Servo-constraints for Inversion of Underactuated Multibody Systems

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieurin (Dr.-Ing.)

genehmigte Dissertation

von
Svenja Drücker

aus
Hamburg

2022

1. Gutachter: Prof. Dr.-Ing. Robert Seifried
2. Gutachter: Prof. Dr. Olivier Brüls

Tag der mündlichen Prüfung: 03. 12. 2021

MuM Notes in Mechanics and Dynamics

Editor: Prof. Dr.-Ing. Robert Seifried
Hamburg University of Technology
Institute of Mechanics and Ocean Engineering (MuM)
www.tuhh.de/mum

Volume 5
Svenja Drücker
“Servo-constraints for Inversion of Underactuated Multibody Systems”
Hamburg, 2022

DOI: 10.15480/882.4089
ORCID: https://orcid.org/0000-0002-2565-4022

Preface

This work results from research conducted at the Institute of Mechanics and Ocean Engineering at Hamburg University of Technology. Many people have contributed to this work in one way or another.

First of all, I would like to thank Prof. Robert Seifried for introducing me to this fascinating research topic. His continuous support and guidance are the basis of this work. Moreover, I would like to thank Prof. Olivier Brüls for giving me the opportunity to stay at his lab in Liège and for acting as second reviewer. Our discussions and his valuable comments had a great impact on this work. I am also grateful to Prof. Otto von Estorff for chairing the examination committee.

The presented research was only possible because of the friendly and collegial atmosphere at the Institute of Mechanics and Ocean Engineering. All colleagues contributed to this working environment and I am grateful for the mutual support, countless coffee breaks and discussions about everything and anything. I learned so much during this memorable time. Thank you all.

Finally, my family was there for me whenever encouragement, motivation or distraction was necessary. I am especially grateful to my husband for his endless patience and support.

Hamburg, January 2022

Svenja Drücker

To my family.

Contents

Abstract

Accurate inverse models are essential for the control of underactuated multibody systems. The servo-constraints approach is applied here, which formulates the inverse model problem as a set of differential-algebraic equations (DAEs). These DAEs have different properties depending on the system type. In this work, the servo-constraints approach is analyzed for all three possible system types. For differentially flat systems, experimental results demonstrate the real-time capabilities of the approach. For minimum phase systems, various integration methods are compared with respect to efficiency and accuracy. For non-minimum phase systems, a simplification of stable inversion is proposed and an output redefinition strategy is introduced for flexible bodies.

Introduction

Industrial manufacturers have an increasing demand for higher manufacturing accuracies, higher process velocities, more efficiency and extensive automation. The following system designs contribute to these goals. Light and slender machines enable higher process velocities and less energy consumption compared to conventional machines [DwivedyEberhard06]. Introducing compliance to industrial robots enables human-robot interaction and safe incorporation of machines into human work space [BicchiPeshkinColgate08]. Cable manipulators combine the dexterity of industrial robots with the ability to carry high payloads [HeydenWoernle06]. The aforementioned design trends usually result in underactuated systems with more degrees of freedom than independent control inputs. With advancing machine design, the software and control component must be developed simultaneously to adapt to new system properties. Therefore, efficient control strategies for underactuated systems are subject to current research [LiuYu13, Seifried14].

1.1 Background and Motivation

Accurate mathematical models are essential for analysis, simulation and controller design of mechanical systems. Multibody system dynamics is a suitable method for modeling rigid mechanical systems [Schiehlen14] and is frequently used for robotic applications [Spong06] and machine dynamics [Hollburg07]. However, the rigidity assumption might not be fulfilled for light and slender structures undergoing elastic deformations. In that case, flexibility must be taken into account in the mathematical model, for example by applying methods from flexible multibody system dynamics [Bauchau11].
Flexible systems usually belong to the class of underactuated multibody systems, since the elastic deformation often cannot be actuated directly. Therefore, the systems possess more degrees of freedom than independent control inputs. Generally, underactuation can arise due to several reasons, such as flexibility, intentional design or actuator failure [Seifried14]. Control of underactuated systems is challenging and ongoing research [LiuYu13, Seifried14].
Two design degree of freedom control is an efficient control strategy for trajectory tracking [Skogestad04]. It consists of a feedforward controller responsible for trajectory tracking and a feedback controller responsible for disturbance rejection. Ideally, the feedforward controller is designed as an inverse model of the

system to cancel out all known dynamics. If the model described the real system perfectly, the feedforward control would result in perfect tracking of the desired trajectory. However, there are always parameter uncertainties, unmodeled dynamics or other disturbances [ThrunBurgardFox06]. A feedback controller is therefore added to stabilize the system around the desired trajectory.

This work focuses on the development of accurate inverse models for underactuated multibody systems. The use of accurate inverse models in the feedforward path results in small tracking errors, which has several advantages. For example, it reduces complexity of the feedback controller design and linear control strategies can be applied. Moreover, noise effects in the feedback loop can be reduced.

Theoretically, inverse models can be computed analytically with methods from nonlinear control theory [Isidori96, Sastry99]. However, this approach is quite burdensome for complex multibody systems. Alternatively, the method of servo-constraints is applied in this work, since it is applicable to general complex systems. Servo-constraints are defined to constrain the system output to be equal to the desired trajectory. They append the equations of motions to form a set of differential-algebraic equations (DAEs), which can be solved numerically for the inverse model [BlajerKołodziejczyk04].

Regarding the inverse model properties, dynamic systems are divided into three classes. Differentially flat systems do not have internal dynamics and the inverse model is purely algebraic [FliessEtAl95]. Differentially flat systems include e.g. cable robots, undamped mass-spring chains and simple vehicle models [MurrayRathinamSluis95]. Minimum phase systems have a dynamic inverse model with stable dynamics [Sastry99]. This property is typical for damped mass-spring chains or flexible joint manipulators [deLucaBook08]. Non-minimum phase systems have a dynamic inverse model with unstable internal dynamics [Sastry99]. This is typical for flexible link manipulators [deLucaBook08]. The system class must be carefully analyzed to choose a suitable solving method for the servo-constraints approach. For differentially flat systems and minimum phase systems, the inverse model DAEs can be integrated forward in time. This is ideally done in real-time on the real system, since the feedforward controller is then able to react to changes in the trajectory or the model itself. For non-minimum phase systems, the set of DAEs cannot be solved forward in time, but a boundary value problem [ChenPaden96] or an optimization problem [BastosSeifried-Brüls13] must be solved.

1.2 Aim of the Work

The aim of this work is to extend the current use of servo-constraints to real-time applications and to more complex mechanical systems. In other words, it is desired to make the servo-constraints approach more suitable for real world problems and to decrease the gap towards industrial application. These goals

are reached by looking at the framework of servo-constraints in a comprehensive manner for all three system types.
For differentially flat systems, real-time capabilities of the approach are analyzed in detail. To this end, comparative studies are given to point out suitable solvers [OttoSeifried18a]. Real-time application of first order as well as higher order integrators is demonstrated in experiments for a large scale cable robot [OttoSeifried18c]. These results contribute to the application of servo-constraints to standard industrial machines in the future.
The analysis of suitable solvers also enables an efficient inversion of minimum phase systems. Therefore, it is possible to consider systems with more degrees of freedom than examined so far. It is shown in simulations for a three-dimensional manipulator that real-time inversion is possible. Thus, the method of servo-constraints is advanced towards real-time integration of complex minimum phase systems.
For non-minimum phase systems, a simplification of the boundary conditions for stable inversion is proposed [DrückerSeifried20]. This simplification makes it possible to consider multibody systems with more flexible elements than studied so far in the context of stable inversion. This extension is demonstrated by inverting a highly flexible manipulator modeled by the absolute nodal coordinate formulation (ANCF). Additionally, a systematic approach for output redefinition is proposed for flexible manipulators, which renders the internal dynamics stable and simplifies the solution process. This makes it possible to treat the flexible manipulator as a minimum phase system and further contributes to the goal of inverting complex multibody systems in real-time.

1.3 Structure of the Work

In Chap. 2, rigid multibody system dynamics as well as the ANCF are presented to model rigid and highly flexible bodies respectively. The mathematical models serve as a basis for controller design in the subsequent chapters. There always exists a trade-off between model accuracy and complexity. It is proposed in this work to use rather complex models for system inversion to obtain accurate feedforward controllers. Complex models can be considered conveniently in the framework of servo-constraints. On the other hand, a more simple model can be used for feedback control design to reduce controller complexity. The chapter is completed with the introduction of application examples, which are used throughout the work to point out relevant properties and aspects of the inversion process.
Several methods for model inversion are presented in Chap. 3. Classical system inversion based on nonlinear control theory, flatness based inversion and inversion based on servo-constraints are considered. Inverse models based on servo-constraints are the focus of this work, since they yield a general approach which is applicable to complex underactuated multibody systems. Analysis methods are presented to classify models with respect to the three system classes of dif-

ferentially flat systems, minimum phase and non-minimum phase systems. For each class, special care must be taken in the selection of suitable DAE solvers. Typical solvers are presented, which are usually applied in the context of servo-constraints. The chapter concludes with a classification of the application examples. The following three chapters are devoted to each of the three system classes.
Chapter 4 concerns differentially flat systems. An undamped torsional oscillator and two cable robot models are analyzed in detail. Numerical studies compare different formulations of the inverse model DAEs. Experimental results for the cable robot demonstrate the real-time capabilities of servo-constraints. Moreover, typical combinations of feedforward control with feedback control are validated in experiments. Thereby, a linear quadratic regulator and sliding mode control are used in the feedback path.
Chapter 5 covers minimum phase systems. Since these systems possess internal dynamics, suitable integration methods must be selected with care. An application example with a combination of first order and second order internal dynamics is analyzed and can serve as a benchmark problem for solver selection. Moreover, the method of servo-constraints is applied to a four-link manipulator with passive joint to demonstrate application to a complex three-dimensional system. Comparative studies of the integration methods are presented.
Chapter 6 discusses non-minimum phase systems. Due to unstable internal dynamics, the inverse models cannot be integrated forward in time. Alternatively, stable inversion is introduced, which defines a boundary value problem to obtain a bounded solution for the inverse model. So far, complex definitions of the boundary conditions restrict the approach to simple academic examples. Simplified boundary conditions are proposed and convergence is validated for a manipulator with one passive joint. Due to these simplifications, more complex systems can be considered for model inversion. This is demonstrated for a highly flexible manipulator modeled by the ANCF. Nevertheless, stable inversion remains a burdensome approach due to the underlying boundary value problem. Hence, a systematic approach is proposed for highly flexible manipulators to redefine the system output, such that the internal dynamics becomes stable. Then, the inverse model can be integrated forward in time, while the original output is tracked approximately.
The presented work is summarized in Chap. 7 and future research directions are pointed out.

Chapter 2

Modeling of Multibody Systems

Accurate mathematical models are essential for analysis, simulation, optimization and controller-design of the underlying mechanical system. Multibody system dynamics is a method for modeling mechanical systems undergoing large translational and rotational motion [Schiehlen14, Seifried14]. It is applied in various fields, such as robotics [Spong06], vehicle dynamics [Popp10], biomechanics [Ambrosio13], machine dynamics [Hollburg07] and others.

Multibody systems consist of rigid and flexible bodies, which are connected by constraint elements and coupling elements [Schiehlen14]. Constraint elements include joints and other position-dependent connectors, while coupling elements include springs, dampers and other force elements. Models are based on certain approximations and assumptions. In rigid multibody dynamics, it is assumed that the bodies do not deform. Efficient simulation is usually possible for rigid multibody systems because they have comparably few degrees of freedom. However, the rigidity assumption might not be valid for lightweight machines or fast-moving robots. In these systems, the structural flexibility might influence the global motion and flexible multibody dynamics is applied [Bauchau11, Géradin01]. There exist different methods for modeling structural flexibility, e.g. the floating frame of reference approach [Schwertassek99], geometrically exact finite element formulations [Simo85] or the ANCF [Shabana11].

Underactuated multibody systems are a subclass of multibody systems. They are defined as systems with less system inputs than degrees of freedom [Seifried14]. Underactuation can arise accidentally due to actuator failure, but it can also be introduced intentionally to enable certain motion principles. This is the case for cable-driven manipulators and crane systems [CamarilloEtAl08, OttoSeifried18c, Woernle13]. Moreover, flexible multibody systems are inherently underactuated, because the elasticity usually cannot be controlled directly. This is for example the case for flexible joint manipulators [Spong87, deLucaBook08] and flexible link manipulators [BurkhardtSeifriedEberhard15, deLucaBook08]. Further examples of underactuated multibody systems are e.g. given in [LiuYu13]. Regarding controller design, the properties of underactuated systems are quite different compared to fully actuated systems. For example, it is not possible to control all degrees of freedom independently. This makes control problems and model inversion a challenging problem [LiuYu13, Seifried14, Spong98].

In this chapter, rigid multibody system dynamics is briefly introduced first. This includes the equations of motion using generalized coordinates as well as redundant coordinates. Then, the ANCF is presented for modeling flexible bodies. Moreover, high-level actuator models for describing system inputs are introduced. Finally, application examples are presented, which are analyzed in the subsequent chapters.

2.1 Rigid Multibody System Dynamics

In rigid multibody system dynamics, the bodies are assumed to be rigid, such that they do not undergo any elastic deformations. This assumption is valid if internal forces due to structural deformations are small with respect to external forces acting on the body.
In the following, the equations of motion are presented for holonomic rigid multibody systems modeled using generalized coordinates as well as redundant coordinates. Afterwards, linearizations of the equations of motion are introduced. The framework is presented according to [Schiehlen14].

2.1.1 Multibody Dynamics in ODE Form

A multibody system with n_f degrees of freedom is considered. In case there are no kinematic loops, a generalized coordinate vector $\boldsymbol{y} \in \mathbb{R}^{n_\mathrm{f}}$ and a generalized velocity vector $\boldsymbol{v} \in \mathbb{R}^{n_\mathrm{f}}$ can be defined to describe the motion of the system. The n_u system inputs are collected in the vector $\boldsymbol{u} \in \mathbb{R}^{n_\mathrm{u}}$.
In order to obtain the equations of motion, Newton's and Euler's law are first derived for each body. Then, d'Alemberts principle is inserted to eliminate the reaction forces. This approach is formalized in the Newton-Euler formalism [Schiehlen14]. The equations of motion arise as ordinary differential equations (ODEs) in the form

$$\dot{\boldsymbol{y}} = \boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} \tag{2.1}$$

$$\boldsymbol{M}(\boldsymbol{y},t)\dot{\boldsymbol{v}} + \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) = \boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y})\boldsymbol{u} \tag{2.2}$$

with $\boldsymbol{Z} : \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$ describing the kinematic relationship between the generalized positions $\boldsymbol{y}$ and velocities $\boldsymbol{v}$, a positive definite and symmetric mass matrix $\boldsymbol{M} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$, the Coriolis, centrifugal and gyroscopic forces $\boldsymbol{k} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{f}}$, the applied forces $\boldsymbol{q} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{f}}$ and the input distribution matrix $\boldsymbol{B} : \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{u}}$. Note that most of the time, the generalized velocities are selected as $\boldsymbol{v} = \dot{\boldsymbol{y}}$ and the kinematics matrix reduces to $\boldsymbol{Z} = \boldsymbol{I}_{n_\mathrm{f}}$ with identity $\boldsymbol{I}_{n_\mathrm{f}}$ of dimension n_f.
For controller design, the system output $\boldsymbol{z} \in \mathbb{R}^{n_\mathrm{u}}$ is assumed to be of the same dimension as the system input and is defined as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{y}) \tag{2.3}$$

with $\boldsymbol{h} : \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{u}}$.

