measurement data set extracted from an experiment conducted on the object of interest. The initial values of that experimental setup can be assumed to be known based on the selected experiment design or estimated in the identification procedure.

To be able to evaluate the fit of the model to the system an optimization criterion, i.e. a *goodness-of-fit* measure, needs to be selected. Starting from an initial guess for the parameters, the second part of the solution process is the application of numerical optimization methods. To improve the fit of the model, the parametrization is changed in an iterative manner, optimizing the goodness-of-fit measure step by step.

Once this process converges onto a final parametrization of the model, the last step of the system identification procedure is the verification of the results. The estimated model

output can be compared to the measured data set. This can be done by visual comparison or limited statistical analysis. In addition to that, separate validation data sets from different experiments can be used to compare the performance of the model under different operating conditions or the identification data can be split up in a process called *cross-validation*.

When dealing with measurement signals perturbed by noise it is of interest to look at the influence the noise has on the identified model parameters. The influence of *gaussian* noise in the data sets as well as the utilization of multiple data sets in order to minimize the influence of noise will be explored.

1.2 Systems and Models

The terms system and model are used in various disciplines. In general the term system refers to a real or imaginary structure consisting of smaller parts or subsystems. The behaviour of the system is defined by the interactions and relations in between those parts. A more comprehensive definition of a system can be found in [1]: the *CESM* model, which defines a system as a structured object consisting of a composition, environment, structure and mechanism.

A model of a system is an artificial representation of that system created for some purpose. The purpose can be as simple as merely being a simplified or more concise depiction of the system or a portrayal of the fundamental working principles of the system.

When models are used to perform experiments on systems that would be too dangerous, expensive or time-consuming to be performed on the real counterpart, or simply impossible to be performed as the real counterpart doesn't exist, we speak of simulation models [2]. Models can be built in various ways. A model can be formulated verbally, as a graphical model or a physical model in form of a real and possibly miniature replication. The feasibility to perform a simulation on a model and extracting meaningful or quantitative results out of it are dependent on the kind of model and various other factors, nonetheless the accuracy with which the model is able to predict the behaviour of the represented system, and the experimental design.

The idea of quantifying features brings us to mathematical models. Assuming we want to build a simulation model, the mathematical model needs to consist of a description of the state of the system and an output signal $y(t)$, in order to be able to extract information out of it. To attribute the observed output of the model to its inner workings, we assign so called *state variables* $x(t)$, which describe the state of the system at any point in time. In many cases a system will not be fully self contained. There will be interactions with the environment. These external influences are realized as input signals $u(t)$, which might be control inputs that a user or operator can define and change at will, or known or unknown disturbances which can't be influenced but that need to be taken into account when de-



Figure 1.1: Block diagram model of a system with an input $u(t)$ and an output $y(t)$.

describing the behaviour of the system. A simple block diagram model of a system S can be seen in Fig. 1.1.

The main difference between a static and a dynamic system is that for a static system the output only depends on the current state and input of the system. In that sense a static system can be described by a simple algebraic equation:

$$y(t) = f(u(t)). \quad (1.1)$$

For a dynamic system the observed output is dependent on the past history of the input, i.e. for any sequence of the input that is applied to the system over time, the system evolves in a different way. The tool that allows us to encode the effect that the state of the system and the input to the system at a certain point in time have on the change of the trajectory is the differential equation. The internal mechanisms of the system are defined by differential equations describing the interaction of the internal properties in between themselves as well as their interaction with the external influences. Higher order differential equations can be transformed into first order ones by assigning additional state variables. This results in the *state space representation* of the dynamic system as a system of first order differential equations [3]:

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (1.2)$$

The output variables $\mathbf{y}(t)$ are the selected properties of the system that we are interested in. They represent the measurements that we are making. The outputs $\mathbf{y}(t)$ at any point in time are related to the external inputs $\mathbf{u}(t)$ at that time as well as the internal state of the system $\mathbf{x}(t)$ at that time. This means that once the state space equation is solved and we know the trajectory of the state variables for the time period that we are interested in, the measurement variables are derived from a regular equation, the *observer equation*

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (1.3)$$

1.3 System Identification and its Place in Control Theory

The three main and integral parts of a controlled system, as seen in Fig. 1.1 are the behaviour of the system S , the control input $u(t)$, and the system output $y(t)$, whereas the behaviour of the system is described by the relations of the internal state variables $\mathbf{x}(t)$.

Having knowledge about any two of these three attributes of a system results in one of three distinguished kinds of problems of determining the third property [4].

The first kind of task is the *simulation problem*. Given a system S and a control input $u(t)$, find the output $y(t)$. As a *forward problem*, this is the simplest kind of the three types of problems that were hinted at above. In general, there exists one unique solution that is the result of solving an initial value problem (IVP) for the system of differential equations describing the system S from the initial state of the system $x(t_0)$ at the initial time t_0 over the time frame of the defined control input $u(t)$.

The second kind of problem is the *control problem*. Given a system S and a desired system output, i.e. behaviour, $y(t)$, find the input $u(t)$. This is an *inverse problem* that may not have a single unique solution. Depending on the exact problem statement, it will result in one or multiple correct solutions. In general these kind of problems are more loosely formulated in respect to the requirements on the desired behaviour of the system. In many cases it is demanded that a system is brought to a specific state, but the specific path to take and the exact time frame in which to achieve this are not exactly specified. In fact there are infinitely many possible paths and corresponding control sequences that move a system from one state to another [5]. In many cases there are also compromises to be made. For example in a classical PID controlled system faster rise time needs to be weighted against the ensuing overshoot and oscillations.

The third and last kind of problem is the *system identification problem*. Given an input $u(t)$ and an output $y(t)$ of a system, find a system model S that is able to describe this observed input-output behaviour. This again is an inverse problem that results in many possible solutions. The quality of the identified model can be assessed in two ways. On the one hand the ability to represent or interpolate the given input-output data, i.e. the goodness-of-fit, is the determining factor in the solution of the system identification problem. On the other hand the suitability of the identified model to be able to make accurate predictions for different operating conditions when applied in a simulation problem might be of great importance. Depending on the intended application of the model, shortcomings in this suitability for extrapolation would rule out models regardless of the goodness of fit to the given identification data.

1.4 Why are Linear Models Preferred over Nonlinear Models?

This work is aimed at the system identification of nonlinear dynamic systems. This makes it necessary to deal with the difficulties that come with nonlinear systems compared to linear ones. Therefore, in this section some of the differences and the available methods to deal with linear and nonlinear dynamic systems are discussed.

The main argument as to why one should always favour modeling a system as a linear model rather than a nonlinear one is that linear models are naturally easier to deal with. As per definition linear systems have a more narrowly defined general structure

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \quad (1.4)$$

compared to a general nonlinear model

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (1.5)$$

Figuratively, if the space of all system models was a sea, the space of linear systems would just be a little island. Because of both the simplicity and the generality of the linear system, the focus of research has always been weighted on this side. Furthermore, many nonlinear problems can only be approached by local approximation of the nonlinear problem by a linear one. Adding to that, naturally, any approach to deal with some type of problem may work on simple tasks but break down on more complicated ones.

This has lead to a lot of different ways to analyse and work with linear systems. For once it is possible to look at linear systems in the time and frequency domain. This is enabled via the *Laplace*- and the \mathcal{Z} -transformation. The Laplace transform enables to solution of a certain class of partial differential equations, which correspond to single input linear time invariant dynamic systems:

$$a_n \frac{\partial^n y}{\partial t^n} + \dots + a_1 \frac{\partial y}{\partial t} + a_0 y(t) = b_m \frac{\partial^m u}{\partial t^m} + \dots + b_1 \frac{\partial u}{\partial t} + b_0 u(t) \quad (1.6)$$

as algebraic equations in the complex s domain:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{a_n s^n + \dots + a_1 s + a_0}{b_m s^m + \dots + b_1 s + b_0}. \quad (1.7)$$

Note that here the initial conditions are set to be zero. The transfer function $H(s)$ is based on the Laplace transform of the impulse response of the system. Any input $u(t)$ can be approximated as a sum of discrete impulses. The response of a linear system to the input $u(t)$ is therefore the superposition of the response to each of these impulses. For an infinitesimally fine discretization this corresponds to a convolution integral in the time domain

$$y(t) = h(t) * u(t), \quad (1.8)$$

where $h(t)$ is the response of the system to a unit impulse. In the s domain, the convolution simplifies to a simple multiplication:

$$Y(s) = H(s)U(s). \quad (1.9)$$

For nonlinear dynamic systems this is less applicable. The input-output behaviour of a nonlinear dynamic system can not simply be classified by an impulse response like in linear systems as the principle of superposition does not apply [6].

There are various different solution methods available to solve the IVP for a linear system. Besides finding an analytic solution, which is arguably easier for linear systems, any linear or nonlinear vector field defined as in equation (1.4) or (1.5) can be numerically integrated. The simplest algorithm to numerically integrate the solution of the IVP is the *Euler* integration. More advanced algorithms like *Runge-Kutta* type methods apply an iterative approach with varying step size in order to get more accurate results. Another approach is to discretize the state variables in time and to apply a numerical differentiation matrix in order to approximate the discretized derivatives of each state variable. For linear systems this transforms the system of linear differential equations into a system of linear equations, which can be solved directly as a least squares problem with equality constraints, which enforce the initial conditions [7]. It is therefore called the *Global Least Squares* solution method. The power of this solution method lies in the fact that it can also be used for the solution of the inverse problem [8]. As was recently shown, the Global Least Squares solution method in conjunction with the *variable projection method* can be used in the time domain system identification of linear state-space models [9].

The Global Least Squares solution method can also be applied to the solution of the IVP for a systems of nonlinear differential equations, but this results in a system of nonlinear equations and the least squares fit of which needs to be optimized in an iterative manner, relying on good initial estimates of the solution [10].

For homogeneous i.e. unforced systems of linear differential equations it is also possible to deploy the *exponential matrix*, which is based on the analytic solution of a linear system, to compute the state of the system at any point in time [11].

When general nonlinear models are considered, they are often represented as a kind of subclass called hybrid models, e.g. a *Wiener* model, which consists of a linear system followed by a static nonlinearity, or a *Hammerstein* model, which consists of a nonlinear actuator that feeds into a linear system [12]. This approach comes with the advantage that techniques for the analysis of linear systems can be applied for the analysis of the respective part of the hybrid model.

To summarize, while there is a whole array of methods available to deal with linear dynamic systems, the general approach to solve the simulation problem for a nonlinear dynamic systems is an approximation of the solution via numerical integration. Furthermore, the fact that the input-output behaviour of a nonlinear dynamic system cannot be classified by a single impulse response has consequences on the selection of identification data for the system identification problem of nonlinear dynamic systems.

Chapter 2

System Identification Theory

In this chapter some of the core principles of the system identification theory are discussed. These include an overview of the system identification procedure, the selection of a model structure whose parameters are to be identified, the experimental design that is used in order to extract information about the object of interest, the goodness-of-fit criterion that is based on the output error of the model, and the verification of the estimated model parametrization.

2.1 System Identification as a Part of System Modeling

A dynamic system evolving over time is described by a mathematical model in the form of input variables $\mathbf{u}(t)$, internal state variables $\mathbf{x}(t)$ and output variables $\mathbf{y}(t)$ as well as a description of the relationships between the inputs, the internal state and the outputs [2]. The task of system modeling in terms of building a mathematical model is the determination of the input variables $\mathbf{u}(t)$, output variables $\mathbf{y}(t)$ and the necessary number of state variables $\mathbf{x}(t)$ to describe the relations between the inputs and outputs.

In theoretical or physical modeling, so called *first principles*, i.e. physical laws, are used to describe the behaviour of a system. For example the motion of an object of mass subject to forces and gravity can be described by Newton's laws. Ohm's law and Kirchhoff's laws can be used to describe the current in an electrical circuit. Larger systems can be built as the combination of smaller subsystems whose behaviours are known. Sometimes this is also called *mechanistic* modeling and can be performed supported by software like Modelica or Simscape [13].

System identification is commonly used in conjunction with physical modeling or as a complement thereof [2]. This means that physical modeling alone doesn't lead to a working simulation model which actually represents a specific real world counterpart. In most cases precise knowledge about a number of parameters ψ is required in order for the simulation model to match the behaviour of the modeled system. Such a model can be

described as

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\boldsymbol{\psi})\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\psi})\mathbf{u}(t) \quad (2.1)$$

$$\mathbf{y}(t) = \mathbf{C}(\boldsymbol{\psi})\mathbf{x}(t) + \mathbf{D}(\boldsymbol{\psi})\mathbf{u}(t) \quad (2.2)$$

for a linear time invariant (LTI) system, or as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\psi}, t) \quad (2.3)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\psi}, t) \quad (2.4)$$

for a general, presumably nonlinear system in state space form. The determination of the parameters $\boldsymbol{\psi}$ for any given model structure is the system identification problem.

2.2 The System Identification Procedure

System identification is a kind of modeling, which relies on observations from the system in order to determine its properties. As such it is also called data driven modeling or experimental modeling. To be more precise, not necessarily the properties of the system but the properties of the system model are determined. Information about the system is extracted in the form of input-output data sets $[\mathbf{u}(t), \mathbf{y}(t)]$. This is done by the means of performing experiments on the object or process of interest. For a controlled dynamic system a defined control sequence $\mathbf{u}(t)$ is applied to the system and the output $\mathbf{y}(t)$ is measured.

This information is used to determine the parameters of a design model. Depending on the amount of a priori information that is used in the determination of the structure of the design model, so called *black-box* models or *universal approximators* can be distinguished from so called *white-box* or *gray-box* models [13].

Once input-output data is available and a design model is decided upon, the system identification problem is reduced to the task of fitting the output of the model for the given input sequence to the output data of the system. For nonlinear dynamic systems this is a nonlinear curve fitting problem, which can be formulated as the optimization of a goodness-of-fit criterion.

2.3 Building a Design Model

2.3.1 Black-Box Models

One area of system identification involves the modeling of a processes or systems using little if any a priori knowledge. The basis of black-box models are general mathematical model structures which can be scaled in their differentiation order and therefore also in their number of parameters. For example, this could be a general LTI system model in the form of a state-space or transfer function representation, or a nonlinear model in the form of a Wiener or Hammerstein model. In general, higher order models are more flexible in the way that they can be more accurately fitted to any arbitrary data. This is similar to the task of fitting a polynomial of varying degree to data. n data points can be approximated in a vertical distance least squares sense by a polynomial of degree $d < n - 1$, but a polynomial of degree $d \geq n - 1$ can fit the n data points exactly.

This process of determining, based in input-output observations, to which class of systems a specific one belongs is also called *system classification* [14]. With access to different model structures of varying order, a process of trial and error can be deployed. Various model structures of various differentiation order are fitted to the observation data in order to make a selection on the best fitting one. Special attention must be paid to this differentiation order. Ideally, the order of the model is set as high as necessary to accomplish sufficient flexibility to be able to accurately fit the input-output data sets while keeping it as low as possible to avoid a potential over fitting.

The advantage to this approach is clear. The underlying physical principles of the system aren't needed to be able to build a model that can replicate the input-output behaviour of the system.

The main disadvantage of black-box models lies in the name and coincides with the property of inverse problems. The solution of the system identification problem is not uniquely defined. Black-box models can be used to accurately describe the relation of the input and the output of a dynamic system, but the internal mechanisms of the model might be different to the ones in the real system. Therefore, even if a model is found that correctly depicts the system behaviour, it might still be difficult to assign meaning to the identified parameters.

Another disadvantage of black-box models is that in their role of universal approximators these general model structures are not necessarily designed to fit a specific system. In order to fit different input-output behaviours, black-box models are typically over parametrized, i.e. they have more flexibility than might be needed for any given task. This does not mean that a black-box model is less able to form a good simulation model, but that it might be more difficult to distinguish necessary and unnecessary parameters in the process of system identification i.e. it might be more difficult to successfully execute a numerical optimization of the larger range of parameters [15].

In summary it can be said that the task of system identification of black-box models requires less information about the object of interest and less work leading up to the process of system identification but also results in a system model that gives less insight into the system that is to be identified.

2.3.2 White- and Grey-Box Models

So called white-box or gray-box models are the result of physical or theoretical modeling [2]. They represent a system model as described by (2.3) with unknown parameters ψ that are to be determined. According to [13] different shades of grey are used to denominate the degree of disparity from a clear-white model, which would be a model based purely on physical laws where all parameters are known. Darker shades reflect the presence of unknown physical constants which are parameters that are to be determined by system identification or a model that is not purely based on physical laws but that also might include empirical models and simplified assumptions as an abstraction to the real system. The advantage of using a so called white-box model is that as much a priori knowledge as possible is used in the formulation of the system model. An engineer can investigate the object of interest, map out all the relevant connections, and use first order principles to quantify them. Therefore each state variable and each parameter can be attributed to actual physical properties. This for one makes it easier to get good initial estimates for the task of system identification. The rough range of at least some of the parameters can be estimated and bounds can be placed on them based on physical limitations (e.g. a spring stiffness or a damping coefficient will never be negative). In addition to that it can be easy to distinguish between right and wrong results (local minima). Starting from a sensible initial guess and having knowledge about the implications of the limitations of the system make it easier to get correct results or handle a failure in the convergence to a good result in the system identification procedure.

The main disadvantage of white-box models compared to black-box models is that every relevant interaction needs to be correctly determined and incorporated into the model. Otherwise the systematic error built into the model will be compensated by a wrong parametrization which leads to the best possible fit.

In Summary it can be said that the task of system identification of white-box models requires a lot of information and knowledge about the kind of system under investigation, but also results in a system model that gives full insight into the internal behaviour of the system and that comes with knowledge about the meaning of the identified parameters.

2.4 Experimental Design

In order to perform a system identification, information in the form of observations of the behaviour of the system that is to be investigated needs to be at hand. In order to extract information out of the system, experiments need to be performed. The design of those experiments can have serious effects on the results of the system identification. Especially when dealing with nonlinear systems, which might behave differently depending on the excitation level, an appropriate experimental design needs to be determined. Some parameters that are to be identified might have little influence on the system output in certain operational regimes. This makes the identification difficult, especially when the noise level in the measurement data is larger than the influence that a specific parameter has on the system output, which might result in a large disturbance induced variation of the identified parameter value. This means that, for a single input system, an input sequence $u(t)$, $t \in [t_0, t_f]$, needs to be selected which leads to a proper excitation of the system. In many cases it might be advantageous to consider the combination of multiple different input-output data sets as basis for the system identification. Different input sequences need to be designed accordingly. In order to extract suitable identification data sets, at least the whole operational regime of the system of interest should be considered.

Furthermore, the experimental design has an effect on the computability of the numerical solution of the system identification problem. In cases where the solution of the simulation problem already represents a challenge, the experiment could be designed in a way which eases this. In cases where it is difficult to find a global minimizer for the nonlinear curve fitting problem, a different experimental setup might lead to improvements in respect to less local minima or at least a clearer distinguishability of local and global minima.

2.5 The *Goodness-Of-Fit* Measure

2.5.1 Vertical Distance Least Squares Fit of the Model Output

Once a design model is determined or decided upon, the task of system identification is reduced to fitting the output of that model to the observed measurement data. The execution of an experiment corresponds to the computation of a simulation of the system model. For any parametrization ψ of the model, an initial value problem can be solved, which corresponds to the execution of the mentioned experiment. The observations that were made at defined points in time during the experiment correspond to discrete points in time of the simulation. Therefore, for each measurement ξ_i at time t_i a model output $y(t_i)$ can be computed. Identifying the optimal parameters ψ^* means finding the best fit of the model output \mathbf{y} to the measurements $\boldsymbol{\xi}$. For measurements at fixed points in time

the sum of the squared vertical distances can be used as the goodness-of-fit Measure. This reduces the problem of fitting a function to each point in the given data to the problem of minimizing a scalar valued optimality criterion $\varepsilon(\boldsymbol{\psi})$:

$$\boldsymbol{\psi}^* = \underset{\boldsymbol{\psi}}{\operatorname{argmin}} \varepsilon(\boldsymbol{\psi}). \quad (2.5)$$

$\varepsilon(\boldsymbol{\psi})$ can be defined in the following equivalent ways for n measurement points:

$$\varepsilon(\boldsymbol{\psi}) = \sum_{i=1}^n [\xi(t_i) - y(t_i)]^2 \quad (2.6)$$

$$\varepsilon(\boldsymbol{\psi}) = \|\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})\|_2^2 \quad (2.7)$$

$$\varepsilon(\boldsymbol{\psi}) = [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})]^T [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})] \quad (2.8)$$

2.5.2 Utilization of Multiple Data Sets

When multiple sets of measurements are available for the task of system identification, it makes sense to utilize all of the available information in order to identify the best possible model parametrization. For example two experiments might lead to the measurement data sets $\boldsymbol{\xi}_A$ and $\boldsymbol{\xi}_B$. If both of these correspond to the same experimental setup with the same system input $u(t)$ then the objective functions that described the sum of squares with respect to each one of them are

$$\varepsilon_A(\boldsymbol{\psi}) = \|\boldsymbol{\xi}_A - \mathbf{y}(\boldsymbol{\psi})\|_2^2 \quad \text{and} \quad (2.9)$$

$$\varepsilon_B(\boldsymbol{\psi}) = \|\boldsymbol{\xi}_B - \mathbf{y}(\boldsymbol{\psi})\|_2^2. \quad (2.10)$$

These two objective functions can be minimized simultaneously by combining them into

$$\varepsilon_{AB}(\boldsymbol{\psi}) = \left\| \begin{bmatrix} \boldsymbol{\xi}_A \\ \boldsymbol{\xi}_B \end{bmatrix} - \begin{bmatrix} \mathbf{y}(\boldsymbol{\psi}) \\ \mathbf{y}(\boldsymbol{\psi}) \end{bmatrix} \right\|_2^2. \quad (2.11)$$

When the results of two different experiments are used, there ought to be two different functions $\mathbf{y}_A(\boldsymbol{\psi})$ and $\mathbf{y}_B(\boldsymbol{\psi})$ for the model output.

$$\varepsilon_{AB}(\boldsymbol{\psi}) = \left\| \begin{bmatrix} \boldsymbol{\xi}_A \\ \boldsymbol{\xi}_B \end{bmatrix} - \begin{bmatrix} \mathbf{y}_A(\boldsymbol{\psi}) \\ \mathbf{y}_B(\boldsymbol{\psi}) \end{bmatrix} \right\|_2^2 \quad (2.12)$$

For a system with multiple measured output variables a similar approach can be applied.

2.5.3 Weighting of the Model Output

In some problems it can be advantageous to put additional emphasis on specific portions of the system output by adding a weight w_i to the least squares difference at the point in time t_i .

$$\varepsilon(\boldsymbol{\psi}) = \sum_{i=1}^n w_i [\xi(t_i) - y(t_i)]^2 \quad (2.13)$$

In matrix form this can be represented by a weighting matrix W with diagonal entries w_i :

$$\varepsilon(\boldsymbol{\psi}) = \|W(\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi}))\|_2^2. \quad (2.14)$$

The weighting of a specific model output can be especially useful when performing the system identification based on multiple data sets. Separate weights can be asserted on the different data sets in order to balance out the effect that each individual data set has on the overall objective function value.

$$\varepsilon_{AB}(\boldsymbol{\psi}) = \left\| \begin{bmatrix} W_A & 0 \\ 0 & W_B \end{bmatrix} \left(\begin{bmatrix} \boldsymbol{\xi}_A \\ \boldsymbol{\xi}_B \end{bmatrix} - \begin{bmatrix} \mathbf{y}_A(\boldsymbol{\psi}) \\ \mathbf{y}_B(\boldsymbol{\psi}) \end{bmatrix} \right) \right\|_2^2 \quad (2.15)$$

2.5.4 Model Uncertainty

Disturbance Induced Variation

The objective function $\varepsilon(\boldsymbol{\psi})$ is positive definite and would be zero if $\boldsymbol{\xi} = \mathbf{y}(\boldsymbol{\psi})$. In general this equality will never be satisfied because of noise perturbations $\boldsymbol{\delta}$ in the measurement, that can't be reproduced by the model [16].

$$\boldsymbol{\xi} = \mathbf{y}_{\text{exact}} + \boldsymbol{\delta} \neq \mathbf{y}(\boldsymbol{\psi}^*) \quad (2.16)$$

If the parameter set $\boldsymbol{\psi}$ corresponding to the global minimum of equation (2.6) is found, one might expect the value of the objective function to be the norm of the noise $\|\boldsymbol{\delta}\|_2^2$. This is true when speaking about the expected value in its statistical meaning. When performing the system identification based on one set of noisy measurements $\boldsymbol{\xi}_i$ resulting in the parameter set $\boldsymbol{\psi}_i^*$, the estimated model output will not optimally interpolate the exact system output $\mathbf{y}_{\text{exact}}$ but rather try to approximate the measurement noise as good as possible, as it is minimizing the sum of squares of $[\boldsymbol{\xi}_i - \mathbf{y}(\boldsymbol{\psi}_i)]$. For an *unbiased estimator* the expected value is the exact value of the parametrization of the underlying system that produced the observations. However, because of the limited available measurement data, the estimated results will deviate from the exact parametrization.

The important insight to take from this is that

$$\varepsilon = \varepsilon(\xi, \psi) = \varepsilon(\mathbf{y}_{exct} + \delta, \psi) \quad \text{and therefore} \quad (2.17)$$

$$\psi^* = \psi^*(\xi) = \psi^*(\mathbf{y}_{exct}, \delta) \quad (2.18)$$

i.e. any identified parameter set ψ depends on the design model $\mathbf{y}(\psi)$ representing the real system, the experimental design used to extract information about the system represented by \mathbf{y}_{exct} , and the accuracy of the measurement represented by δ .

This brings us to the term *model uncertainty*. As the disturbance in the measurement induces a variation of the objective function which results in a variation of the identified parameters, so does this uncertainty of the parameters affect the output of the model. Model uncertainty can usually be counteracted by increasing the measurement time, as this will decrease the signal to noise ratio resulting in less disturbance induced variation [17]. For gaussian measurement noise the sum of squares optimality criterion is the maximum likelihood estimator [18]. Under this condition and with infinite amounts of identification data, the minimization of $\varepsilon(\psi)$ would result in the exact parameter set. The problem is that in practice there will only ever be a finite amount of identification data available.

Influence of the Numerical Optimization

As ψ^* is the results of a numerical optimization of the curve fitting problem, the applied numerical solution methods also has an effect on this solution. For a well posed curve fitting problem that is optimized with a correctly set up numerical optimization method it can be assumed that the influence of a numerical discretization error on the identified parameter set is at least small compared to a disturbance induced variation if not negligible, as long as the iterative solution method is able to converge to a global minimizer of $\varepsilon(\psi)$.

2.6 Verification

The task of verifying the validity of the model in its purpose of describing the behaviour of the object of interest largely depends on the operational range in which the simulation model is to be used. At a bare minimum the simulation model should be able to reproduce the behaviour observed in the experiments used as a basis for the system identification. This can be verified by comparing the estimated system output of the system model to the identification data sets.

Of course, it would be advantageous if the simulation model would make it possible to accurately predict the behaviour of the system under different operating conditions than presented in the given identification data of the system. Therefore, it is useful to have a

verification data set at hand, which was measured in a separate experiment, that was not included in the identification procedure. The simulated model output under the identified parametrization for this experimental setup can then be compared to the corresponding verification data set in order to evaluate the capability of the simulation model not only to reproduce the observed behaviour presented by the identification data sets but also to be able to predict the behaviour of the object of interest under different operating conditions. In general, this procedure is called *cross-validation*, where only a subset of the available data is used for the estimation and the rest is used for the validation.

Chapter 3

Numerical Solution Methods

In this chapter the two main numerical tools that can be used in the system identification of nonlinear dynamic systems are looked at: The numerical approximation of the solution of initial value problems and the numerical solution of nonlinear curve fitting problems. A selected number of the available algorithms for each of these problems are discussed. At the end of this chapter an overview of one implementation of an algorithm for the system identification of nonlinear dynamic systems is presented.