2.1.2 Multibody Dynamics in DAE Form

It might be convenient to omit the use of generalized coordinates and to use redundant coordinates instead. This choice of coordinates can simplify the modeling process. For example, in case the multibody system has a kinematic loop, it is not straightforward to select generalized coordinates to describe the motion of the system.
Then, redundant coordinates $\boldsymbol{y} \in \mathbb{R}^{n_\mathrm{y}}$, $\boldsymbol{v} \in \mathbb{R}^{n_\mathrm{y}}$ with $n_\mathrm{y} = n_\mathrm{f} + n_\mathrm{c}$ can be used to describe the constrained dynamics of a multibody system with n_f degrees of freedom and n_c constraints. The equations of motion arise as DAEs of the form

$$\dot{\boldsymbol{y}} = \boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} \tag{2.4}$$

$$\boldsymbol{M}(\boldsymbol{y},t)\dot{\boldsymbol{v}} + \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) = \boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{C}(\boldsymbol{y},t)^\mathsf{T}\boldsymbol{\lambda} + \boldsymbol{B}(\boldsymbol{y})\boldsymbol{u} \tag{2.5}$$

$$\boldsymbol{c}(\boldsymbol{y},t) = \boldsymbol{0} \tag{2.6}$$

with $\boldsymbol{Z} : \mathbb{R}^{n_\mathrm{y}} \to \mathbb{R}^{n_\mathrm{y}\times n_\mathrm{y}}$ describing the kinematic relationship between the positions $\boldsymbol{y}$ and velocities $\boldsymbol{v}$, a positive definite and symmetric mass matrix $\boldsymbol{M} : \mathbb{R}^{n_\mathrm{y}}\times\mathbb{R} \to \mathbb{R}^{n_\mathrm{y}\times n_\mathrm{y}}$, the Coriolis, centrifugal and gyroscopic forces $\boldsymbol{k} : \mathbb{R}^{n_\mathrm{y}}\times\mathbb{R}^{n_\mathrm{y}}\times\mathbb{R} \to \mathbb{R}^{n_\mathrm{y}}$, the applied forces $\boldsymbol{q} : \mathbb{R}^{n_\mathrm{y}}\times\mathbb{R}^{n_\mathrm{y}}\times\mathbb{R} \to \mathbb{R}^{n_\mathrm{y}}$ and the input distribution matrix $\boldsymbol{B} : \mathbb{R}^{n_\mathrm{y}} \to \mathbb{R}^{n_\mathrm{y}\times n_\mathrm{u}}$. The Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^{n_\mathrm{c}}$ enforce the constraints $\boldsymbol{c} : \mathbb{R}^{n_\mathrm{y}}\times\mathbb{R} \to \mathbb{R}^{n_\mathrm{c}}$ with the constraint gradient $\boldsymbol{C} : \mathbb{R}^{n_\mathrm{y}}\times\mathbb{R} \to \mathbb{R}^{n_\mathrm{c}\times n_\mathrm{y}}$. A regular constraint gradient $\boldsymbol{C}$ is assumed, i.e. all constraints act in independent directions. For notational simplicity, the same notation is used for generalized coordinates as well as redundant coordinates. The specific choice of coordinates is referenced in the subsequent chapters and examples.
For later reference, the derivatives of the algebraic constraints $\boldsymbol{c}(\boldsymbol{y},t)$ are

$$\dot{\boldsymbol{c}}(\boldsymbol{y},\boldsymbol{v},t) = \underbrace{\frac{\partial \boldsymbol{c}(\boldsymbol{y},t)}{\partial \boldsymbol{y}}\boldsymbol{Z}(\boldsymbol{y})}_{\boldsymbol{C}(\boldsymbol{y},t)}\boldsymbol{v} + \frac{\partial \boldsymbol{c}(\boldsymbol{y},t)}{\partial t} = \boldsymbol{0}\,, \tag{2.7}$$

$$\ddot{\boldsymbol{c}}(\boldsymbol{y},\boldsymbol{v},\dot{\boldsymbol{v}},t) = \boldsymbol{C}(\boldsymbol{y},t)\dot{\boldsymbol{v}} + \dot{\boldsymbol{C}}(\boldsymbol{y},\boldsymbol{v},t)\boldsymbol{v} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \boldsymbol{c}(\boldsymbol{y},t)}{\partial t}\right) = \boldsymbol{0}\,. \tag{2.8}$$

For controller design, the system output $\boldsymbol{z} \in \mathbb{R}^{n_\mathrm{u}}$ is assumed to be of the same dimension as the system input and is defined as

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{y}) \tag{2.9}$$

with $\boldsymbol{h} : \mathbb{R}^{n_\mathrm{y}} \to \mathbb{R}^{n_\mathrm{u}}$.
Differential-algebraic equations, such as equations (2.4)–(2.6), are often classified by the differentiation index. The differentiation index is defined as the number of derivatives to be taken of the constraints $\boldsymbol{c}(\boldsymbol{y},t)$ until the equations can be transformed to ordinary differential equations [CampbellGear95, Hairer02]. There exist several other index concepts, which are compared and discussed in [CampbellGear95]. In the scope of this work, the index refers to the differentiation

index. It is possible to show that the equations of motion (2.4)–(2.6) have differentiation index 3 due to invertibility of the matrix $\boldsymbol{C}(\boldsymbol{y},t)\boldsymbol{M}(\boldsymbol{y},t)^{-1}\boldsymbol{C}(\boldsymbol{y},t)^{\mathsf{T}}$.

2.1.3 Linearization of the Equations of Motion

For linear controller design and local analysis of the equations of motion, they are linearized around a stationary linearization point [Schiehlen14, Seifried14]. The linearization is based on Taylor series and performed for both the equations of motion in ODE form as well as DAE form. Here, only autonomous systems are considered, such that the linearization around a stationary point yields linear time-invariant equations of motion.
First, the equations of motion (2.1)–(2.2) in ODE form are considered. The linearization is considered around the constant equilibrium point $\boldsymbol{y}_{\mathrm{eq}}, \boldsymbol{v}_{\mathrm{eq}}, \boldsymbol{u}_{\mathrm{eq}}$ defined by

$$\begin{bmatrix} \dot{\boldsymbol{y}}_{\mathrm{eq}} \\ \dot{\boldsymbol{v}}_{\mathrm{eq}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}(\boldsymbol{y}_{\mathrm{eq}})\boldsymbol{v}_{\mathrm{eq}} \\ \boldsymbol{M}(\boldsymbol{y}_{\mathrm{eq}})^{-1}\left(\boldsymbol{q}(\boldsymbol{y}_{\mathrm{eq}},\boldsymbol{v}_{\mathrm{eq}}) - \boldsymbol{k}(\boldsymbol{y}_{\mathrm{eq}},\boldsymbol{v}_{\mathrm{eq}}) + \boldsymbol{B}(\boldsymbol{y}_{\mathrm{eq}})\boldsymbol{u}_{\mathrm{eq}}\right) \end{bmatrix} = \boldsymbol{0}\,. \tag{2.10}$$

The linearized equations of motion for small values $\widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{y}_{\mathrm{eq}}$, $\widetilde{\boldsymbol{v}} = \boldsymbol{v} - \boldsymbol{v}_{\mathrm{eq}}$, $\widetilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_{\mathrm{eq}}$ read

$$\dot{\widetilde{\boldsymbol{y}}} = \boldsymbol{Z}_{\mathrm{lin}}^{\mathrm{y}}\,\widetilde{\boldsymbol{y}} + \boldsymbol{Z}_{\mathrm{lin}}\,\widetilde{\boldsymbol{v}} \tag{2.11}$$

$$\boldsymbol{M}_{\mathrm{lin}}\dot{\widetilde{\boldsymbol{v}}} + \boldsymbol{D}_{\mathrm{lin}}\widetilde{\boldsymbol{v}} + \boldsymbol{K}_{\mathrm{lin}}\widetilde{\boldsymbol{y}} = \bar{\boldsymbol{B}}_{\mathrm{lin}}\widetilde{\boldsymbol{u}} \tag{2.12}$$

with the constant matrices

$$\boldsymbol{Z}_{\mathrm{lin}}^{\mathrm{y}} = \left.\frac{\partial \boldsymbol{Z}\boldsymbol{v}}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}_{\mathrm{eq}},\boldsymbol{v}_{\mathrm{eq}}}, \tag{2.13}$$

$$\boldsymbol{Z}_{\mathrm{lin}} = \boldsymbol{Z}(\boldsymbol{y}_{\mathrm{eq}})\,, \tag{2.14}$$

$$\boldsymbol{M}_{\mathrm{lin}} = \boldsymbol{M}(\boldsymbol{y}_{\mathrm{eq}})\,, \tag{2.15}$$

$$\boldsymbol{D}_{\mathrm{lin}} = \left.\left(\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{v}} - \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{v}}\right)\right|_{\boldsymbol{y}_{\mathrm{eq}},\boldsymbol{v}_{\mathrm{eq}}}, \tag{2.16}$$

$$\boldsymbol{K}_{\mathrm{lin}} = \left.\left(\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{y}} - \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{y}} - \frac{\partial \boldsymbol{B}\,\boldsymbol{u}_{\mathrm{eq}}}{\partial \boldsymbol{y}}\right)\right|_{\boldsymbol{y}_{\mathrm{eq}},\boldsymbol{v}_{\mathrm{eq}}}, \tag{2.17}$$

$$\bar{\boldsymbol{B}}_{\mathrm{lin}} = \boldsymbol{B}(\boldsymbol{y}_{\mathrm{eq}}) \tag{2.18}$$

of dimension $\boldsymbol{Z}_{\mathrm{lin}}^{\mathrm{y}}, \boldsymbol{Z}_{\mathrm{lin}}, \boldsymbol{M}_{\mathrm{lin}}, \boldsymbol{D}_{\mathrm{lin}}, \boldsymbol{K}_{\mathrm{lin}} \in \mathbb{R}^{n_{\mathrm{f}} \times n_{\mathrm{f}}}$ and $\bar{\boldsymbol{B}}_{\mathrm{lin}} \in \mathbb{R}^{n_{\mathrm{f}} \times n_{\mathrm{u}}}$. The linearized equations of motion are summarized in

$$\begin{bmatrix} \dot{\widetilde{\boldsymbol{y}}} \\ \dot{\widetilde{\boldsymbol{v}}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{Z}_{\mathrm{lin}}^{\mathrm{y}} & \boldsymbol{Z}_{\mathrm{lin}} \\ -\boldsymbol{M}_{\mathrm{lin}}^{-1}\boldsymbol{K}_{\mathrm{lin}} & -\boldsymbol{M}_{\mathrm{lin}}^{-1}\boldsymbol{D}_{\mathrm{lin}} \end{bmatrix}}_{\boldsymbol{A}_{\mathrm{lin}}} \begin{bmatrix} \widetilde{\boldsymbol{y}} \\ \widetilde{\boldsymbol{v}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}_{\mathrm{lin}}^{-1}\bar{\boldsymbol{B}}_{\mathrm{lin}} \end{bmatrix}}_{\boldsymbol{B}_{\mathrm{lin}}} \widetilde{\boldsymbol{u}} \tag{2.19}$$

with $\boldsymbol{A}_{\text{lin}} \in \mathbb{R}^{2n_\text{f} \times 2n_\text{f}}$ and $\boldsymbol{B}_{\text{lin}} \in \mathbb{R}^{2n_\text{f} \times n_\text{u}}$. The linearized output for small variations $\widetilde{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{z}_{\text{eq}}$ around the output $\boldsymbol{z}_{\text{eq}}$ at equilibrium is

$$\widetilde{\boldsymbol{z}} = \underbrace{\begin{bmatrix} \left.\dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}_{\text{eq}}} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{H}_{\text{lin}}} \begin{bmatrix} \widetilde{\boldsymbol{y}} \\ \widetilde{\boldsymbol{v}} \end{bmatrix} \tag{2.20}$$

with $\boldsymbol{H}_{\text{lin}} \in \mathbb{R}^{n_\text{u} \times 2n_\text{f}}$.
Secondly, the equations of motion (2.4)–(2.6) in DAE form are considered. The equilibrium point $\boldsymbol{y}_{\text{eq}}, \boldsymbol{v}_{\text{eq}}, \boldsymbol{u}_{\text{eq}}, \boldsymbol{\lambda}_{\text{eq}}$ is defined equivalently to equation (2.10). Following [BrülsBastosSeifried13, GonzálezEtAl17], the linearization for small values $\widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{y}_{\text{eq}}$, $\widetilde{\boldsymbol{v}} = \boldsymbol{v} - \boldsymbol{v}_{\text{eq}}$, $\widetilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_{\text{eq}}$, $\widetilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda} - \boldsymbol{\lambda}_{\text{eq}}$ yields

$$\dot{\widetilde{\boldsymbol{y}}} = \boldsymbol{Z}^{\text{y}}_{\text{lin}}\, \widetilde{\boldsymbol{y}} + \boldsymbol{Z}_{\text{lin}}\, \widetilde{\boldsymbol{v}} \tag{2.21}$$

$$\boldsymbol{M}_{\text{lin}} \dot{\widetilde{\boldsymbol{v}}} + \boldsymbol{D}_{\text{lin}} \widetilde{\boldsymbol{v}} + \boldsymbol{K}_{\text{lin}} \widetilde{\boldsymbol{y}} = \boldsymbol{C}^{\text{T}}_{\text{lin}} \widetilde{\boldsymbol{\lambda}} + \bar{\boldsymbol{B}}_{\text{lin}} \widetilde{\boldsymbol{u}} \tag{2.22}$$

$$\boldsymbol{C}^{\text{y}}_{\text{lin}} \widetilde{\boldsymbol{y}} = \boldsymbol{0} \tag{2.23}$$

with the constant matrices

$$\boldsymbol{Z}^{\text{y}}_{\text{lin}} = \left.\frac{\partial \boldsymbol{Z}\boldsymbol{v}}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{v}_{\text{eq}}}, \tag{2.24}$$

$$\boldsymbol{Z}_{\text{lin}} = \boldsymbol{Z}(\boldsymbol{y}_{\text{eq}}), \tag{2.25}$$

$$\boldsymbol{M}_{\text{lin}} = \boldsymbol{M}(\boldsymbol{y}_{\text{eq}}), \tag{2.26}$$

$$\boldsymbol{D}_{\text{lin}} = \left.\left(\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{v}} - \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{v}}\right)\right|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{v}_{\text{eq}}}, \tag{2.27}$$

$$\boldsymbol{K}_{\text{lin}} = \left.\left(\frac{\partial \boldsymbol{k}}{\partial \boldsymbol{y}} - \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{y}} - \frac{\partial \boldsymbol{B}\, \boldsymbol{u}_{\text{eq}}}{\partial \boldsymbol{y}} - \frac{\partial \boldsymbol{C}^{\text{T}} \boldsymbol{\lambda}_{\text{eq}}}{\partial \boldsymbol{y}}\right)\right|_{\boldsymbol{y}_{\text{eq}}, \boldsymbol{v}_{\text{eq}}}, \tag{2.28}$$

$$\boldsymbol{C}_{\text{lin}} = \boldsymbol{C}(\boldsymbol{y}_{\text{eq}}), \tag{2.29}$$

$$\boldsymbol{C}^{\text{y}}_{\text{lin}} = \left.\frac{\partial \boldsymbol{c}}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}_{\text{eq}},}, \tag{2.30}$$

$$\bar{\boldsymbol{B}}_{\text{lin}} = \boldsymbol{B}(\boldsymbol{y}_{\text{eq}}) \tag{2.31}$$

of dimension $\boldsymbol{Z}^{\text{y}}_{\text{lin}}, \boldsymbol{Z}_{\text{lin}}, \boldsymbol{M}_{\text{lin}}, \boldsymbol{D}_{\text{lin}}, \boldsymbol{K}_{\text{lin}} \in \mathbb{R}^{n_\text{y} \times n_\text{y}}$, $\bar{\boldsymbol{B}}_{\text{lin}} \in \mathbb{R}^{n_\text{y} \times n_\text{u}}$ and $\boldsymbol{C}^{\text{y}}_{\text{lin}}, \boldsymbol{C}_{\text{lin}} \in \mathbb{R}^{n_\text{c} \times n_\text{y}}$. The linearized equations of motion are summarized as

$$\underbrace{\begin{bmatrix} \boldsymbol{I}_{n_\text{y}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{\text{lin}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{E}_{\text{lin}}} \begin{bmatrix} \dot{\widetilde{\boldsymbol{y}}} \\ \dot{\widetilde{\boldsymbol{v}}} \\ \dot{\widetilde{\boldsymbol{\lambda}}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{Z}^{\text{y}}_{\text{lin}} & \boldsymbol{Z}_{\text{lin}} & \boldsymbol{0} \\ -\boldsymbol{K}_{\text{lin}} & -\boldsymbol{D}_{\text{lin}} & \boldsymbol{C}^{\text{T}}_{\text{lin}} \\ \boldsymbol{C}^{\text{y}}_{\text{lin}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{A}_{\text{lin}}} \begin{bmatrix} \widetilde{\boldsymbol{y}} \\ \widetilde{\boldsymbol{v}} \\ \widetilde{\boldsymbol{\lambda}} \end{bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \bar{\boldsymbol{B}}_{\text{lin}} \\ \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{B}_{\text{lin}}} \widetilde{\boldsymbol{u}} \tag{2.32}$$

with $\boldsymbol{E}_{\mathrm{lin}}, \boldsymbol{A}_{\mathrm{lin}} \in \mathbb{R}^{(2n_{\mathrm{y}}+n_{\mathrm{c}})\times(2n_{\mathrm{y}}+n_{\mathrm{c}})}$ and $\boldsymbol{B}_{\mathrm{lin}} \in \mathbb{R}^{(2n_{\mathrm{y}}+n_{\mathrm{c}})\times n_{\mathrm{u}}}$. The linearized output for small variations $\widetilde{\boldsymbol{z}} = \boldsymbol{z} - \boldsymbol{z}_{\mathrm{eq}}$ around the output $\boldsymbol{z}_{\mathrm{eq}}$ at equilibrium is

$$\widetilde{\boldsymbol{z}} = \underbrace{\begin{bmatrix} \left.\dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}\right|_{\boldsymbol{y}_{\mathrm{eq}}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{H}_{\mathrm{lin}}} \begin{bmatrix} \widetilde{\boldsymbol{y}} \\ \widetilde{\boldsymbol{v}} \\ \widetilde{\boldsymbol{\lambda}} \end{bmatrix} \tag{2.33}$$

with $\boldsymbol{H}_{\mathrm{lin}} \in \mathbb{R}^{n_{\mathrm{u}}\times(2n_{\mathrm{y}}+n_{\mathrm{c}})}$.

2.2 Flexible Multibody System Dynamics

The assumption of rigid bodies might not be valid for light and fast-moving bodies which undergo elastic deformations. Then, structural flexibility must be modeled. Here, the focus lies on beam formulations, which are suitable to model flexible manipulators. Several approaches exist in flexible multibody system dynamics to model structural flexibility, such as the floating frame of reference formulation, geometrically exact formulations or the ANCF. A brief overview of these approaches is given in the following. Afterwards, a beam model based on the ANCF is introduced.

2.2.1 Overview of Different Approaches

The floating frame of reference approach couples large rigid body motion with small linear elastic deformations with respect to the reference motion [Schwertassek99]. A strong advantage lies in the possibility of linear modal reduction by taking into account only the relevant modes for a certain problem. However, for large deformations, modal reduction is in general not applicable and the method becomes inefficient [GerstmayrSugiyamaMikkola13, WasfyNoor03].
The geometrically exact formulation describes the kinematic relations by introducing a moving frame which considers finite translations and rotations [Simo85, Bauchau11]. The method is well-established and yields accurate results for large deformations [BauchauEtAl15]. However, the numerical discretization of the continuous equations is not straightforward [Romero08]. This is due to the interpolation of rotation parameters. Advanced interpolation and integration techniques must be implemented to deal with large rotations [Bauchau11, Romero04]. Thus, a direct integration into existing flexible multibody dynamics code is difficult [Romero08].
The ANCF is proposed in [Shabana98] to represent large rigid body rotations as well as large deformations. While rigid body motion can be represented exactly, the method does not rely on rotation parameters. Thus, integration into existing multibody dynamics code is simple [Romero08]. However, locking problems are reported for ANCF elements and there exist advanced strategies and element types to circumvent locking [DufvaSopanenMikkola06, García-

VallejoMikkolaEscalona07, GerstmayrShabana06]. Application examples of the ANCF are cables, flexible pendulums [EscalonaHussienShabana98], rubber chains [MaquedaMohamedShabana10], tyres [SugiyamaSuda09] and flexible manipulators [NachbagauerZehetnerGerstmayr12, TianEtAl07, VoharKeglRen08].