3.1 Numerical Solution to the IVP

A dynamic system described in state space form is defined as a vector field

$$\dot{\mathbf{x}}(t) = \frac{\partial \mathbf{x}}{\partial t} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t). \quad (3.1)$$

The problem of finding a solution to this system of first order differential equations for a control input $\mathbf{u}(t)$ and an initial state $\mathbf{x}(t = t_0) = \mathbf{x}_0$ is called the initial value problem. For simple systems, this can be done analytically. However, for most nonlinear systems there is no known analytic solution and the best approach to solving them is to develop a numerical approximation which is good enough [19].

Given the inputs $\mathbf{u}(t)$ for $t \in [t_0, t_f]$ and the state $\mathbf{x}(t)$ of the system at time t_0 the solution of the initial value problem for any vector field as defined in equation (3.1) can be approximated by numerical integration.

3.1.1 Euler-Integration

Numerical integration is the iterative extrapolation of the state trajectory $\mathbf{x}(t)$ from an initial point $\mathbf{x}(t_0)$ based on gradient information given by $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)$. For a given $\mathbf{u}(t)$ this simplifies to $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$.

Euler integration is one of the simplest numerical integration methods. It is based on the

approximation of the increment $\Delta \mathbf{x} = \mathbf{x}(t + \Delta t) - \mathbf{x}(t)$ with the help of the derivative $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$. The equation for it can be derived by truncating the *Taylor series expansion* of $\mathbf{x}(t)$ after the first derivative term:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \dot{\mathbf{x}}(t) \quad (3.2)$$

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \mathbf{f}(\mathbf{x}(t), t) \quad (3.3)$$

This is the explicit Euler integration method as the approximation of the state at time $t + \Delta t$ explicitly depends on the state of the system at time t .

By expanding the Taylor series from the point $t + \Delta t$ backwards a slightly different equation for the approximation of the increment can be derived:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \dot{\mathbf{x}}(t + \Delta t) \quad (3.4)$$

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \mathbf{f}(\mathbf{x}(t + \Delta t), t + \Delta t). \quad (3.5)$$

This is an implicit equation in $\mathbf{x}(t + \Delta t)$ and is therefore called the implicit Euler integration formula. It is a more stable numerical integration method but comes with the disadvantage of the computational effort to solve this implicit equation for $\mathbf{x}(t + \Delta t)$ in every step.

On a side note, equation (3.2) corresponds to a forward finite difference formula for the approximation of the derivative:

$$\dot{\mathbf{x}}(t) \approx \frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t} \quad (3.6)$$

and equation (3.4) corresponds to a backwards finite difference formula:

$$\dot{\mathbf{x}}(t_0 + \Delta t) \approx \frac{\mathbf{x}(t_0 + \Delta t) - \mathbf{x}(t_0)}{\Delta t}. \quad (3.7)$$

3.1.2 Higher Order Numerical Integration Methods

Higher order numerical integration methods make use of higher order finite difference formulas and function evaluations at multiple intermediate steps in time. Runge–Kutta methods are one-step multistage methods, which make use of multiple function evaluations for each step Δt in order to achieve a better approximation of the state of the system at time $t + \Delta t$ [20].

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \Delta t \sum_{i=1}^n w_i \mathbf{k}_i \quad (3.8)$$

For example, the fourth order Runge-Kutta method uses four evaluations of $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$ for every step Δt [21]:

$$\mathbf{k}_1 = \Delta t \mathbf{f}(\mathbf{x}(t), t) \quad (3.9)$$

$$\mathbf{k}_2 = \Delta t \mathbf{f}\left(\mathbf{x}(t) + \frac{1}{2}\mathbf{k}_1, t + \frac{1}{2}\Delta t\right) \quad (3.10)$$

$$\mathbf{k}_3 = \Delta t \mathbf{f}\left(\mathbf{x}(t) + \frac{1}{2}\mathbf{k}_2, t + \frac{1}{2}\Delta t\right) \quad (3.11)$$

$$\mathbf{k}_4 = \Delta t \mathbf{f}(\mathbf{x}(t) + \mathbf{k}_3, t + \Delta t) \quad (3.12)$$

The approximation of $\mathbf{x}(t + \Delta t)$ is computed with $w_1 = w_4 = \frac{1}{6}$ and $w_2 = w_3 = \frac{1}{3}$:

$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \frac{1}{6}\mathbf{k}_1 + \frac{1}{3}\mathbf{k}_2 + \frac{1}{3}\mathbf{k}_3 + \frac{1}{6}\mathbf{k}_4. \quad (3.13)$$

The MATLAB® function `ode45()` is an implementation of a Runge Kutta Dormand Prince algorithm [22]. It used two different formulas to estimate $\mathbf{x}(t + \Delta t)$. As the name suggests, these are a fourth and a fifth order Runge-Kutta type formula. By design, it only needs six evaluations of $\mathbf{f}(\mathbf{x}(t), t)$ in order to compute both of these approximations. By comparing the results of the fourth and fifth order algorithm, it is possible to estimate the error due to the discrete step in time Δt , i.e. the local discretization error. This error estimate is used to adapt the step size to smaller values where necessary in order to achieve the demanded numerical accuracy as specified in the solver options or to increase the step size where possible in order to improve the computation time.

3.2 Numerical Optimization

In any curve fitting problem a function $y(x, \psi)$ is to be fitted to a number of N given data points ξ . If the location of the data points respect to the independent variable x is assumed to be exactly known, then only the vertical distance of the function $y(x, \psi)$ evaluated at these location $[x_1, \dots, x_N]$ to those data points ξ needs to be optimized by adjusting the parameters ψ .

$$\xi = \begin{bmatrix} \xi_i \\ \vdots \\ \xi_N \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} y(x_i, \psi) \\ \vdots \\ y(x_N, \psi) \end{bmatrix} = \mathbf{y}(x, \psi) = \mathbf{y}(\psi) \quad (3.14)$$

The fact that this equality cannot be met due to measurement noise leads to the definition of a scalar valued optimality criterion $\varepsilon(\psi)$ as described in section 2.5.

3.2.1 Topology of the Objective Function

Before presenting some of the algorithms that can be used to numerically optimize this objective function, the topology of $\varepsilon(\boldsymbol{\psi})$ for some simple problems is looked at in order to discuss the difference between linear and nonlinear curve fitting problems.

In Linear Curve Fitting

In linear curve fitting, a design function $y(x)$ that is to be fitted to the data is a linear combination of n known basis functions $f_i(x)$.

$$y(x, \boldsymbol{\psi}) = \boldsymbol{\psi}_1 f_1(x) + \cdots + \boldsymbol{\psi}_n f_n(x) = \sum_{i=1}^n \boldsymbol{\psi}_i f_i(x) \quad (3.15)$$

The parameters that are determined in the solution of the linear curve fitting problem are the n linear coefficients

$$\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\psi}_1 & \boldsymbol{\psi}_2 & \cdots & \boldsymbol{\psi}_n \end{bmatrix}^T. \quad (3.16)$$

For N given data points ξ_i at their respective locations x_i with respect to the independent variable x , one optimality criterion that can be used for the fit is the sum of the squared vertical distances between the data points x_i and the design function evaluated at those locations $y(x_i)$, i.e. the *vertical distance least squares fit*. This means that in order to achieve the best fit under the assumption of gaussian measurement noise, the following functional is to be minimized.

$$\varepsilon(\boldsymbol{\psi}) = \sum_{i=1}^N (\xi_i - y(x_i, \boldsymbol{\psi}))^2 \quad (3.17)$$

The same functional can also be formulated as

$$\varepsilon(\boldsymbol{\psi}) = [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})]^T [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})] \quad (3.18)$$

with the discrete values ξ_i of the data that is to be fitted stacked on top of each other in the vector

$$\boldsymbol{\xi} = \begin{bmatrix} \xi(x_1) & \xi(x_2) & \cdots & \xi(x_N) \end{bmatrix}^T \quad (3.19)$$

and the vectorized $y(x_i, \boldsymbol{\psi})$

$$\mathbf{y}(\boldsymbol{\psi}) = \begin{bmatrix} y(x_1, \boldsymbol{\psi}) & y(x_2, \boldsymbol{\psi}) & \cdots & y(x_N, \boldsymbol{\psi}) \end{bmatrix}^T. \quad (3.20)$$

For a linear combination of known basis functions $f_i(x)$, $\mathbf{y}(\boldsymbol{\psi})$ can also be described as the matrix product

$$\mathbf{y}(\boldsymbol{\psi}) = \mathbf{V}\boldsymbol{\psi} \quad (3.21)$$

where V is a matrix similar to a *Vandermonde* matrix with n columns containing the basis functions evaluated at the discrete values of the abscissae x . A Vandermonde matrix typically only consists of evaluated basis functions which are monomials.

$$V = \begin{bmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_N) & \dots & f_n(x_N) \end{bmatrix} \quad (3.22)$$

This means that for the linear curve fitting problem the objective function can be described as

$$\varepsilon(\psi) = [\xi - V\psi]^T [\xi - V\psi]. \quad (3.23)$$

The parameters ψ that lead to the best fit in the vertical distance least squares sense are the ones that minimize this function.

$$\psi^* = \underset{\psi}{\operatorname{argmin}} \varepsilon(\psi) \quad (3.24)$$

By differentiating equation (3.23) with respect to ψ and setting the derivative to zero an extrema can be found. The resulting criteria for this extrema is

$$\xi = V\psi \quad (3.25)$$

which means that the the optimum is found as

$$\psi^* = V^+ \xi \quad (3.26)$$

where V^+ is the *Moore-Penrose* pseudo inverse

$$V^+ = (V^T V)^{-1} V^T. \quad (3.27)$$

The important insight to take from this is that for the linear curve fitting problem an explicit solution can be formulated that directly leads to the optimal solution. As will be seen later, for nonlinear problems this is not the case. Iterative solution methods have to be applied, which – starting from an initial guess – try to minimize the objective function (3.18) step by step. The progress of the iterative solution method is determined by the numerical optimization algorithm as well as by the topology of the objective function.

These iterative solvers can of course also be applied to linear problems. To discuss the topology of the linear curve fitting problem, an example problem is looked at. The design function that is to be fitted to some data is

$$y(x, \psi) = \psi_1 x^2 + \psi_2 x. \quad (3.28)$$

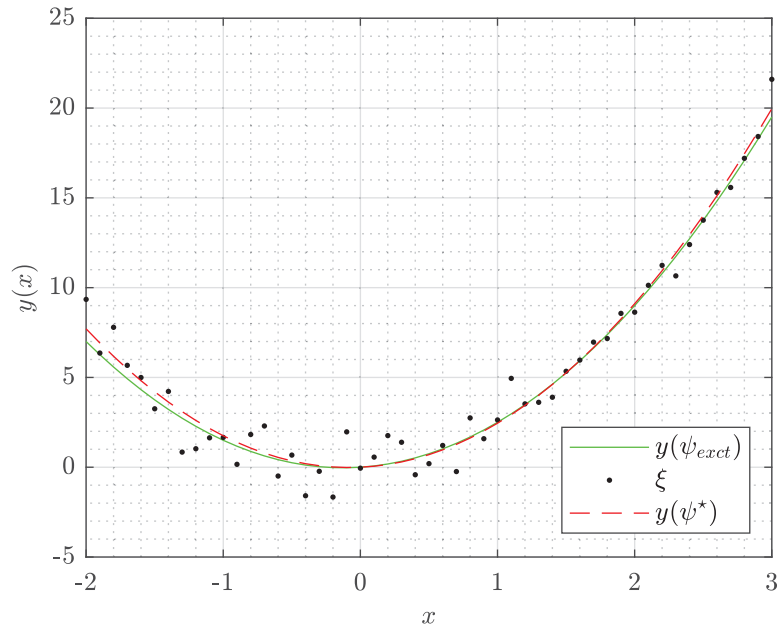


Figure 3.1: Fitting a polynomial to noisy data.

The data ξ is generated by evaluating $y(x, \psi)$ for $\psi_{exact} = [2, \frac{1}{2}]^T$ at 51 evenly spaced points x_i sampled from the interval $x \in [-2, 3]$, i.e. with a step size of $\Delta x = 0.1$. Furthermore, noise δ was added to the measurement data. This noise was sampled from the normal distribution $\mathcal{N}(0, 1)$.

These data points as well as the fitted curve can be seen in Fig. 3.1. As equation (3.28) has two parameters ψ_1 and ψ_2 , the value of the goodness-of-fit criterion $\varepsilon(\psi)$ can be visualized as a contour plot for a region of the two dimensional parameter space. This can be seen in Fig. 3.2 for $\psi_1 \in [0, 5]$ and $\psi_2 \in [-2.5, 2.5]$. As is typical for linear curve fitting problems with linearly independent basis functions, the cost function forms an ellipsoidal valley with one global minimum. This is due to the fact that the objective function $\varepsilon(\psi)$ is the result of squaring linear functions in ψ .

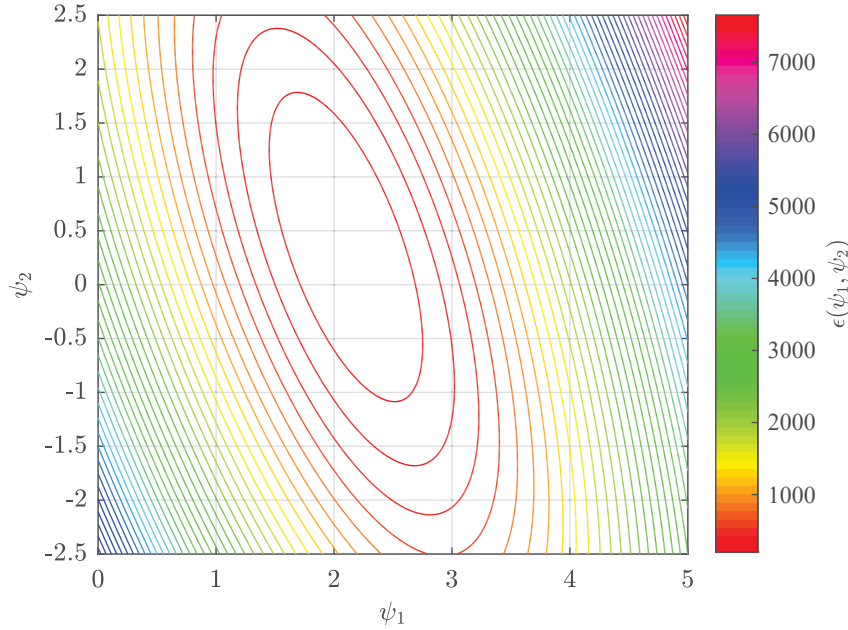


Figure 3.2: Contour plot of the objective function for a linear curve fitting problem in two parameters ψ_1 and ψ_2 .

In Nonlinear Curve Fitting

Any iterative optimization method needs to be initialized with an initial guess ψ_0 . The algorithm will in general compute steps in the parameter space that lead to a decrease in the objective function value $\varepsilon(\psi_0)$. It is quite clear that for a linear curve fitting problem with linearly independent basis functions taking downhill steps starting from any initial point in the parameter space will eventually lead to a convergence towards the one global minimum due to the convex topology of the objective function as visualized in Fig. 3.2. For nonlinear curve fitting problems, this is not always the case. There might exist multiple local minima which make convergence to the global minimum difficult. As an example the function

$$y(x, \psi) = \sin(\psi x) \quad (3.29)$$

is to be looked at. The data ξ that is to be fitted was generated by evaluating $y(x, \psi)$ for $\psi = \psi_{exact} = 3$ at 51 evenly spaced points $x \in [0, 2\pi]$. The objective function $\varepsilon(\psi)$ for this curve fitting problem that can be formulated as in (3.28) evaluated for $\psi \in [-10, 10]$ can be seen in Fig. 3.3. There is clearly one global minimum at $\psi = \psi_{exact}$. But there are also other local minima. This means that any initial guess that is not near the exact value might lead to a wrong result as the numerical optimization will converge into the wrong valley.

Equation (3.29) can be augmented by adding a linear coefficient making it a function with two parameters

$$y(x, \psi) = \psi_1 \sin(\psi_2 x). \quad (3.30)$$

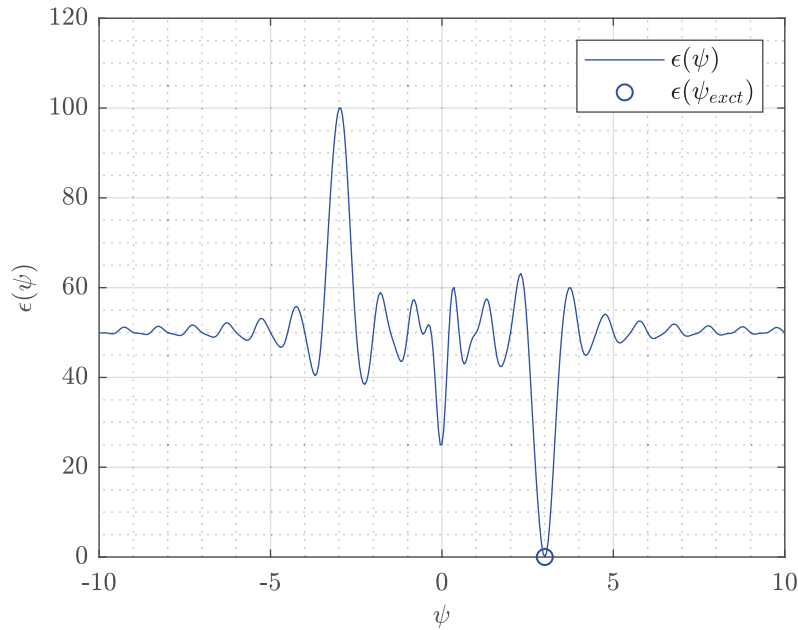


Figure 3.3: The objective function for the nonlinear curve fitting problem in one parameter for $y(x) = \sin(\psi x)$.

Again, a curve fitting problem can be defined based on data points ξ generated by evaluating equation (3.30) for $\psi = \psi_{exact} = [1, 3]$ at 51 evenly spaced points $x \in [0, 2\pi]$. The objective function $\varepsilon(\psi)$ for this example problem is visualized in Fig. 3.4 as a contour plot for $\psi_1 \in [-4, 4]$ and $\psi_2 \in [-5, 5]$. In this case there are actually two global minimum at $\psi = \psi_{exact}$ and at $\psi = -\psi_{exact}$ as sine is an odd function. Making a slice through this contour plot at $\psi_1 = 1$ would result in the two dimensional plot of $\varepsilon(\psi_1 = 1, \psi_2)$ over ψ_2 as in Fig. 3.3.

As can be seen in Fig. 3.4 for one dimension and in Fig. 3.4 for two dimensions, the topology of a nonlinear curve fitting problem might look uncanny on a global scale. But near a minima it actually is shaped similar to a linear curve fitting problem. This is in fact true for any neighbourhood around a specific local point in ψ . Any differentiable function can be approximated by a Taylor series expansion around a local point. This means that a linear curve fitting problem can be defined, using the linearised model derived from a first order Taylor series expansion. The objective function of this linear curve fitting problem will of course look similar to the nonlinear curve fitting problem around the point of linearisation.

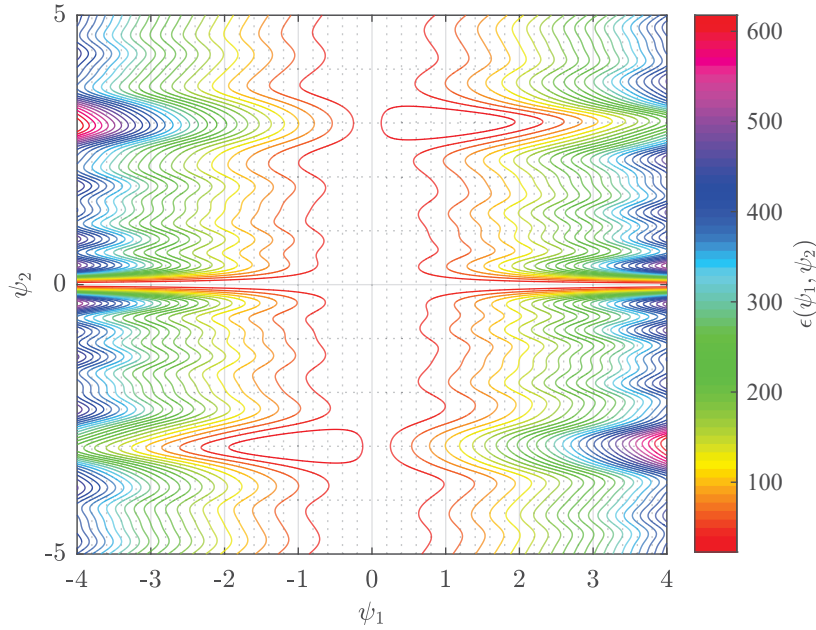


Figure 3.4: The objective function for the nonlinear curve fitting problem in two parameters for $y(x) = \psi_1 \sin(\psi_2 x)$.

A second example that is looked at here is the exponential function

$$y(x) = e^{\psi x}. \quad (3.31)$$

The data ξ that is to be fitted was generated by evaluating $y(x, \psi)$ for $\psi = \psi_{exact} = 0.5$ at 51 evenly spaced points $x \in [-2, 3]$. The evaluation of the objective function $\epsilon(\psi)$ for this curve fitting problem for $\psi \in [-1, 1]$ can be seen in Fig. 3.5. There is clearly one global minimum at $\psi = \psi_{exact}$. As can be seen in the figure, this curve fitting problem results in a convex objective function similar to the linear curve fitting problem in section 3.2.1. Therefore the solution of the numerical optimization of $\epsilon(\psi)$ will hardly depend on variations of the initial guess ψ_0 . The exponential function defined in equation (3.31) can be expanded by adding a second parameter in the form of a linear coefficient.

$$y(x) = \psi_1 e^{\psi_2 x} \quad (3.32)$$

As before, a curve fitting problem can be defined based on data points ξ . These data points are sampled from the solution of equation (3.32) at 51 evenly spaced points $x \in [-2, 3]$ for $\psi_{exact} = [2, 0.5]^T$. The objective function $\epsilon(\psi)$ for this example problem is visualized in Fig. 3.6 as a contour plot for $\psi_1 \in [-2, 3]$ and $\psi_2 \in [-1, 1]$. As can be seen, $\epsilon(\psi)$ again appears to be convex, as can be expected by adding a linear coefficient to the already convex results of the exponential function (3.31), which depends on a single parameter. As is shown by this example, just because a problem is labelled as nonlinear does not

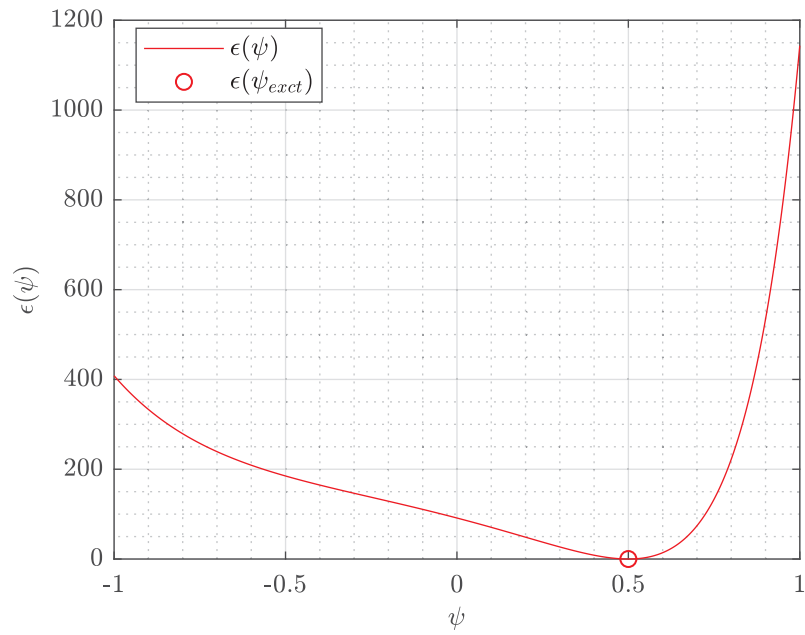


Figure 3.5: The objective function for the nonlinear curve fitting problem in one parameter for $y(x) = e^{\psi x}$.

mean that it needs to be ill behaved. Nevertheless, one has to be mindful of potential local minima and verify the results of a nonlinear curve fitting problem accordingly.

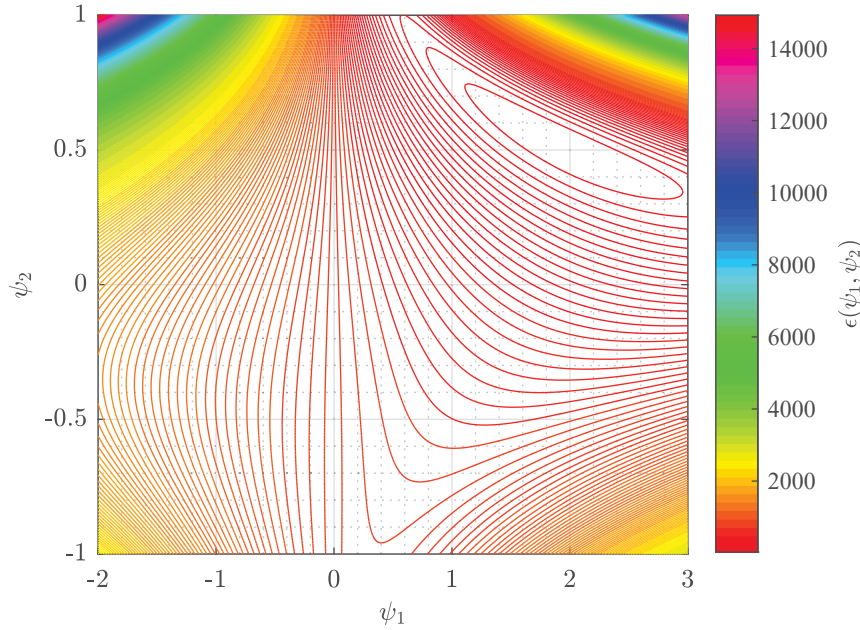


Figure 3.6: The objective function for the nonlinear curve fitting problem in two parameters for $y(x) = \psi_1 e^{\psi_2 x}$.

3.2.2 Gradient Based Numerical Optimization

As was already discussed, in a linear curve fitting problem a function $y(x, \psi)$, which is a weighted sum of basis functions, is fitted to some given data points ξ .

$$y(x, \psi) = \psi_1 f_1(x) + \cdots + \psi_n f_n(x) = \sum_{i=1}^n \psi_i f_i(x) \quad (3.33)$$

The parameters that are to be determined are the constant coefficients, the weights, of the known basis functions $f_i(x)$. The function $y(x, \psi)$ might be nonlinear in the independent variable x as some of the basis functions are nonlinear in x , but $y(x, \psi)$ is linear in the parameters ψ .

$$\left. \frac{\partial y(x, \psi)}{\partial \psi_i} \right|_{x=x_k} = \text{const.} \quad (3.34)$$

In a nonlinear curve fitting problem $y(x, \psi)$ is a nonlinear function in ψ .

$$\left. \frac{\partial y(x, \psi)}{\partial \psi_i} \right|_{x=x_k} = g(\psi) \quad (3.35)$$

This means that it is not possible to formulate an explicit solution for ψ as it is for the linear curve fitting problem. Different numerical solution methods can be applied, in order to minimize the objective function $\varepsilon(\psi)$ in an iterative manner. Starting at an initial guess ψ_0 the gradient at that position in the parameter space can be evaluated. Different methods use this gradient information in different ways in order to derive a step $\Delta\psi$ towards the

optimal parametrization $\boldsymbol{\psi}^*$, which is a minimizer of $\varepsilon(\boldsymbol{\psi})$.

Gradient Descent

If $y(x, \boldsymbol{\psi})$ is differentiable with respect to $\boldsymbol{\psi}$, then the *gradient descent* or *steepest descent* method can be used for the numerical minimization of $\varepsilon(\boldsymbol{\psi})$. Starting from an initial guess $\boldsymbol{\psi}_0$ for the parameters, an updated parameter set can be found by making a step $\Delta\boldsymbol{\psi}$ into the negative direction of the gradient of $\varepsilon(\boldsymbol{\psi})$.

$$\Delta\boldsymbol{\psi} = -\alpha \left. \frac{\partial \varepsilon(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} \right|_{\boldsymbol{\psi}=\boldsymbol{\psi}_0} \quad (3.36)$$

For the objective function

$$\varepsilon(\boldsymbol{\psi}) = [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})]^T [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})] \quad (3.37)$$

this results in

$$\frac{\partial \varepsilon(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} = 2[\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})]^T \frac{\partial}{\partial \boldsymbol{\psi}} [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})] \quad (3.38)$$

$$= -2[\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})]^T \frac{\partial \mathbf{y}(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} \quad (3.39)$$

$$= -2[\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})]^T \mathbf{J}(\boldsymbol{\psi}) \quad (3.40)$$

Therefore a step from $\boldsymbol{\psi}_0$ in the direction of the negative gradient is given by

$$\Delta\boldsymbol{\psi} = \mathbf{J}^T(\boldsymbol{\psi}_0) [\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi}_0)] \quad (3.41)$$

where $\mathbf{J}(\boldsymbol{\psi}_0)$ is the *Jacobian* of $\mathbf{y}(\boldsymbol{\psi})$ evaluated at $\boldsymbol{\psi}_0$

$$\mathbf{J}(\boldsymbol{\psi}_0) = \begin{bmatrix} \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_1} \right|_{x=x_1, \boldsymbol{\psi}=\boldsymbol{\psi}_0} & \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_2} \right|_{x=x_1, \boldsymbol{\psi}=\boldsymbol{\psi}_0} & \cdots & \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_n} \right|_{x=x_1, \boldsymbol{\psi}=\boldsymbol{\psi}_0} \\ \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_1} \right|_{x=x_2, \boldsymbol{\psi}=\boldsymbol{\psi}_0} & \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_2} \right|_{x=x_2, \boldsymbol{\psi}=\boldsymbol{\psi}_0} & \cdots & \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_n} \right|_{x=x_2, \boldsymbol{\psi}=\boldsymbol{\psi}_0} \\ \vdots & \vdots & \ddots & \vdots \\ \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_1} \right|_{x=x_N, \boldsymbol{\psi}=\boldsymbol{\psi}_0} & \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_2} \right|_{x=x_N, \boldsymbol{\psi}=\boldsymbol{\psi}_0} & \cdots & \left. \frac{\partial y(x, \boldsymbol{\psi})}{\partial \psi_n} \right|_{x=x_N, \boldsymbol{\psi}=\boldsymbol{\psi}_0} \end{bmatrix} \quad (3.42)$$

and the next point in the parameter space is

$$\boldsymbol{\psi}_1 = \boldsymbol{\psi}_0 + \alpha \Delta\boldsymbol{\psi}_0. \quad (3.43)$$

The gradient descent method is easy to implement but has the disadvantage that it offers little control of the convergence performance beside scaling the step size with the coefficient α [23]. Selecting a value that is too small might lead to poor convergence

due to making little progress in the parameter space. Selecting a values that is too high might lead to poor convergence due to overshooting of the minimum or the trough of the ellipsoidal valley.

Gauss-Newton

The *Gauss-Newton* method is a nonlinear curve fitting algorithm, which works by approximating the nonlinear design function $y(x, \psi)$ by a first order Taylor series expansion around the current point ψ_0 in the parameter space. In each iteration a step $\Delta\psi$ is computed, which – according to this linearised model – minimizes the sum of squares optimality criterion.

For any inital guess ψ_0 there will be a discrepancy between ξ and $y(\psi_0)$.

$$\xi = y(\psi_0) + \Delta y \quad (3.44)$$

The Gauss-Newton methods tries to find a step $\Delta\psi$ based on the first order Taylor series expansion of $y(\psi)$ around ψ_0 , which minimizes this discrepancy.

$$y(\psi_0 + \Delta\psi) \approx y(\psi_0) + J(\psi_0)\Delta\psi \quad (3.45)$$

$$\xi = y(\psi_0) + J(\psi_0)\Delta\psi \quad (3.46)$$

where $J(\psi_0)$ is the Jacobian of $y(\psi)$ evaluated at ψ_0 . Equation (3.46) is a system of N linear equations in the n unknowns $\Delta\psi = [\Delta\psi_1, \dots, \Delta\psi_n]$. If the columns of the Jacobian are linearly independent then the unique solution is

$$\Delta\psi = J^+(\psi_0)(\xi - y(\psi_0)). \quad (3.47)$$

Which is of course the minimum of the linear curve fitting problem, that arises when using the linearised model (3.45):

$$\varepsilon(\Delta\psi) = \|\xi - y(\psi_0) - J(\psi_0)\Delta\psi\|_2^2. \quad (3.48)$$

The next point in the parameter space can be computed by

$$\psi_1 = \psi_0 + \alpha\Delta\psi. \quad (3.49)$$

The relaxation factor α is typically set to be $\alpha < 1$. This is due to the fact that the linearisation of the nonlinear function $y(x, \psi)$ might only be a good approximation in a close proximity around ψ_0 . Making too large steps $\alpha\Delta\psi$ might lead to poor convergence due to overshooting past the minimum of $\varepsilon(\psi)$.

Levenberg-Marquardt

The *Levenberg-Marquardt* algorithm is based on the idea of K. Levenberg in 1944 and later D. Marquardt in 1963 to use a damped Gauss Newton method for the numerical solution of a nonlinear curve fitting problems [24, 25].

In the Gauss-Newton method the step $\Delta\psi$ is computed from the system of linear equations

$$J^T J \Delta\psi = J^T (\xi - \mathbf{y}(\psi_0)) \quad (3.50)$$

which is equation (3.46) multiplied with J^T from the left. In the Levenberg-Marquardt algorithm this equation is modified by adding a damping term λI to the matrix product on the left hand side:

$$(J^T J + \lambda I) \Delta\psi = J^T (\xi - \mathbf{y}(\psi_0)) \quad (3.51)$$

where λ is a scalar and I is a unity matrix of appropriate size. This equation can be derived by adding a penalty term for the step $\Delta\psi$ in the parameter space to the cost function (3.48) of the Gauss-Newton method:

$$\varepsilon(\Delta\psi) = \|\xi - \mathbf{y}(\psi_0) - J(\psi_0)\Delta\psi\|_2^2 + \lambda \|\Delta\psi\|_2^2. \quad (3.52)$$

Setting the derivative of this cost function with respect to $\Delta\psi$ to zero results in equation (3.51). Depending on the coefficient λ this leads to a different convergence behaviour. For large values of λ the algorithm makes small steps in the direction of the steepest descent

$$\Delta\psi \approx \frac{1}{\lambda} J^T(\psi_0) [\xi - \mathbf{y}(\psi_0)]. \quad (3.53)$$

For small values of λ the algorithm makes steps according to the Gauss-Newton method

$$\Delta\psi \approx J^+(\psi_0) [\xi - \mathbf{y}(\psi_0)]. \quad (3.54)$$

The damping coefficient λ does not have fixed values and an important aspect of the Levenberg-Marquardt algorithm is the way in which it is manipulated during the iteration procedure. Usually λ is initialized as a larger value to ensure a quick descent from the initial guess $\varepsilon(\psi_0)$ along the direction of the gradient. Smaller values of λ ensure a good convergence in the region near a minima, where the Gauss-Newton method shows quadratic convergence if the objective function value at the minimum is approximately zero: $\varepsilon(\psi^*) \approx 0$ [26].

Numerical Approximation of the Jacobian

In the curve fitting problem a function $y(x, \psi)$ is to be fitted to the data points ξ given at their locations $\mathbf{x} = [x_1, \dots, x_N]$. Therefore the evaluation of $y(x, \psi)$ at these location is

described as the vector valued function $\mathbf{y}(\boldsymbol{\psi})$. The Jacobian \mathbf{J} , which is needed in some algorithms for the numerical optimization of the sum of squares goodness-of-fit criterion $\varepsilon(\boldsymbol{\psi})$, contains the derivatives of $\mathbf{y}(\boldsymbol{\psi})$ with respect to each of the n parameters ψ .

$$\mathbf{J}(\boldsymbol{\psi}_0) = \left[\left. \frac{\partial \mathbf{y}(\boldsymbol{\psi})}{\partial \psi_1} \right|_{\boldsymbol{\psi}=\boldsymbol{\psi}_0} \quad \left. \frac{\partial \mathbf{y}(\boldsymbol{\psi})}{\partial \psi_2} \right|_{\boldsymbol{\psi}=\boldsymbol{\psi}_0} \quad \cdots \quad \left. \frac{\partial \mathbf{y}(\boldsymbol{\psi})}{\partial \psi_n} \right|_{\boldsymbol{\psi}=\boldsymbol{\psi}_0} \right] \quad (3.55)$$

For the system identification problem, $\mathbf{y}(\boldsymbol{\psi})$ – the output of the system model – is not known analytically. It can merely be evaluated for any selected point in the parameter space $\boldsymbol{\psi}$ by numerical integration of the initial value problem. Therefore, in order to apply a gradient based optimization method, a numerical approximation of the Jacobian \mathbf{J} needs to be found.

This is done by applying a finite difference formula and by varying one parameter at a time for each column of the Jacobian \mathbf{J} .

$$\left. \frac{\partial \mathbf{y}(\boldsymbol{\psi})}{\partial \psi_i} \right|_{\boldsymbol{\psi}=\boldsymbol{\psi}_0} \approx \frac{\mathbf{y}(\boldsymbol{\psi}_0 + \boldsymbol{\delta}_i) - \mathbf{y}(\boldsymbol{\psi}_0)}{\Delta \psi_i} \quad \dots \quad \text{forward} \quad (3.56)$$

$$\approx \frac{\mathbf{y}(\boldsymbol{\psi}_0 + \boldsymbol{\delta}_i) - \mathbf{y}(\boldsymbol{\psi}_0 - \boldsymbol{\delta}_i)}{2\Delta \psi_i} \quad \dots \quad \text{central} \quad (3.57)$$

$$\approx \frac{\mathbf{y}(\boldsymbol{\psi}_0) - \mathbf{y}(\boldsymbol{\psi}_0 - \boldsymbol{\delta}_i)}{\Delta \psi_i} \quad \dots \quad \text{backward} \quad (3.58)$$

with

$$\boldsymbol{\delta}_i = \Delta \psi_i \mathbf{e}_i \quad (3.59)$$

where \mathbf{e}_i is a n dimensional unity vector pointing in the direction of ψ_i in the parameter space.

$$\mathbf{e}_i = \left[e_1 \quad \dots \quad e_n \right]^T = (e_j) \quad \text{with} \quad e_j = \begin{cases} e_j = 1 & \text{if } i = j \\ e_j = 0 & \text{if } i \neq j \end{cases} \quad (3.60)$$

The central finite difference formula needs $2n$ evaluations of $\mathbf{y}(\boldsymbol{\psi})$ for each approximation of the Jacobian \mathbf{J} , where the forward and backward formula suffice with $(n + 1)$ evaluations.

Regardless of whether a one sided or the central formula is used, the numerical approximation of the Jacobian using finite differences is computationally expensive [27]. As an alternative the Broyden rank-1 update formula can be used to estimate the Jacobian for a new parameter based on the known Jacobian for a nearby point in the parameter space.

$$\mathbf{J}(\boldsymbol{\psi} + \Delta \boldsymbol{\psi}) = \mathbf{F}(\mathbf{J}(\boldsymbol{\psi}), \Delta \boldsymbol{\psi}) \quad (3.61)$$

The disadvantage of this method is, that the successive updates of the Jacobian via Broyden's formula lead to divergence from the real value. Therefore is it advisable to at least

after some steps perform a more precise approximation via a finite difference formula [23].

3.2.3 Numerical Optimization Without Gradient Information

Finding the minimum of the scalar valued optimality criterion $\varepsilon(\psi)$ can also be performed without gradient information. In section 3.2.1 a region of the parameter space was investigated simply by plotting the objective function values for the purpose of visualization. If this evaluated region of the parameter space includes a point ψ which leads to a $\varepsilon(\psi)$ which is smaller than for all surrounding points, then an approximate solution for one minimum of the objective function has already been found. A subsequent evaluation of $\varepsilon(\psi)$ near that minima could be used to further improve on the accuracy of the solution. Of course, a visual inspection of the objective function in order to find a minimum is both computationally expensive and rather difficult to do on a two dimensional graphical representation of $\varepsilon(\psi)$, as soon as the dimension of the parameter space ψ is larger than two.

One algorithm that improves on this idea of simply comparing the evaluated objective function values for different points in the parameter space is the *Nelder-Mead simplex method*.

The Nelder-Mead Simplex Method

Similar to gradient based numerical optimization methods, the Nelder-Mead simplex method is also an iterative numerical optimization method that starts at an initial point in the parameter space. Instead of utilizing gradient information in order to iteratively find a way towards a minimum, the Nelder-Mead algorithm purely relies in function evaluations. For a n dimensional parameter space, $n + 1$ function evaluations around the initial guess ψ_0 lead to the vertices of a so called *simplex* [28]. The algorithm optimizes the objective function by trying out new points and comparing their function value to the ones of the current simplex. By replacing the point with the highest value a new simplex is formed and the process repeated.

The MATLAB[®] function `fminsearch()` is an implementation of the Simplex method. The advantage of using this algorithm compared to gradient based methods lies in the simple application as no gradient information has to be provided.

This MATLAB[®] function was tested on different problems during the work on this thesis, but in comparison to gradient based methods like the Gauss-Newton and Levenberg-Marquardt algorithms it demonstrated worse performance, especially on higher dimensional problems. This might be due to the fact that the simplex method can show poor convergence in the neighbourhood of a minima when small convergence tolerances are demanded as is mentioned in [28]. A discussion about the convergence properties of the

Nelder Mead simplex method in low dimensions can be found in [29].

3.2.4 Improving the Performance of the Numerical Optimization

Once a numerical solver for the IVP and the numerical optimization method has been selected, the successful solution of the nonlinear curve fitting problem of determining the parametrization of a given model structure based on some given identification data mainly comes down to finding a close enough initial guess ψ_0 for the set of parameters that are to be identified. Other factors that influence the results are the selected settings and convergence criteria for the numerical solution methods. However, the performance of the numerical optimization can be further improved by implementing the following procedures, where applicable.

Variable Projection Method

The variable projection methods was introduced by [30] for the solution of nonlinear least square problems whose variables separate. This means that the design function $y(\psi, x)$ of the curve fitting problem can be described as a linear combination of nonlinear basis functions $f_i(\beta, x)$:

$$y(\psi, x) = y(\alpha, \beta, x) = \sum_{i=1}^m \alpha_i f_i(\beta, x). \quad (3.62)$$

In this case the sum of squares can be denoted as

$$\varepsilon(\alpha, \beta) = \sum_{j=1}^n \left[\xi(x_j) - \sum_{i=1}^m (\alpha_i f_i(\beta, x_j)) \right]^2 \quad (3.63)$$

The problem of minimizing this functional is also called a separable least squares problem. It can be solved in two steps by applying the variable projection method as shown in [31]:

As the basis functions $f_i(\beta, x)$ only depend on the set of parameters β , for any choice of β a matrix $V(\beta)$ similar to a Vandermonde matrix containing the evaluated nonlinear basis functions can be defined:

$$V(\beta) = \begin{bmatrix} f_1(\beta, x_1) & \dots & f_m(\beta, x_1) \\ \vdots & \ddots & \vdots \\ f_1(\beta, x_n) & \dots & f_m(\beta, x_n) \end{bmatrix}. \quad (3.64)$$

This means that the objective function $\varepsilon(\alpha, \beta)$ of the separable least squares problem can be written as

$$\varepsilon(\alpha, \beta) = \|\xi - V(\beta)\alpha\|_2^2. \quad (3.65)$$

For any β the linear coefficients α can be computed as the solution of a linear least squares problem

$$\alpha(\beta) = V^+(\beta)\xi. \quad (3.66)$$

Substituting equation (3.66) into (3.65) results in

$$\varepsilon(\beta) = \|\xi - V(\beta)V^+(\beta)\xi\|_2^2 \quad (3.67)$$

which is a function solely dependent on β . Therefore the two steps of solving the separable nonlinear least squares problem are:

1.

$$\beta^* = \underset{\beta}{\operatorname{argmin}} \varepsilon(\alpha, \beta) \Big|_{\alpha=V^+(\beta)\xi} \quad (3.68)$$

, which is a nonlinear optimization problem for the determination of β^* . Subsequently, the optimal weighting α^* of the nonlinear basis functions $f_i(\beta, x)$ can be evaluated as

2.

$$\alpha^* = V^+(\beta^*)\xi. \quad (3.69)$$

The advantage of the variable projection method is that number of parameters, that need to be estimated in the nonlinear optimization, is reduced. This leads to better computational efficiency and a greater likelihood of finding a global minimum, rather than a local one [30, 32].

The variable projection method has found application various disciplines, see for example [31] for an overview. It was also applied in system identification problems. For example in [33] it is applied in the identification of a *Wiener* model, which consists of a discrete LTI system followed by a static non-linearity. In [15], a nonlinear system is identified as a composite local linear state space model, a composition of discrete LTI systems, with the help of the variable projection method. The application in [9] for the system identification of a continuous LTI system utilizing the Global Least Squares solution method was already mentioned in section 1.4.

Unfortunately, the application of the variable projection methods on the system identification of nonlinear state space models appears to be severely limited by the fact that without access to an analytic solution the state trajectory of a nonlinear dynamic system has to be approximated by a numerical solver. This means that in most cases there is no sum of basis functions that can be superimposed, no projection matrix that can be formulated and no separation of variables into ones with linear and nonlinear influence on the basis functions. It is of course possible to design a nonlinear dynamic system, whose output is the superposition of two independent dynamic processes. In such a case it would be advantageous to apply the variable projection method on the system identification problem.

However, for the problems that are studied in chapter 4 this is not the case.

Normalization of the Parameter Space

In some problems the parameter space occupied by ψ^* is badly scaled or skewed in a way that for example one parameter ψ_i lies on a completely different scale of magnitude compared to another parameter ψ_j .

$$\frac{\psi_i}{\psi_j} = 10^n, \quad n \gg 1 \quad (3.70)$$

If the general range or order of magnitude of some parameters values is known then a simple linear transformation can be applied in order to put them on a similar scale. For example when $\hat{\psi}_i$ is the expected value of ψ_i , then a parameter transformation which projects $\hat{\psi}_i \rightarrow 1$ but keeps zero at zero could be used:

$$\begin{bmatrix} 0 & \hat{\psi}_i \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 \end{bmatrix} \quad (3.71)$$

When ψ_{min} are the minimum values of the parameters ψ and ψ_{max} are the maximum values, then each parameter can be projected onto the interval $[0, 1]$ by

$$\phi(\psi) = \frac{\psi - \psi_{min}}{\psi_{max} - \psi_{min}}. \quad (3.72)$$

The above case where $\psi_i = 0$ remains at $\psi_i = 0$ can of course be achieved by setting ψ_{min} to be the zero vector.

A normalization or rescaling of the parameter space can help with the convergence of numerical optimizations methods. One convergence criteria for such an algorithm is often defined to be a lower limit of the acceptable step size $\|\Delta\psi\|$. If the exact value of one of the parameters is of smaller or equal order compared to the smallest accepted step size, then that parameter might be identified less precisely. The extent of this numerical discretization error of course also depends on the influence that variations of this parameter have on the system output and on the noise disturbance in the identification data.

One way of counteracting this is to set the mentioned convergence criteria to smaller values. This can be imagined as using a scope to look at the topology of the objective function in the neighbourhood of the minimum in more detail. But a rescaling of the one problematic parameter takes on the problem in a slightly different way. Instead of investing more computational resources in order to essentially resolve the whole parameter space on a smaller scale, the parameter space is stretched in that one problematic dimension to make convergence easier.

3.3 Implementation of a System Identification Algorithm in MATLAB®

In the system identification problem the parameters of a system model are to be determined such that the behaviour of the model reflects the observed behaviour of the system. This problem can be solved numerically by utilizing numerical integration and numerical optimization methods.

The system model is given by the state-space and observer equations, e.g. for a single input single output system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), u(t), t, \mathbf{p}) \quad (3.73)$$

$$y(t) = g(\mathbf{x}(t), u(t), t, \mathbf{p}) \quad (3.74)$$

or in the case that it is given by higher order differential equations, it needs to be transformed into a state-space representation.

The given n observational data points

$$\boldsymbol{\xi} = [\xi_0, \xi_1, \dots, \xi_{n-1}]^T, \quad (3.75)$$

which correspond to a given input sequence $u(t)$, were measured at the given points in time

$$\mathbf{t} = [t_0, t_2, \dots, t_{n-1}]^T. \quad (3.76)$$

This means that for any set of system parameters \mathbf{p} and initial conditions $\mathbf{x}_0 = \mathbf{x}(t_0)$ a numerical solution method can be applied in order to compute the response of the system model to the given input sequence $u(t)$ at these points in time \mathbf{t} :

$$\mathbf{y}(\boldsymbol{\psi}) = [y(\boldsymbol{\psi}, t_0), y(\boldsymbol{\psi}, t_1), \dots, y(\boldsymbol{\psi}, t_{n-1})]^T, \quad (3.77)$$

where $\boldsymbol{\psi}$ contains the model parameters \mathbf{p} and the initial conditions \mathbf{x}_0 that are to be determined in the system identification. If all the initial conditions are assumed to be known, then $\boldsymbol{\psi} = \mathbf{p}$.

The goal here is to define a function in MATLAB®, which – provided with the input of any value for the set of parameters $\boldsymbol{\psi}$ that are to be identified – returns an estimate of the system output $\mathbf{y}(\boldsymbol{\psi})$ for this parametrization. Any ODE-solver can be applied for the solution of the initial value problem for equation (3.73). For general problems the MATLAB® function `ode45()` can be applied. For the solution of stiff systems `ode23s()` might be more advantageous. The observer equation (3.74) can then be used to extract the system output $\mathbf{y}(\boldsymbol{\psi})$ from the state trajectories $[\mathbf{x}_1(\boldsymbol{\psi}), \mathbf{x}_2(\boldsymbol{\psi}) \dots]$. Once the system output $\mathbf{y}(\boldsymbol{\psi})$ is defined as a function in the parameters $\boldsymbol{\psi}$ that are to be identified,

it is also possible to implement a function which approximates the jacobian $J(\boldsymbol{\psi})$ of $\mathbf{y}(\boldsymbol{\psi})$ by the use of finite difference formulae as described in section 3.2.2. This is necessary in the case where a gradient based numerical optimization technique like the Gauss-Newton or Levenberg-Marquardt algorithm is applied for the solution of the nonlinear curve fitting problem. These algorithms usually have the sum of squares optimality criterion implemented in their code and require the identification data $\boldsymbol{\xi}$, the function that is to be fit to that data $\mathbf{y}(\boldsymbol{\psi})$, the gradient information in the form of $J(\boldsymbol{\psi})$ as well as an initial guess for the parameter set $\boldsymbol{\psi}_0$ as input. The Nelder-Mead Simplex method implemented in the MATLAB® function `fminsearch()` only requires the objective function $\varepsilon(\boldsymbol{\psi})$ and an initial guess $\boldsymbol{\psi}_0$ as an input. In this case the approximation of the gradient is not required but the objective function needs to be implemented in MATLAB® utilizing the previously defined function which returns $\mathbf{y}(\boldsymbol{\psi})$:

$$\varepsilon(\boldsymbol{\psi}) = \|\boldsymbol{\xi} - \mathbf{y}(\boldsymbol{\psi})\|_2^2. \quad (3.78)$$

There are different numerical solution methods for ODEs as well as different numerical optimization methods for nonlinear curve fitting problems, some of which were mentioned previously. Any reader of this thesis should feel encouraged to combine any ODE solver and any numerical optimization technique of their liking into an algorithm for the system identification of nonlinear dynamic systems.

To summarize, here is a short overview of the implemented system identification procedure:

1. Define the identification data $[\boldsymbol{\xi}, \mathbf{t}]$ as well as the corresponding system input $u(t)$.
2. Define the system model, whose parameters \mathbf{p} are to be estimated:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), u(t), t, \mathbf{p}) \\ y(t) &= g(\mathbf{x}(t), u(t), t, \mathbf{p}) \end{aligned}$$

3. Determine whether the initial conditions of the experiment $\mathbf{x}_0 = \mathbf{x}(t_0)$ are known or to be estimated
 - (a) if \mathbf{x}_0 is known: $\boldsymbol{\psi} = \mathbf{p}$
 - (b) if \mathbf{x}_0 is to be estimated: $\boldsymbol{\psi} = [\mathbf{p}^T, \mathbf{x}_0^T]^T$
4. Select or implement a suitable ODE solver.
5. Implement a function, e.g. `estimateModelOutput($\boldsymbol{\psi}$)`, which utilizes the chosen ODE solver and returns the estimated model output $\mathbf{y}(\boldsymbol{\psi})$ at the points in time \mathbf{t} for the given system input $u(t)$ and any given parameter set $\boldsymbol{\psi}$.

6. Select or implement a suitable numerical optimization algorithm.
 - (a) In the case of a gradient based numerical optimization method:
Implement a function, which approximates the jacobian $J(\boldsymbol{\psi})$ of $\mathbf{y}(\boldsymbol{\psi})$ by utilizing `estimateModelOutput($\boldsymbol{\psi}$)`.
 - (b) In the case of a numerical optimization method that does not rely on gradient information: Implement a function which evaluates $\varepsilon(\boldsymbol{\psi})$ by utilizing `estimateModelOutput($\boldsymbol{\psi}$)`.
7. Select an initial guess $\boldsymbol{\psi}_0$ for the parameters that are to be identified.
8. Perform the numerical optimization by providing the selected algorithm with the necessary inputs: $\varepsilon(\boldsymbol{\psi})$ and $\boldsymbol{\psi}_0$ or $\boldsymbol{\xi}$, $\mathbf{y}(\boldsymbol{\psi})$ and $\boldsymbol{\psi}_0$.
This should result in the identified parameter set $\boldsymbol{\psi}^*$.
9. Validate the results.

Chapter 4

Case Studies

In this chapter the presented system identification algorithm is applied to a number of example problems. These include an object in free fall with drag due to air, a nonlinear mass and spring system and a nonlinear dynamic friction model. In each of these case studies the model representing the actual system is described and the experimental design in order to extract information about the object of interest is investigated. The system identification is performed on the basis of synthetic measurement data. This means that the exact parametrization of the system model is decided upon and the measurement data is generated as the simulation output of the exactly parametrized system. Different aspects and problems that arise during the system identification of nonlinear dynamics systems are investigated and illustrated with respect to the mentioned example problems. These include the aforementioned experimental design, the selection of a fitting initial guess for the set of parameters that are to be identified and the utilization of multiple data sets in order to achieve an accurate estimate of those parameters.