Several works compare the three approaches. The floating frame of reference approach is compared to the ANCF in [DiboldGerstmayrIrschik09]. The ANCF is compared to the geometrically exact formulation and to experimental data in [BauchauEtAl15]. Different aspects of both methods, such as accuracy, numerical implementation and efficiency are compared in [Romero08]. The authors conclude that the geometrically exact formulation is more efficient and more accurate than the ANCF. However, implementation and integration into existing code is easier for the ANCF.

Here, highly flexible manipulators undergoing large deformations are considered for model inversion. Due to more simple implementation and integration into existing code, the ANCF is applied to model highly flexible manipulators. In subsequent chapters, it will be shown that an analytical Jacobian of the equations of motion is indispensable for the applied methods. Therefore, the ANCF beam formulation introduced in [OmarShabana01, García-VallejoEtAl04] is applied in the context of this work. Note that the methods considered in this work are generally also applicable to other beam formulations and the chosen ANCF beam formulation only serves as an application example.

2.2.2 Absolute Nodal Coordinate Formulation

The two-dimensional beam element is introduced following [OmarShabana01, García-VallejoEtAl04]. The beam element allows for shear deformation and the cross-section does not necessarily stay perpendicular to the neutral axis. One beam element of length L is shown in the deformed and undeformed configuration in Fig. 2.1. A coordinate system is introduced with $x \in [0; L]$ describing the axis in beam direction in the undeformed configuration and y describing the direction perpendicular to x. For notational simplicity, scaled coordinates are introduced with $\xi(x) = \frac{x}{L}$ and $\eta(y) = \frac{y}{L}$.

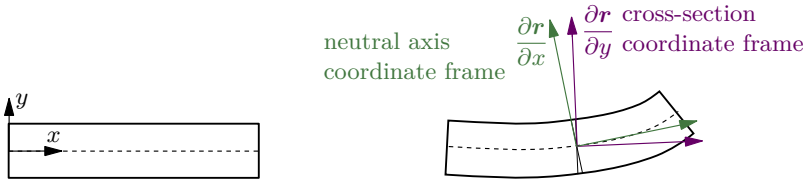


Figure 2.1: Deformed and undeformed configuration of an ANCF element.

In the following, the equations of motion are stated for one beam element with two nodes. They can then be assembled for N elements. The position $\boldsymbol{r}$ of an arbitrary point on the beam element is written as

$$\boldsymbol{r}(\boldsymbol{e}) = \begin{bmatrix} r_1(\boldsymbol{e}) \\ r_2(\boldsymbol{e}) \end{bmatrix} = \boldsymbol{\Phi}(x, y)\, \boldsymbol{e}\,. \tag{2.34}$$

Thereby, the function $\boldsymbol{\Phi} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{2\times 12}$ contains the global shape functions and $\boldsymbol{e} \in \mathbb{R}^{12}$ is the generalized coordinate vector of one element with

$$\boldsymbol{y} = \boldsymbol{e} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} \end{bmatrix}^{\mathsf{T}}\,. \tag{2.35}$$

The generalized coordinates $\boldsymbol{e}$ are chosen to describe the absolute positions of the nodes and the global slopes of the position vector $\boldsymbol{r}(\boldsymbol{e})$. The generalized coordinates e_i with $i = 1, 2, \ldots, 6$ at the left node are expressed as

$$e_1 = r_1\Big|_{x=0}\,, \quad e_2 = r_2\Big|_{x=0}\,, \tag{2.36}$$

$$e_3 = \frac{\partial r_1}{\partial x}\bigg|_{x=0}\,, \quad e_4 = \frac{\partial r_2}{\partial x}\bigg|_{x=0}\,, \quad e_5 = \frac{\partial r_1}{\partial y}\bigg|_{x=0}\,, \quad e_6 = \frac{\partial r_2}{\partial y}\bigg|_{x=0} \tag{2.37}$$

and the coordinates e_i with $i = 7, 8, \ldots, 12$ at the right node are

$$e_7 = r_1\Big|_{x=L}\,, \quad e_8 = r_2\Big|_{x=L}\,, \tag{2.38}$$

$$e_9 = \frac{\partial r_1}{\partial x}\bigg|_{x=L}\,, \quad e_{10} = \frac{\partial r_2}{\partial x}\bigg|_{x=L}\,, \quad e_{11} = \frac{\partial r_1}{\partial y}\bigg|_{x=L}\,, \quad e_{12} = \frac{\partial r_2}{\partial y}\bigg|_{x=L}\,. \tag{2.39}$$

The global shape function matrix $\boldsymbol{\Phi}(x, y)$ is defined as

$$\boldsymbol{\Phi}(x, y) = \begin{bmatrix} \phi_1(x, y)\boldsymbol{I} & \phi_2(x, y)\boldsymbol{I} & \phi_3(x, y)\boldsymbol{I} & \phi_4(x, y)\boldsymbol{I} & \phi_5(x, y)\boldsymbol{I} & \phi_6(x, y)\boldsymbol{I} \end{bmatrix} \tag{2.40}$$

with identity $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ and the shape functions

$$\phi_1(x, y) = 1 - 3\xi(x)^2 + 2\xi(x)^3\,, \quad \phi_2(x, y) = L\left(\xi(x) - 2\xi(x)^2 + \xi(x)^3\right)\,, \tag{2.41}$$

$$\phi_3(x, y) = L\left(\eta(y) - \xi(x)\,\eta(y)\right)\,, \quad \phi_4(x, y) = 3\xi(x)^2 - 2\xi(x)^3\,, \tag{2.42}$$

$$\phi_5(x, y) = L\left(-\xi(x)^2 + \xi(x)^3\right)\,, \quad \phi_6(x, y) = L\,\xi(x)\,\eta(y)\,. \tag{2.43}$$

The influence of the coordinates e_i with $i = 1, 2, \ldots, 6$ on the deformation field is visualized in Fig. 2.2. Note that for visualization purposes, the scaling is different for the different deformations. The coordinates e_i with $i = 7, 8, \ldots, 12$ have an equivalent influence on the deformation field at the right hand node.

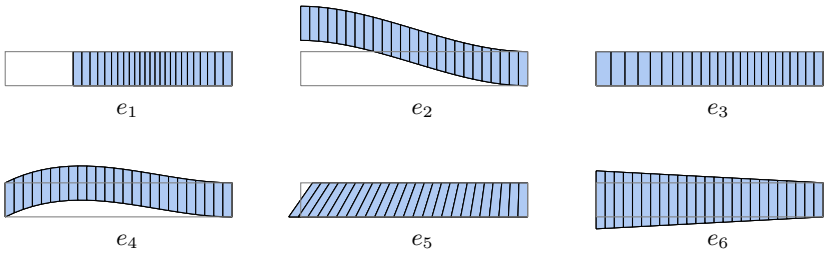


Figure 2.2: Influence of the coordinates e_i on the deformation field of a beam element for $i = 1, \ldots, 6$.

The mass matrix is derived in [OmarShabana01] from the kinetic energy of the beam as

$$\boldsymbol{M} = \int_V \rho\, \boldsymbol{\Phi}(x,y)^{\mathsf{T}}\, \boldsymbol{\Phi}(x,y)\, \mathrm{dV} \tag{2.44}$$

with element density ρ and volume V. The mass matrix $\boldsymbol{M}$ is constant and can be evaluated before the simulation. The vector of Coriolis, centrifugal and gyroscopic forces vanishes

$$\boldsymbol{k} = \boldsymbol{0}\,. \tag{2.45}$$

The derivation of the applied forces $\boldsymbol{q}(\boldsymbol{e})$ is presented according to [García-VallejoEtAl04] and is based on a general continuum mechanics approach. The Green-Lagrange strain tensor $\boldsymbol{\varepsilon} : \mathbb{R}^{12} \to \mathbb{R}^{2\times 2}$ reads

$$\boldsymbol{\varepsilon}(\boldsymbol{e}) = \frac{1}{2}\left(\bar{\boldsymbol{J}}(\boldsymbol{e})^{\mathsf{T}}\bar{\boldsymbol{J}}(\boldsymbol{e}) - \boldsymbol{I}_2\right)\,. \tag{2.46}$$

Thereby, the deformation gradient $\bar{\boldsymbol{J}} : \mathbb{R}^{12} \to \mathbb{R}^{2\times 2}$ is

$$\bar{\boldsymbol{J}}(\boldsymbol{e}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{r}_0} \tag{2.47}$$

with initial position $\boldsymbol{r}_0 = \boldsymbol{\Phi}(x,y)\,\boldsymbol{e}_0$ of a point on the beam in initial configuration $\boldsymbol{e}_0$. For a linear elastic material model, the strain energy is given by

$$U_e(\boldsymbol{e}) = \int_V \frac{\lambda + 2G}{2}\left(\varepsilon_{11}(\boldsymbol{e})^2 + \varepsilon_{22}(\boldsymbol{e})^2\right) + \lambda\,\varepsilon_{11}(\boldsymbol{e})\varepsilon_{22}(\boldsymbol{e}) + 2\,G\,\varepsilon_{12}(\boldsymbol{e})^2\,\mathrm{dV} \tag{2.48}$$

with the components $\varepsilon_{11}(\boldsymbol{e})$, $\varepsilon_{22}(\boldsymbol{e})$ and $\varepsilon_{12}(\boldsymbol{e})$ of the symmetric strain tensor $\boldsymbol{\varepsilon}(\boldsymbol{e})$. The material constants are

$$\lambda = \frac{E\,\nu}{(1-2\nu)(1+\nu)}\,, \quad G = \frac{E}{2(1+\nu)} \tag{2.49}$$

with Young's modulus E and Poisson's ratio ν. The applied forces $\boldsymbol{q}(\boldsymbol{e})$ arise from the variation of the strain energy U_e as

$$\boldsymbol{q}(\boldsymbol{e}) = -\frac{\partial U_e}{\partial \boldsymbol{e}} \,. \tag{2.50}$$

It is shown in [García-VallejoEtAl04] that the applied forces can be written as

$$\boldsymbol{q}(\boldsymbol{e}) = -\boldsymbol{K}(\boldsymbol{e})\,\boldsymbol{e} \tag{2.51}$$

with a state-dependent nonlinear stiffness matrix $\boldsymbol{K}(\boldsymbol{e})$. Furthermore, invariant matrices are introduced in [García-VallejoEtAl04] for simplified calculation of the stiffness matrix $\boldsymbol{K}(\boldsymbol{e})$ to avoid solving volume integrals in each time step. Moreover, the presented formulation makes it possible to derive an analytical expression for the Jacobian of the elastic forces.

2.3 High-level Actuator Models

System inputs $\boldsymbol{u}$ are usually realized by actuators, such as servomotors. Servomotor controllers can be employed in torque-controlled or velocity-controlled mode. In the torque-controlled case, a reference torque is given to the motor controller. Cascaded internal control loops control the actual torque to be equal to the reference torque, e.g. by measuring the motor current. In the velocity-controlled mode, the internal control loops control the velocity to be equal to the reference velocity. For both cases, it is assumed here that the internal control loops are much faster than the system dynamics. If this assumption is not valid, the actuator dynamics must be modeled in more detail.
Concerning the mechanical model, a force-controlled actuator enters the model as an applied force. The velocity-controlled actuator can be modeled using different relationships between the reference input u and the velocity $\dot{x}$ of the body. A few high-level models applied in the context of this work are summarized in the following based on [Bishop08]. In a first approximation, a zero order relationship can be assumed. Then, the reference velocity u directly corresponds to the body velocity $\dot{x}$, such that the motor model is

$$\dot{x} = u \,. \tag{2.52}$$

Alternatively, a first order relationship can be assumed. A first order model is

$$\tau\ddot{x} + \dot{x} = u \,. \tag{2.53}$$

The time constant τ describes how fast a reference velocity u is reached and must be identified on the experimental setup. First order models are successfully applied in [KnierimKriegerSawodny10, Radisch11]. A second order model is

$$\dddot{x} + 2\zeta\omega_\mathrm{n}\ddot{x} + \omega_\mathrm{n}^2\dot{x} = \omega_\mathrm{n}^2 u \tag{2.54}$$

with the parameters ω_n and ζ to be identified. Second order dynamics can oscillate and can feature an overshoot in the dynamics. All models are displayed in Fig. 2.3 for exemplary parameters $\tau = 0.02\,\mathrm{s}$ and $\zeta = 0.3$, $\omega_\mathrm{n} = 70\,\frac{\mathrm{rad}}{\mathrm{s}}$ and a reference system input $u = 1\,\frac{\mathrm{m}}{\mathrm{s}}$ applied at time $t = 0\,\mathrm{s}$. Some properties of the inverse model depend on the chosen actuator model, see also the discussion in [OttoSeifried18b].
Furthermore, a time delay exists between the time instant of setting the reference velocity and the motor reacting to the command. However, in this thesis, it is assumed that the time delay is small compared to the dynamics of the system and is therefore not modeled.

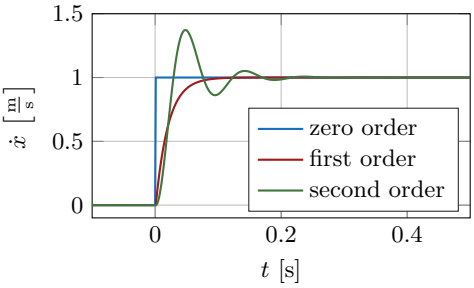


Figure 2.3: High-level models for velocity-controlled actuators.

2.4 Application Examples

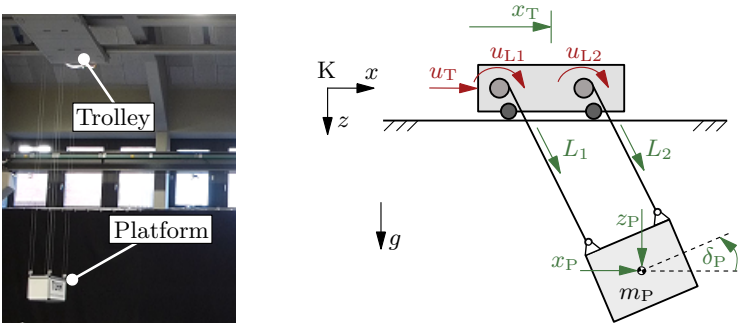
Different application examples are considered throughout this thesis. Each example has specific properties regarding its inverse model. The models serve as small computational examples in Chap. 3 in order to demonstrate theoretical results. Simulation results and detailed analyses are shown in Chaps. 4–6. In the following, the considered application examples are introduced. They include two cable robot models, a torsional oscillator, a mass-on-car system, two manipulators with one passive joint and two flexible manipulators modeled by the ANCF.

2.4.1 Cable Robot

The methods considered in this thesis are validated by experiments on the experimental cable robot setup shown in Fig. 2.4(a). The cable robot consists of a trolley which can move on a set of parallel axis. A load platform is attached by four cables. It is possible to actuate each cable individually in order to control the orientation of the load platform. Depending on the motion of the cables, two models are considered and presented in the following.

Complex Cable Robot Model

The complex cable robot model denotes a two-dimensional model considering a rotational degree of freedom of the load platform. For derivation of the model, it is assumed that two cables each move synchronously and form a set. Each set of cables can move independently from the other, such that the load platform can rotate around one axis. Thus, the three-dimensional system can be modeled in two dimensions and the model is shown in Fig. 2.4(b), see also [OttoSeifried17]. It consists of two rigid bodies connected by two cables. The first body is the trolley moving horizontally, while the second body is the load platform.



(a) Experimental setup. (b) Model of the experimental setup.

Figure 2.4: Experimental setup and model of a cable robot [OttoSeifried17].