4.1 Free Fall with Drag due to Air

4.1.1 The *Riccati* Equation

The *Riccati* equation is a general nonlinear differential equation of first order. It holds a special place in the theory of differential equations because of its relation to the general linear differential equation of second order [34]. The general form of the Riccati equation can be defined as

$$\dot{x} + q(t)x + r(t)x^2 = p(t). \quad (4.1)$$

A special case of equation (4.1) can be obtained by setting the coefficient functions $q(t)$ and $r(t)$ as well as the disturbing function $p(t)$ to constant values. This results in the simplified description of a body in vertical free fall. The state variable was changed from x to v , as it in this case describes the velocity of a body of mass m subject to constant

gravitational acceleration g and a friction force proportional to the square of the velocity v^2 .

$$\dot{v} - \frac{c}{m}v^2 = -g \quad (4.2)$$

The general solution to this equation is known and can for example be found in [34]:

$$v(t) = v_{lim} \left(\frac{e^{rt} - ke^{-rt}}{e^{rt} + ke^{-rt}} \right) \quad (4.3)$$

with the terminal velocity v_{lim} as the velocity at the stationary state where the gravitational and the frictional force cancel each other out:

$$v_{lim} = -\sqrt{\frac{mg}{c}} \quad \text{and} \quad (4.4)$$

$$r = \sqrt{\frac{cg}{m}}. \quad (4.5)$$

The special solution for any initial velocity $v(t=0) = v_0$ is

$$v(t) = \frac{v_0 + v_{lim} \tanh rt}{1 + \frac{v_0}{v_{lim}} \tanh rt}. \quad (4.6)$$

Integrating this function will result in the analytic solution for the position $s(t)$.

$$s(t) = s_0 + \int_0^t v(t) dt \quad (4.7)$$

$$s(t) = s_0 - \frac{v_{lim}^2}{g} \left(\cosh \frac{gt}{v_{lim}} + \frac{v_0}{v_{lim}} \sinh \frac{gt}{v_{lim}} \right) \quad (4.8)$$

$$\text{with } s_0 = s(t=0) \quad (4.9)$$

These nonlinear equations (4.6) and (4.8) as the solution to the problem of the body in free fall show that even such a relatively simple nonlinear dynamic system might present an interesting object of study for the task of system identification.

4.1.2 System Modeling

The same differential equation as (4.2) that resulted as a special case of the Riccati equation can also be obtained by applying Newton's second law of motion. The body of mass m is subject to a constant downward accelerating gravitational force $F_g = mg$. To account for a correct behaviour for all initial conditions the friction force has to be defined in a slightly different way. The resistive friction force due to air drag that is defined as $F_f = cv^2$ as in (4.2) is changed to $F_f = -cv^2 \text{sign}(v)$ to account for a possible positive

initial velocity v_0 . This results in the following equation of motion.

$$m\ddot{s} = \sum F_i = F_f - F_g \quad (4.10)$$

$$m\ddot{s} = -cs^2 \text{sign}(\dot{s}) - mg \quad (4.11)$$

As the position of the mass $s(t)$ isn't explicitly present in the equation, a change of variables can be performed. This results in an equation in the velocity, similar to equation (4.2).

$$\dot{s} = v \quad (4.12)$$

$$\ddot{s} = \dot{v} \quad (4.13)$$

$$m\dot{v} = -cv^2 \text{sign}(v) - mg \quad (4.14)$$

Whether the equation of motion (4.10) depending on the position $s(t)$ or equation (4.12) depending on the velocity $v(t)$ is used for the purpose of system identification, depends on what measurement data is available. The transformation of (4.10) into state-space representation with $x_1(t)$ as the position and $x_2(t)$ as the velocity can be used in any case:

$$\dot{x}_1 = x_2 \quad (4.15)$$

$$\dot{x}_2 = -cx_2^2 \text{sign}(x_2) - mg. \quad (4.16)$$

These two first order differential equations are combined into matrix form:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -cx_2^2 \text{sign}(x_2) - mg \end{bmatrix}. \quad (4.17)$$

The observer equation is

$$y(t) = \mathbf{C}\mathbf{x}(t) = \begin{bmatrix} c_{11} & 0 \\ 0 & c_{22} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad (4.18)$$

with $c_{11} = 1$ and $c_{22} = 0$ for the output $y(t) = x_1(t)$ or $c_{11} = 0$ and $c_{22} = 1$ for the output $y(t) = x_2(t)$.

4.1.3 Experimental Design

In general industrial applications the determination of drag coefficients is usually based on measurements of the drag force F_f in wind tunnel experiments. Nevertheless, this simple system of the body in free fall presents a good starting point for the application of system identification techniques. For that purpose the experimental design will consist of dropping an object of mass m from a specific height h and using the position data as the

basis for the parameter estimation. It is assumed that both the initial position $s_0 = h$ and the initial velocity v_0 as well as the mass m of the object and the gravitational acceleration g are known. The only parameter that is to be determined is the drag coefficient c . It is also assumed that the object is falling vertically straight down and doesn't tumble during the fall so that the drag coefficient remains constant over time.

One way of tackling this problem would be to find a model that can estimate the time it takes for the mass to fall down the height h . This would mean that we are looking for the solution to the boundary value problem

$$s(t_0) = h \quad t_0 = 0 \quad (4.19)$$

$$s(t_f) = 0 \quad t_f \dots \text{extracted from the measurement} \quad (4.20)$$

The disadvantage of this is that we are only taking one measurement, the final time t_f into consideration.

Another approach would be to design the experiment in a way that the object reaches the stationary state of terminal velocity v_{lim} . If the terminal velocity can be estimated, then it is trivial to estimate the drag coefficient from that using equation (4.4).

However, in the following section the system identification approach of minimizing the output error of the model in relation to the observed position measurement as described in chapter 2 will be applied. As such we want to minimize the functional

$$\varepsilon(\xi, \psi) = \|\xi - \mathbf{y}(\psi)\|_2^2. \quad (4.21)$$

The position measurements $\xi(t_i)$ observed at the discrete points in time t_i are contained in the vector ξ . The parameter ψ to be estimated is the drag coefficient c . The model output $y(\psi, t)$ can be computed in two ways. For any experimental data ξ , where the initial velocity $v_0 \leq 0$, the analytic solution (4.8) can be used. Alternatively the solution to the differential equation in state-space representation (4.17) can be computed by numerical integration of the initial value problem. For other problems involving nonlinear differential equations there might not be any analytic solution available. Therefore the path of applying a numerical solution method will also be pursued in this case. The analytic solution might still come in handy in case it is necessary to confirm the accuracy of numerical approximation of the solution of the initial value problem.

Generating Measurement Data

The specific experiment designed to extract information about the drag coefficient $c = \psi$ of an object of mass $m = 10\text{kg}$ is a drop from the height $h = 30\text{m}$ with an initial velocity $v_0 = 0$ and subject to the gravitational constant of acceleration $g = 9.81\text{m/s}^2$.

The observed measurement is the position $\xi(t) \approx s(t)$. The exact position trajectory was generated from the analytic solution (4.8) for the exact parameter $\psi_{\text{exact}} = 0.5 \text{ kg/m}$. This exact system output $y_{\text{exact}}(t)$ is sampled every $t_s = 0.1 \text{ s}$ over the time interval $[t_0, t_f] = [0, 3]$ resulting in the vector $\mathbf{y}_{\text{exact}} = \mathbf{y}(\psi_{\text{exact}})$. Gaussian measurement noise δ that is added to this measurement is generated by sampling pseudorandom values from the normal distribution $\mathcal{N}(\mu, \sigma)$. These disturbances are selected to be rather large with standard deviation of $\sigma = 1$ and zero mean $\mu = 0$. Any experiment i that is performed this way is assumed to result in an observed system output based on the same exact solution $\mathbf{y}_{\text{exact}}$ disturbed by a different sequence of independent random variables δ_i sampled from the same normal distribution.

$$\xi_i = \mathbf{y}_{\text{exact}} + \delta_i \quad (4.22)$$

4.1.4 System Identification Based on One Data Set

The first execution of the described experimental procedure is labeled as experiment A and the corresponding set of observations as

$$\xi_A = \mathbf{y}_{\text{exact}} + \delta_A. \quad (4.23)$$

Based on equation (2.7) the cost function of the system identification problem based on experiment A can be defined as

$$\varepsilon_A(\psi) = \varepsilon(\xi_A, \psi) \quad (4.24)$$

$$\varepsilon_A(\psi) = \|\xi_A - \mathbf{y}(\psi)\|_2^2. \quad (4.25)$$

For problems like this, which involve only one parameter ψ that is to be estimated, it is possible to visualize the objective function $\varepsilon(\psi)$ over a selected region of the parameter space. Figure 4.1 shows the cost function $\varepsilon_A(\xi_A, \psi)$ based on experiment A as well as the theoretical cost function $\varepsilon(\mathbf{y}_{\text{exact}}, \psi)$ based on a measurement without any noise disturbances. Both are evaluated for $\psi \in [0, 1]$. As can be seen, for this problem $\varepsilon(\psi)$ appears to be a convex function with one global minimum. The location of this minima is found by numerical optimization of the objective function.

$$\psi_A = \underset{\psi}{\operatorname{argmin}} \varepsilon_A(\psi) = \underset{\psi}{\operatorname{argmin}} \|\xi_A - \mathbf{y}(\psi)\|_2^2 \quad (4.26)$$

Table 4.1 shows the results of the numerical optimization of the above cost function using the Simplex and the Gauss-Newton method. $\mathbf{y}(\psi)$ was computed from the state space representation (4.17) by numerical integration using the Runge-Kutta method. For the solution using the Gauss-Newton method the derivative of $\mathbf{y}(\psi)$ with respect to ψ was computed as a numerical approximation using a finite difference formula. The initial star-

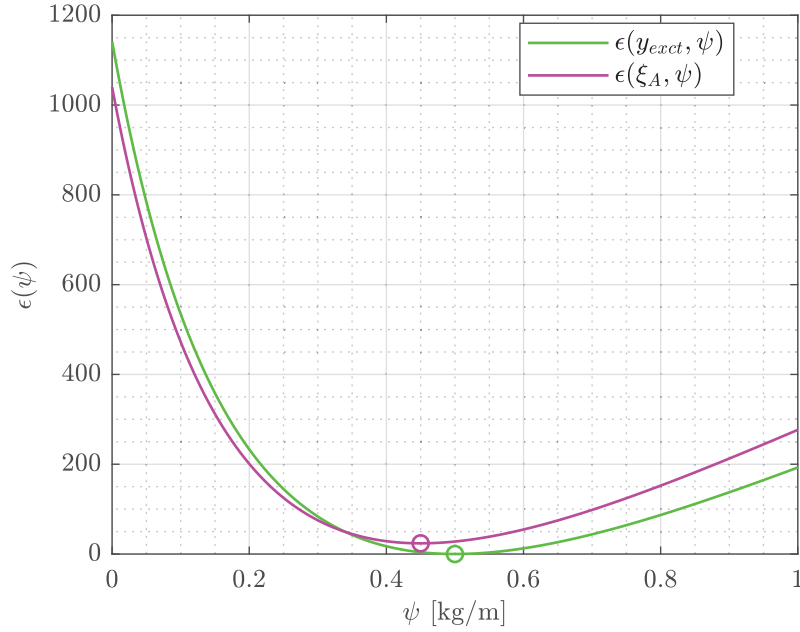


Figure 4.1: Local minima of the objective function for the free fall experiment A.

ing point was set at $\psi_0 = 0$. Because of the mentioned convex form of the cost function, convergence to the minimum from that initial point in the parameter space isn't a problem. The deviation of these two results is not of special relevance here. It is due to the different path in the parameter space that the applied numerical methods take during their iterative solution procedure and due to the different convergence criteria that stop the iteration. The important result is that both methods lead to approximately the same result which deviates from the exact parameter $\psi_{exact} = 0.5 \text{ kg/m}$. This error in the identified parameter is not due to wrong convergence or getting trapped in a wrong local minima. The global minimum of the objective function $\varepsilon(\xi_A, \psi)$ actually lies at the identified location ψ_A . The error in the identified parameter with respect to the exact one is due to the disturbance induced variation of the cost function. This is also visible in Fig. 4.1. The minima of each curve, which are marked by a circle, are not the same. The large deviation is to be expected for the large noise level that was purposefully selected to achieve this clear distinction.

	Simplex	Gauss-Newton	exact
ψ_A [kg/m]	0.450063	0.449959	0.5

Table 4.1: Results of the system identification based on the data from the free fall experiment A.

Fig. 4.2 shows the exact system output $y_{exact}(t)$, the noise perturbed observations ξ_A and the estimated output of the model $y(\psi_A, t)$. As the exact parametrization is known, it is also possible to compare the respective velocity trajectories $x_2(\psi_{exact}, t)$ and $x_2(\psi_A, t)$,

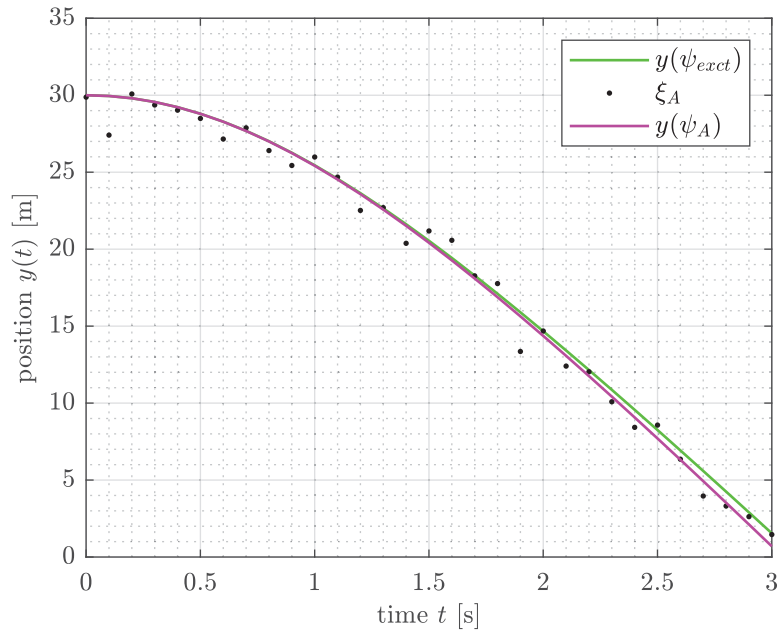


Figure 4.2: Position data and estimate of the object in free fall with drag due to air resistance.

which are shown in Fig. 4.3. The portrayed performance of the model seems to be the best that is possible, based in the data that is given. Ways of improving the model uncertainty would be to improve the measurement accuracy or to reduce the sample time t_s or to increase the measurement interval $[t_0, t_f]$ by increasing t_f in the hope of reducing the signal to noise ratio in order to reduce the disturbance induced variation of the identified parameters [17]. This implies changing the experimental setup in a way to be able to make better or more observations. Another way of improving the model uncertainty without changing the experimental setup is explored in the next section.

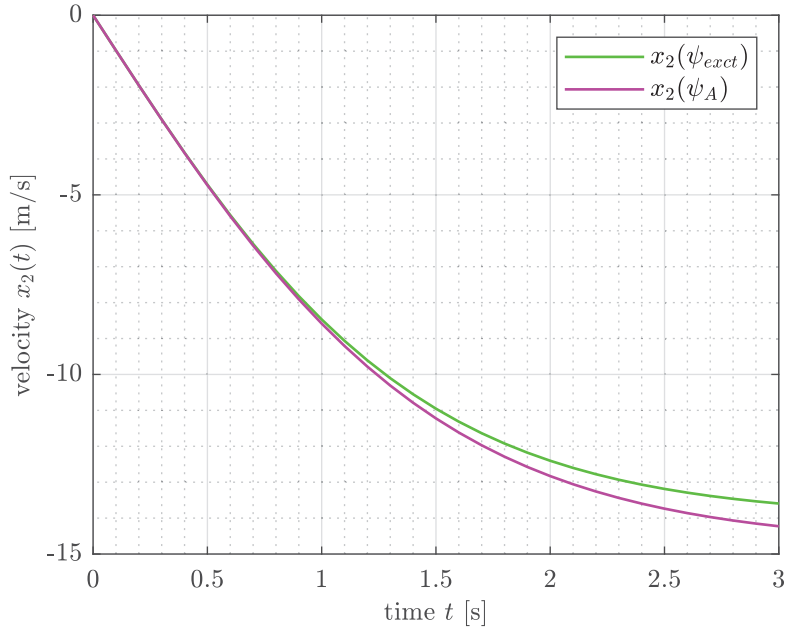


Figure 4.3: Velocity estimate of the object in free fall with drag due to air resistance.

4.1.5 System Identification Based on Two Data Sets

A second experiment B with identical setup to experiment A is performed. This results in the measurement data

$$\xi_A = \mathbf{y}_{exact} + \delta_A \quad (4.27)$$

$$\xi_B = \mathbf{y}_{exact} + \delta_B \quad (4.28)$$

to be available for the estimation of the drag coefficient ψ . The cost function

$$\varepsilon_B(\psi) = \|\xi_B - \mathbf{y}(\psi)\|_2^2 \quad \text{with} \quad (4.29)$$

$$\psi_B = \underset{\psi}{\operatorname{argmin}} \varepsilon_B(\psi) \quad (4.30)$$

can be defined for the measurement ξ_B like equation (4.25). In a similar way, a cost function for the simultaneous vertical distance least squares fit to both data sets ξ_A and ξ_B can be defined by stacking the measurement vectors on top of each other.

$$\varepsilon_{A,B}(\psi) = \varepsilon(\xi_A, \xi_B, \psi) \quad (4.31)$$

$$\varepsilon_{A,B}(\psi) = \left\| \begin{bmatrix} \xi_A \\ \xi_B \end{bmatrix} - \begin{bmatrix} \mathbf{y}(\psi) \\ \mathbf{y}(\psi) \end{bmatrix} \right\|_2^2 \quad (4.32)$$

$$\psi_{A,B} = \underset{\psi}{\operatorname{argmin}} \varepsilon_{A,B}(\psi) \quad (4.33)$$

The question that is to be investigated is what kind of improvement can be achieved by optimizing the fit to both the data sets of experiment A and B . Necessarily one of the measurements will be better than the other as they won't be identical. If we wouldn't have knowledge about the exact parameter ψ_{exct} there would be no way of telling which one it is. For the one dimensional parameter space the only thing that can be said with certainty is that either

$$\psi_{exct} < \psi_1 < \psi_2 \quad (4.34)$$

or

$$\psi_1 < \psi_2 < \psi_{exct} \quad (4.35)$$

or

$$\psi_1 < \psi_{exct} < \psi_2 \quad (4.36)$$

where ψ_1 is the smaller and ψ_2 is the larger value of ψ_A and ψ_B . As we are necessarily combining a better and a worse measurement, it can be expected that $\psi_{A,B}$ is at least a better parametrization than the parameter identified based on the worse measurement. The question that arises from this is whether $\psi_{A,B}$ is always in between ψ_A and ψ_B . If true, this would mean that $\psi_{A,B}$ is a kind of average result of experiment A and B and that only in the case of (4.36) could $\psi_{A,B}$ possibly be better than both ψ_A and ψ_B . However, without knowledge about the exact parameter ψ_{exct} , which wouldn't be available in a real system identification problem, the only distinction that can be made between ψ_A and ψ_B is that one of them is larger and the other one smaller. In this case $\psi_{A,B}$ would still be the best guess of a parametrization that depicts the behaviour of the system most accurately. As ξ_A and ξ_B contain measurements samples drawn at the same point in time in reference to the time interval $[t_0, t_f]$ of the experiment, another data set can be generated by taking the mean of those two measurement sets and defining a fourth cost function accordingly.

$$\xi_{\overline{AB}} = \frac{\xi_A + \xi_B}{2} \quad (4.37)$$

$$\varepsilon_{\overline{AB}}(\psi) = \varepsilon(\xi_{\overline{AB}}, \psi) \quad (4.38)$$

$$\psi_{\overline{AB}} = \underset{\psi}{\operatorname{argmin}} \varepsilon_{\overline{AB}}(\psi) \quad (4.39)$$

As was presented in Fig. 4.1 for the first data set ξ_A , Fig. 4.4 shows the cost functions ε_A , ε_B , $\varepsilon_{A,B}$ and $\varepsilon_{\overline{AB}}$ for the region $\psi \in [0, 1]$ of the parameter space. For visual clarity the vertical axis is set to a logarithmic scale. For the system identification procedure, equation (4.17) was again numerically integrated as described in section 4.1.4. The values of the parameters that were identified using the Gauss-Newton method can be seen in table 4.2. The resulting estimated state trajectories based on the identified parameters can be seen in Fig. 4.5 and 4.6.

	ψ_A	ψ_B	$\psi_{A,B}$	$\psi_{\overline{AB}}$	ψ_{exct}
[kg/m]	0.449959	0.521219	0.484568	0.484568	0.5

Table 4.2: Results of the system identification based on the data from the free fall experiment A and B .

As it turns out $\psi_{A,B}$ is identical to $\psi_{\overline{AB}}$ with respect to not only the six digits displayed here but up to the eleventh digit. To analyse this result, we will look at one step of the Gauss-Newton minimization of the cost function $\varepsilon_{\overline{AB}}(\psi)$ that was defined in (4.32).

$$\begin{bmatrix} \xi_A - \mathbf{y}(\psi_i) \\ \xi_B - \mathbf{y}(\psi_i) \end{bmatrix} = \begin{bmatrix} \mathbf{J}(\psi_i) \\ \mathbf{J}(\psi_i) \end{bmatrix} \Delta\psi_i \quad (4.40)$$

Solving this system of equations is the same as simultaneously solving the two systems of equations

$$\xi_A - \mathbf{y}(\psi_i) = \mathbf{J}(\psi_i)\Delta\psi_i \quad (4.41)$$

$$\xi_B - \mathbf{y}(\psi_i) = \mathbf{J}(\psi_i)\Delta\psi_i \quad (4.42)$$

which can be added to gain

$$\xi_A + \xi_B - 2\mathbf{y}(\psi_i) = 2\mathbf{J}(\psi_i)\Delta\psi_i \quad (4.43)$$

$$\frac{\xi_A + \xi_B}{2} - \mathbf{y}(\psi_i) = \mathbf{J}(\psi_i)\Delta\psi_i \quad (4.44)$$

$$\xi_{\overline{AB}} - \mathbf{y}(\psi_i) = \mathbf{J}(\psi_i)\Delta\psi_i \quad (4.45)$$

which is a step in the Gauss-Newton minimization of the cost function $\varepsilon_{\overline{AB}}(\psi)$. Therefore, each step in the minimization of $\varepsilon_{A,B}(\psi)$ is the same as in the minimization of $\varepsilon_{\overline{AB}}$. This implies that both iterative solution procedures should lead to the same results, as has been observed. Deviations from this identical result might occur due to different activation of the convergence criteria. While the minima of $\varepsilon_{A,B}(\psi)$ and $\varepsilon_{\overline{AB}}(\psi)$ lie at the same place in the parameter space ψ , the value of $\varepsilon_{A,B}(\psi) \neq \varepsilon_{\overline{AB}}(\psi)$ as can be seen in Fig. 4.4. Therefore, a limit set on the minimum allowed change of the objective function value might trigger the stop of the iterative solution method for one but not the other.

To summarize, this means that fitting a model to multiple measurement data sets which are based on the same experimental setup is the same as fitting the model to the mean of those data sets.

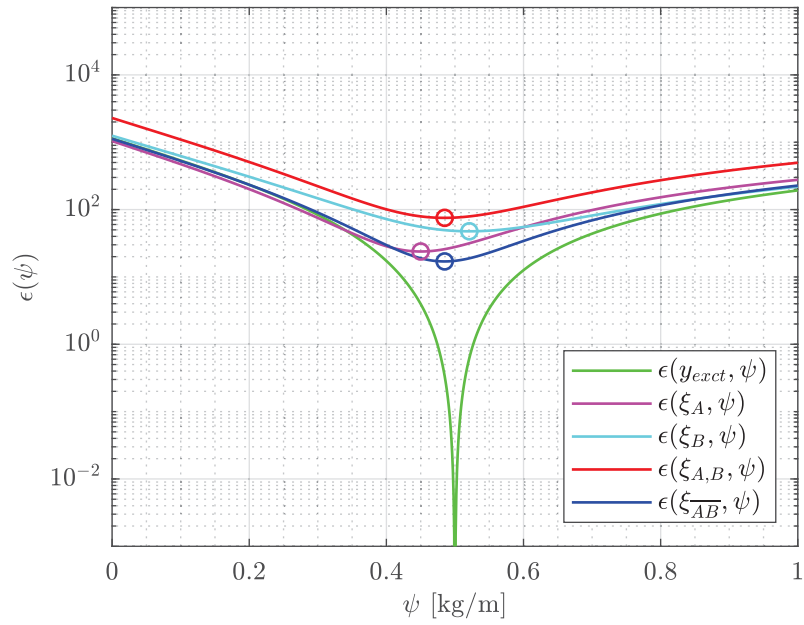


Figure 4.4: Local minima of the objective functions for the free fall experiment A and B.

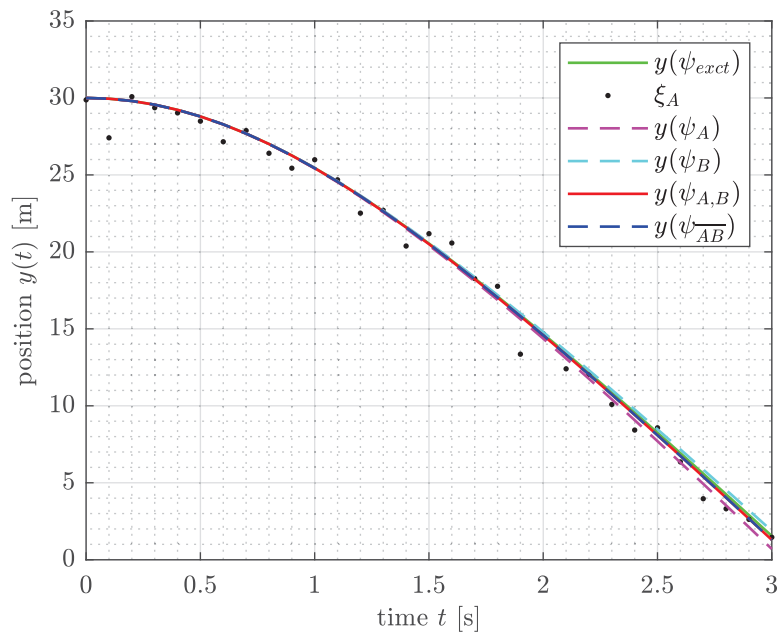


Figure 4.5: Estimated position trajectories for the free fall experiment based in the data sets A and B.

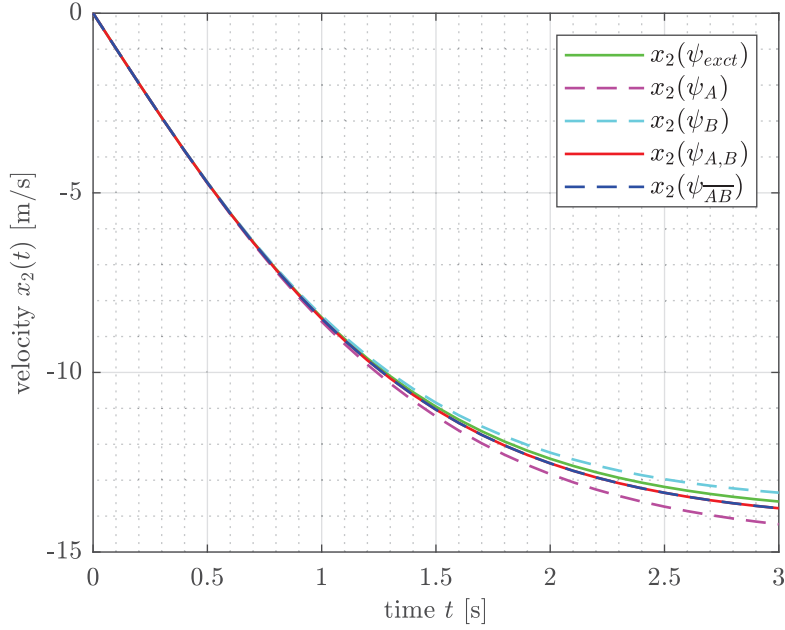


Figure 4.6: Estimated velocity trajectories for the free fall experiment based in the data sets A and B .

4.1.6 Statistical Analysis

To analyse the influence of the measurement noise on the identified parameter ψ , a *Monte-Carlo* simulation is performed. For this purpose the system identification process of estimating the drag coefficient based on one data set corresponding to an experiment A as in section 4.1.4 and based on two data sets corresponding to two experiments A and B as in section 4.1.5 are performed $n = 10^4$ times. Different noise disturbances were generated for each run by sampling the normal distribution $\mathcal{N}(0, 1)$ as was done before. The goal of the Monte-Carlo simulation is be able to analyse the resulting distribution of the identified parameters. To check if the gaussian noise in the identification data results in normally distributed parameters the *Kolmogorov-Smirnov* test can be used. In this test the largest vertical deviation of the empirical cumulative distribution function of a data set from the normal cumulative distribution function is used as a metric in order to evaluate the probability that this data set stems from the normal distribution. In order to be able to apply the Kolmogorov-Smirnov test, the results of the Monte-Carlo simulations have to be normalized as the Kolmogorov-Smirnov test tests for the null hypothesis that a given data sets stems from the standard normal distribution. This is performed via the following formula, using the sample mean $\bar{\psi}_i$ and sample standard deviation s_{ψ_i} for each set of n identified parameters ψ_A , ψ_B , $\psi_{A,B}$ and ψ_{AB} :

$$\tilde{x} = \frac{x - \bar{x}}{s_x}. \quad (4.46)$$

Table 4.3 shows the resulting p-values $p_{KS}(\psi_i)$ in addition to the sample mean $\bar{\psi}_i$ and sample standard deviation s_{ψ_i} corresponding each set of identified parameters ψ_A , ψ_B , $\psi_{A,B}$ and $\psi_{\overline{AB}}$. In particular the results for ψ_A and $\psi_{A,B}$ can be seen in Fig. 4.7, where they are visualized in the form of empirical and normal cumulative distribution functions (CDF). In addition to that Fig. 4.8 and 4.9 show the comparison of a histogram and the normal distribution computed with the sample mean and sample standard deviation for each of these two identified parameter sets. The important insight that is to be taken from this is that gaussian measurement noise apparently results in a normal distribution of the identified parameters due to disturbance induced variation. This is also confirmed by the Kolmogorov-Smirnov tests for a significance level of $\alpha = 0.05 < p_{KS}(\psi_i)$.

As is exhaustively demonstrated – at least for this example problem – normally distributed disturbances lead to normally distributed identified parameters. One thing to note is that the results obtained here are based on the numerical optimization of the objective function. This means that the error of the identified parameter ψ compared to the exact one ψ_{exact} is not only due to the disturbance induced variation of the objective function $\varepsilon(\psi)$ but also due to the numerical accuracy of the solution method. However, as long as the disturbance induced variation is large compared to the numerical accuracy, the identified parameters should remain to be approximately normally distributed.

	ψ_A	ψ_B	$\psi_{A,B}$	$\psi_{\overline{AB}}$
$\bar{\psi}_i$	0.500277	0.500331	0.500041	0.500040
s_{ψ_i}	0.0258864	0.0262175	0.018495	0.0184945
p_{KS}	0.256843	0.142756	0.424425	0.42518

Table 4.3: Results of the Monte-Carlo simulation of the system identification of the free fall experiment based on one and two data sets.

In practice, where only limited identification data is available, a so called *bootstrapping* method can be applied. Synthetic measurements of N data points can be generated from a single set of measurements of N data points by random drawing with replacement. These synthetic data sets can then be used as basis for a Monte-Carlo simulation in order to get an insight into the statistical distribution of the to be estimated parameters [18]. An issue that may arise is that not every data point has the same influence on the estimated parameters as the excitation of a controlled dynamic system changes over time through the applied input.

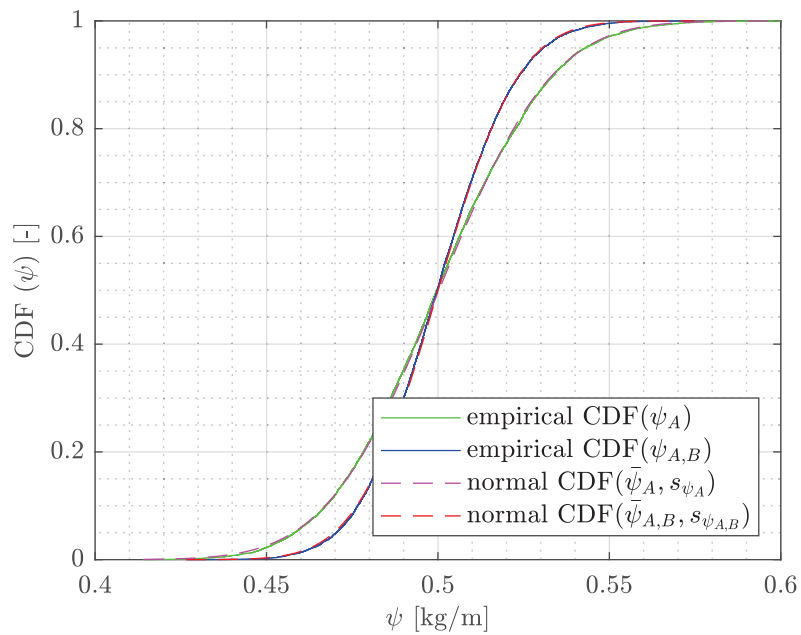


Figure 4.7: Cumulative distribution function of the results of the Monte-Carlo simulation for the free fall experiment based on one and two data sets A and B .

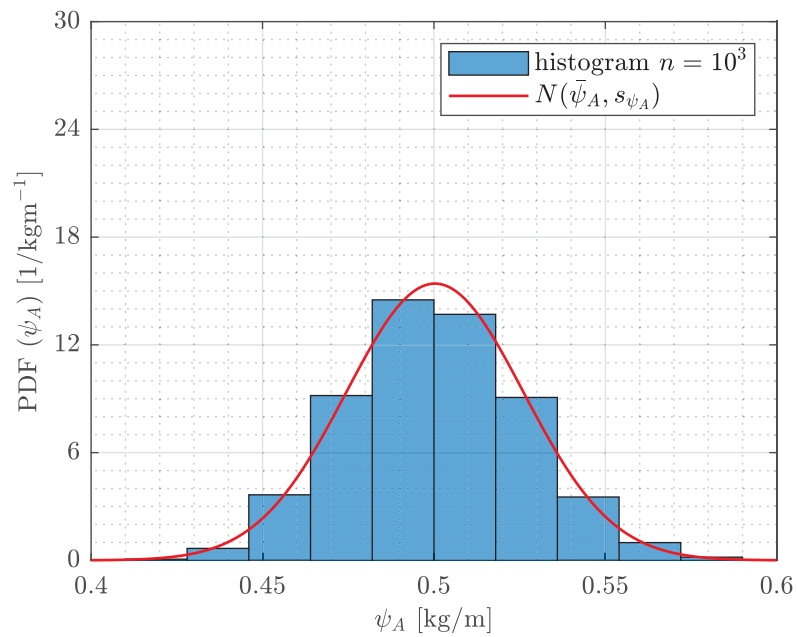


Figure 4.8: Histogram of the results of the Monte-Carlo simulation for the free fall experiment based on one data set A .

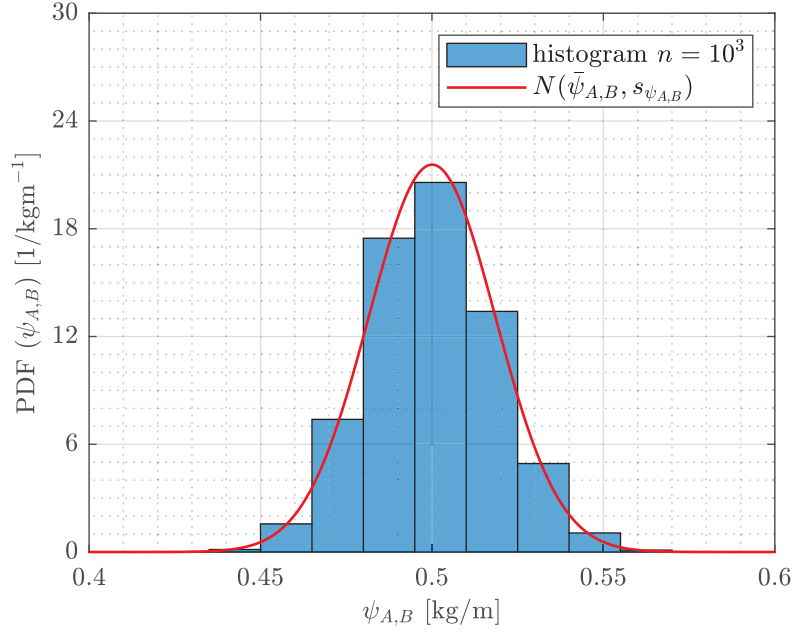


Figure 4.9: Histogram of the results of the Monte-Carlo simulation for the free fall experiment based on two data sets A and B .

4.1.7 System Identification Based on More Than Two Data Sets

In the previous section the effect of combining the information of two data sets on the disturbance induced variation of the identified parameter was investigated. This idea is expanded to include more than two data sets in the identification procedure. This time $n = 1000$ Monte-Carlo simulations are performed for the system identification utilizing $n_{exp} \in \{x | x \in \mathbb{N}, x \leq 6\}$ different data sets, all of which are observations based on the same experiment. The sample standard deviation $s_{\psi}(n_{exp})$ of the results of the Monte-Carlo simulation corresponding to the system identification based on n_{exp} data sets can be seen in Fig. 4.10. A vertical distance least squares fit of the function

$$s_{\psi}(n_{exp}) = \frac{k}{\sqrt{n_{exp}}} \quad (4.47)$$

to the sample standard deviations $s_{\psi}(n_{exp})$ that resulted in the coefficient $k_{LSQ} = 0.0259 \approx s_{\psi}(n_{exp} = 1)$ can also be seen in the figure. This function seems to coincide with the equation

$$s_{\psi}(n_{exp}) = \frac{s_{\psi}(n_{exp} = 1)}{\sqrt{n_{exp}}} \quad (4.48)$$

which is to be expected under the hypothesis that the least squares fit of one function to multiple data sets is equal to the fit of the function to the mean of those data sets. In this case the disturbance induced variation is reduced by the same factor $\frac{1}{\sqrt{n_{exp}}}$, which follows from the reduction of the variation of the mean of the noise disturbances based on n_{exp}

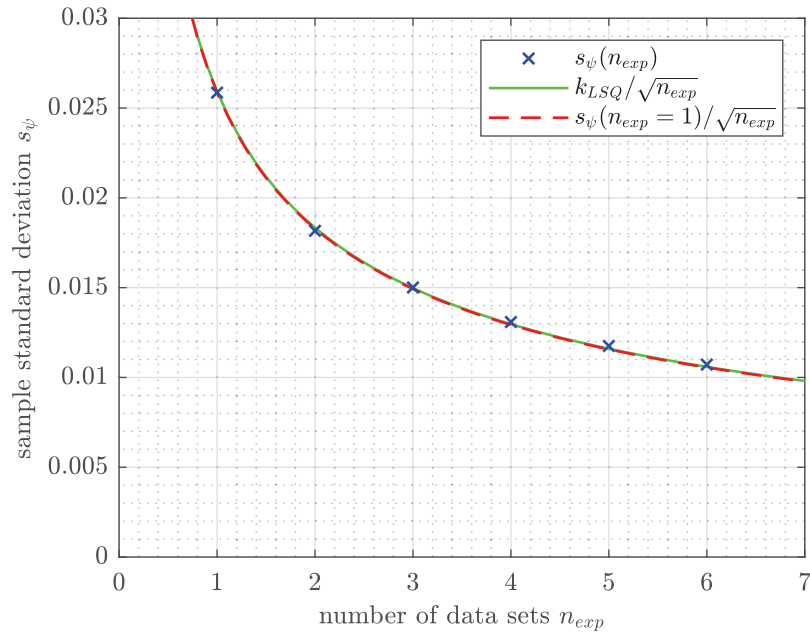


Figure 4.10: Expected standard deviation due to disturbance induced variation of the parameter ψ when combining multiple data sets. The results are based on a Monte-Carlo simulation.

samples according to the *central limit theorem*.

4.1.8 System Identification of the Model Parameter and the Initial Conditions of the Experiment

The model parameters of the system described by equation (4.17) include the mass of the object m , the gravitational constant of acceleration g and the drag coefficient c . The former two of these parameters are assumed to be known. The other two constants that influence the result of the IVP are the initial conditions of the state variables: the initial position $x_1(t_0) = s_0$ and the initial velocity $x_2(t_0) = v_0$. When the exact values of these initial values aren't precisely known, they can be included as parameters that are to be estimated by the system identification.

One can easily imagine an experiment that allows for the precise determination of all the initial conditions. This would mean that these initial values do not need to be estimated. But what would happen when the experiment and the measurement are not coordinated correctly. A shift in time between the start of the experiment and the start of the measurement would introduce an error into the data that cannot be compensated by fixed initial conditions. This is not a problem in this example as the measurement identification data is artificially generated at precise points in time. Nevertheless this section is about the investigation of the effects of adding variable initial conditions to the task of system identification.

The set of all parameters that are to be identified are combined into one column vector ψ .

$$\psi = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{bmatrix} = \begin{bmatrix} c \\ s_0 \\ v_0 \end{bmatrix} \quad (4.49)$$

The new cost function defined for the system identification problem based on one data set ξ_A is now dependent on these three parameters ψ .

$$\varepsilon_A(\psi) = \varepsilon(\xi_A, \psi) \quad (4.50)$$

$$\varepsilon_A(\psi) = \|\xi_A - \mathbf{y}(\psi)\|_2^2. \quad (4.51)$$

This makes it more difficult to analyse the topology of the cost function in order to get an insight about potential local minima. If one of the parameters is set to a fixed value or expressed as a combination of the other two, the objective function can be visualized as a three dimensional surface plot over a two dimensional region of the parameter space. This can be seen as a contour plot of $\varepsilon(\psi_1, \psi_2)$ for $\psi_1 \in [0, 1]$, $\psi_2 \in [25, 35]$ and $\psi_3 = 0$ in Fig. 4.11 and for $\psi_1 = 0.5$, $\psi_2 \in [25, 35]$ and $\psi_3 \in [-5, 5]$ in Fig. 4.12. These two visualizations only represent two orthogonal slices through the three dimensional parameter space. Nevertheless, the objective function appears to be monotonically falling towards one global minimum. This means that neither larger deviations in the initial guess for the parameter c nor for the initial conditions s_0 or v_0 should affect the results of the identified parameter set.

To evaluate the effect that the addition of the initial conditions to the parameter set has on the results of the drag coefficient, another Monte-Carlo simulation was performed based on the same data sets as in section 4.1.4. As the initial conditions were previously fixed at the exact values and are now free to be estimated, it is to be expected that the disturbance induced variation of the initial conditions will also have a negative effect on the estimated drag coefficient c . The numerical optimization of $\varepsilon_A(\psi)$ will use the freedom in the initial conditions to get a better fit on the noisy ξ_A which results in a worse fit to the exact system output \mathbf{y}_{exact} compared to the optimization of the cost function that was fixed to the exact initial conditions as in section 4.1.4. The sample mean and standard deviation of the identified parameters based on the results of $N = 1000$ identification procedures are shown in table 4.4. Also listed are the p-values p_{KS} of the Kolmogorov-Smirnov tests performed on the normalized results.

As was expected, the standard deviation of the drag coefficient s_{ψ_1} is larger than in the results of the Monte-Carlo simulation in section 4.1.4, where only c was identified. This leads to the conclusion that it might be better to fix the initial conditions of the simulation model to a best estimate based on the experimental setup instead of including them into the identification parameter set, if this best estimate is more precise than the disturbance

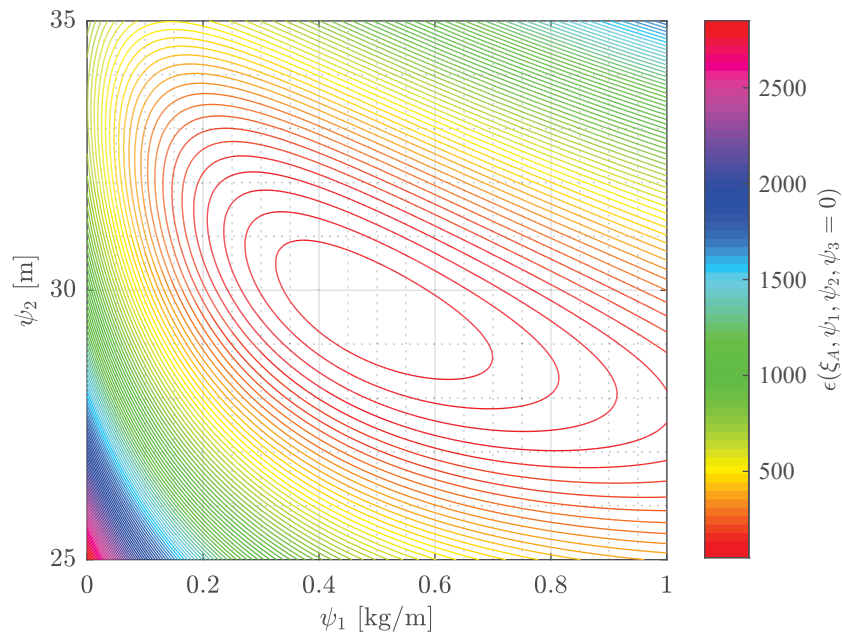


Figure 4.11: Contour plot of the objective function $\epsilon(\psi)$ evaluated for a two dimensional slice of the parameter space: $\psi_1 \in [0, 1]$, $\psi_2 \in [25, 35]$ and $\psi_3 = 0$.

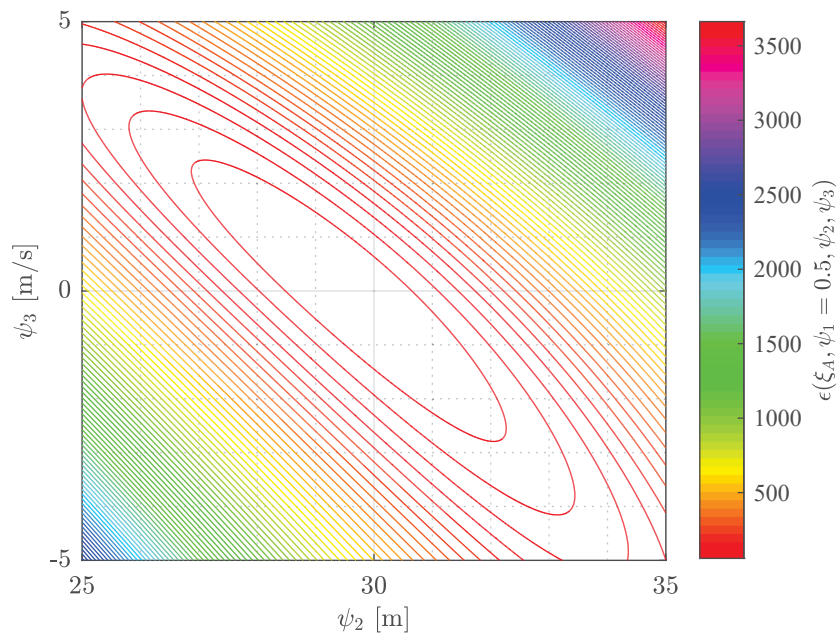


Figure 4.12: Contour plot of the objective function $\epsilon(\psi)$ evaluated for a two dimensional slice of the parameter space: $\psi_1 = 0.5$, $\psi_2 \in [25, 35]$ and $\psi_3 \in [-5, 5]$.

induced variation cause by the additional freedom of the objective function. When dealing with more accurate measurement data, the disturbance induced variation is less impactful. Therefore it might be less likely to find a better estimate for the initial conditions based on the experimental setup compared to the system identification of those initial conditions.

	ψ_1 [kg/m]	ψ_2 [m]	ψ_3 [m/s]
$\bar{\psi}_i$	0.50304	30.0256	-0.0611812
s_{ψ_i}	0.0571966	0.4999	0.688684
p_{KS}	0.279727	0.924865	0.112976

Table 4.4: Results of the Monte-Carlo simulation of the system identification of the parameter and the initial conditions of free fall experiment.

4.2 The *Duffing* Equation and the Nonlinear Mass and Spring System

The second case study looked at in this chapter is about the system identification of a nonlinear mass and spring system. The equations of motion for the mass and spring system can be derived by applying Newton's second law by equating the sum of forces acting on the mass m . For a mass m restricted to motions in one horizontal direction there may be the force of the spring F_{sp} , a resistive friction force F_f and an external force F acting on it. This results in the equation of motion

$$m\ddot{x} + F_{sp} + F_f = F. \quad (4.52)$$

This seemingly simple system can be used to investigate a range of different nonlinear systems. Both the force of the spring F_{sp} as well as the friction force F_f can be modeled as linear or with varying degrees of nonlinearity [35]. For example the following equations describe a linear, a hardening, and a softening spring respectively.

$$F_{sp} = kx \quad (4.53)$$

$$F_{sp} = k[1 + (hx)^2]x \quad (4.54)$$

$$F_{sp} = k[1 - (sx)^2]x \quad (4.55)$$

This nonlinear behaviour is illustrated in Fig. 4.13 for a spring coefficient $k = 64 \text{ N/m}$ and a hardening or softening coefficient of $h = s = 2 \text{ m}^{-1}$ respectively. Of course, any other relation between the displacement of the mass x and the spring force F_{sp} in form of a function could be used as well.

The general form of the so called Duffing equation can be formulated as

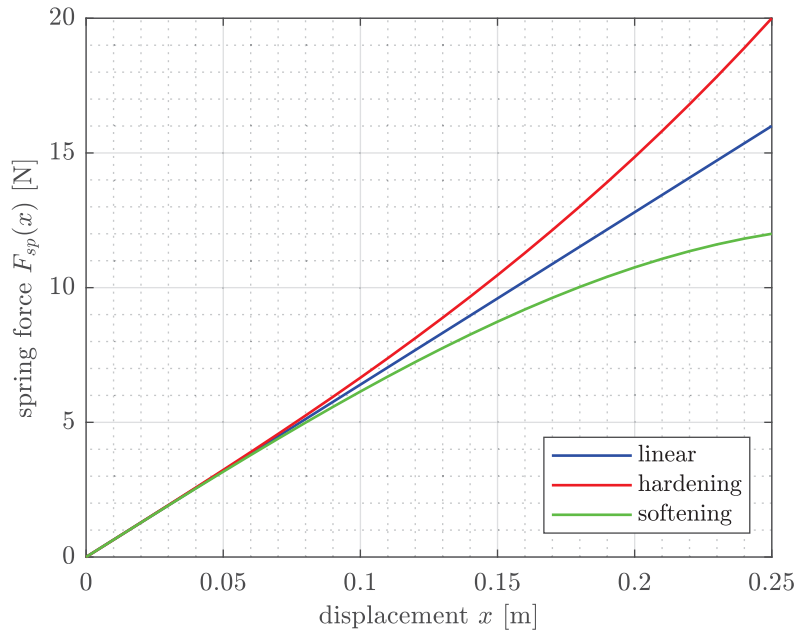


Figure 4.13: Spring characteristic for a linear, a hardening and a softening spring.

$$\ddot{x} + \alpha\dot{x} + \beta x + \gamma x^3 = F. \quad (4.56)$$

Assuming that the friction force is linear viscous $F_f = b\dot{x}(t)$ results in an equivalent equation. For example

$$m\ddot{x} + k[1 + (hx)^2]x + b\dot{x} = F. \quad (4.57)$$

for a hardening spring.

4.2.1 Simulation Model

The system of the mass and hardening spring that is to be investigated is described by equation (4.57). For $F = u(t)$, $x(t) = x_1(t)$ and $\dot{x}(t) = x_2(t)$ the state space representation of this system is

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{m} [u - k[1 + (hx_1)^2]x_1 - bx_2] \end{bmatrix} = \mathbf{f}(\mathbf{x}, u) \quad (4.58)$$

with the observer equation

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x} = x_1. \quad (4.59)$$

For any set of model parameters

$$\mathbf{p} = \begin{bmatrix} k & h & b \end{bmatrix}^T \quad (4.60)$$

and any set of initial conditions

$$\mathbf{x}_0 = \begin{bmatrix} x_{1,0} & x_{2,0} \end{bmatrix}^T = \mathbf{x}(t=0) \quad (4.61)$$

and any input $u(t)$ an approximate solution of the initial value problem can be computed by numerical integration, e.g. by applying the MATLAB[®] function `ode45()`.

4.2.2 System Identification Based on Two Data Sets A and B

For the purpose of identifying the parameters of the mass and hardening spring system two experiments labeled A and B are defined. The sinusoidal input $u_A(t)$ and step input $u_B(t)$ corresponding to those experiments are shown in Fig. 4.14. These input sequences result in a system excitation on two different levels. The goal of this section is to analyse the effect that these different experimental setups have on the identified parameters.

The simulation output of each experiment is labelled as

$$\mathbf{y}_A(\boldsymbol{\psi}) = [y(\boldsymbol{\psi}, u_A(t))|_{t=t_0}, y(\boldsymbol{\psi}, u_A(t))|_{t=t_0+t_{step}}, \dots, y(\boldsymbol{\psi}, u_A(t))|_{t=t_f}]^T \quad (4.62)$$

$$\mathbf{y}_B(\boldsymbol{\psi}) = [y(\boldsymbol{\psi}, u_B(t))|_{t=t_0}, y(\boldsymbol{\psi}, u_B(t))|_{t=t_0+t_{step}}, \dots, y(\boldsymbol{\psi}, u_B(t))|_{t=t_f}]^T \quad (4.63)$$

The system output is sampled every $t_{step} = 0.02$ s. Each experiment lasts ten seconds: $[t_0, t_f] = [0\text{ s}, 10\text{ s}]$.