The model has $n_\mathrm{f} = 4$ degrees of freedom. There are $n_\mathrm{u} = 3$ system inputs, namely the actuator moving the trolley and two actuators turning the winches to vary the cable lengths. Thus, the system is underactuated with one unactuated degree of freedom, which is the sway motion of the load platform. Due to the kinematic loop, the equations of motion are derived in DAE form, see Sec. 2.1.2. The equations of motion are stated in terms of the redundant coordinates

$$\boldsymbol{y} = \begin{bmatrix} x_\mathrm{T} & L_1 & L_2 & x_\mathrm{P} & z_\mathrm{P} & \delta_\mathrm{P} \end{bmatrix}^\mathsf{T} \tag{2.55}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}} \tag{2.56}$$

with x_T describing the position of the trolley, L_1 and L_2 denoting the length of each cable and x_P, z_P and δ_P denoting the position and orientation of the load platform respectively, see Fig. 2.4(b). The actuators are employed in velocity-controlled mode, see Sec. 2.3, and are collected in the vector

$$\boldsymbol{u} = \begin{bmatrix} u_\mathrm{T} & u_\mathrm{L1} & u_\mathrm{L2} \end{bmatrix}^\mathsf{T} . \tag{2.57}$$

Thereby, u_T denotes the input of the trolley actuator and u_L1 and u_L2 denote the inputs of the cable actuators. These reference velocities are directly transformed to the dimension of the trolley velocity $\dot{x}_\mathrm{T}$ and cable velocities $\dot{L}_1$ and $\dot{L}_1$

respectively. The first order model (2.53) is applied to model the influence of the system input $\boldsymbol{u}$. The time constants τ_T and $\tau_{\mathrm{L}1} = \tau_{\mathrm{L}2}$ of the trolley and cable actuators are identified on the experimental setup.
Combining the actuator dynamics with Newton's and Euler's law for the platform dynamics yields the matrices of the equations of motion (2.4)–(2.6) as

$$\boldsymbol{M} = \begin{bmatrix} \tau_\mathrm{T} & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_{\mathrm{L}1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \tau_{\mathrm{L}2} & 0 & 0 & 0 \\ 0 & 0 & 0 & m_\mathrm{P} & 0 & 0 \\ 0 & 0 & 0 & 0 & m_\mathrm{P} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_\mathrm{P} \end{bmatrix}, \tag{2.58}$$

$$\boldsymbol{k}(\dot{\boldsymbol{y}}) = \begin{bmatrix} \dot{x}_\mathrm{T} \\ \dot{L}_1 \\ \dot{L}_2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ m_\mathrm{P} g \\ 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2.59}$$

Thereby, m_P and I_P denote platform mass and moment of inertia respectively and g denotes the gravitational constant. For the derivation of the algebraic constraints $\boldsymbol{c}(\boldsymbol{y}, t)$ and the Jacobian $\boldsymbol{C}(\boldsymbol{y}, t)$, the geometry of the cable robot model is displayed in Fig. 2.5. The rotation matrix from the body-fixed coordinate system $\mathrm{K}' : \{\mathrm{O}', x', y', z'\}$ to the inertial coordinate system $\mathrm{K}: \{\mathrm{O}, x, y, z\}$ is

$$\boldsymbol{S}(\boldsymbol{y}) = \begin{bmatrix} \cos(\delta_\mathrm{P}) & 0 & \sin(\delta_\mathrm{P}) \\ 0 & 1 & 0 \\ -\sin(\delta_\mathrm{P}) & 0 & \cos(\delta_\mathrm{P}) \end{bmatrix} \tag{2.60}$$

with $\boldsymbol{S} : \mathbb{R}^{n_\mathrm{y}} \to \mathbb{R}^{3\times 3}$. The algebraic constraints represent the cable lengths

$$\boldsymbol{c}(\boldsymbol{y}) = \begin{bmatrix} L_1 - \|\boldsymbol{r}_1(\boldsymbol{y})\| \\ L_2 - \|\boldsymbol{r}_2(\boldsymbol{y})\| \end{bmatrix} = \boldsymbol{0} \tag{2.61}$$

with the position vectors

$$\boldsymbol{r}_0(\boldsymbol{y}) = \begin{bmatrix} x_\mathrm{T} - x_\mathrm{P} \\ 0 \\ -z_\mathrm{P} \end{bmatrix}, \tag{2.62}$$

$$\boldsymbol{r}_1(\boldsymbol{y}) = \boldsymbol{r}_0(\boldsymbol{y}) + \frac{1}{2}\begin{bmatrix} -L_{\mathrm{T},x} \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\,\boldsymbol{S}(\boldsymbol{y}) \begin{bmatrix} L_{\mathrm{P},x} \\ 0 \\ L_{\mathrm{P},z} \end{bmatrix}, \tag{2.63}$$

$$\boldsymbol{r}_2(\boldsymbol{y}) = \boldsymbol{r}_0(\boldsymbol{y}) + \frac{1}{2}\begin{bmatrix} L_{\mathrm{T},x} \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\,\boldsymbol{S}(\boldsymbol{y}) \begin{bmatrix} -L_{\mathrm{P},x} \\ 0 \\ L_{\mathrm{P},z} \end{bmatrix}, \tag{2.64}$$

where $L_{\mathrm{T},x}$ and $L_{\mathrm{P},x}$, $L_{\mathrm{P},z}$ describe the size of trolley and platform respectively, see Fig. 2.5. The algebraic constraints (2.61) are enforced by the Lagrange multipliers $\boldsymbol{\lambda} = \begin{bmatrix}\lambda_1 & \lambda_2\end{bmatrix}^{\mathsf{T}}$, which represent the cable forces.
The angle between the center of mass of the platform and the center of mass of the trolley is denoted by φ^* and is a measure of the sway motion. The system output is the position and orientation of the platform, such that

$$\boldsymbol{z} = \begin{bmatrix} x_{\mathrm{P}} & z_{\mathrm{P}} & \delta_{\mathrm{P}} \end{bmatrix}^{\mathsf{T}} . \tag{2.65}$$

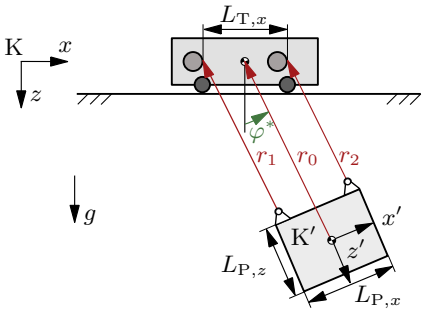


Figure 2.5: Geometry of the cable robot model.

Reduced Cable Robot Model

In case both cables L_1 and L_2 move synchronously, the complex model can be simplified. For $L_1 = L_2$ and $u_{\mathrm{L}1} = u_{\mathrm{L}2}$, the load platform stays horizontal, such that $\delta_{\mathrm{P}} = 0°$. Therefore, the platform reduces to a point mass in the model. The reduced model in shown in Fig. 2.6. It has $n_{\mathrm{f}} = 3$ degrees of freedom and $n_{\mathrm{u}} = 2$ system inputs. Thus, it is underactuated with one unactuated degree of freedom, which is the sway motion of the load platform. The reduced model is analyzed in detail in [OttoSeifried18c].

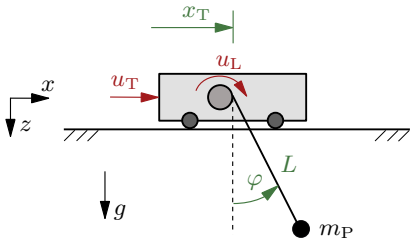


Figure 2.6: Reduced model of the cable robot [OttoSeifried18c].

The equations of motion of the reduced cable robot model can be stated in ODE form with the vector of generalized coordinates

$$\boldsymbol{y} = \begin{bmatrix} x_\mathrm{T} & L & \varphi \end{bmatrix}^\mathsf{T} \tag{2.66}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}}\,. \tag{2.67}$$

Thereby, L denotes the single cable length and φ describes the sway angle of the platform, see Fig. 2.6. The system input is collected in

$$\boldsymbol{u} = \begin{bmatrix} u_\mathrm{T} & u_\mathrm{L} \end{bmatrix}^\mathsf{T} \tag{2.68}$$

with u_T denoting the trolley actuator and u_L denoting the cable actuator. The actuator dynamics is again modeled with the first order model (2.53) with time constants τ_T and τ_L for trolley and cable respectively. The matrices of the equations of motion (2.1)–(2.2) are

$$\boldsymbol{M}(\boldsymbol{y}) = \begin{bmatrix} \tau_\mathrm{T} & 0 & 0 \\ 0 & \tau_\mathrm{L} & 0 \\ \cos(\varphi) & 0 & L \end{bmatrix}, \quad \boldsymbol{k}(\dot{\boldsymbol{y}}) = \begin{bmatrix} \dot{x}_\mathrm{T} \\ \dot{L} \\ 2\,\dot{\varphi}\,\dot{L} \end{bmatrix}, \tag{2.69}$$

$$\boldsymbol{q}(\boldsymbol{y}) = \begin{bmatrix} 0 \\ 0 \\ -g\sin(\varphi) \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \tag{2.70}$$

The equations of motion of the reduced cable robot can also be stated in DAE form by describing the position of the platform with the coordinates x_P, z_P. Thus, the vector of redundant coordinates is

$$\boldsymbol{y} = \begin{bmatrix} x_\mathrm{T} & L & x_\mathrm{P} & z_\mathrm{P} \end{bmatrix}^\mathsf{T} \tag{2.71}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}}\,. \tag{2.72}$$

The matrices of the equations of motion (2.4)–(2.6) are

$$\boldsymbol{M} = \begin{bmatrix} \tau_\mathrm{T} & 0 & 0 & 0 \\ 0 & \tau_\mathrm{L} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{k}(\dot{\boldsymbol{y}}) = \begin{bmatrix} \dot{x}_\mathrm{T} \\ \dot{L} \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \end{bmatrix}, \tag{2.73}$$

$$\boldsymbol{C}(\boldsymbol{y}) = \begin{bmatrix} 0 \\ 0 \\ 2\,(x_\mathrm{P} - x_\mathrm{T}) \\ 2z_\mathrm{P} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{2.74}$$

The algebraic constraint reads

$$c(\boldsymbol{y}) = z_\mathrm{P}^2 + (x_\mathrm{P} - x_\mathrm{T})^2 - L^2 = 0 \tag{2.75}$$

which enforces the cable length. The scalar Lagrange multiplier λ corresponds to the cable force.

For the reduced model, the system output is the position of the platform, which can be expressed in generalized as well as redundant coordinates as

$$\boldsymbol{z} = \begin{bmatrix} x_\mathrm{T} + L\sin(\varphi) \\ L\cos(\varphi) \end{bmatrix} = \begin{bmatrix} x_\mathrm{P} \\ z_\mathrm{P} \end{bmatrix} . \tag{2.76}$$

2.4.2 Torsional Oscillator

The torsional oscillator is a simple model for a drive train or drill string. It can be interpreted as a generalization of the elastic joint models of flexible joint manipulators. The system is a typical example in the context of model inversion, whereby usually equivalent translational mass-spring-damper chains are considered [AltmannHeiland17, FumagalliEtAl10]. The torsional oscillator consists of N rotating disks which are connected by linear springs and dampers with coefficients k_i, d_i for $i = 1, 2, \ldots, N-1$. The model is shown in Fig. 2.7. It has $n_\mathrm{f} = N$ degrees of freedom, while it is actuated by a single system input. The system input u is a torque applied on disk 1. For $N > 1$, the system is underactuated with $N - 1$ unactuated degrees of freedom.

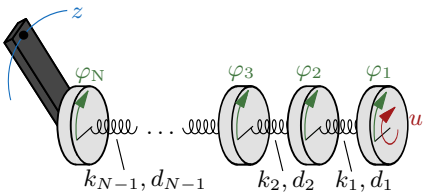


Figure 2.7: Torsional oscillator with N disks.

For deriving the equations of motion in ODE form, the vector of generalized coordinates is selected as

$$\boldsymbol{y} = \begin{bmatrix} \varphi_1 & \varphi_2 & \ldots & \varphi_{\mathrm{N}-1} & \varphi_\mathrm{N} \end{bmatrix}^\mathsf{T} \tag{2.77}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}} , \tag{2.78}$$

where φ_i describes the rotation of disk i for $i = 1, 2, \ldots, N$. The matrices of the linear equations of motion (2.11)–(2.12) are

$$\boldsymbol{M}_{\text{lin}} = \begin{bmatrix} I_1 & 0 & & & \\ 0 & I_2 & 0 & & \\ & \ddots & \ddots & \ddots & \\ & & & I_{\text{N}-1} & 0 \\ & & & 0 & I_{\text{N}} \end{bmatrix}, \tag{2.79}$$

$$\boldsymbol{D}_{\text{lin}} = \begin{bmatrix} d_1 & -d_1 & 0 & & \\ -d_1 & d_1 + d_2 & -d_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -d_{\text{N}-2} & d_{\text{N}-2} + d_{\text{N}-1} & -d_{\text{N}-1} \\ & & & -d_{\text{N}-1} & d_{\text{N}-1} \end{bmatrix}, \tag{2.80}$$

$$\boldsymbol{K}_{\text{lin}} = \begin{bmatrix} k_1 & -k_1 & 0 & & \\ -k_1 & k_1 + k_2 & -k_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -k_{\text{N}-2} & k_{\text{N}-2} + k_{\text{N}-1} & -k_{\text{N}-1} \\ & & & -k_{\text{N}-1} & k_{\text{N}-1} \end{bmatrix}, \quad \bar{\boldsymbol{B}}_{\text{lin}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \tag{2.81}$$

Thereby, I_i denotes the moment of inertia of disk i for $i = 1, 2, \ldots, N$. The system output z is defined as the rotation of the Nth disk with

$$z = \varphi_{\text{N}}\,. \tag{2.82}$$

2.4.3 Mass-on-car System

A mass-on-car system is introduced in [SeifriedBlajer13] in order to illustrate various properties and configurations arising for inverse model problems. It is extended in [OttoSeifried18b] with an additional mass. The extended model is introduced in the following.

The mass-on-car system consists of three masses with mass m_i each and is shown in Fig. 2.8. The masses are coupled by linear spring-damper combinations with coefficients k_i, d_i for $i = 1, 2$. The first two masses are moving on a horizontal base, while the third mass moves on an inclined plane. The plane is fixed on the second mass and is inclined by the constant angle α. The system has $n_{\text{f}} = 3$ degrees of freedom and $n_{\text{u}} = 1$ system input. The system input u is modeled as a force applied horizontally on the first mass. The system is underactuated with two unactuated degrees of freedom, which describe the motion of mass 2 and 3. The system is modeled in ODE form with the generalized coordinates

$$\boldsymbol{y} = \begin{bmatrix} x_1 & s_1 & s_2 \end{bmatrix}^{\mathsf{T}} \tag{2.83}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}} \tag{2.84}$$

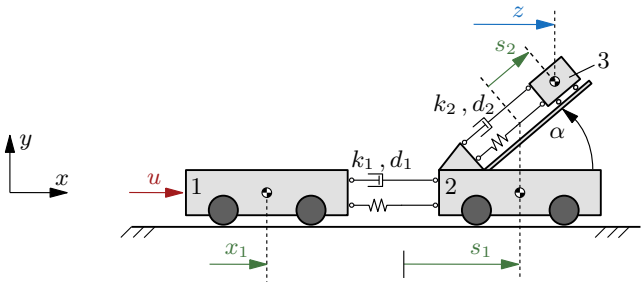


Figure 2.8: Model of the mass-on-car system [OttoSeifried18b].

with x_1 describing the position of mass 1, and s_1 and s_2 describing the positions of masses 2 and 3 relative to mass 1. The matrices of the equation of motion (2.1)–(2.2) are

$$\boldsymbol{M} = \begin{bmatrix} m_1 + m_2 + m_3 & m_2 + m_3 & m_3 \cos(\alpha) \\ m_2 + m_3 & m_2 + m_3 & m_3 \cos(\alpha) \\ m_3 \cos(\alpha) & m_3 \cos(\alpha) & m_3 \end{bmatrix}, \tag{2.85}$$

$$\boldsymbol{k} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{q}(\boldsymbol{y}, \dot{\boldsymbol{y}}) = - \begin{bmatrix} 0 \\ d_1 \dot{s}_1 + k_1 s_1 \\ d_2 \dot{s}_2 + k_2 s_2 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{2.86}$$

The system output z is the horizontal position of the third mass with

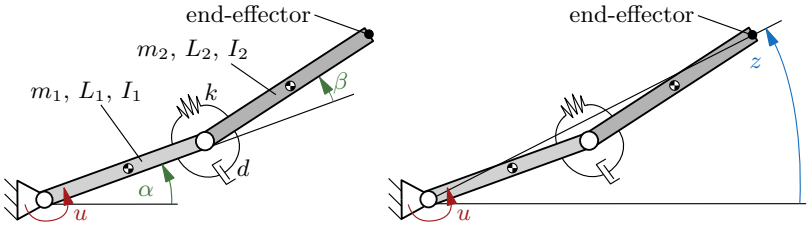
$$z = x_1 + s_1 + s_2 \cos \alpha \,. \tag{2.87}$$

2.4.4 Manipulators with one Passive Joint

A manipulator with one passive joint is the most simple model of a flexible manipulator. The structural flexibility is modeled by a single elastic mode, which is represented by a linear spring-damper combination parallel to a passive joint. In the following, a simple two-dimensional manipulator with two links and a complex three-dimensional manipulator with four links are considered.

Two-dimensional Manipulator with two Links

The two-dimensional manipulator with two links and one passive joint can be described by compact equations of motion. Due to its simplicity, an analytical analysis of the system is possible in subsequent chapters.
The model is shown in Fig. 2.9. It consists of two links with mass m_i, length L_i and moment of inertia I_i respectively. They are connected by a spring-damper combination with stiffness and damping coefficients k and d. The first link is



(a) Robot model. (b) System output z.

Figure 2.9: Model of a two-link manipulator with one passive joint.

fixed at a pin joint. The manipulator model has $n_\mathrm{f} = 2$ degrees of freedom and is actuated with the system input torque u. Thus, it is underactuated with one unactuated degree of freedom, which is the motion of the second link.
For modeling the system in ODE form, the generalized coordinates are selected as

$$\boldsymbol{y} = \begin{bmatrix} \alpha & \beta \end{bmatrix}^\mathsf{T} \tag{2.88}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}}\,. \tag{2.89}$$

Thereby, α describes the angle between the first link and the horizontal line and β is the relative angle between first and second link. The matrices of the equations of motion (2.1)–(2.2) are

$$\boldsymbol{M}(\boldsymbol{y}) = \begin{bmatrix} a_1 + a_2 + 2a_3 + L_1^2\, m_2 + I_1 + I_2 & a_2 + a_3 + I_2 \\ a_2 + a_3 + I_2 & a_2 + I_2 \end{bmatrix}, \tag{2.90}$$

$$\boldsymbol{k}(\boldsymbol{y}, \dot{\boldsymbol{y}}) = \begin{bmatrix} -\dfrac{1}{2}\, L_1\, L_2 \dot{\beta}\, m_2 \left(2\dot{\alpha} + \dot{\beta}\right) \sin(\beta) \\ \dfrac{1}{2}\, L_1\, L_2\, \dot{\alpha}^2\, m_2\, \sin(\beta) \end{bmatrix}, \tag{2.91}$$

$$\boldsymbol{q}(\boldsymbol{y}, \dot{\boldsymbol{y}}) = \begin{bmatrix} 0 \\ -k\,\beta - d\,\dot{\beta} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{2.92}$$

with the abbreviations

$$a_1 = \frac{m_1 L_1^2}{4}, \quad a_2 = \frac{m_2 L_2^2}{4}, \quad a_3 = \frac{L_1\, L_2\, m_2 \cos(\beta)}{2}. \tag{2.93}$$

The control goal is the tracking of the horizontal end-effector position r_x. The position is given by

$$r_\mathrm{x} = L_1 \sin(\alpha) + L_2 \sin(\alpha + \beta)\,. \tag{2.94}$$

A linearization for small angles β yields

$$r_{\mathrm{x,lin}} = (L_1 + L_2)\sin(z) \tag{2.95}$$

with the auxiliary angle z shown in Fig. 2.9(b). The linearized end-effector position $r_{\mathrm{x,lin}}$ can be recovered from the auxiliary angle z. Therefore, the system output is chosen as the auxiliary angle z with

$$z = \alpha + \frac{L_2}{L_1 + L_2}\,\beta\,, \tag{2.96}$$

which follows directly from the linearization of the end-effector position r_{x}. This choice of linear system output is discussed in detail in [Seifried12]. Such a linear system output simplifies an analytical analysis of the inverse model properties.