The parameters that are to be identified are the model parameters \mathbf{p} as well as the initial conditions of the respective experiment $\mathbf{x}_{0,A}$ and $\mathbf{x}_{0,B}$. This can be summarized as an identification parameter vector

$$\boldsymbol{\psi} = \begin{bmatrix} \mathbf{p}^T & \mathbf{x}_{0,A}^T & \mathbf{x}_{0,B}^T \end{bmatrix}^T. \quad (4.64)$$

The synthetic measurement data sets $\boldsymbol{\xi}_A$ and $\boldsymbol{\xi}_B$ are generated by adding Gaussian measurement noise, sampled from the normal distribution $\mathcal{N}(\mu = 0, \sigma = 0.01)$, to the response of the system to $u_A(t)$ and $u_B(t)$ under the exact parametrization

$$\mathbf{p}_{exct} = \begin{bmatrix} k_{exct} & h_{exct} & b_{exct} \end{bmatrix}^T = \begin{bmatrix} 64\text{ N/m} & 2\text{ m}^{-1} & 0.1\text{ kg/s} \end{bmatrix}^T \quad (4.65)$$

with the exact initial conditions

$$\mathbf{x}_{0,A,exct} = \mathbf{x}_{0,B,exct} = \begin{bmatrix} 0\text{ m} & 0\text{ m/s} \end{bmatrix}^T. \quad (4.66)$$

The objective functions that are to be minimized for the purpose of system identification

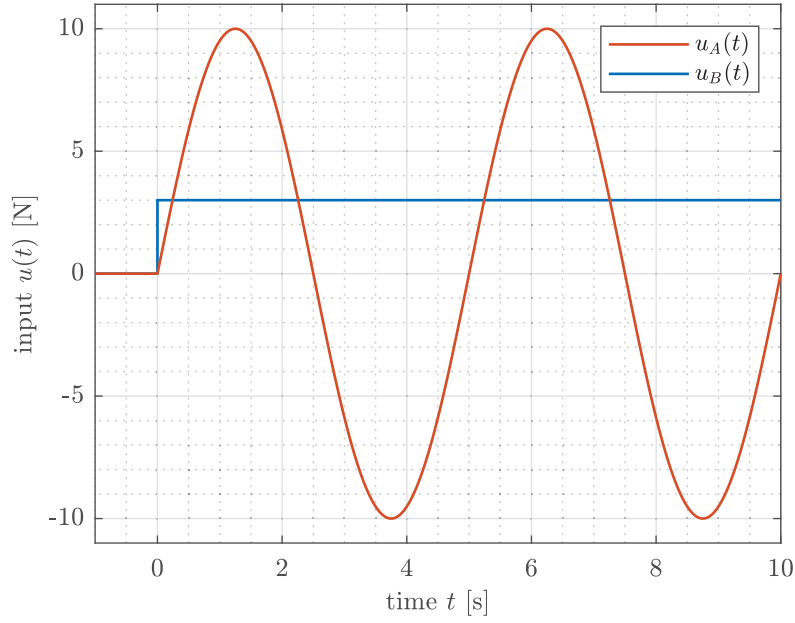


Figure 4.14: System inputs $u_A(t)$ and $u_B(t)$ for the experiments A and B simulated on the mass and hardening spring system.

based on the identification data sets ξ_A , ξ_B or both of them simultaneously are

$$\varepsilon_A(\psi) = \|\xi_A - \mathbf{y}_A(\psi)\|_2^2, \quad (4.67)$$

$$\varepsilon_B(\psi) = \|\xi_B - \mathbf{y}_B(\psi)\|_2^2 \text{ and} \quad (4.68)$$

$$\varepsilon_{AB}(\psi) = \left\| \begin{bmatrix} \xi_A \\ \xi_B \end{bmatrix} - \begin{bmatrix} \mathbf{y}_A(\psi) \\ \mathbf{y}_B(\psi) \end{bmatrix} \right\|_2^2. \quad (4.69)$$

The initial guess for the parameters that are to be identified is

$$\psi_0 = \left[50 \text{N/m} \quad 0.1 \text{m}^{-1} \quad 0.01 \text{kg/s} \quad \xi_A(1) \quad 0 \text{m/s} \quad \xi_B(1) \quad 0 \text{m/s} \right]^T. \quad (4.70)$$

The results of the numerical optimization are denoted as

$$\psi_A = \underset{\psi}{\operatorname{argmin}} \varepsilon_A(\psi), \quad (4.71)$$

$$\psi_B = \underset{\psi}{\operatorname{argmin}} \varepsilon_B(\psi) \text{ and} \quad (4.72)$$

$$\psi_{AB} = \underset{\psi}{\operatorname{argmin}} \varepsilon_{AB}(\psi). \quad (4.73)$$

Fig. 4.15 shows the identification data set ξ_A as well as the exact system output $\mathbf{y}_A(\psi_{\text{exact}})$ on which that data is based on and a comparison of the estimated system outputs $\mathbf{y}_A(\psi_A)$, $\mathbf{y}_A(\psi_B)$ and $\mathbf{y}_A(\psi_{AB})$ based on the results of the system identification procedure. Fig. 4.16 shows the same for experiment B . As can be seen, the estimated system output of

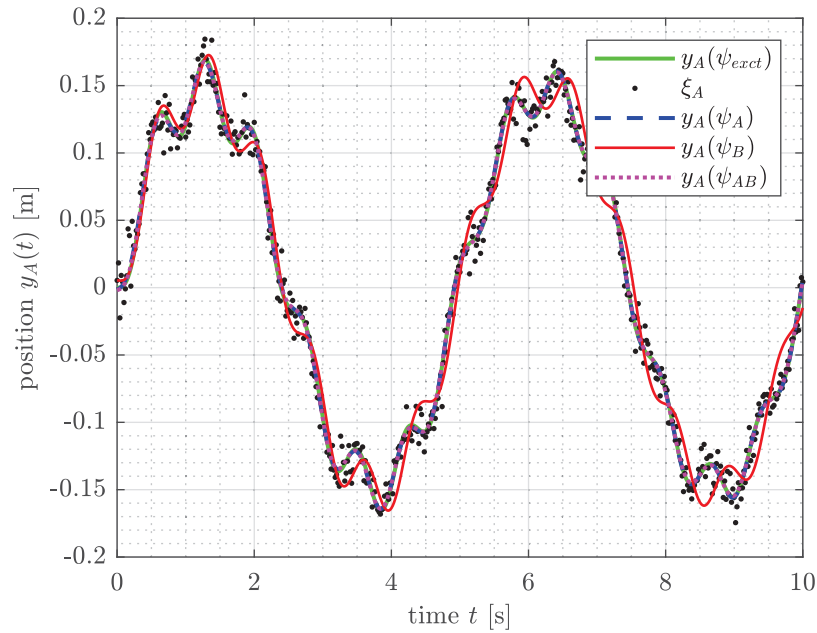


Figure 4.15: Comparison of the estimated system outputs of experiment A based on the results ψ_A , ψ_A and ψ_{AB} of the system identification.

$y_A(\psi_A)$ and $y_B(\psi_B)$ fit rather nicely to the respective output of the exactly parametrized system. However, the estimate of y_A based on ψ_B as well as the estimate of y_B based on ψ_A results in rather poor performance. The estimates based on ψ_{AB} for each experiment are similar to both $y_A(\psi_A)$ and $y_B(\psi_B)$.

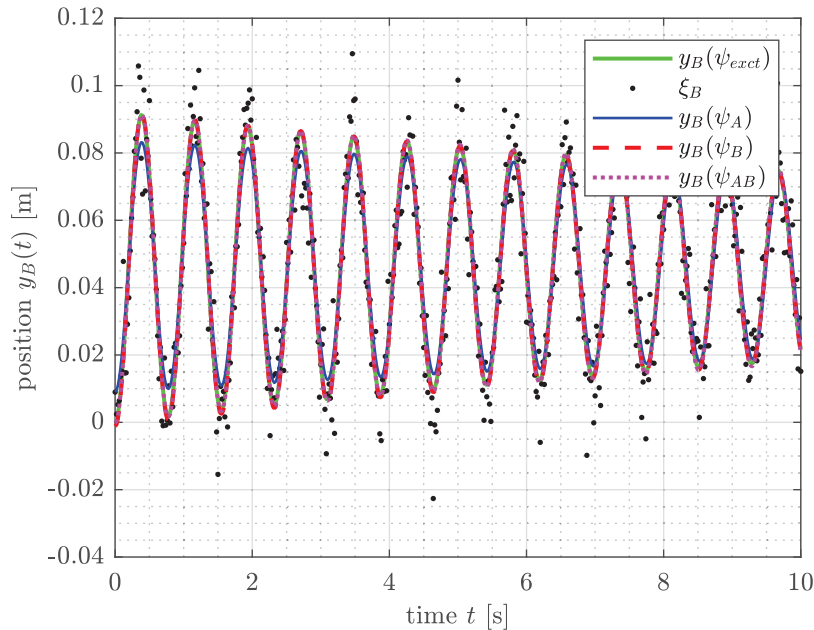


Figure 4.16: Comparison of the estimated system outputs of experiment B based on the results ψ_A , ψ_A and ψ_{AB} of the system identification.

4.2.3 Statistical Analysis – Monte-Carlo Simulation

As was presented in the last section, the combination of multiple data sets can be advantageous in the identification of dynamic systems. Especially when dealing with nonlinear dynamic systems it is important to take this into consideration when designing the experiments that are used to extract information about a system.

To gain more insight into the effect that the combination of the information of two data sets ξ_A and ξ_B has on the system identification of the nonlinear mass and spring system, a Monte-Carlo simulation is performed. The same procedure of optimizing ε_A , ε_B and ε_{AB} as in section 4.2.2 is performed $n = 1000$ times. New noise disturbances at the same noise level were generated for each run. A comparison of the resulting system outputs $y_A(\psi)$ and $y_B(\psi)$ for all parameter sets ψ_A , ψ_A and ψ_{AB} identified in the Monte-Carlo simulation can be seen in Fig. 4.17 and 4.18. The same discussion about the accuracy and inaccuracy of ψ_A , ψ_A and ψ_{AB} in the estimation of y_A and y_B as at the end of section 4.2.2 can be made again here. However, some major outlier trajectories can be made out under the results of the Monte Carlo simulation. All $n = 1000$ successfully converged in parameter sets ψ_A , ψ_A and ψ_{AB} . But some of those apparently lie at wrong local minima.

As was already mentioned in section 3.2.1 and section 4.1, a visual analysis of the topology of an objective function $\varepsilon(\psi)$ in a more than two dimensional parameter spaces ψ is difficult. However, conclusions can be drawn from the results of the numerical optimization, especially when combining a larger number of results, as for example after

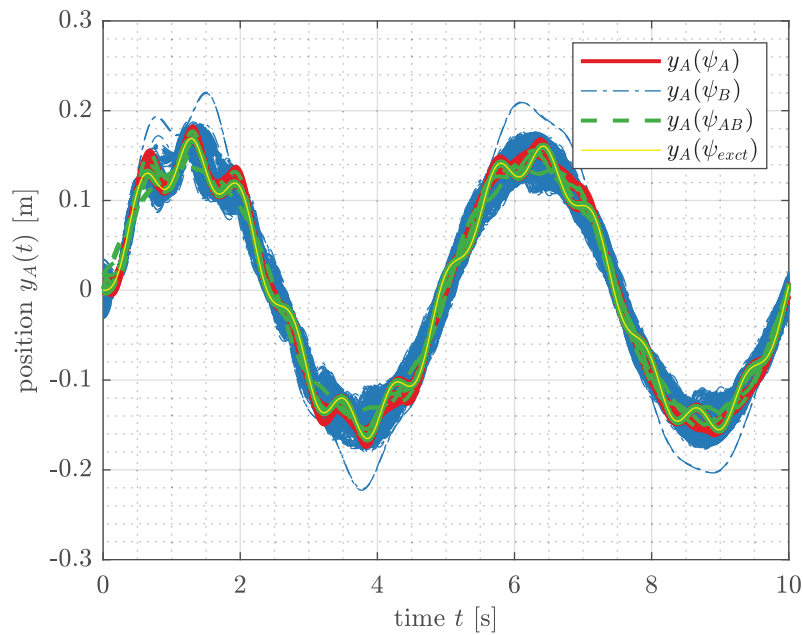


Figure 4.17: Comparison of the simulation output of experiment A corresponding to all data sets ψ_A , ψ_A and ψ_{AB} identified in the Monte-Carlo simulation.

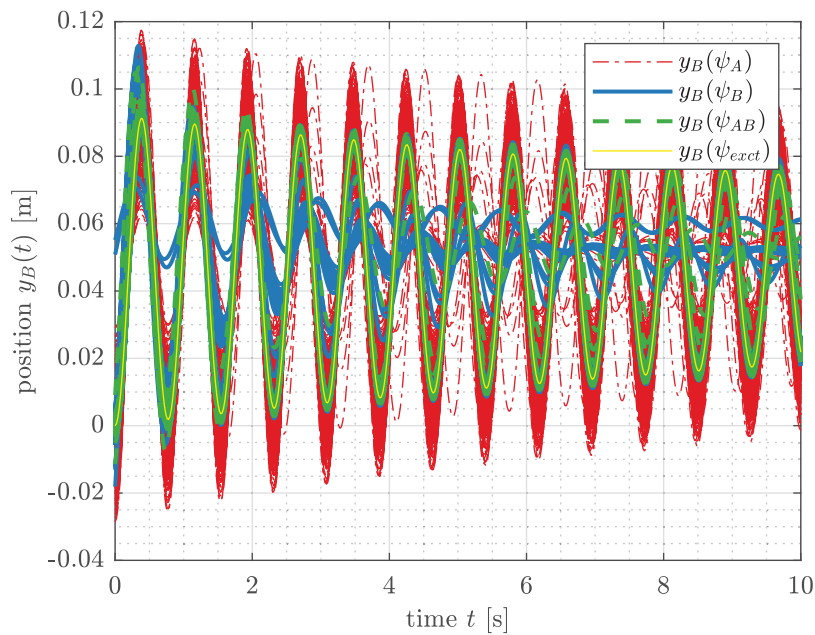


Figure 4.18: Comparison of the simulation output of experiment B corresponding to all data sets ψ_A , ψ_A and ψ_{AB} identified in the Monte-Carlo simulation.

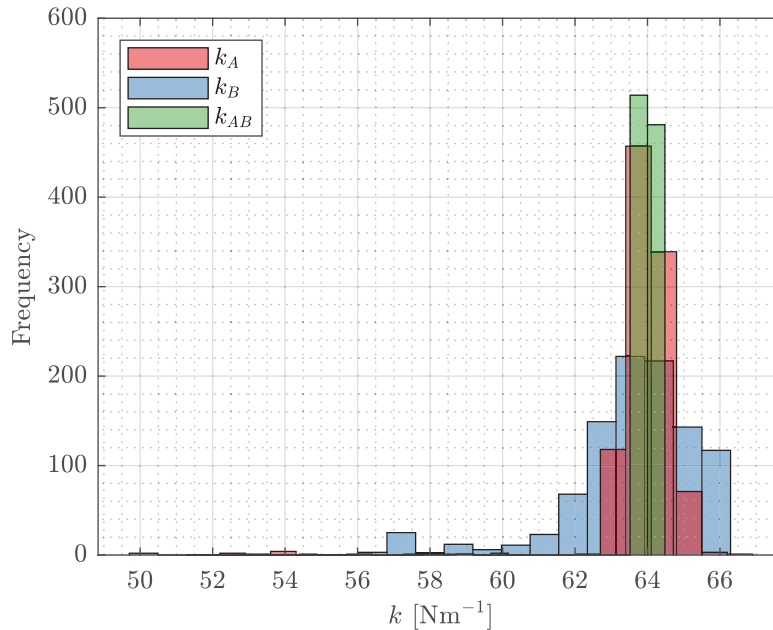


Figure 4.19: Comparison of the results of the Monte-Carlo simulation in the form of histograms for the identified parameter k based on the identification data generated from experiment A and B as well as both of them combined.

performing a Monte-Carlo simulation.

One thing of note is that as the objective function $\varepsilon(\psi)$ depends on the identification data ξ , which is slightly different in each run of the Monte-Carlo simulation that was performed here. Nevertheless, unless when dealing with ill-posed problems, the topology of the objective function should be dominated by the system model and the experimental design, i.e. the input $u(t)$, rather than the measurement noise. This means that a single outlier trajectory might be caused by a disturbance induced local minima of the respective objective function. But if there is an accumulation of identified parameter sets in a region of the parameter space that is not the global minimum, then there is probably a local minima regardless of the noise disturbance.

This can be visually investigated by looking at the histogram of a selected parameters. As the spring stiffness k has a dominant effect in the behaviour of the system, that parameter was selected to be looked at more closely. Fig. 4.19 shows a comparison of the histograms of the identified values of k based on the identification data of experiment A , B and the utilization of both data sets. As can be seen, both k_A and k_B show a wider approximately bell shaped distribution around the exact value of $k_{exact} = 64\text{N/m}$ as well as a larger amount of outliers. The reason for this might be that local minima of $\varepsilon_A(\psi)$ or $\varepsilon_B(\psi)$ that might exist in certain regions of the parameter space are either not existent for $\varepsilon_{AB}(\psi)$ or that this objective function is shaped in a way that the path in the parameter space that is taken in the iterative numerical optimization just does not fall into those local minima but successfully converges to the global minimum for most cases. In fact, the

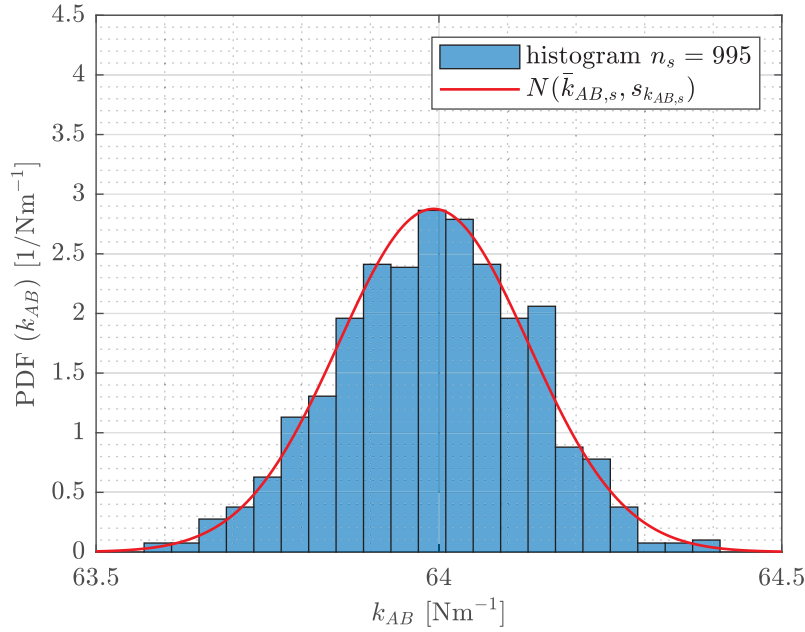


Figure 4.20: Histogram of the identified parameter k_{AB} after the removal of outliers.

histogram of k_{AB} shows hardly any outliers. Based on the histogram of k_{AB} shown in Fig. 4.19 a range of $62 \text{ N/m} < k_{AB,s} < 66 \text{ N/m}$ can be selected, which does not seem to contain any vast outliers. $n_s = 995$ of the $n = 1000$ results of the Monte-Carlo simulation fall into that range. Fig. 4.20 shows a detailed histogram of these selected parameters $k_{AB,s}$ as well as the normal distribution based on the sample mean $\bar{k}_{AB,s}$ and sample standard deviation $s_{k_{AB,s}}$.

When trying to differentiate between outliers due to wrong convergence and parameter induced variation a wider range has to be selected for the results of k_A and k_B , e.g. $60 \text{ N/m} < k_{A,s}, k_{B,s} < 68 \text{ N/m}$. Table 4.5 shows the results of a Kolmogorov-Smirnov test performed on the normalized data set of the selected parameter sets $k_{A,s}$, $k_{B,s}$ and $k_{AB,s}$ in the form of p -values p_{KS} . The data sets were normalized according to the sample mean \bar{k} and sample standard deviation s_k , which are also shown in the table below. Performing the test on the whole parameter sets including all outliers failed for k_A , k_B as well as for k_{AB} ($p_{KS} \ll 1\%$).

	$k_{A,s}$ [N/m]	$k_{B,s}$ [N/m]	$k_{AB,s}$ [N/m]
n_s	990	949	995
\bar{k}	64.0158	63.9046	63.9921
s_k	0.5184	1.2272	0.1387
p_{KS}	0.3490	0.0845	0.8819

Table 4.5: Results of the Monte-Carlo simulation for the nonlinear mass and spring system after the removal of outliers.

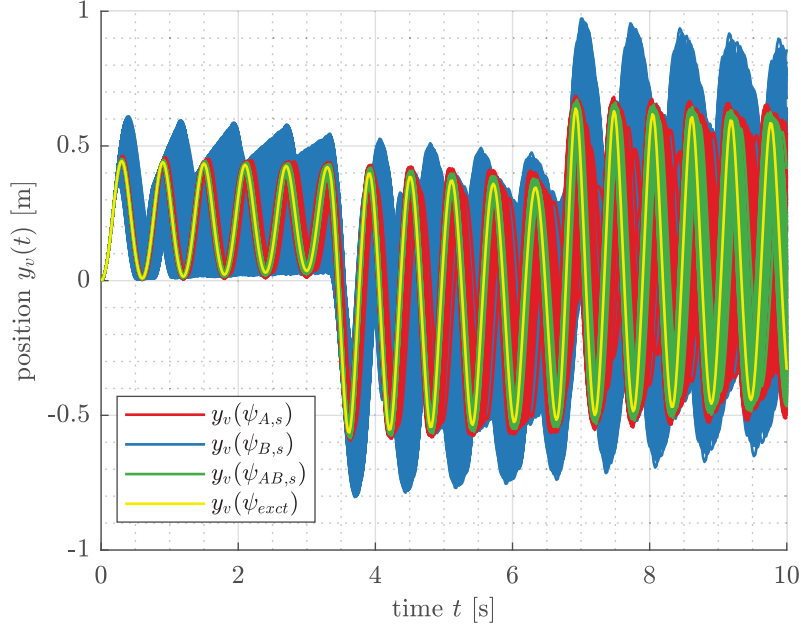


Figure 4.21: Comparison of the system response to the verification test case with input $u_v(t)$. The system is parameterized according to the selected results $\psi_{A,s}$, $\psi_{B,s}$ and $\psi_{AB,s}$ as well as ψ_{exact} .

Fig. 4.21 shows a qualitative visualisation of the fact that the identification based on two data sets ξ_A and ξ_B leads to better results than the identification based on a single data set. Here the system response to a verification test case is presented. The input $u_v(t)$ is a multi-step function

$$u_v(t) = \begin{cases} 10 & \text{if } t < \frac{t_f}{3} \\ -10 & \text{if } \frac{t_f}{3} \leq t < \frac{2t_f}{3} \\ 20 & \text{if } t \geq \frac{2t_f}{3} \end{cases} \quad (4.74)$$

for $t \in [t_0, t_f]$. The presented system responses correspond to all the parameter sets ψ_A , ψ_B and ψ_{AB} , which were identified in the Monte-Carlo simulation excluding those outliers that were detected in the previous look at the histogram of the identified spring stiffness k . As can be seen, the estimation of the system output $y_v(\psi_{AB})$ results in the best approximation of the behaviour of the exactly parametrized system, even though the excitation level of this verification test case lies further above experiment A and B.

4.2.4 Multiple Starting Point Procedure

As was seen in the results of section 4.2.3, the objective functions $\varepsilon_A(\psi)$, $\varepsilon_B(\psi)$ and $\varepsilon_{AB}(\psi)$ appear to not be of a perfect convex shape. Therefore, depending on the selected initial guess ψ_0 and the selected numerical solver, the numerical optimization of any of these scalar-valued optimality criteria might result in a wrong local minima. When encountering such a situation, the simplest way of getting around the issue is to have

another try at the parameter estimation after selecting a new, potentially better initial guess. For system identification problems, where it is difficult to come up with a well-founded initial guess ψ_0 , a multiple starting point procedure can be applied. This means that a set of initial guesses is selected and the results of each of those can be used in order to come to a consensus on local and global minima or to at least achieve a single correct convergence to the global minimum.

In this section the application of a multiple starting point procedure for the identification of the nonlinear mass and spring system based on two experiments A and B is investigated. For this purpose the same data sets ξ_A and ξ_B as in section 4.2.2 are used. The parameters that are to be determined are

$$\psi = \left[k \quad h \quad b \quad x_{0,A} \quad \dot{x}_{0,A} \quad x_{0,B} \quad \dot{x}_{0,B} \right]^T. \quad (4.75)$$

The set of initial guesses for each of those parameters are:

$$k \in \{1, 10, 25, 50, 75, 100, 150\}, \quad (4.76)$$

$$h \in \{0, 0.1, 0.5, 1, 2, 5, 10\}, \quad (4.77)$$

$$b \in \{0, 0.01, 0.1, 0.5\}, \quad (4.78)$$

$$x_{0,A} \in \{0, \xi_A(1)\}, \quad (4.79)$$

$$\dot{x}_{0,A} = 0, \quad (4.80)$$

$$x_{0,B} \in \{0, \xi_B(1)\} \text{ and} \quad (4.81)$$

$$\dot{x}_{0,B} = 0. \quad (4.82)$$

This means that there are 784 different combinations for the initial guess ψ_0 in the numerical minimization of $\varepsilon_{AB}(\psi)$, as defined in equation 4.69.

In this example problem every initial guess $\psi_{0,i}$ lead to convergence to an identified parameter set $\psi_{AB,i}$. This does not need to be the case in every problem. For example the ODE-solver might fail to compute a solution for certain parameter sets. This needs to be handled in the algorithm of the multiple starting point procedure.

In order to evaluate the goodness-of-fit of each results, the objective function value $\varepsilon_{AB}(\psi_{AB})$ is looked at. Fig. 4.22 presents $\varepsilon_{AB}(\psi_{AB,i})$ on a logarithmically scaled vertical axis over the the index i which corresponds to a unique initial guess $\psi_{0,i}$ that lead to the result $\psi_{AB,i}$. As can be seen, there are accumulations of results on different levels of the objective function value. These probably correspond to different local minima. A majority of the results seems to approximately lie on the same lowest level which is clearly separated from the other results. Therefore, a simple way of trying to distinguish between parameter sets that correspond to the global minimum of $\varepsilon_{AB}(\psi)$ and ones that belong to different local minima would be to put a threshold on the goodness-of-fit. In this case a value of $\varepsilon_{max} = 0.07$ is selected. This results in 682 out of the 784 parameter sets

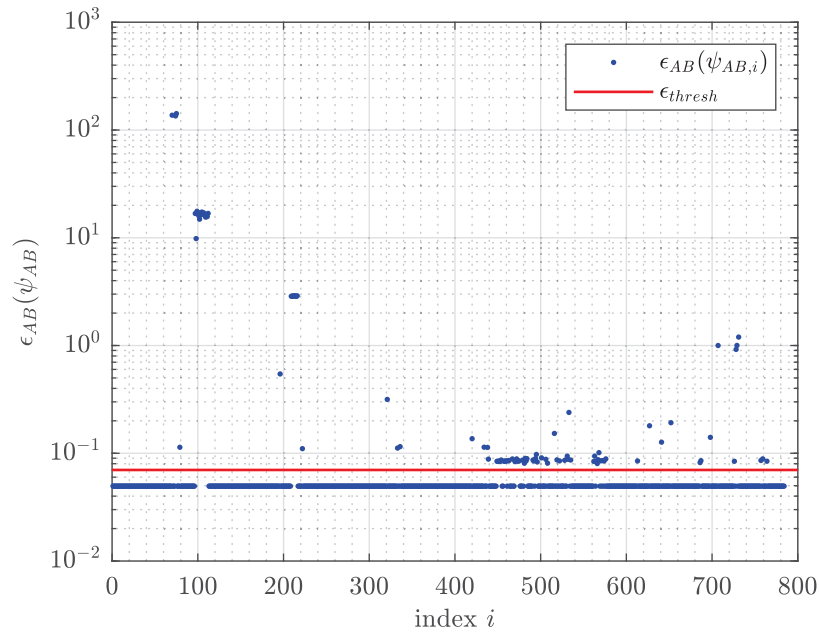


Figure 4.22: Comparison of the objective function values $\epsilon(\psi_{AB})$ for each identified parameter set ψ_{AB} based on a different initial guess that lead to a convergence of the numerical optimization.

which correspond to $\epsilon(\psi) < \epsilon_{thresh}$. Fig. 4.23 shows the estimated system output based on these selected parameters sets compared to the identification data and the response of the exactly parametrized system. As can be seen, all identified parameter sets with objective function value below the selected threshold results in a practically identical system behaviour.

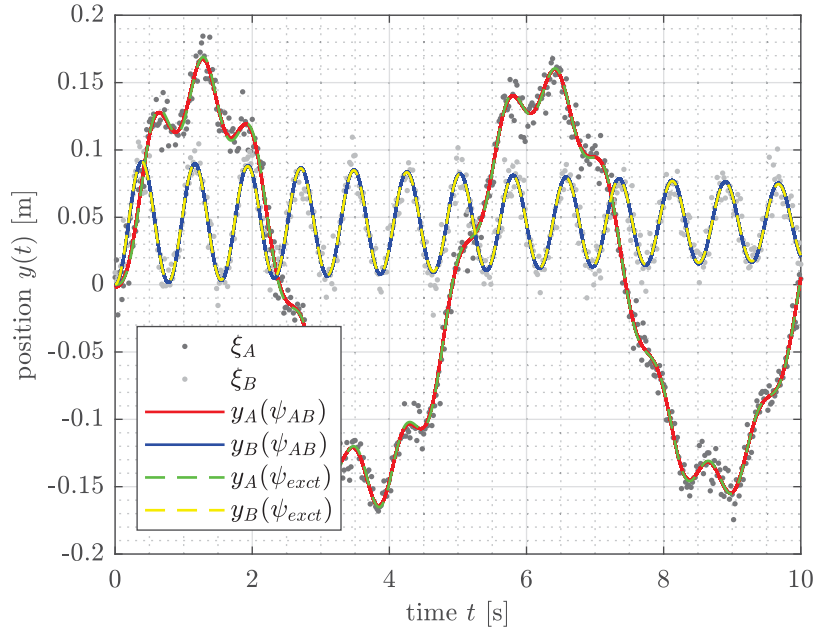


Figure 4.23: Comparison of the simulation output of experiment *A* and *B* for all 682 identified parameter sets corresponding to $\varepsilon(\psi) < \varepsilon_{thresh}$ compared to the identification data.