Three-dimensional Manipulator with Four Links

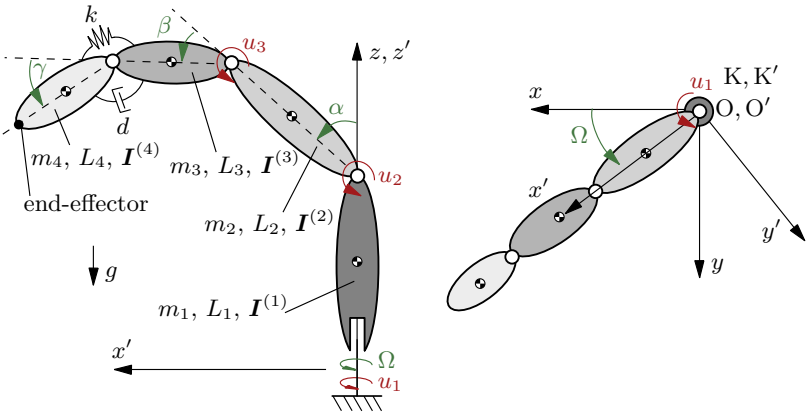
The two-link manipulator considered before is extended to a three-dimensional robot with four links. Figures 2.10(a) and 2.10(b) visualize the front and top view of the model respectively. Link 1 rotates around the z-axis with angle Ω. Link 2 is connected to the end of link 1 by an actuated joint. The rotation of link 2 around the local y'-axis is denoted by α. Similarly, link 3 is connected to link 2 and its rotation around the local y'-axis is denoted by β, which is measured relatively to angle α. Finally, the passive joint connects links 3 and 4 by a spring-damper combination. The rotation of the joint is measured by angle γ relatively to angle β. The system has $n_{\mathrm{f}} = 4$ degrees of freedom and $n_{\mathrm{u}} = 3$ system inputs actuate the system. System input u_1 rotates link 1 around the z-axis, while u_2 and u_3 actuate the links 2 and 3 respectively. Thus, the system is underactuated with one unactuated coordinate, which is represented by the motion of the fourth link.

The equations of motion are derived in ODE form with the generalized coordinates chosen as

$$\boldsymbol{y} = \begin{bmatrix}\Omega & \alpha & \beta & \gamma\end{bmatrix}^{\mathsf{T}} \tag{2.97}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}}\,. \tag{2.98}$$

The explicit equations are omitted here due to their complexity. They can be derived using the position vector $\boldsymbol{r}_i$ of the center of gravity and the inertia



(a) Front view. (b) Top view.

Figure 2.10: Model of a three-dimensional manipulator with one passive joint.

tensor $\boldsymbol{I}^{(i)}$ of each link i for $i = 1, 2, \ldots, 4$. The position vectors are

$$\boldsymbol{r}_1 = \begin{bmatrix} 0 \\ 0 \\ \frac{L_1}{2} \end{bmatrix}, \qquad \boldsymbol{r}_2(\boldsymbol{y}) = \boldsymbol{S}_{\mathrm{KK}'}(\Omega) \begin{bmatrix} \frac{L_2}{2}\sin(\alpha) \\ 0 \\ L_1 + \frac{L_2}{2}\cos(\alpha) \end{bmatrix}, \tag{2.99}$$

$$\boldsymbol{r}_3(\boldsymbol{y}) = \boldsymbol{S}_{\mathrm{KK}'}(\Omega) \begin{bmatrix} L_2\sin(\alpha) + \frac{L_3}{2}\sin(\alpha+\beta) \\ 0 \\ L_1 + L_2\cos(\alpha) + \frac{L_3}{2}\cos(\alpha+\beta) \end{bmatrix}, \tag{2.100}$$

$$\boldsymbol{r}_4(\boldsymbol{y}) = \boldsymbol{S}_{\mathrm{KK}'}(\Omega) \begin{bmatrix} L_2\sin(\alpha) + L_3\sin(\alpha+\beta) + \frac{L_4}{2}\sin(\alpha+\beta+\gamma) \\ 0 \\ L_1 + L_2\cos(\alpha) + L_3\cos(\alpha+\beta) + \frac{L_4}{2}\cos(\alpha+\beta+\gamma) \end{bmatrix} \tag{2.101}$$

with the rotation matrix $\boldsymbol{S}_{\mathrm{KK}'} : \mathbb{R} \to \mathbb{R}^{3\times 3}$ from the local coordinate system $\mathrm{K}' : \{\mathrm{O}', x', y', z'\}$ to the inertial coordinate system $\mathrm{K}: \{\mathrm{O}, x, y, z\}$

$$\boldsymbol{S}_{\mathrm{KK}'}(\Omega) = \begin{bmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.102}$$

The inertia tensors in the body-fixed coordinate systems $\mathrm{K}^{(i)} : \{\mathrm{O}^{(i)}, x^{(i)}, y^{(i)}, z^{(i)}\}$ are

$$\boldsymbol{I}^{(i)} = \begin{bmatrix} I_{i,x^{(i)}} & 0 & 0 \\ 0 & I_{i,y^{(i)}} & 0 \\ 0 & 0 & I_{i,z^{(i)}} \end{bmatrix}, \tag{2.103}$$

which are transformed into the inertial coordinate system K: $\{\mathrm{O}, x, y, z\}$ by

$$\boldsymbol{I}_i = \left(\boldsymbol{S}_{\mathrm{KK}'}(\Omega)\boldsymbol{S}_{\mathrm{K}'\mathrm{K}^{(i)}}(\Gamma_i)\right) \boldsymbol{I}^{(i)} \left(\boldsymbol{S}_{\mathrm{KK}'}(\Omega)\boldsymbol{S}_{\mathrm{K}'\mathrm{K}^{(i)}}(\Gamma_i)\right)^{\mathsf{T}} . \tag{2.104}$$

Thereby, $\boldsymbol{S}_{\mathrm{K}'\mathrm{K}^{(i)}} : \mathbb{R} \to \mathbb{R}^{3\times 3}$ denotes the rotation matrices from the body-fixed frames into the coordinate system $\mathrm{K}' : \{\mathrm{O}', x', y', z'\}$ with

$$\boldsymbol{S}_{\mathrm{K}'\mathrm{K}^{(i)}}(\Gamma_i) = \begin{bmatrix} \cos(\Gamma_i) & 0 & \sin(\Gamma_i) \\ 0 & 1 & 0 \\ -\sin(\Gamma_i) & 0 & \cos(\Gamma_i) \end{bmatrix} . \tag{2.105}$$

The angle Γ_i for each link is given by $\Gamma_1 = 0$, $\Gamma_2 = \alpha$, $\Gamma_3 = \alpha + \beta$ and $\Gamma_4 = \alpha + \beta + \gamma$.
The three-dimensional system output is the end-effector position in the inertial coordinate system K: $\{\mathrm{O}, x, y, z\}$ with

$$\boldsymbol{z} = \boldsymbol{S}_{\mathrm{KK}'}(\Omega) \begin{bmatrix} L_2\sin(\alpha) + L_3\sin(\alpha+\beta) + L_4\sin(\alpha+\beta+\gamma) \\ 0 \\ L_1 + L_2\cos(\alpha) + L_3\cos(\alpha+\beta) + L_4\cos(\alpha+\beta+\gamma) \end{bmatrix} . \tag{2.106}$$

A linearization of the end-effector position is omitted since the equations of motion are too complex to analyze them analytically.

2.4.5 Flexible Manipulators Modeled Using the ANCF

The previously introduced manipulators with one passive joint are simple and allow an analytical analysis of the inverse model properties. However, a simple model might not be sufficiently accurate for highly flexible manipulators and the flexibility is here modeled using the ANCF beam element described in Sec. 2.2.2. In the following, two flexible manipulators are considered. First, a flexible one-link manipulator is introduced for analysis of the proposed computation schemes. Then, a two-link manipulator with one flexible link is considered.

Flexible Manipulator with one Link

The flexible manipulator with one link is shown in Fig. 2.11. Similar to the manipulator with one passive joint, it is fixed on the left end by an actuated joint. The degrees of freedom depend on the considered number of ANCF beam elements. Due to $n_\mathrm{u} = 1$ system input and many degrees of freedom, the model is underactuated with many unactuated degrees of freedom.

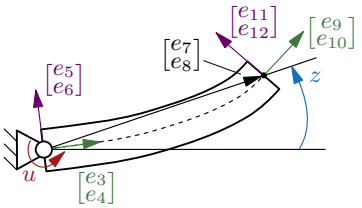


Figure 2.11: Model of a flexible manipulator with one link shown for one ANCF element.

In Sec. 2.2.2, the equations of motion for a free beam element are given in equations (2.44), (2.45) and (2.51). The free flexible manipulator modeled with N ANCF elements has $n_\mathrm{f} = 6(N+1)$ degrees of freedom. In the context of this work, N is considered in the range $N = 1\,, 2 \ldots 10$.
For the manipulator in Fig. 2.11, some additional algebraic constraints arise. Four different constraint formulations are considered and are discussed in the following. For all considered formulations, the pin joint on the left hand side of the flexible manipulator is enforced by the constraint

$$\boldsymbol{c}_1(\boldsymbol{y}) = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{2.107}$$

As mentioned in Sec. 2.2.2, the ANCF captures shear deformation of the beam and the cross-section does not necessarily stay perpendicular to the neutral axis. Nevertheless, it might be desired to constrain the cross-section to be perpendicular to the neutral axis on the left node, e.g. to model a bearing. Such a constraint reads

$$c_2(\boldsymbol{y}) = -e_3\, e_5 - e_4\, e_6 = 0\,. \tag{2.108}$$

As discussed in Sec. 2.3, actuators can be either used in force-controlled or velocity-controlled mode. Both configurations are considered for the flexible manipulator. Considering a force-controlled actuator, the system input is applied by the torque u_tor. For $N = 1$ element, the input is distributed in the equations of motion by the matrix

$$\boldsymbol{B}(\boldsymbol{y}) = -\begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{e_6}{f_{56}^2} & -\dfrac{e_5}{f_{56}^2} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T} \tag{2.109}$$

with the abbreviation

$$f_{56} = \sqrt{e_5^2 + e_6^2}\,. \tag{2.110}$$

The matrix $\boldsymbol{B} : \mathbb{R}^{12\times 1} \to \mathbb{R}^{12\times 1}$ is derived in [OmarShabana01]. In case the actuator is employed in velocity-controlled mode, the system input is the actuator velocity u_vel. The actuator velocity is assumed to be equal to the angular

velocity $\dot{\gamma}$ of the cross-section at the left node. This corresponds to the zero order actuator model (2.52) and is described by the constraint

$$c_3(\boldsymbol{y}) = u_{\mathrm{vel}} - \dot{\gamma} = u_{\mathrm{vel}} + \frac{e_6\,\dot{e}_5 - e_5\,\dot{e}_6}{f_{56}^2} = 0\,. \tag{2.111}$$

In contrast to the torque-controlled mode, the velocity-controlled actuator model reduces the degrees of freedom by one. Table 2.1 summarizes the considered formulations and degrees of freedom of the flexible manipulator models.
For all considered formulations, the system output z is the angle between right and left node of the flexible manipulator, see Fig. 2.11, with

$$z = \arctan\left(\frac{e_{6(N+1)-4}}{e_{6(N+1)-5}}\right)\,. \tag{2.112}$$

Table 2.1: Overview of considered configurations for N ANCF elements.

Case	Cross-section at joint	System input type	Active constraints	Degrees of freedom n_{f}
1	free	force-controlled	$\boldsymbol{c}_1$	$6(N+1)-2$
2	free	velocity-controlled	$\boldsymbol{c}_1,\ c_3$	$6(N+1)-3$
3	constrained	force-controlled	$\boldsymbol{c}_1,\ c_2$	$6(N+1)-3$
4	constrained	velocity-controlled	$\boldsymbol{c}_1,\ c_2,\ c_3$	$6(N+1)-4$

Flexible Manipulator with two Links

The flexible manipulator with two links consists of a rigid and a flexible link, see Fig. 2.12. The rigid link with length L_1 is attached to an actuated pin joint on the left end. The flexible link is attached to the rigid link by an actuated joint. The flexible link with length L_2 is modeled by the ANCF. The model has $n_{\mathrm{u}} = 2$ system inputs and many unactuated degrees of freedom depending on the number of considered ANCF beam elements. Thus, the system is underactuated.

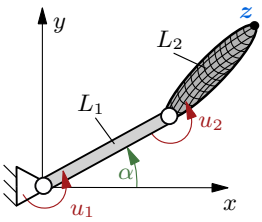


Figure 2.12: Model of a flexible two-link manipulator.

The redundant coordinate vector is chosen as

$$\boldsymbol{y} = \begin{bmatrix} \alpha & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 & e_{10} & e_{11} & e_{12} \end{bmatrix}^\mathsf{T} \tag{2.113}$$

$$\boldsymbol{v} = \dot{\boldsymbol{y}} \tag{2.114}$$

for $N = 1$ ANCF element. Thereby, the motion of the rigid link is coupled to the motion of the flexible link. Thus, several constraints are necessary. The joint between link 1 and link 2 is modeled by the constraint

$$\boldsymbol{c}_1(\boldsymbol{y}) = \begin{bmatrix} L_1 \cos(\alpha) \\ L_1 \sin(\alpha) \end{bmatrix} - \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{2.115}$$

Both actuators u_1 and u_2 are modeled as velocity-controlled actuators with zero order models (2.52). Thus, the constraints are

$$\boldsymbol{c}_2(\boldsymbol{y}) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} \dot{\alpha} \\ -\dfrac{e_6\,\dot{e}_5 - e_5\,\dot{e}_6}{f_{56}^2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} . \tag{2.116}$$

In total, the system has $6(N+1) - 3$ degrees of freedom for N elements.
The system output is defined as the end-effector position and is expressed by

$$\boldsymbol{z} = \begin{bmatrix} e_{6(N+1)-5} \\ e_{6(N+1)-4} \end{bmatrix} . \tag{2.117}$$

Model Inversion Concepts for Feedforward Control

Accurate control is essential for the successful use of modern mechanical and mechatronic systems. With advancing system development, the systems become more complex in design and more flexible in application. At the same time, their control becomes more difficult and is an active field of research, see e.g. the survey [LiuYu13]. Generally, control objectives can be divided into two main categories. Stabilizing control focuses on controlling a system around a stationary operating point, while trajectory control moves the system in space on a predefined trajectory.

The present work focuses on trajectory control of underactuated multibody systems. For trajectory control, a two design degree of freedom control structure is an efficient approach and is visualized in Fig. 3.1 [Skogestad04]. The complete controller consists of two parts. The feedforward part calculates the feedforward input $\boldsymbol{u}_{\mathrm{ffw}}$, which is responsible for generating large motion of the system. The feedback part $\boldsymbol{u}_{\mathrm{fb}}$ is responsible for rejecting disturbances and thereby stabilizes the system around the desired output trajectory $\boldsymbol{z}_{\mathrm{d}}(t)$. For this purpose, the feedback controller can utilize the desired state trajectory $\boldsymbol{y}_{\mathrm{d}}(t)$, which is also computed by the feedforward control.

It is desired to apply a precise feedforward control, which accurately moves the system on the desired trajectory and which yields small tracking errors. In that case, a simple feedback control strategy is sufficient for accurate tracking, see e.g. the experimental results in [OttoSeifried18c]. Moreover, small tracking errors result in small correcting action of the feedback controller and sensor noise is not amplified. Thus, the focus of this work is to design accurate feedforward controllers for underactuated multibody systems. Ideally, the feedforward control is chosen as an inverse model of the system, since it cancels out all modeled dynamics. While a forward simulation computes the motion of a system based on prescribed forces and torques, the inverse model computes the forces and torques required to induce a desired motion, see Fig. 3.2 [SicilianoEtAl09]. Since underactuated systems have more degrees of freedom than independent control inputs, deriving the inverse model is not straightforward.

During controller design, there is generally a trade-off between model accuracy and controller complexity [Skogestad04]. A detailed system model can cover all dynamic effects of a system, but results in a difficult controller design. On the other hand, a reduced model yields a simple control design, but might not cover all important dynamic effects of the system. Concerning the two design degree

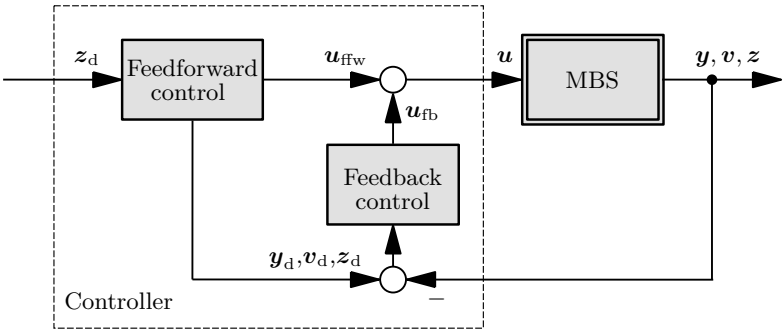


Figure 3.1: Two design degree of freedom control structure.

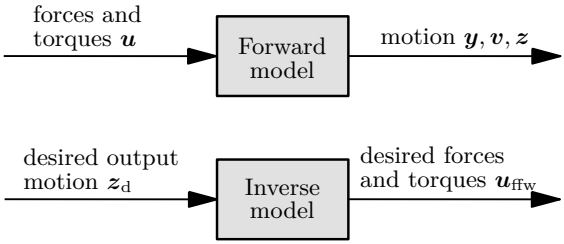


Figure 3.2: Schematics of forward model and inverse model.

of freedom control structure, it seems favorable to use a rather detailed model for the inverse model and a more simplified model or a model-free approach for the feedback part. In the course of this work, it is shown that the method of servo-constraints [BlajerKołodziejczyk04] can be applied with reasonable effort to compute the inverse models of considerably complex systems.

This chapter introduces theoretical background as well as various approaches and analysis methods for the model inversion of multibody systems. Throughout the chapter, the application examples introduced in Sec. 2.4 serve as small computational examples to illustrate theoretical results. First, nonlinear control approaches are briefly presented and their limitations regarding complex underactuated systems are discussed. Then, the framework of servo-constraints is introduced, which is suitable for complex underactuated multibody systems. Afterwards, numerical methods are presented to solve the servo-constraints approach in real-time environments. It is also shown how the combination of an inverse model with a feedback controller reduces effort in the feedback part and improves controller performance. Finally, the application examples introduced in Chap. 2 are classified with respect to properties of their inverse models.

3.1 Nonlinear Control Approaches for Model Inversion

In the following, model inversion approaches from nonlinear control theory are discussed. First, the inversion of fully actuated systems is introduced, which is also known as method of computed torques in the robotics community [Siciliano08, SicilianoEtAl09]. The limitation of the method to fully actuated systems is lifted in some instances by model inversion of differentially flat systems [FliessEtAl95], which is discussed next. Finally, a systematic approach for the inversion of nonlinear underactuated systems based on the Byrnes-Isidori normal form is presented [Isidori96, Sastry99]. All presented methods have certain limitations which make them unpractical for complex real world systems. The respective limitations are discussed for each method.

3.1.1 Model Inversion for Fully Actuated Multibody Systems

Conventional robotic manipulators are usually modeled as fully actuated multibody systems, since all links are modeled as rigid bodies and there exists an actuator on each joint to control the motion of the following link [SicilianoEtAl09]. Here, model inversion for fully actuated systems is briefly introduced according to [Spong06] in order to highlight some similarities to the more general approach of the Byrnes-Isidori normal form discussed later.

A fully actuated multibody system has a control input for each degree of freedom, such that $n_\mathrm{u} = n_\mathrm{f}$. The inverse model problem for fully actuated systems consists of two steps. First, the desired state trajectory $\boldsymbol{y}_\mathrm{d}(t), \boldsymbol{v}_\mathrm{d}(t)$ is computed from the desired output trajectory $\boldsymbol{z}_\mathrm{d}(t)$. This is called inverse kinematics. Then, during the inverse dynamics step, the desired system input $\boldsymbol{u}_\mathrm{ffw}$ is computed from the desired state trajectory $\boldsymbol{y}_\mathrm{d}(t), \boldsymbol{v}_\mathrm{d}(t)$. In the following formulation, both steps are performed in combination. The inverse model is introduced for the equations of motion (2.1)–(2.2) in ODE form. The equations of motion are first reformulated in the output coordinates $\boldsymbol{z}$, which are defined in equation (2.3). The first two derivatives of the output are

$$\dot{\boldsymbol{z}} = \underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{y})}{\partial \boldsymbol{y}} \boldsymbol{Z}(\boldsymbol{y})}_{\boldsymbol{H}(\boldsymbol{y})} \boldsymbol{v}\,, \tag{3.1}$$

$$\ddot{\boldsymbol{z}} = \boldsymbol{H}(\boldsymbol{y})\dot{\boldsymbol{v}} + \underbrace{\dot{\boldsymbol{H}}(\boldsymbol{y},\boldsymbol{v})\boldsymbol{v}}_{\bar{\boldsymbol{h}}(\boldsymbol{y},\boldsymbol{v})} \tag{3.2}$$

with the output Jacobian $\boldsymbol{H} : \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$, its derivative $\dot{\boldsymbol{H}} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$ and the local velocity $\bar{\boldsymbol{h}} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R}^{n_\mathrm{f}} \to \mathbb{R}^{n_\mathrm{f}}$. Replacing the acceleration $\dot{\boldsymbol{v}}$ by the equation of motion (2.2) yields the equations of motion in output coordinates. All states $\boldsymbol{y}, \boldsymbol{v}$ have to be substituted by the output coordinates $\boldsymbol{z}, \dot{\boldsymbol{z}}$

using the inverse functions of equations (2.3) and (3.1) respectively to yield

$$\ddot{\boldsymbol{z}} = \boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{H}\boldsymbol{M}^{-1}\left(\boldsymbol{q} - \boldsymbol{k}\right) + \bar{\boldsymbol{h}}\,. \tag{3.3}$$

The function arguments are omitted here for better readability. Due to the assumptions stated above, the output Jacobian $\boldsymbol{H}(\boldsymbol{y})$, the system input distribution $\boldsymbol{B}(\boldsymbol{y})$ as well as the mass matrix $\boldsymbol{M}(\boldsymbol{y}, t)$ are square and regular matrices. Thus, the coupling matrix

$$\boldsymbol{P}(\boldsymbol{y}, t) = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y}, t)^{-1}\boldsymbol{B}(\boldsymbol{y}) \tag{3.4}$$

with $\boldsymbol{P} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{f} \times n_\mathrm{f}}$ is invertible and the system input $\boldsymbol{u}$ is obtained as

$$\boldsymbol{u} = \left(\boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{B}\right)^{-1}\left(\ddot{\boldsymbol{z}} - \boldsymbol{H}\boldsymbol{M}^{-1}\left(\boldsymbol{q} - \boldsymbol{k}\right) - \bar{\boldsymbol{h}}\right)\,. \tag{3.5}$$

Substituting the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ and its derivatives into all arguments in equation (3.5) yields the feedforward control input

$$\boldsymbol{u}_\mathrm{ffw} = \left(\boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{B}\right)^{-1}\left(\ddot{\boldsymbol{z}}_\mathrm{d}(t) - \boldsymbol{H}\boldsymbol{M}^{-1}\left(\boldsymbol{q} - \boldsymbol{k}\right) - \bar{\boldsymbol{h}}\right). \tag{3.6}$$

The feedforward control law cancels out the modeled system dynamics. Note that the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ must be twice continuously differentiable for the control law to exist.
The introduced approach for inversion of fully actuated multibody systems cannot be applied to underactuated systems. This becomes immediately evident in the feedforward control law (3.6) since the matrices $\boldsymbol{H}(\boldsymbol{y})$ and $\boldsymbol{B}(\boldsymbol{y})$ are not square and the coupling matrix $\boldsymbol{P}(\boldsymbol{y}, t)$ is not necessarily invertible for underactuated systems. However, the feedforward control law (3.6) expresses the system input $\boldsymbol{u}$ as a function of the system output $\boldsymbol{z}$ and its derivatives $\dot{\boldsymbol{z}}$, $\ddot{\boldsymbol{z}}$. This property is generalized in the following and results in differentially flatness.

3.1.2 Flatness-based Approach

Differential flatness is a system property introduced in [FliessEtAl95] for general nonlinear systems, see also [RothfußRudolphZeitz97]. A system is called differentially flat if it is possible to describe the system input $\boldsymbol{u}$ as a function of a differentially flat fictitious output $\boldsymbol{z}$ and its derivatives, such that

$$\boldsymbol{u} = \boldsymbol{\Theta}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \dots) \tag{3.7}$$

with a nonlinear function $\boldsymbol{\Theta} : \mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{u}} \times \cdots \to \mathbb{R}^{n_\mathrm{u}}$. In the scope of this work, it is assumed that the fictitious output is the physical system output that should be tracked. The control law (3.6) for a fully actuated multibody system can be interpreted as such a function $\boldsymbol{\Theta}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}})$. Thus, a fully actuated system with the above assumptions is differentially flat. However, in general there does not exist a simple systematic approach to find a flat output $\boldsymbol{z}$, to derive the function $\boldsymbol{\Theta}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ddot{\boldsymbol{z}}, \dots)$ or to make a statement about its existence. A

detailed discussion on necessary and sufficient conditions for differential flatness with focus on mechanical systems can be found in [Knoll16]. The derivation of the function (3.7) is demonstrated in the following example.

Example 3.1 The equations of motion of the undamped 2-disk torsional oscillator from Sec. 2.4.2 with parameters $I_1 = I_2 = I$ and $k_1 = k_2 = k$ are

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix} + \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{3.8}$$

with system output $z = \varphi_2$. From the first equation it can be verified that the system input u is a function of the coordinates φ_1, φ_2 and $\ddot{\varphi}_1$ with

$$u = I\,\ddot{\varphi}_1 + k\,(\varphi_1 - \varphi_2)\ . \tag{3.9}$$

Solving the second part of equation (3.8) for the angle φ_1 and substituting the system output z yields

$$\varphi_1 = \frac{1}{k}\,(I\ddot{z} + k\,z)\ , \tag{3.10}$$

which only depends on the output z and its second derivative. The expression can be differentiated twice to obtain an expression for $\ddot{\varphi}_1$. Substituting all information into term (3.9) shows that the undamped torsional oscillator with 2 disks is differentially flat with the input-output relationship

$$u = \Theta\left(\ddot{z}, z^{(4)}\right) = \frac{I^2}{k} z^{(4)} + 2\,I\,\ddot{z}\,. \tag{3.11}$$

The property of differential flatness also holds for an arbitrary number of N disks for the undamped oscillator. This can be verified by applying the presented substitution recursively.

Differential flatness is an advantageous property for controller design [HagenmeyerZeitz04]. Equation (3.7) is the feedforward control $\boldsymbol{u}_{\mathrm{ffw}}$ when the desired state trajectory $\boldsymbol{z}_{\mathrm{d}}(t)$ is substituted for the system output $\boldsymbol{z}$ with

$$\boldsymbol{u}_{\mathrm{ffw}} = \boldsymbol{\Theta}(\boldsymbol{z}_{\mathrm{d}}(t), \dot{\boldsymbol{z}}_{\mathrm{d}}(t), \ddot{\boldsymbol{z}}_{\mathrm{d}}(t), \dots)\,. \tag{3.12}$$

The control law (3.12) is a purely algebraic expression for the system input. Thereby, a certain number of derivatives of the output $\boldsymbol{z}_{\mathrm{d}}$ is required for its evaluation. This restricts the choice of feasible desired trajectories since they must be smooth up to the order of the highest derivative appearing in the control law (3.12).

For linear systems, differential flatness is equivalent to controllability. General differentially flat systems can be transformed to linear controllable systems by quasi-static feedback [DelaleauRudolph98], which denotes that the feedback law does not only depend on the new transformed states, but also the new system input and its derivatives. Application of differentially flat control strategies is for example presented for overhead cranes in [DelaleauRudolph98, KnierimKrieger-Sawodny10, NeupertEtAl06]. Moreover, differentially flatness is utilized for the control of cable robots in [HeydenWoernle06, Woernle13] and for vehicles in [FuchshumerSchlacherRittenschober05, RouchonEtAl93]. A catalog of differentially flat mechanical systems is presented in [MurrayRathinamSluis95]. Since the derivation of the input-output relationship is not straightforward for large systems, a systematic approach to obtain the relationship is introduced in the following.

3.1.3 Byrnes-Isidori Normal Form

For general nonlinear systems, a systematic approach for deriving the input-output relationship is developed in [Isidori96, Sastry99] and is called Byrnes-Isidori normal form. It is formalized for underactuated multibody systems in [Seifried14] and is generalized for multibody systems in DAE form in [Berger17, Labisch13]. In the following, it is briefly summarized for single-input single-output (SISO) systems in the input-affine ODE form

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u \tag{3.13}$$

$$z = h(\boldsymbol{x}) \tag{3.14}$$

with state vector $\boldsymbol{x} \in \mathbb{R}^{2n_\mathrm{f}}$, function of generalized forces $\boldsymbol{f} : \mathbb{R}^{2n_\mathrm{f}} \to \mathbb{R}^{2n_\mathrm{f}}$, input distribution function $\boldsymbol{g} : \mathbb{R}^{2n_\mathrm{f}} \to \mathbb{R}^{2n_\mathrm{f}}$ and a scalar output function $h : \mathbb{R}^{2n_\mathrm{f}} \to \mathbb{R}$. Note that the equations of motion (2.1)–(2.2) with output (2.3) can be transformed into form (3.13). The notation of Lie derivatives is used to represent the derivatives of the system output z, see e.g. [Sastry99, SlotineLi91]. The Lie derivatives are the directional derivatives of a function $h(\boldsymbol{x})$ along the vector field $\boldsymbol{f}(\boldsymbol{x})$ with

$$L^0_{\boldsymbol{f}} h(\boldsymbol{x}) = h(\boldsymbol{x})\,, \tag{3.15}$$

$$L^i_{\boldsymbol{f}} h(\boldsymbol{x}) = \frac{\partial \left(L^{i-1}_{\boldsymbol{f}} h(\boldsymbol{x}) \right)}{\partial \boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x})\,, \tag{3.16}$$

$$L_{\boldsymbol{g}} L_{\boldsymbol{f}} h(\boldsymbol{x}) = \frac{\partial \left(L_{\boldsymbol{f}} h(\boldsymbol{x}) \right)}{\partial \boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x})\,. \tag{3.17}$$

Taking Lie derivatives of the system output z yields

$$\dot{z} = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}} = L_{\boldsymbol{f}} h(\boldsymbol{x}) + \underbrace{L_{\boldsymbol{g}} h(\boldsymbol{x})}_{=0} u\,, \tag{3.18}$$

$$\ddot{z} = L_{\boldsymbol{f}}^2 h(\boldsymbol{x}) + \underbrace{L_{\boldsymbol{g}} L_{\boldsymbol{f}} h(\boldsymbol{x})}_{=0} u\,, \tag{3.19}$$

$$\vdots$$

$$z^{(r-1)} = L_{\boldsymbol{f}}^{r-1} h(\boldsymbol{x}) + \underbrace{L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{r-2} h(\boldsymbol{x})}_{=0} u\,, \tag{3.20}$$

$$z^{(r)} = L_{\boldsymbol{f}}^{r} h(\boldsymbol{x}) + \underbrace{L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{r-1} h(\boldsymbol{x})}_{\neq 0} u\,. \tag{3.21}$$

The number of derivatives which have to be taken until the term $L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{r-1} h(\boldsymbol{x})$ is not equal to zero is called the relative degree r. Roughly speaking, it determines through how many integrators the input influences the output. For the systems (3.13)–(3.14), the relative degree is $r \leq 2\,n_{\mathrm{f}}$.

Example 3.2 In order to highlight similarities to the fully actuated case, the Byrnes-Isidori normal form is stated directly for the equations of motion (2.1)–(2.2) in ODE form with $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{y}^{\mathsf{T}} & \boldsymbol{v}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$. Then it is

$$\begin{aligned}
\dot{z} &= L_{\boldsymbol{f}} h(\boldsymbol{x}) + \underbrace{L_{\boldsymbol{g}} h(\boldsymbol{x})}_{=0} u = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{v}\,, \\
\ddot{z} &= L_{\boldsymbol{f}}^2 h(\boldsymbol{x}) + \underbrace{L_{\boldsymbol{g}} L_{\boldsymbol{f}} h(\boldsymbol{x})}_{\neq 0} u \\
&= \boldsymbol{H}\boldsymbol{M}^{-1}\left(\boldsymbol{q} - \boldsymbol{k}\right) + \bar{\boldsymbol{h}} + \boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{B}u\,.
\end{aligned}$$

The non-vanishing term $L_{\boldsymbol{g}} L_{\boldsymbol{f}} h(\boldsymbol{x})$ corresponds to the decoupling matrix $\boldsymbol{P} = \boldsymbol{H}\boldsymbol{M}^{-1}\boldsymbol{B}$, see also similarities to equation (3.5). Thus, SISO fully actuated systems have relative degree $r = 2$.