4.3 Dynamic Friction Models – Stribeck Effect

In this section the system identification of a nonlinear dynamic friction model, which embodies the Stribeck effect, is investigated.

4.3.1 Modeling of Friction

The modeling of friction is a whole field of science on its own. In general the resistive force F_f is dependent on the state of the mass and the contact with the environment. The phenomenon that friction decreases with increasing velocity is called the *Stribeck Effect* [36]. In simple models of friction between two bodies that are in contact, F_f can have components due to static and dynamic Coulomb friction. The exact modeling of the transition from static to dynamic friction presents one of the biggest challenges of friction modeling, both in the correct representation of reality and in the numerical solution of the resulting mathematical models because of the ensuing discontinuity.

In the motionless state, static friction is described by a static friction coefficient μ_0 and the normal contact force F_N . It is also called the *stiction* force or break away force. F_f opposes the sum of the forces acting on the body and its orientation must therefore be determined accordingly.

$$\|F_f\| \leq \mu_0 F_N \quad \text{if } \dot{x} = 0. \quad (4.83)$$

Only once the body is in motion, a simpler description is given by the dynamic friction coefficient $\mu < \mu_0$:

$$F_f = \mu F_N \text{sign}(\dot{x}) \quad \text{if } \dot{x} \neq 0. \quad (4.84)$$

If a moving body is in contact with a viscous medium like gases or fluids, a nonlinear viscous friction term can be added:

$$F_f = b|\dot{x}^{\delta_x}| \text{sign} \dot{x} \quad (4.85)$$

which simplifies to

$$F_f = b\dot{x} \quad (4.86)$$

for linear viscous friction ($\delta_x = 1$). The combination of these effects is defined in the Stribeck friction model:

$$F_f = \mu F_n \text{sign}(\dot{x}) + b\dot{x} + F_s(\dot{x}) \quad (4.87)$$

where $F_s(\dot{x})$ is a function describing the Stribeck effect [37]. In some simulation environments the characteristic (v, F_f) curve resulting from the Stribeck effect is implemented as a lookup table [38]. The accurate representation in the form of mathematical models has been attempted in different ways. One of which is the LuGre friction model described in the next section.

4.3.2 The LuGre Friction Model

The *LuGre* friction model is a dynamic friction model first described in [39], which introduces a new state variable $z(t)$ in order to represent real world friction characteristics in an accurate way. For a body of mass m sliding on a surface and subject to friction, the interface at the point of contact is envisioned as a little bristle. This bristle acts both as a spring and a damper. The new state variable $z(t)$ describes the deflection of that bristle. For a stiffness σ_0 and a damping coefficient σ_1 of the bristle the force acting on the body due to the deflection of the bristle can be described as

$$F = \sigma_0 z(t) + \sigma_1 \dot{z}(t). \quad (4.88)$$

By further adding a linear viscous friction term proportional to the velocity of the body $v(t)$ as in equation (4.86) the dynamic friction force acting on the body is described as

$$F_f = \sigma_0 z(t) + \sigma_1 \dot{z}(t) + bv(t). \quad (4.89)$$

The dynamics of the bristle are described by

$$\dot{z}(t) = \frac{\partial z}{\partial t} = v(t) - \frac{|v|}{g(v)} z(t) \quad (4.90)$$

with $g(v)$ describing the transition from static to dynamic Coulomb friction as an exponentially decaying function with respect to $v(t)^2$:

$$\sigma_0 g(v) = F_C + (F_S - F_C) e^{-\left(\frac{v}{v_s}\right)^2} \quad (4.91)$$

where $F_S = \mu_0 F_N$ is the static friction force and $F_C = \mu F_N$ is the dynamic Coulomb friction force. The Stribeck velocity v_s determines the rate at which $g(v)$ transitions from $\frac{F_S}{\sigma_0}$ to $\frac{F_C}{\sigma_0}$ with increasing velocity $v(t)$.

4.3.3 The Simulation Model

For the purpose of investigating the system identification of the LuGre friction model, a system of a mass m lying on a horizontal surface is looked at. The body of mass is driven by a unidirectional force $u(t)$. The system is subject to friction. This results in the equation of motion

$$m\ddot{x}(t) = u(t) - F_f(t) \quad (4.92)$$

where $x(t)$ is the position of the mass m . The friction force F_f is modelled according to the LuGre model including the linear viscous term. The exact system parameters are assumed to be

$m = 1 \text{ kg}$... mass

$\sigma_0 = 1 \times 10^5 \frac{\text{N}}{\text{m}}$... stiffness of the bristle

$v_0 = 0.01 \frac{\text{m}}{\text{s}}$... Stribeck velocity

$b = 0.4 \frac{\text{kg}}{\text{s}}$... linear viscous friction coefficient

$\mu_0 = 0.15$... static Coulomb friction coefficient

$\mu = 0.10$... dynamic Coulomb friction coefficient

The damping coefficient σ_1 of the bristle is set to critical damping as suggested by [39]:

$$\sigma_1 = 2\sqrt{\frac{\sigma_0}{m}} \quad (4.93)$$

The normal force F_N is assumed to be constant and due to gravity: $F_N = mg$ with $g = 9.81 \text{ m/s}^2$.

For the numerical approximation of the system response the MATLAB® functions `ode45()` and `ode23s()` were tested. As both of them lead to practically the same results `ode45()` was selected as the method of choice to compute the results shown in the following sections.

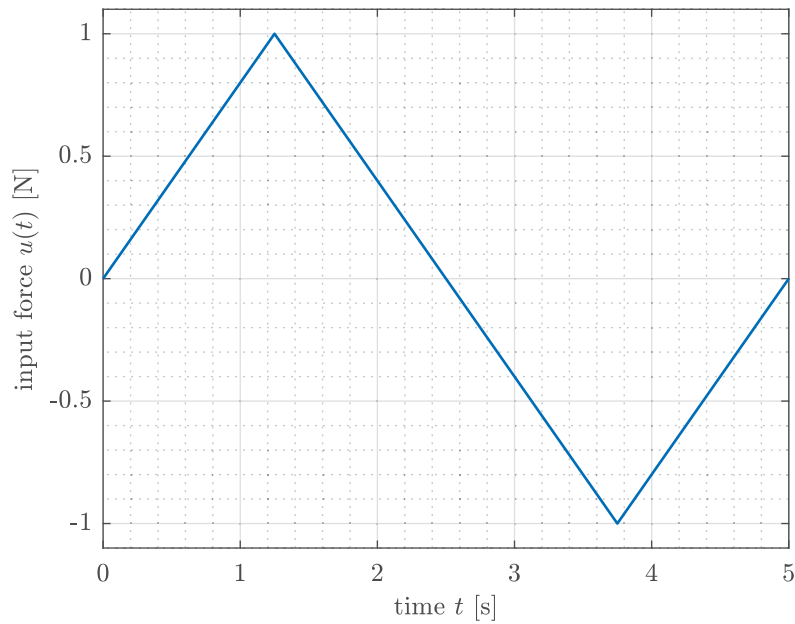


Figure 4.24: Triangular signal $u(t)$ that is applied to the LuGre friction model in order to analyse its properties.

The Influence of Varying System Excitation Levels

The LuGre friction model is able to produce system behaviour which is similar to experimental observations of real world friction characteristics [37, 39, 40]. The LuGre model accounts for micro or *pre-sliding* displacements corresponding to an applied excitation force $u \ll \mu F_N$ as well as friction hysteresis of varying form in cyclic processes which depends not only on the parametrization but also on the rate of change due to the nature of the dynamic friction model .

For the purpose of identifying the system parameters of the LuGre friction model, it might therefore be important to design experiments that result in measurement data which gives insight into these different behaviours at smaller and higher excitation levels of the system.

A downside of the LuGre model is, that it can pose difficulties in the numeric integration due to the stiff parametrization of the bristle [37, 38]. Therefore it would be advantageous if the experimental design results in a setup that is as easy a possible to solve numerically. To investigate the behaviour of the model, a triangular signal $u(t)$ as shown in Fig. 4.24 with varying amplitude \hat{u} is applied to the system. A single cycle of $u(t)$ is looked at with a period of five seconds.

Fig. 4.25 shows the system response to the described input signal for a selection of small amplitudes $\hat{u} < \mu_0 mg$. These results show a small pre-sliding displacements $x(t)$ which the LuGre model allows for due to the fact that it is not perfectly rigid before the static friction force is overcome. Fig. 4.26 shows the system response to larger input signals

$\hat{u} \geq \mu_0 mg$. As can be seen, once the static friction force is overcome, the system exhibits macro level displacements $x(t)$. A small displacement goes together with a velocity $v > 0$, which lowers the friction force due to the Stribeck effect, which results in faster acceleration. As can be seen by comparing the two figures, the system response for different excitation levels may lie on a completely different magnitude.

The important insight to take away from this is that there exists a sort of boundary between the system responding with micro or macro level displacements, which is related to the applied excitation force $u(t)$ but which of course also depends on the system parameters.

For example, a smaller bristle stiffness σ_0 would result in a larger pre-sliding displacement and therefore a higher velocity $v(t)$ as a response to the same input force $u(t)$. A smaller v_0 would result in a more rapid lowering of the friction force with rising velocity $v(t)$. These parameter changes would result in the overcoming of the static friction force at lower input levels. Of course a lower μ_0 would result in the same effect.

During the system identification procedure, starting from an initial guess ψ_0 , a numerical optimization algorithm iterates over the parameter space in order to find the optimal parameter set, which results in the fitting model output $y(\psi)$ to the measured system response ξ . When the measurement data corresponds to macro level displacements, but the initial guess results in micro level displacements or the other way around, correct convergence might be poor or even impossible. Therefore one has to be careful when working with identification data sets that lie close to that boundary, i.e. where small variations in the parameters cause the overcoming of that boundary and consequently a drastic change in the system behaviour.

To conclude the discussion of the system behaviour at varying excitation levels the following figures present the (x, F_f) and (v, F_f) characteristic for the same triangular waveform inputs $u(t)$ as discussed so far. F_f was computed according to equation (4.89). Fig. 4.27 and Fig. 4.28 show the friction hysteresis (x, F_f) at lower and higher excitation of the system. Fig. 4.29 shows the (v, F_f) hysteresis for small excitation levels as well as one flaw of the LuGre model. As the friction is modelled as a dynamic system via the state $z(t)$, the friction force F_f does not always point against the direction of motion $\text{sign}(v(t))$. Fig. 4.30 shows the (v, F_f) characteristic for higher excitation levels. This results in the typical Stribeck curve where F_f starts at $\mu_0 mg$, sharply falls off to μmg as the motion begins and rises with the viscous friction term $bv(t)$.

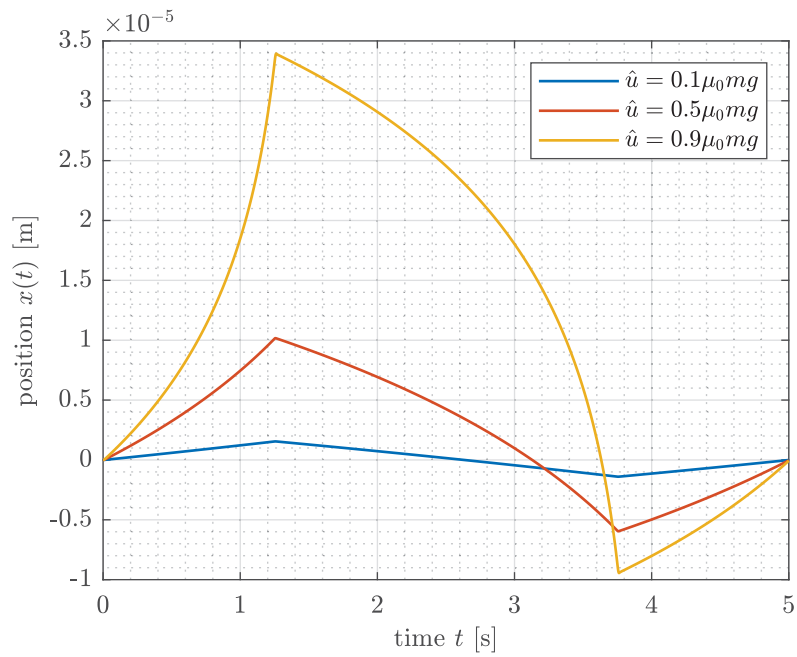


Figure 4.25: System response $x(t)$ to the triangular input signal $u(t)$ at smaller excitation levels.

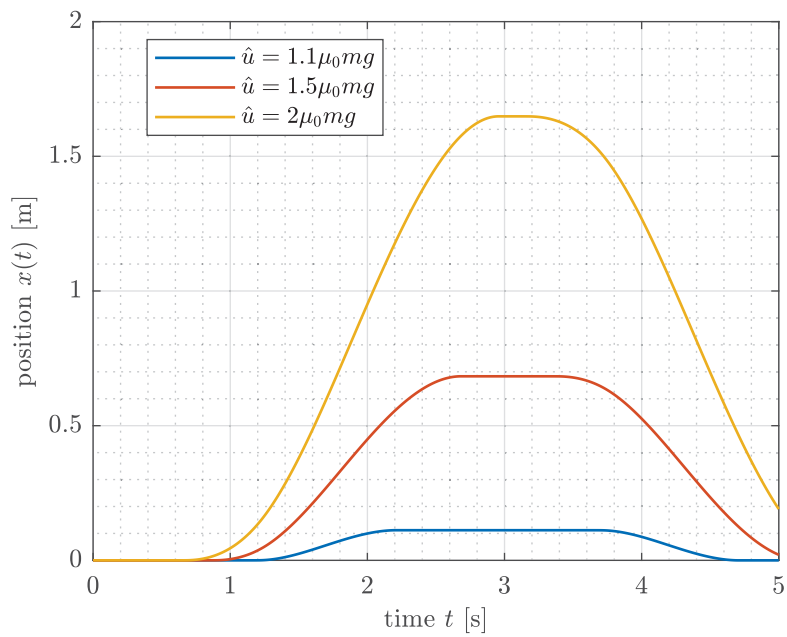


Figure 4.26: System response $x(t)$ to the triangular input signal $u(t)$ at larger excitation levels.

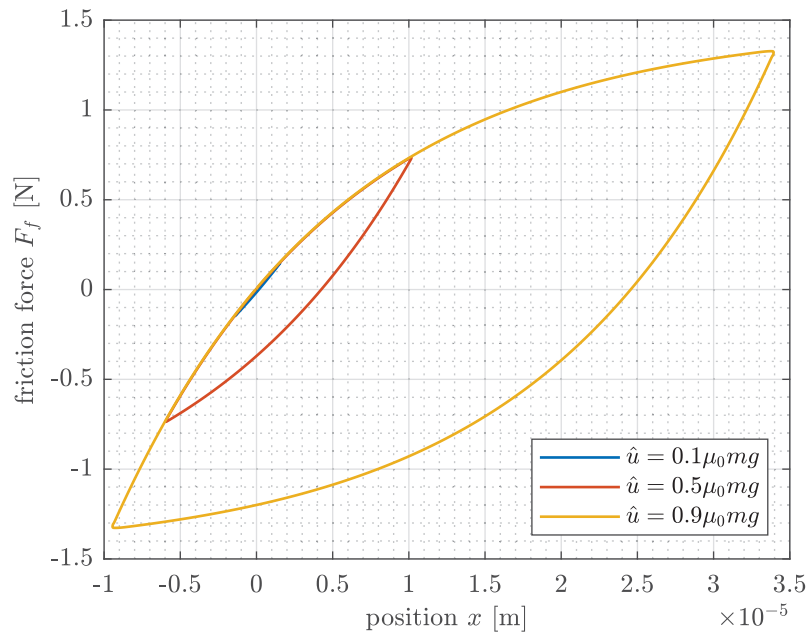


Figure 4.27: $(x(t), F_f(t))$ characteristic for smaller excitation levels $u(t) < \mu_0 mg$.

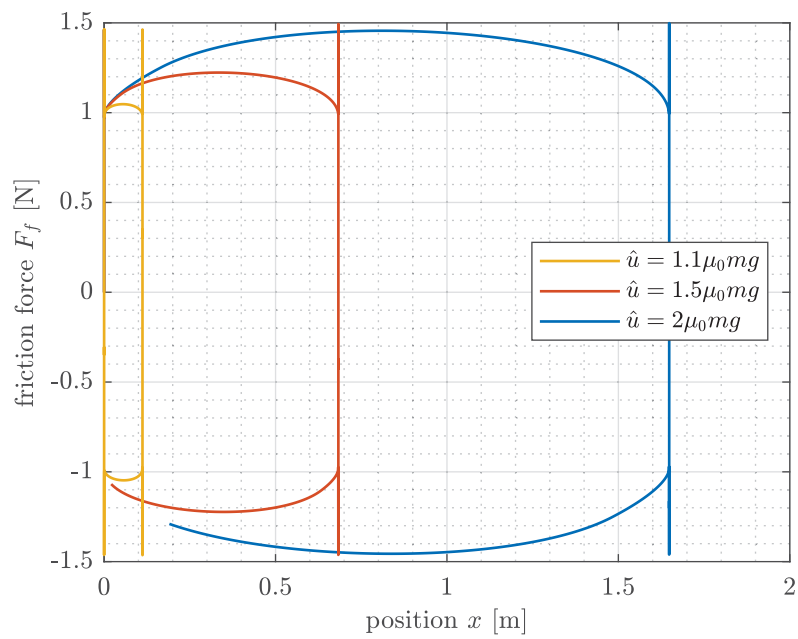


Figure 4.28: $(x(t), F_f(t))$ characteristic for higher excitation levels $u(t) > \mu_0 mg$.

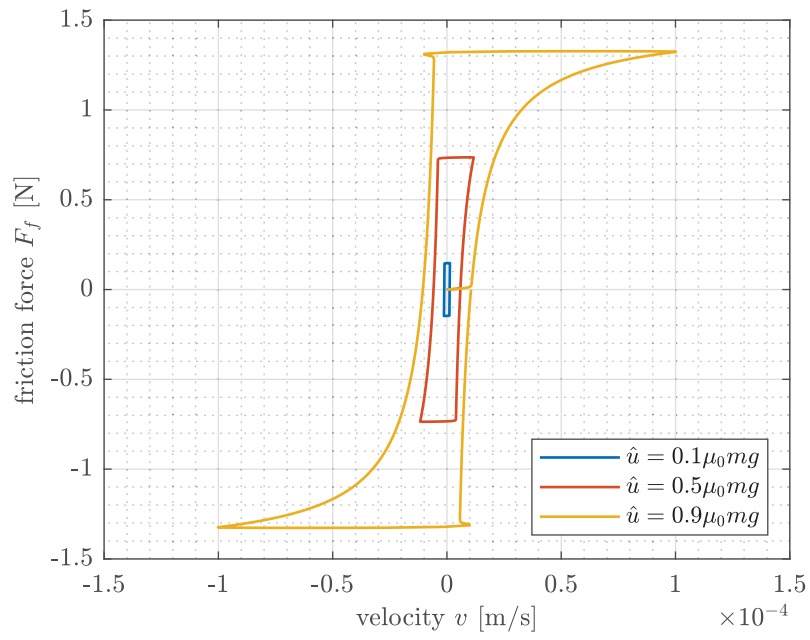


Figure 4.29: $(v(t), F_f(t))$ characteristic for smaller excitation levels $u(t) < \mu_0 mg$.

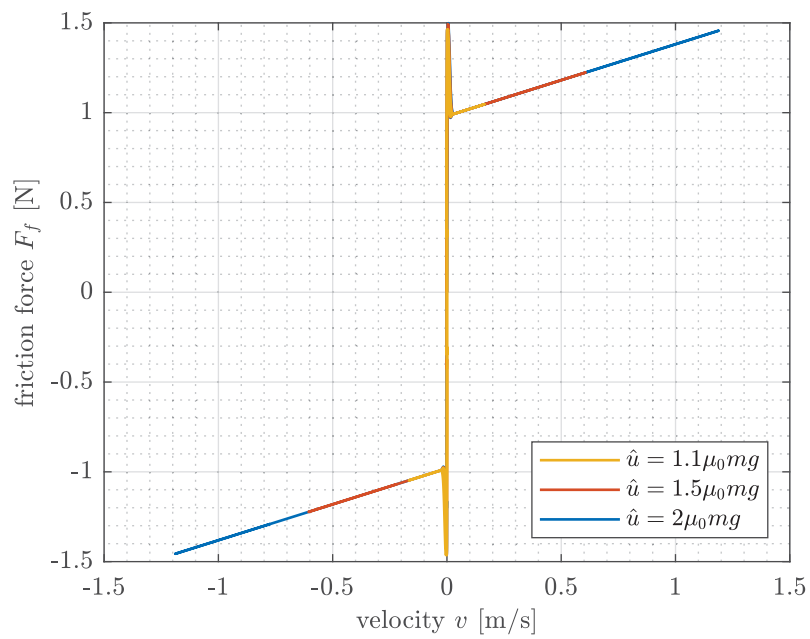


Figure 4.30: $(v(t), F_f(t))$ characteristic for higher excitation levels $u(t) > \mu_0 mg$.

The Influence of Varying Parameters

To look at the effect that a small variation of the parameters has on the system response, the numerical approximation of the Jacobian $J(\boldsymbol{\psi})$ of the system response $x(t)$ is now looked at. The parameters under investigation are

$$\boldsymbol{\psi} = \left[b \quad \sigma_0 \quad v_s \quad \mu_0 \quad \mu \right]^T. \quad (4.94)$$

The Jacobian is approximated by a central finite difference formula for small changes around $\boldsymbol{\psi}_{excit}$ as specified in the beginning of this section 4.3.3. As some of the parameters lie on a different scale of magnitude compared to others, an individual step size of $\Delta\psi_i = 1\% \psi_i$ was selected for each parameter ψ_i .

For a small excitation with the triangular signal $u(t)$ with amplitude $\hat{u} = 0.5\mu_0mg$ the Jacobian is presented in Fig. 4.31. For this input, the system shows a larger change in the system response for a variation of σ_0 and μ_0 . The variation of the other parameters shows less of an effect. For clearer visibility, these components of the Jacobian are shown in detail in Fig. 4.32. Note that the variation of v_s shows a similar but not the same change in the response as a variation of μ .

Fig. 4.33 shows the numerically approximated Jacobian for the same input form but with a higher amplitude $\hat{u} = 1.5\mu_0mg$. As can be seen, for this test case the variation of σ_0 results in the smallest variation of the response.

Based on this analysis it can be assumed that the system identification based in pre-sliding displacement data could result in better estimation of σ_0 compared to macro level displacement data due to the larger effect of small variations on the value of the objective function.

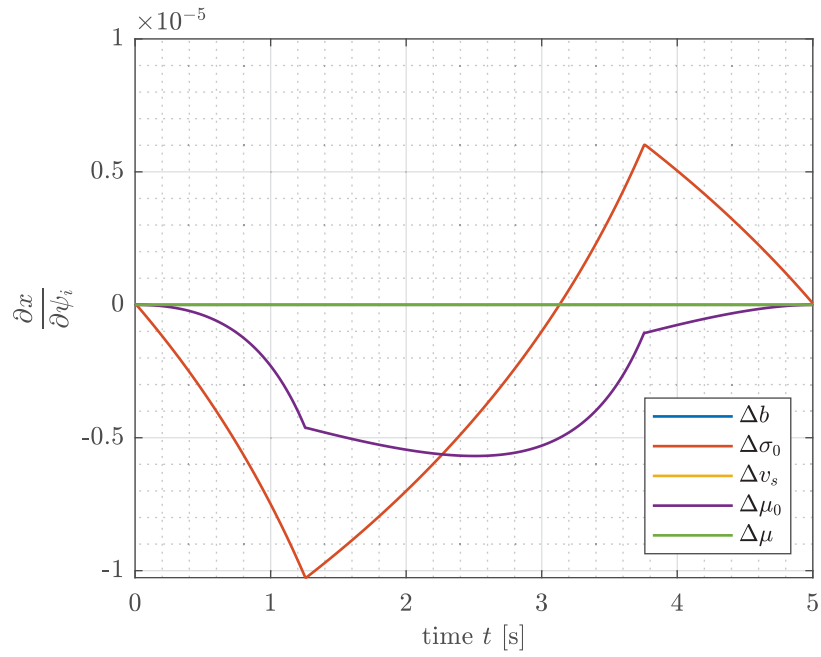


Figure 4.31: Numerically approximated Jacobian for a triangular input signal $u(t)$ with $\hat{u} = 0.5\mu_0 mg$.

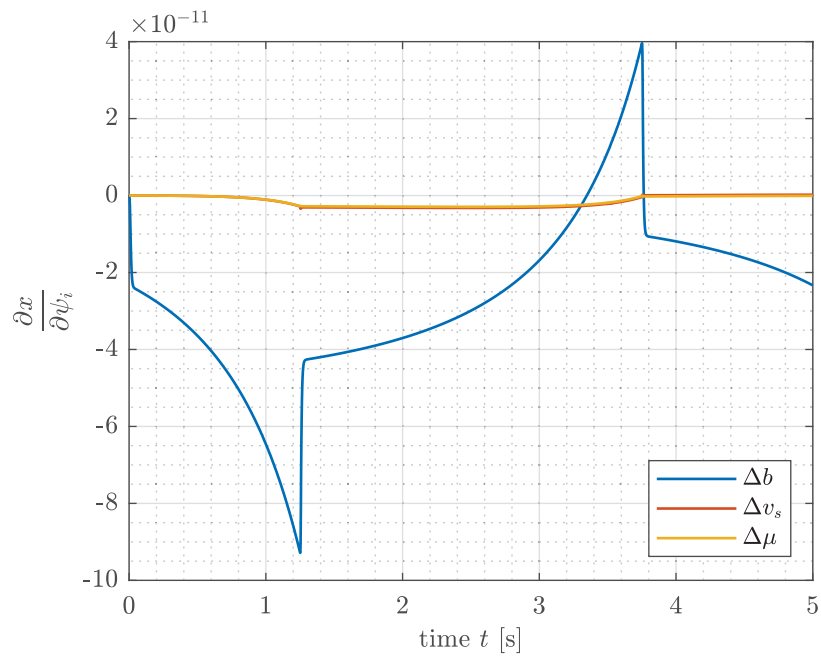


Figure 4.32: Selected components of the numerically approximated Jacobian for a triangular input signal $u(t)$ with $\hat{u} = 0.5\mu_0 mg$.