The previous computations form the basis for a coordinate transformation to new coordinates which include the output z and its first $r-1$ derivatives. The

new coordinates $\bar{\boldsymbol{x}} \in \mathbb{R}^{2n_\mathrm{f}}$ are chosen as

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_r \\ \bar{\boldsymbol{x}}_{(r+1):2n_\mathrm{f}} \end{bmatrix} = \boldsymbol{\Psi}(\boldsymbol{x}) = \begin{bmatrix} z \\ \dot{z} \\ \vdots \\ z^{(r-1)} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} h(\boldsymbol{x}) \\ L_{\boldsymbol{f}} h(\boldsymbol{x}) \\ \vdots \\ L_{\boldsymbol{f}}^{r-1} h(\boldsymbol{x}) \\ \boldsymbol{\Psi}_{(r+1):2n_\mathrm{f}}(\boldsymbol{x}) \end{bmatrix} \tag{3.22}$$

with a function $\boldsymbol{\Psi} : \mathbb{R}^{2n_\mathrm{f}} \to \mathbb{R}^{2n_\mathrm{f}}$. The last part $\boldsymbol{\eta} \in \mathbb{R}^{2n_\mathrm{f}-r}$ of the new states $\bar{\boldsymbol{x}}$ must be chosen such that the transformation $\boldsymbol{\Psi}(\boldsymbol{x})$ is a local diffeomorphism [Sastry99]. As a rule of thumb, the unactuated coordinates are a good choice for the states $\boldsymbol{\eta}$ in case the system output function $h(\boldsymbol{x})$ contains all actuated coordinates [Seifried14]. Substituting the new coordinates $\bar{\boldsymbol{x}}$ into the system dynamics (3.13) yields the Byrnes-Isidori normal form

$$\dot{\bar{\boldsymbol{x}}} = \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \vdots \\ \dot{\bar{x}}_{r-1} \\ \dot{\bar{x}}_r \\ \dot{\bar{\boldsymbol{x}}}_{(r+1):2n_\mathrm{f}} \end{bmatrix} = \begin{bmatrix} \bar{x}_2 \\ \bar{x}_3 \\ \vdots \\ \bar{x}_r \\ \alpha(\bar{\boldsymbol{x}}) + \beta(\bar{\boldsymbol{x}}) u \\ \boldsymbol{\rho}(\bar{\boldsymbol{x}}) + \boldsymbol{\sigma}(\bar{\boldsymbol{x}}) u \end{bmatrix} = \begin{bmatrix} \bar{x}_2 \\ \bar{x}_3 \\ \vdots \\ \bar{x}_r \\ L_{\boldsymbol{f}}^r h(\bar{\boldsymbol{x}}) + L_{\boldsymbol{g}} L_{\boldsymbol{f}}^{r-1} h(\bar{\boldsymbol{x}}) u \\ \boldsymbol{\rho}(\bar{\boldsymbol{x}}) + \boldsymbol{\sigma}(\bar{\boldsymbol{x}}) u \end{bmatrix} \tag{3.23}$$

with the functions $\alpha, \beta : \mathbb{R}^{2n_\mathrm{f}} \to \mathbb{R}$ and $\boldsymbol{\rho}, \boldsymbol{\sigma} : \mathbb{R}^{2n_\mathrm{f}} \to \mathbb{R}^{2n_\mathrm{f}-r}$. Thereby, the terms $\alpha(\bar{\boldsymbol{x}})$ and $\beta(\bar{\boldsymbol{x}})$ are expressed in the new coordinates $\bar{\boldsymbol{x}}$. The functions $\boldsymbol{\rho}(\bar{\boldsymbol{x}})$ and $\boldsymbol{\sigma}(\bar{\boldsymbol{x}})$ are given by the coordinate transformation $\boldsymbol{\Psi}(\boldsymbol{x})$ for a specific choice $\boldsymbol{\eta}$. The inverse model for tracking a desired trajectory $z_\mathrm{d}(t)$ can be derived from the Byrnes-Isidori normal form by solving the r^{th} part of equation (3.23) for the system input u. Substituting the desired trajectory $z_\mathrm{d}(t)$ into $\bar{\boldsymbol{x}} = \begin{bmatrix} z & \dot{z} & \dots & z^{(r-1)} & \boldsymbol{\eta} \end{bmatrix}^\mathsf{T}$ and into equation (3.23) yields the feedforward control input with

$$u_\mathrm{ffw} = \frac{1}{\beta\left(z_\mathrm{d}, \dot{z}_\mathrm{d}, \dots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta}\right)} \left(z_\mathrm{d}^{(r)} - \alpha\left(z_\mathrm{d}, \dot{z}_\mathrm{d}, \dots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta}\right) \right) . \tag{3.24}$$

The invertibility of the term $\beta(z_\mathrm{d}, \dot{z}_\mathrm{d}, \dots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta})$ follows directly from the definition of the relative degree, see equation (3.21). Equivalently to the differentially flat case, a possibly high derivative of the desired output $\boldsymbol{z}_\mathrm{d}(t)$ appears in the feedforward control law (3.24). Therefore, the desired output trajectory must be r times continuously differentiable for the control law to exist. However, in contrast to the differentially flat case, the control input u_ffw is not only

a function of the desired output $z_\mathrm{d}(t)$ and its derivatives, but also a function of the coordinates $\boldsymbol{\eta}$. The coordinates $\boldsymbol{\eta}$ describe the internal dynamics

$$\begin{aligned}\dot{\boldsymbol{\eta}} = \boldsymbol{\rho}\left(z_\mathrm{d}, \dot{z}_\mathrm{d}, \ldots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta}\right) + \\ \frac{\boldsymbol{\sigma}\left(z_\mathrm{d}, \dot{z}_\mathrm{d}, \ldots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta}\right)}{\beta\left(z_\mathrm{d}, \dot{z}_\mathrm{d}, \ldots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta}\right)}\left(z_\mathrm{d}^{(r)} - \alpha\left(z_\mathrm{d}, \dot{z}_\mathrm{d}, \ldots, z_\mathrm{d}^{(r-1)}, \boldsymbol{\eta}\right)\right),\end{aligned} \tag{3.25}$$

which are obtained by substituting equation (3.24) into the last part of equation (3.23). The complete inverse model is then given by equations (3.24) and (3.25). Note that the inverse model is now a dynamical system due to the internal dynamics (3.25), which must be solved to evaluate the control law (3.24). The presented approach can be generalized for multi-input multi-output (MIMO) systems, see e.g. [Sastry99, Seifried14]. Then, a decoupling matrix must be found to determine the relationship between each input and output channel. The concept of the relative degree is extended to a vector relative degree $\boldsymbol{r} = \{r_1, r_2, \ldots r_{n_\mathrm{u}}\}$, where each output is differentiated until an input appears and a regular decoupling matrix exists between the input and output channels [Isidori96]. Note that for $r = \Sigma r_i = 2n_\mathrm{f}$, the internal dynamics vanishes and the inverse model is completely algebraic. This is the case for fully actuated multibody systems considered above and for differentially flat systems.
The internal dynamics (3.25) are ODEs, which are driven by the desired trajectory $z_\mathrm{d}(t)$. Stability analysis is usually performed using the concept of zero dynamics. The zero dynamics is defined by zeroing the output $z(t) = 0$ for all t, such that

$$\dot{\boldsymbol{\eta}} = \boldsymbol{\rho}(0, 0, \ldots, 0, \boldsymbol{\eta}) - \frac{\boldsymbol{\sigma}(0, 0, \ldots, 0, \boldsymbol{\eta})}{\beta(0, 0, \ldots, 0, \boldsymbol{\eta})}\,\alpha(0, 0, \ldots, 0, \boldsymbol{\eta}) = \boldsymbol{\gamma}(\boldsymbol{\eta}) \tag{3.26}$$

with $\boldsymbol{\gamma} : \mathbb{R}^{2n_\mathrm{f}-r} \to \mathbb{R}^{2n_\mathrm{f}-r}$. Zero dynamics can be interpreted as the possible motion of a system with the output $\boldsymbol{z}$ hold constant [Miu91]. The linearization of the zero dynamics for small values $\widetilde{\boldsymbol{\eta}}$ around the equilibrium $\boldsymbol{\eta}_\mathrm{eq}$ is

$$\dot{\boldsymbol{\eta}} = \left.\frac{\partial \boldsymbol{\gamma}(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}\right|_{\boldsymbol{\eta}_\mathrm{eq}} \widetilde{\boldsymbol{\eta}} = \boldsymbol{A}_\eta\, \widetilde{\boldsymbol{\eta}}\,. \tag{3.27}$$

An eigenvalue analysis of the linearized system matrix $\boldsymbol{A}_\eta \in \mathbb{R}^{(2n_\mathrm{f}-r)\times(2n_\mathrm{f}-r)}$ determines stability of the zero dynamics. Systems with stable internal dynamics are called minimum phase systems, while systems with unstable internal dynamics are denoted non-minimum phase systems [SlotineLi91]. Note that the relative degree and zero dynamics are nonlinear extensions to the concept of pole excess and invariant zeros of linear systems, see e.g. [Svaricek06]. Application of the above procedure is shown in the following example.

Example 3.3 The mass-on-car system introduced in Sec. 2.4.3 is considered in configuration $\alpha = 0$ and without the first mass. The equations of motion are

$$\begin{bmatrix} m_1 + m_2 & m_2 \\ m_2 & m_2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{s} \end{bmatrix} + \begin{bmatrix} 0 \\ ks + d\dot{s} \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} . \tag{3.28}$$

The system output is the position of the second mass with $z = x + s$. Differentiating the system output yields

$$\dot{z} = \dot{x} + \dot{s} , \tag{3.29}$$

$$\ddot{z} = \ddot{x} + \ddot{s} = -\frac{ks + d\dot{s}}{m_2} , \tag{3.30}$$

$$\dddot{z} = -\frac{k\dot{s} + d\ddot{s}}{m_2} = -\frac{k\dot{s}}{m_2} + \frac{d}{m_1\, m_2} \left(u + \frac{(m_1 + m_2)(ks + d\dot{s})}{m_2} \right) . \tag{3.31}$$

Since the system input u first appears in $\dddot{z}$, the system has relative degree $r = 3$ and internal dynamics of dimension $2\, n_\mathrm{f} - r = 1$ remain. According to the rule of thumb in [Seifried14], the coordinate η of the internal dynamics is chosen as the unactuated coordinate, $\eta = s$, since the output z contains all actuated coordinates. Thus, the new coordinates are

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \end{bmatrix} = \begin{bmatrix} z \\ \dot{z} \\ \ddot{z} \\ \eta \end{bmatrix} = \begin{bmatrix} x + s \\ \dot{x} + \dot{s} \\ \ddot{x} + \ddot{s} \\ s \end{bmatrix} . \tag{3.32}$$

Expressing the dynamics in the new coordinates $\bar{\boldsymbol{x}}$ yields the Byrnes-Isidori normal form

$$\dot{\bar{\boldsymbol{x}}} = \begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \\ \dot{\bar{x}}_3 \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \bar{x}_2 \\ \bar{x}_3 \\ \dfrac{d}{m_2\, m_1} \left(u - \bar{x}_3 (m_1 + m_2) \right) + \dfrac{k\, (\bar{x}_3 m_2 + k\eta)}{d\, m_2} \\ -\dfrac{m_2\, \bar{x}_3 + k\, \eta}{d} \end{bmatrix} . \tag{3.33}$$

The feedforward control input is obtained from substituting the desired trajectory into the third equation of (3.33) as

$$u_\mathrm{ffw} = \frac{m_1\, m_2}{d} \left(\dddot{z}_\mathrm{d}(t) + \frac{k\, (m_2 \ddot{z}_\mathrm{d}(t) + k\eta)}{d\, m_2} - \frac{d(m_1 + m_2) \ddot{z}_\mathrm{d}(t)}{m_1\, m_2} \right) . \tag{3.34}$$

Thereby, the coordinate η is obtained by integrating the internal dynamics in the fourth part of equation (3.33).

The Byrnes-Isidori normal form is quite burdensome to derive for general multibody systems due to several reasons. The derivatives of the output $\boldsymbol{z}$ must be computed in algebraic form and the system must be given in the input-affine form (3.13). Moreover, the nonlinear state transformation $\boldsymbol{\Psi}(\boldsymbol{x})$ must be found. Therefore, flexible multibody systems with multiple inputs and outputs and many unactuated degrees of freedom are difficult to handle using this approach.

3.2 Framework of Servo-constraints for Model Inversion

The above approaches have limitations in their application to either fully actuated multibody systems or systems with few degrees of freedom. Alternatively, the inverse model can be computed with the servo-constraints approach. Servo-constraints are also called control-, path- or program-constraints, see e.g. [Campbell95]. A general framework for the use of servo-constraints in the context of multibody system dynamics is introduced in [BlajerKołodziejczyk04, BlajerKołodziejczyk07]. Moreover, the concept is formulated in the framework of Maggi's approach in [Chen08]. In the proposed approach, the equations of motion are appended by algebraic constraints which constrain the system output $\boldsymbol{z}$ to be equal to the desired output trajectory $\boldsymbol{z}_\mathrm{d}(t)$. This results in DAEs, which have to be solved numerically for the feedforward control input $\boldsymbol{u}_\mathrm{ffw}$.
While some references solve the inverse model DAEs with backwards differentiation formulas [FumagalliEtAl10, MobergHanssen10], the implicit Euler scheme is used most of the time to solve the arising set of DAEs [BlajerKołodziejczyk04, OttoSeifried18c, SeifriedBlajer13]. This is due to its advantages such as simple implementation, real-time applicability and stability properties [Hairer02]. Most common application examples are differentially flat systems, which include cranes [OttoSeifried18c, AltmannBetschYang16, BetschQuasemUhlar09] and mass-spring chains [AltmannHeiland17, OttoSeifried18b, FumagalliEtAl10]. In the differentially flat case, the inverse model has no dynamics and the implicit Euler is a reasonable choice because its numerical damping does not perturb the solution. However, servo-constraints are also applicable to non-flat systems, which have either stable or unstable internal dynamics. Minimum phase systems with stable internal dynamics are considered for example in [SeifriedBlajer13, OttoSeifried18a]. In that case, the solver must be chosen carefully to avoid non-physical damping of the internal dynamics. Non-minimum phase systems with unstable internal dynamics are considered in [BrülsBastosSeifried13, BastosSeifriedBrüls17, DrückerSeifried20]. The servo-constraints approach must be formulated as a boundary value problem, because the unstable internal dynamics prohibit an integration forward in time. The adaption of the servo-constraints framework for non-minimum phase systems is presented in Chap. 6.
The inverse model DAEs are generally of higher differentiation index [Campbell95]. For example, the index is 5 for overhead cranes [BlajerKołodziejczyk04]. For mass-spring chains, it increases by two for each additional mass [OttoSeifried18b]. However, most of the DAE solvers are developed for index 1

systems. The problem becomes ill-conditioned for larger differentiation indices and index reduction methods are advisable to reduce the index [Hairer02]. In the framework of servo-constraints, minimal extension [AltmannBetschYang16, YangBetschZhang20] and projection [BlajerKołodziejczyk04] are popular choices for index reduction. In order to avoid solving a set of DAEs, a reformulation as an optimization problem is proposed in [AltmannHeiland17].
In the following, the general framework of servo-constraints is presented and different system configurations arising in this context are introduced. Moreover, methods to compute the system zeros and poles are reviewed in order to determine if a model is differentially flat, minimum phase or non-minimum phase. The relationship between relative degree and the differentiation index of the inverse model DAEs is presented afterwards. Since the differentiation index can be considerably higher than 1, various index reduction approaches are presented which are commonly used in this context.

3.2.1 General Framework

The general framework of servo-constraints in the context of multibody systems is introduced according to [BlajerKołodziejczyk04]. The approach is motivated from describing a general multibody system in redundant coordinates. Similar to classical geometric constraints $\boldsymbol{c}(\boldsymbol{y}, t)$, the servo-constraints

$$\boldsymbol{s}(\boldsymbol{y}, t) = \boldsymbol{z} - \boldsymbol{z}_\mathrm{d}(t) = \boldsymbol{h}(\boldsymbol{y}) - \boldsymbol{z}_\mathrm{d}(t) = \boldsymbol{0} \tag{3.35}$$

constrain the system output $\boldsymbol{z}$ to be equal to the desired output $\boldsymbol{z}_\mathrm{d}(t)$ with $\boldsymbol{s} : \mathbb{R}^{n_\mathrm{f}+n_\mathrm{c}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{u}}$. Adding the servo-constraints $\boldsymbol{s}(\boldsymbol{y}, t)$ to the equations of motion (2.4)–(2.6) yields the DAEs

$$\dot{\boldsymbol{y}} = \boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} \tag{3.36}$$

$$\boldsymbol{M}(\boldsymbol{y}, t)\,\dot{\boldsymbol{v}} + \boldsymbol{k}(\boldsymbol{y}, \boldsymbol{v}, t) = \boldsymbol{q}(\boldsymbol{y}, \boldsymbol{v}, t) + \boldsymbol{C}(\boldsymbol{y}, t)^\mathsf{T}\boldsymbol{\lambda} + \boldsymbol{B}(\boldsymbol{y})\boldsymbol{u} \tag{3.37}$$

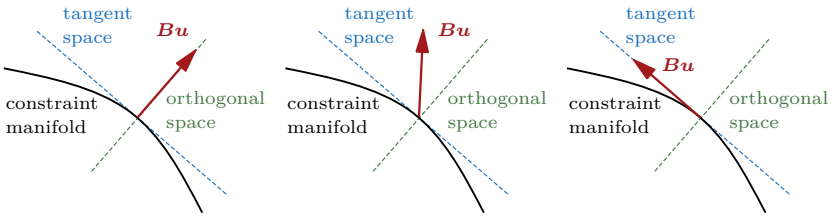
$$\boldsymbol{c}(\boldsymbol{y}, t) = \boldsymbol{0} \tag{3.38}$$

$$\boldsymbol{s}(\boldsymbol{y}, t) = \boldsymbol{0}\,. \tag{3.39}$$

The set of equations (3.36)–(3.39) has $2n_\mathrm{f} + 3n_\mathrm{c} + n_\mathrm{u}$ equations and the same number of unknowns. The unknowns are the position coordinates $\boldsymbol{y}$, the velocity coordinates $\boldsymbol{v}$, the Lagrange multipliers $\boldsymbol{\lambda}$ and the system input $\boldsymbol{u}$. In the context of this work, it is assumed that there exists a unique solution to the DAEs (3.36)–(3.39). Some remarks about the existence of solutions for DAEs are given in Sec. 3.3.1. Since the solution contains the system input $\boldsymbol{u}$, it can be directly applied as a feedforward system input $\boldsymbol{u}_\mathrm{ffw}$. Thus, the DAEs (3.36)–(3.39) describe the inverse model and will be referred to as inverse model DAEs in the following. In contrast to the inversion approaches presented before, the inverse model is solved numerically. Algebraic manipulations are not necessary to compute the feedforward control input $\boldsymbol{u}_\mathrm{ffw}$. However, some manipulations help to gain further insight into the problem properties and are discussed in the following.