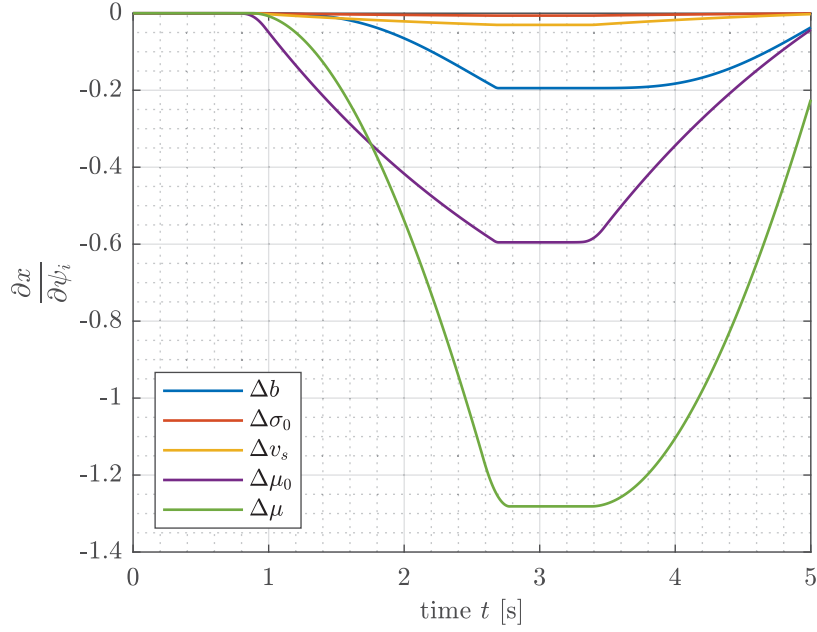


Figure 4.33: Numerically approximated Jacobian for a triangular input signal $u(t)$ with $\hat{u} = 1.5\mu_0 mg$.

4.3.4 System Identification of the LuGre Friction Model

The system under investigation is the same as described on section 4.3.3. As σ_1 is assumed to be dependent on m and σ_0 via the relation of critical damping (4.93) it is excluded from the set of parameters that are to be identified. The mass m is also assumed to be known. Therefore the parameter set that is to be identified is

$$\psi = [b \quad \sigma_0 \quad v_s \quad \mu_0 \quad \mu]^T. \quad (4.95)$$

with

$$\psi_{\text{exct}} = \begin{bmatrix} b_{\text{exct}} \\ \sigma_{0,\text{exct}} \\ v_{s,\text{exct}} \\ \mu_{0,\text{exct}} \\ \mu_{\text{exct}} \end{bmatrix} = \begin{bmatrix} 0.4 \text{ kg/s} \\ 1 \times 10^5 \text{ N/m} \\ 1 \times 10^{-2} \text{ m/s} \\ 0.15 \\ 0.10 \end{bmatrix} \quad (4.96)$$

System Identification Based on a Single Data Set ξ_A : Pre-Sliding Displacement

In the previous section the cyclic behaviour of the nonlinear friction model was investigated. Testing has shown that attempting the system identification based a corresponding input-output data set results in problems in the convergence of the nonlinear curve fitting problem. Therefore, in this section a slightly simplified excitation force $u(t)$, that only points in one direction is selected for the experimental design. This eliminates zero crossings from the state trajectory of the position $x(t)$.

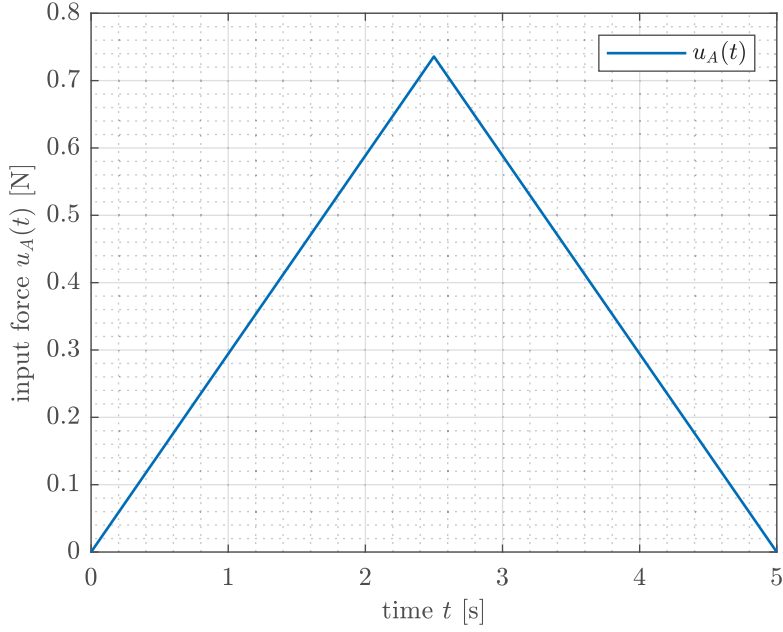


Figure 4.34: Triangular input signal $u_A(t)$ with $\hat{u} = 0.5\mu_0mg$ that is applied to the LuGre friction model on order to generate identification data.

An identification data set ξ_A is generated based on a triangular input signal $u_A(t)$ consisting of a single peak with amplitude $\hat{u}_A = 0.5\mu_{0,exct}mg$ and a period of five seconds as shown in Fig. 4.34. This input force will not be high enough to overcome the static friction. Therefore the movement $x(t)$ is limited to pre-sliding displacement.

For the identification the system response is sampled every $t_{step} = 0.2$ s. The response of an arbitrarily parametrized system to the described input $u_A(t)$ at these points in time is denoted as $y_A(\psi)$. The identification data set ξ_A is generated by adding gaussian measurement noise δ_A , which is sampled from the normal distribution $\mathcal{N}(0, 2.5 \times 10^{-7})$, to the response of the exactly parametrized system:

$$\xi_A = y_A(\psi_{exct}) + \delta_A. \quad (4.97)$$

The resulting data points can be seen in Fig. 4.34. The initial guess for the system parameters is selected as

$$\psi_0 = \left[0.3 \text{ kg/s} \quad 0.75 \times 10^5 \text{ N/m} \quad 2 \times 10^{-2} \text{ m/s} \quad 0.12 \quad 0.12 \right]^T. \quad (4.98)$$

As the system response of to the input $u_A(t)$ lies on a reactively small scale, the objective function is defined as

$$\varepsilon_A(\psi) = \|W_A (\xi_A - y_A(\psi))\|_2^2, \quad (4.99)$$

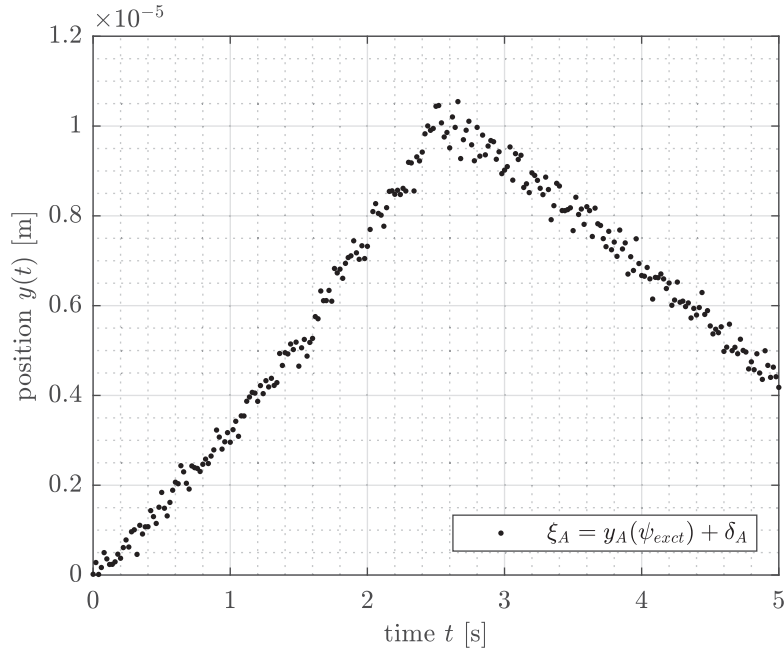


Figure 4.35: Identification data set generated by adding noise to the sampled response of the exactly parameterized system to the input signal $u_A(t)$.

where W_A is a diagonal weighting matrix

$$W_A = \frac{1}{\max(\xi_A)} I_N \quad (4.100)$$

with I_N as a unity matrix of size N , the number of data points. Additionally, in order to further improve the convergence of the numerical optimization the parameter space was normalized by a linear transformation with the following mapping:

$$[\mathbf{0}, \psi_{exact}] \rightarrow [\mathbf{0}, \mathbf{1}]. \quad (4.101)$$

Applying a Levenberg-Marquardt algorithm results in the identified parameter set

$$\psi_A = \begin{bmatrix} 0.30059575 \text{ kg/s} \\ 99889.258 \text{ N/m} \\ 0.010017185 \text{ m/s} \\ 0.15033127 \\ 0.12001962 \end{bmatrix}. \quad (4.102)$$

As can be seen, the bristle stiffness σ_0 , the Stribeck velocity v_s and the static friction coefficient μ_0 are identified accurately. However, this can not be said for the identified values of the viscous friction coefficient b and the dynamic friction coefficient μ .

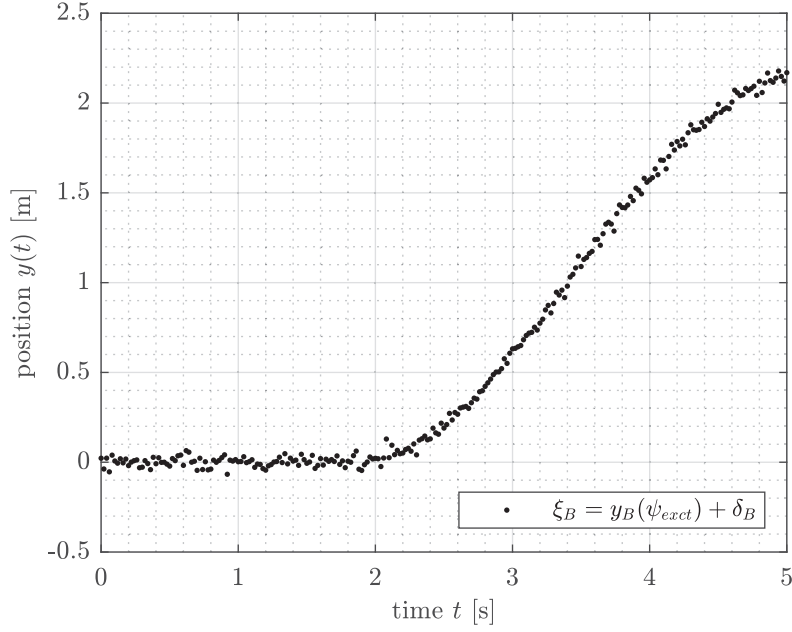


Figure 4.36: Identification data set generated by adding noise to the sampled response of the exactly parametrized system to the input signal $u_B(t)$.

System Identification Based on a Single Data Set ξ_B : Macro Level Displacement

A second identification data set ξ_B is generated based on a triangular input signal $u_B(t)$. The input has the same form as $u_A(t)$, which was shown in Fig. 4.34, consisting of a single peak but with larger amplitude $\hat{u}_B = 1.5\mu_{0,exct}mg$, which will overcome the static friction and lead to macro level displacements. The input is again applied over a period of five seconds.

As previously, for the identification the system response is sampled every $t_{step} = 0.2$ s. The response of an arbitrarily parametrized system to the described input $u_B(t)$ at these points in time is denoted as $\mathbf{y}_B(\boldsymbol{\psi})$. In order to achieve a certain comparability to the results of the system identification based on the pre-sliding experiment, a similar signal to noise ratio is selected. The identification data set ξ_B is generated by adding gaussian measurement noise δ_B , which is sampled from the normal distribution $\mathcal{N}(0, 2.5 \times 10^{-2})$, to the response of the exactly parametrized system:

$$\xi_B = \mathbf{y}_B(\boldsymbol{\psi}_{exct}) + \delta_B. \quad (4.103)$$

The resulting data points can be seen in Fig. 4.36. The initial guess for the system parameters is selected as $\boldsymbol{\psi}_0$ as before. The objective function that needs to be optimized in order to solve this system identification problem is defined as

$$\varepsilon_B(\boldsymbol{\psi}) = \|\mathbf{W}_B(\xi_B - \mathbf{y}_B(\boldsymbol{\psi}))\|_2^2, \quad (4.104)$$

where W_B is a diagonal weighting matrix

$$W_B = \frac{1}{\max(\xi_B)} I_N \quad (4.105)$$

with I_N as a unity matrix of size N , the number of data points. The same rescaling of the parameter space as before was applied again. Applying a Levenberg-Marquardt algorithm to minimize $\epsilon_B(\psi)$ results in the identified parameter set

$$\psi_B = \begin{bmatrix} 0.45650327 \text{ kg/s} \\ 67374.558 \text{ N/m} \\ 0.010472251 \text{ m/s} \\ 0.15327804 \\ 0.09514150 \end{bmatrix}. \quad (4.106)$$

For this setup, the viscous friction coefficient b was identified more accurately, as can be expected for a data set corresponding to a movement at higher velocities. The dynamics of the bristle, which have only minor effects on the macro level motion of the system, represented by the bristle stiffness σ_0 are overshadowed by the measurement noise. Therefore this parameter was not identified very accurately. The static as well as the dynamic friction coefficient μ_0 and μ were both identified with reasonable precision.

System Identification Based on Multiple Response Data Sets: ξ_A and ξ_B

As has already been discussed in section 4.1.5 multiple identification data sets can be used simultaneously in order to make a more accurate prediction of the exact system parameters. In this section the observations of micro level displacements ξ_A and of macro level displacements ξ_B are utilized and the feasibility of overcoming the shortcomings in the identified parameter sets when using those data sets individually is investigated.

The objective function for the identification of the parameters ψ based on two data sets can be defined as

$$\epsilon_{AB}(\psi) = \left\| \begin{bmatrix} \xi_A \\ \xi_B \end{bmatrix} - \begin{bmatrix} y_A(\psi) \\ y_B(\psi) \end{bmatrix} \right\|_2^2. \quad (4.107)$$

As the system responses to $u_A(t)$ and $u_B(t)$ lie on a completely different scale an individual weighting for the corresponding data sets is added to the objective function.

$$\epsilon_{AB}(\psi) = \left\| \begin{bmatrix} W_A & 0 \\ 0 & W_B \end{bmatrix} \left(\begin{bmatrix} \xi_A \\ \xi_B \end{bmatrix} - \begin{bmatrix} y_A(\psi) \\ y_B(\psi) \end{bmatrix} \right) \right\|_2^2 \quad (4.108)$$

with the same weighting matrices W_A and W_B , that were defined in the previous sections. Again, a Levenberg-Marquardt Algorithm was used to minimize this objective function,

which resulted in the identified parameter set

$$\psi_{AB} = \begin{bmatrix} 0.49962043 \text{ kg/s} \\ 104811.425 \text{ N/m} \\ 0.083936576 \text{ m/s} \\ 0.13775571 \\ 0.09121928 \end{bmatrix}. \quad (4.109)$$

Compared to the exact parameters, these results are relatively good estimates of σ_0 , μ_0 and μ , a less accurate identified v_s and b .

The resulting estimated system output for ψ_A , ψ_B and ψ_{AB} compared to the identification data and the system response with exact parametrization ψ_{exct} can be seen in Fig. 4.37 for $u_A(t)$ and Fig. 4.38 for $u_B(t)$. While the parameter set ψ_{AB} identified based on both data sets ξ_A and ξ_B does not produce the best performance in the test case A , it is still a major improvement compared to the evaluation of the test case A with the parameter set ψ_B or ψ_A in the test case B .

To summarize, the case study of the LuGre friction model clearly demonstrates the possible difficulty that can be encountered when only limited identification data is available for the system identification of nonlinear dynamic systems. The correct estimation of the model parameters is not necessarily only a question of achieving a high enough excitation in the experiments that are used to extract information about the object of interest. It is rather dependent on getting enough data at the right levels of system excitation. All in all it was shown that different identification data sets can be combined in a system identification procedure in order to be able to compute an improved estimate of all the model parameters.

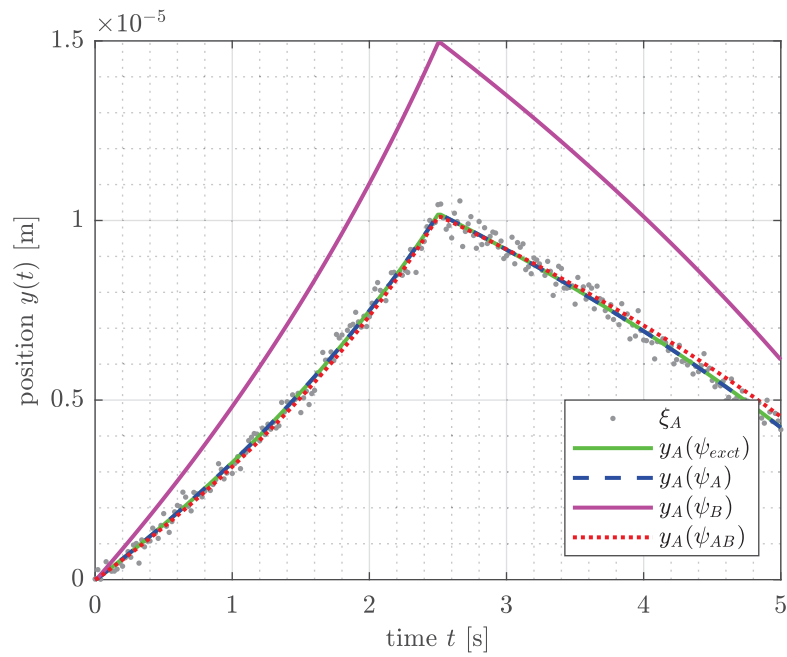


Figure 4.37: Comparison of the system response to the input signal $u_A(t)$ for the exact and the identified parameter sets.

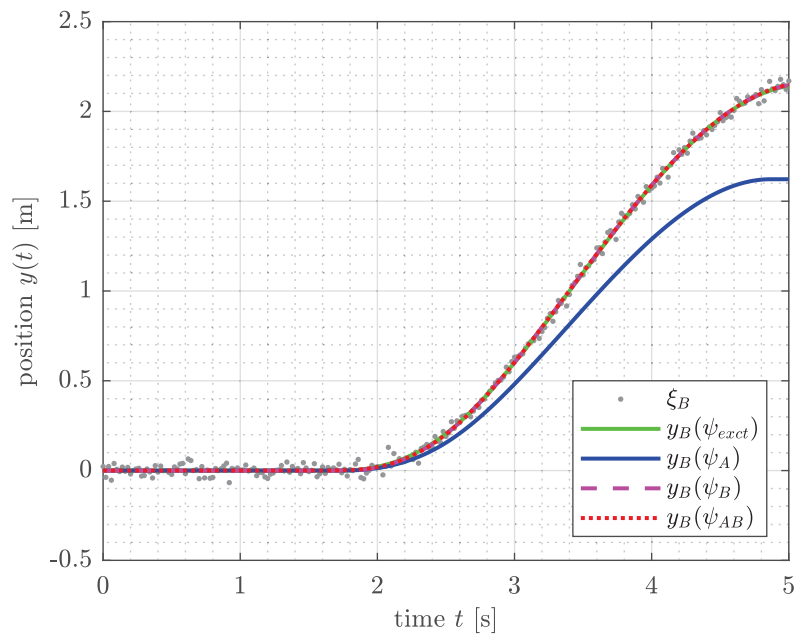


Figure 4.38: Comparison of the system response to the input signal $u_B(t)$ for the exact and the identified parameter sets.

Chapter 5

Conclusion

System identification of nonlinear dynamic systems is a challenging task because dealing with nonlinear systems makes the system identification process more demanding compared to linear dynamic systems. Both the problem of solving the differential equations as well as the curve fitting problem of matching the obtained solution to the measured data become more difficult. This is due to the fact that nonlinear dynamic system may show a wide range of different behaviours depending on the parametrization as well as the applied excitation. On the one hand, small changes in some parameters might lead in large deviations of the system output, which makes it detrimental to have a good guess about the probable range of the parameters. In most cases it should be possible to use a priori knowledge about the system to select a good initial guess for the parameters. On the other hand, disturbance induced variations might dominate the identified value of other parameters. While the general approach of using the solution of the simulation problem as basis for a nonlinear curve fitting problem can be followed every time, any object of interest needs to be studied on a case by case basis. Appropriate experiments to extract observational information need to be defined and a fitting simulation model needs to be determined.

In this thesis the system identification of three nonlinear dynamic systems as presented in chapter 4 based on synthetic measurement data was performed successfully. For this task MATLAB® functions for the numerical integration of ODEs, mainly `ode45()`, in conjunction with numerical optimization methods like the Gauss-Newton, Levenberg-Marquardt and the Simplex method, the later of which is implemented in the MATLAB® function `fminsearch()`, were incorporated in an algorithm for the system identification based on the output error of the simulation model. Various aspects of the system identification problem with respect to those example problems were discussed and illustrated.

In the case study of the object in free fall subject to drag due to air the system identification of a single parameter - the friction coefficient - as well as the identification of this parameter and the initial conditions was performed. As has been shown, in both cases the

nonlinear curve fitting problem is well posed as the objective function appears to be of convex shape over a wide range of the parameter space around the global minimum. The influence of gaussian measurement noise on the identified parameters was investigated. This resulted in the conclusion that for this example problem, a gaussian measurement noise results in gaussian disturbance induced variation of the identified parameters. However, it is important to keep in mind that this will not be the case for every nonlinear system identification problem. Furthermore, the compensation of the disturbance induced variation by means of utilizing multiple identification data sets was investigated. This resulted in the conclusion that, based on the results on a Monte-Carlo simulation, the standard deviation of the identified parameters can be reduced by a factor of the inverse of the square root of the number of data sets incorporated into the system identification procedure.

In the case study of the nonlinear mass and spring system the utilization of observational data from two different experimental setups in the system identification of both the model parameters as well as the initial conditions of each experiment was investigated. This example problem shows that the inclusion of multiple different data sets not only leads to better results in terms of the disturbance induced variation of the identified parameters, but can also improve the convergence of the nonlinear curve fitting problem. In addition to that, the application of a multiple starting point procedure in order to find a way around local minima to a global minimizer was presented.

In the third case study, the system identification of the *LuGre* friction model demanded special attention to the experimental design, as the system exhibits a vastly different behaviour depending on the excitation level. As was shown, by combining identification data sets corresponding to micro and macro level excitations of the system, a proper estimation of all the model parameters can be achieved. Even though the technical feasibility of executing these proposed experiments on the same system might cause practical problems, the advantage of combining the information contained in multiple data sets in order to accurately identify a set of model parameters was clearly demonstrated.

Bibliography

- [1] M. Bunge. *Emergence and Convergence: Qualitative Novelty and the Unity of Knowledge*. Toronto studies in philosophy. University of Toronto Press, 2003.
- [2] Lennart Ljung and Torkel Glad. *Modeling of Dynamic Systems*. Prentice-Hall, Inc., USA, 1994.
- [3] Gene F. Franklin, Michael L. Workman, and Dave Powell. *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co., Inc., 3rd edition, 1997.
- [4] Brian Douglas. The fundamentals of control theory. Available at <https://www.engineeringmediallc.com/> (2020/01/14), 07 2019.
- [5] Donald E. Kirk. *Optimal Control Theory : An Introduction*. Dover Publications, April 2004.
- [6] Amar G. Bose. *A Theory of Nonlinear Systems*. PhD thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering, June 1956.
- [7] G. Rath and M. Harker. Global least squares solution for lti system response. In *2017 International Conference on Applied Electronics (AE)*, pages 1–4, Sep. 2017.
- [8] P. O’Leary and M. Harker. A framework for the evaluation of inclinometer data in the measurement of structures. *IEEE Transactions on Instrumentation and Measurement*, 61(5):1237–1251, May 2012.
- [9] M. Harker and G. Rath. Global least squares for time-domain system identification of state-space models. In *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–6, 2018.
- [10] M. Harker. *Fractional Differential Equations: Numerical Methods for Applications*. Studies in Systems, Decision and Control. Springer International Publishing, 2020.
- [11] Bernard Friedland. *Control System Design: An Introduction to State-Space Methods*. Dover Publications, 2005.

- [12] Lennart Ljung. Some aspects on nonlinear system identification. *IFAC Proceedings Volumes*, 39(1):553 – 564, 2006. 14th IFAC Symposium on Identification and System Parameter Estimation.
- [13] Johan Schoukens and Lennart Ljung. Nonlinear system identification: A user-oriented roadmap. *ArXiv*, abs/1902.00683, 2019.
- [14] L. A. Zadeh. From circuit theory to system theory. *Proceedings of the IRE*, 50(5):856–865, 1962.
- [15] José Borges, Vincent Verdult, Michel Verhaegen, and Miguel Ayala Botto. Separable least squares for projected gradient identification of composite local linear state-space models. 07 2004.
- [16] Klaus Schittkowski. *Numerical Data Fitting in Dynamical Systems: A Practical Introduction with Applications and Software*. Kluwer Academic Publishers, USA, 2002.
- [17] Rolf Isermann and Marco Münchhof. *Identification of Dynamic Systems: An Introduction with Applications*. Springer Publishing Company, Incorporated, 2014.
- [18] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. Cambridge University Press, USA, 1992.
- [19] W. Richard Kolk and Robert A. Lerman. *Analytic Solutions to Nonlinear Differential Equations*, pages 23–60. Springer US, Boston, MA, 1992.
- [20] R. Ashino, M. Nagase, and R. Vaillancourt. Behind and Beyond the Matlab ODE Suite. *Computers and Mathematics with Applications*, 40(4):491 – 512, 2000.
- [21] Eric W. Weisstein. Runge-kutta method. From MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/Runge-KuttaMethod.html> (2020/06/10).
- [22] Lawrence F. Shampine and Mark W. Reichelt. The matlab ode suite. *SIAM J. Sci. Comput.*, 18(1):1–22, January 1997.
- [23] Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. *Methods for Non-Linear Least Squares Problems (2nd ed.)*. 2004. PDF available at http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3215/pdf/imm3215.pdf (2020/02/11).
- [24] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168, jul 1944.

- [25] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [26] C.T. Kelley. *Iterative Methods for Optimization*. Frontiers in Applied Mathematics. Society for Industrial and Applied Mathematics, 1999. PDF available at https://archive.siam.org/books/textbooks/fr18_book.pdf (2020/01/17).
- [27] Henri P. Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. Available at <http://people.duke.edu/~hpgavin/ce281/lm.pdf> (2019/12/06), 2013.
- [28] John A. Nelder and Roger Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
- [29] Jeffrey Lagarias, James Reeds, Margaret Wright, and Paul Wright. Convergence properties of the nelder–mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9:112–147, 12 1998.
- [30] G.H. Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on Numerical Analysis*, 10:413–432, 04 1973.
- [31] Gene Golub and Victor Pereyra. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Problems*, 19:R1–R26(1), 01 2003.
- [32] Dianne P. O’Leary and Bert W. Rust. Variable projection for nonlinear least squares problems. *Computational Optimization and Applications*, 54(3):579–593, Apr 2013.
- [33] J. Bruls, C.T. Chou, B.R.J. Haverkamp, and M. Verhaegen. Linear and non-linear system identification using separable least-squares. *European Journal of Control*, 5(1):116 – 128, 1999.
- [34] Harold T. Davis. *Introduction to Nonlinear Differential and Integral Equations*. Dover, New York, 1962.
- [35] H.K. Khalil. *Nonlinear Control, Global Edition*. Pearson Education Limited, 2015.
- [36] B. Armstrong-Hélouvry. *Control of Machines with Friction*. The Springer International Series in Engineering and Computer Science. Springer US, 1991.
- [37] V. van Geven. A study of friction models and friction compensation. Available at <http://www.mate.tue.nl/mate/pdfs/11194.pdf> (2019/12/18), 2009. traineeship report, Technical University Eindhoven, Department Mechanical Engineering, Dynamics and Control Technology Group.

- [38] Andreas Krämer and Joachim Kempkes. Modellierung und simulation von nicht-linearen reibungseffekten bei der lageregelung von servomotoren. *FHWS Science Journal*, 1(2):47 – 57, 2013.
- [39] Carlos Canudas de Wit, Henrik Olsson, Karl Johan Åström, and Pablo Lischinsky. A new model for control of systems with friction. *IRE Transactions on Automatic Control*, 40(3), 1995.
- [40] T. Piatkowski. Dahl and lugre dynamic friction models — the analysis of selected properties. *Mechanism and Machine Theory*, 73:91 – 100, 2014.