3.2.2 Configurations of Servo-constraint Problems

The inverse model DAEs (3.36)–(3.39) remain structurally similar to the set of DAEs (2.4)–(2.6) which describe the forward dynamics of a multibody system. While the Lagrange multipliers $\boldsymbol{\lambda}$ enforce the geometric constraints $\boldsymbol{c}(\boldsymbol{y}, t)$, the system input $\boldsymbol{u}$ enforces the servo-constraints $\boldsymbol{s}(\boldsymbol{y}, t)$. Thus, the terms $\boldsymbol{C}(\boldsymbol{y}, t)^{\mathsf{T}}\boldsymbol{\lambda}$ and $\boldsymbol{B}(\boldsymbol{y})\boldsymbol{u}$ have some similarities. However, there is also a difference. While the matrix $\boldsymbol{C}(\boldsymbol{y}, t)$ is the Jacobian of the constraints $\boldsymbol{c}(\boldsymbol{y}, t)$, this is generally not true for the input distribution matrix $\boldsymbol{B}$. While the generalized reaction forces $\boldsymbol{\lambda}$ act orthogonal to the tangent of the constraint manifold, this might not be the case for the system input $\boldsymbol{u}$. Possible configurations are distinguished according to [BlajerKołodziejczyk04]. If the system input $\boldsymbol{Bu}$ is orthogonal to the tangent of the constraint manifold, the inverse model is in ideal orthogonal configuration. In case the system input $\boldsymbol{Bu}$ includes directions of both the orthogonal and tangential manifold, the inverse system is in non-ideal orthogonal configuration. The system is in tangential configuration in case the system input lies completely on the tangent of the constraint manifold. Figure 3.3 visualizes the different configurations. For MIMO systems, there can be orthogonal and tangential configurations for individual input and output channels, which yields a mixed tangential-orthogonal realization.



(a) Orthogonal. (b) Non-ideal orthogonal. (c) Tangential.

Figure 3.3: Configurations of the system input $\boldsymbol{Bu}$ with respect to the tangent of the constraint manifold [BlajerKołodziejczyk04].

In order to determine the system configuration, some manipulations of the inverse model DAEs are necessary. To this end, both algebraic variables $\boldsymbol{\lambda}$ and $\boldsymbol{u}$, their distribution matrices as well as both constraints are collected in

$$\boldsymbol{\mu}_{\mathrm{aug}} = \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{u} \end{bmatrix}, \quad \boldsymbol{G}_{\mathrm{aug}}(\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{C}(\boldsymbol{y}) \\ \boldsymbol{B}(\boldsymbol{y})^{\mathsf{T}} \end{bmatrix}, \quad \boldsymbol{c}_{\mathrm{aug}}(\boldsymbol{y}, t) = \begin{bmatrix} \boldsymbol{c}(\boldsymbol{y}, t) \\ \boldsymbol{s}(\boldsymbol{y}, t) \end{bmatrix}. \tag{3.40}$$

Thereby is $\boldsymbol{\mu}_{\text{aug}} \in \mathbb{R}^{n_\text{c}+n_\text{u}}$, $\boldsymbol{G}_{\text{aug}} : \mathbb{R}^{n_\text{y}} \to \mathbb{R}^{(n_\text{c}+n_\text{u})\times n_\text{y}}$, $\boldsymbol{c}_{\text{aug}} : \mathbb{R}^{n_\text{y}} \times \mathbb{R} \to \mathbb{R}^{n_\text{c}+n_\text{u}}$ with $n_\text{y} = n_\text{f} + n_\text{c}$. The inverse model DAEs are summarized as

$$\dot{\boldsymbol{y}} = \boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} \tag{3.41}$$

$$\boldsymbol{M}(\boldsymbol{y},t)\dot{\boldsymbol{v}} + \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) = \boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{G}_{\text{aug}}(\boldsymbol{y})^\mathsf{T}\,\boldsymbol{\mu}_{\text{aug}} \tag{3.42}$$

$$\boldsymbol{c}_{\text{aug}}(\boldsymbol{y},t) = \boldsymbol{0}\,. \tag{3.43}$$

Differentiating the constraints yields

$$\dot{\boldsymbol{c}}_{\text{aug}}(\boldsymbol{y},\boldsymbol{v},t) = \underbrace{\frac{\partial \boldsymbol{c}_{\text{aug}}(\boldsymbol{y},t)}{\partial \boldsymbol{y}}\boldsymbol{Z}(\boldsymbol{y})}_{\boldsymbol{C}_{\text{aug}}(\boldsymbol{y},t)}\boldsymbol{v} + \underbrace{\frac{\partial \boldsymbol{c}_{\text{aug}}(\boldsymbol{y},t)}{\partial t}}_{\boldsymbol{c}_{\text{aug,t}}(\boldsymbol{y},t)} = \boldsymbol{0}\,, \tag{3.44}$$

$$\ddot{\boldsymbol{c}}_{\text{aug}}(\boldsymbol{y},\boldsymbol{v},\dot{\boldsymbol{v}},t) = \boldsymbol{C}_{\text{aug}}(\boldsymbol{y},t)\,\dot{\boldsymbol{v}} + \dot{\boldsymbol{C}}_{\text{aug}}(\boldsymbol{y},\boldsymbol{v},t)\,\boldsymbol{v} + \dot{\boldsymbol{c}}_{\text{aug,t}}(\boldsymbol{y},t) = \boldsymbol{0} \tag{3.45}$$

with $\boldsymbol{C}_{\text{aug}} : \mathbb{R}^{n_\text{y}} \times \mathbb{R} \to \mathbb{R}^{(n_\text{c}+n_\text{u})\times n_\text{y}}$ and $\boldsymbol{c}_{\text{aug,t}} : \mathbb{R}^{n_\text{y}} \times \mathbb{R} \to \mathbb{R}^{n_\text{c}+n_\text{u}}$. Note that due to the servo-constraints, it is $\frac{\partial \boldsymbol{c}_{\text{aug}}(\boldsymbol{y},t)}{\partial \boldsymbol{y}}\boldsymbol{Z}(\boldsymbol{y}) = \boldsymbol{C}_{\text{aug}}(\boldsymbol{y},t) \neq \boldsymbol{G}_{\text{aug}}(\boldsymbol{y},t)$. Substituting the constraint $\ddot{\boldsymbol{c}}_{\text{aug}}(\boldsymbol{y},\boldsymbol{v},\dot{\boldsymbol{v}},t)$ on acceleration level for the constraint $\boldsymbol{c}_{\text{aug}}(\boldsymbol{y},t)$ on position level leads to the formulation

$$\underbrace{\begin{bmatrix} \boldsymbol{I}_{n_\text{y}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}(\boldsymbol{y},t) & -\boldsymbol{G}_{\text{aug}}(\boldsymbol{y})^\mathsf{T} \\ \boldsymbol{0} & \boldsymbol{C}_{\text{aug}}(\boldsymbol{y},t) & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{\Gamma}(\boldsymbol{y},t)} \begin{bmatrix} \dot{\boldsymbol{y}} \\ \dot{\boldsymbol{v}} \\ \boldsymbol{\mu}_{\text{aug}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} \\ \boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) \\ -\dot{\boldsymbol{C}}_{\text{aug}}(\boldsymbol{y},\boldsymbol{v},t)\,\boldsymbol{v} - \dot{\boldsymbol{c}}_{\text{aug,t}}(\boldsymbol{y},t) \end{bmatrix} \tag{3.46}$$

with the matrix function $\boldsymbol{\Gamma} : \mathbb{R}^{n_\text{y}} \times \mathbb{R} \to \mathbb{R}^{(2n_\text{f}+3n_\text{c}+n_\text{u})\times(2n_\text{f}+3n_\text{c}+n_\text{u})}$. For a regular matrix $\boldsymbol{\Gamma}(\boldsymbol{y},t)$ with rank $\text{rk}(\boldsymbol{\Gamma}) = 2n_\text{f}+3n_\text{c}+n_\text{u}$, equation (3.46) can be solved for the algebraic variables $\boldsymbol{\mu}_{\text{aug}}$. In that case, the inverse model is in ideal or non-ideal orthogonal realization [BrülsBastosSeifried13, FumagalliEtAl10]. For rank $2n_\text{f} + 3n_\text{c} < \text{rk}(\boldsymbol{\Gamma}) < 2n_\text{f} + 3n_\text{c} + n_\text{u}$, the system is in mixed tangential-orthogonal realization, where $\text{rk}(\boldsymbol{\Gamma}) - 2n_\text{f} - 3n_\text{c}$ components of the output can be reached directly. In case the rank is $\text{rk}(\boldsymbol{\Gamma}) = 2n_\text{f} + 3n_\text{c}$, the system is in tangential configuration and no output components can be influenced directly. A tangential configuration is often characteristic for differentially flat systems, but is not restricted to them [SeifriedBlajer13, OttoSeifried18b].

For systems written in ODE form (2.1)–(2.2) with added servo-constraint (3.35), the following matrix conditions can also be used to determine the configuration [BlajerKołodziejczyk04]. Differentiating the system output yields

$$\dot{\boldsymbol{z}} = \underbrace{\frac{\partial \boldsymbol{h}(\boldsymbol{y})}{\partial \boldsymbol{y}}\boldsymbol{Z}(\boldsymbol{y})}_{\boldsymbol{H}(\boldsymbol{y})}\,\boldsymbol{v}\,, \tag{3.47}$$

$$\ddot{\boldsymbol{z}} = \boldsymbol{H}(\boldsymbol{y})\dot{\boldsymbol{v}} + \dot{\boldsymbol{H}}(\boldsymbol{y},\dot{\boldsymbol{y}})\boldsymbol{v}\,. \tag{3.48}$$

The accelerations $\dot{\boldsymbol{v}}$ can be obtained from the equations of motion (2.2) and are substituted into equation (3.48) to yield

$$\ddot{\boldsymbol{z}} = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y},t)^{-1}\left(\boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y})\boldsymbol{u}\right) + \dot{\boldsymbol{H}}(\boldsymbol{y},\dot{\boldsymbol{y}})\boldsymbol{v}\,. \tag{3.49}$$

Equivalently to the concepts described in Sec. 3.1, the matrix

$$\boldsymbol{P}(\boldsymbol{y},t) = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y},t)^{-1}\boldsymbol{B}(\boldsymbol{y}) \tag{3.50}$$

with $\boldsymbol{P} : \mathbb{R}^{n_\mathrm{f}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ describes the input-output relationship. The rank of the matrix $\boldsymbol{P}(\boldsymbol{y},t)$ is a measure for the system configuration [BlajerKołodziejczyk04, SeifriedBlajer13]. For full rank $\mathrm{rk}(\boldsymbol{P}) = n_\mathrm{u}$, the system input $\boldsymbol{u}$ directly influences all system outputs and the matrix $\boldsymbol{P}(\boldsymbol{y},t)$ is invertible. Thus, the system is of ideal or non-ideal configuration. For this case, the differentiation index of the DAEs (3.36)–(3.39) is 3. In case the matrix $\boldsymbol{P}(\boldsymbol{y},t)$ is rank deficient with $0 < \mathrm{rk}(\boldsymbol{P}) < n_\mathrm{u}$, only $\mathrm{rk}(\boldsymbol{P})$ components of the output can be directly influenced by the system input. This is a mixed tangential-orthogonal realization. In case the rank is $\mathrm{rk}(\boldsymbol{P}) = 0$, no output can be directly reached by an input, and the inverse system is in tangential configuration.
The system configuration gives important information about the DAE index. For an ideal or non-ideal configuration, the inverse model DAEs (3.36)–(3.39) have differentiation index 3. For mixed tangential-orthogonal or tangential configuration, the DAE index is larger than 3. Besides the DAE index, the relative degree and stability of the zero dynamics are important properties of the system and are analyzed in the following.

3.2.3 Computation of System Zeros and Poles

The zero dynamics introduced in Sec. 3.1.3 is a nonlinear extension to zeros of linear systems [Svaricek06]. An overview of zeros for linear systems is given e.g. in [HoaggBernstein07], while a general survey of the historic development is given in [SchraderSain89]. System zeros of mechanical systems, such as flexible beams, are interpreted and analyzed in [Miu91, Williams92]. According to [HoaggBernstein07], a zero can have different effects on the behavior of a system. For SISO systems, each zero in the left-half plane blocks a certain input signal and for each zero in the right-half plane, a certain input signal becomes unbounded. Moreover, zeros can cause over- and undershoots of the transient response. The extension of those effects for linear MIMO systems is similar, but more involved [HoaggBernstein07].
Briefly looking at the linear SISO case highlights the important properties for the following discussion. In frequency domain with $s \in \mathbb{C}$, the scalar transfer function $G : \mathbb{C} \to \mathbb{C}$ connects the scalar input U and output Y with

$$Y = G(s)\,U\,. \tag{3.51}$$

The transfer function consists of the numerator and denominator polynomials $p_\mathrm{n} : \mathbb{C} \to \mathbb{C}$ and $p_\mathrm{d} : \mathbb{C} \to \mathbb{C}$ respectively, such that

$$G(s) = \frac{p_\mathrm{n}(s)}{p_\mathrm{d}(s)}\,. \tag{3.52}$$

The roots of the numerator polynomial $p_\mathrm{n}(s)$ are called zeros, while the roots of the denominator polynomial $p_\mathrm{d}(s)$ are called poles of the system $G(s)$. The inverse model of the system dynamics (3.51) is

$$U = G(s)^{-1}\, Y = \frac{p_\mathrm{d}(s)}{p_\mathrm{n}(s)}\, Y\,. \tag{3.53}$$

The zeros of the forward dynamics become the poles of the inverse dynamics and vice versa. In the following, the notation of zero and pole refers to a zero and pole of the forward dynamics, if not denoted otherwise. For zeros in the left-half plane, the inverse model has poles in the left-half plane and is stable. The system is called minimum phase. A zero in the right-half plane becomes a pole in the right-half plane of the inverse system. The inverse system becomes unstable and is called non-minimum phase. For differentially flat systems, there are no system zeros and the inverse $G(s)^{-1}$ is a purely algebraic relation. Note the equivalence to the discussion of the stability of zeros dynamics in Sec. 3.1.3. Exact tracking is not possible for non-minimum phase systems [SlotineLi91].
While zeros are easy to define for SISO systems, the concept is more involved for MIMO systems and multiple notions of zeros are developed [SchraderSain89]. Here, the concepts of invariant and transmission zeros are briefly introduced. For linear MIMO systems of the form (2.32), the Rosenbrock matrix is given by

$$R(s) = \begin{bmatrix} s\boldsymbol{E}_\mathrm{lin} - \boldsymbol{A}_\mathrm{lin} & \boldsymbol{B}_\mathrm{lin} \\ -\boldsymbol{H}_\mathrm{lin} & \boldsymbol{0} \end{bmatrix} \tag{3.54}$$

with $\boldsymbol{R} : \mathbb{C} \to \mathbb{C}^{(2n_\mathrm{f}+3n_\mathrm{c}+n_\mathrm{u})\times(2n_\mathrm{f}+3n_\mathrm{c}+n_\mathrm{u})}$. An invariant zero is defined as the complex value $\bar{z} \in \mathbb{C}$, for which the Rosenbrock matrix $R(\bar{z})$ loses rank [HoaggBernstein07, SchraderSain89]. The values $\bar{z}$ correspond to the eigenvalues of a generalized eigenvalue problem for the matrices $\boldsymbol{A}_\mathrm{lin}$, $\boldsymbol{B}_\mathrm{lin}$, $\boldsymbol{H}_\mathrm{lin}$ [HoaggBernstein07]. Transmission zeros are the zeros concerning the input-output behavior and are defined based on the transfer function. They are a subset of the invariant zeros. Both sets coincide for a minimal realization of the system dynamics, since pole-zero cancellations cannot occur in this case.
With regard to the inverse model DAEs (3.36)–(3.39), stability of the zero dynamics and therefore position of the invariant zeros must be known in order to choose a specific approach for solving the inverse model DAEs. For minimum phase systems, the DAEs (3.36)–(3.39) can be integrated forward in time. For non-minimum phase systems, this is not possible and the approach of stable inversion is introduced in Chap. 6. However, a complete analysis based on the Byrnes-Isidori form is often not desired due to the large analytical effort. Thus,

the definition of invariant zeros based on a generalized eigenvalue problem is adapted for the inverse model described by equations (3.36)–(3.39). Linearizing the inverse model DAEs (3.41)–(3.43) according to Sec. 2.1.3 yields

$$\dot{\widetilde{\boldsymbol{y}}} = \boldsymbol{Z}_{\mathrm{lin}}^{\mathrm{y}} \widetilde{\boldsymbol{y}} + \boldsymbol{Z}_{\mathrm{lin}} \widetilde{\boldsymbol{v}} \tag{3.55}$$

$$\boldsymbol{M}_{\mathrm{lin}} \dot{\widetilde{\boldsymbol{v}}} + \boldsymbol{D}_{\mathrm{lin}} \widetilde{\boldsymbol{v}} + \boldsymbol{K}_{\mathrm{lin}} \widetilde{\boldsymbol{y}} = \boldsymbol{G}_{\mathrm{aug,lin}}^{\mathsf{T}} \widetilde{\boldsymbol{\mu}}_{\mathrm{aug}} \tag{3.56}$$

$$\boldsymbol{C}_{\mathrm{aug,lin}}^{\mathrm{y}} \widetilde{\boldsymbol{y}} = \boldsymbol{0}\,. \tag{3.57}$$

with the matrices

$$\boldsymbol{C}_{\mathrm{aug,lin}}^{\mathrm{y}} = \left. \frac{\partial \boldsymbol{c}_{\mathrm{aug}}}{\partial \boldsymbol{y}} \right|_{\boldsymbol{y}_{\mathrm{eq}}} \quad \text{and} \quad \boldsymbol{G}_{\mathrm{aug,lin}} = \boldsymbol{G}_{\mathrm{aug}}(\boldsymbol{y}_{\mathrm{eq}})\,. \tag{3.58}$$

The linearized equations are summarized as

$$\underbrace{\begin{bmatrix} \boldsymbol{I}_{n_{\mathrm{y}}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{\mathrm{lin}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{E}^\star} \begin{bmatrix} \dot{\widetilde{\boldsymbol{y}}} \\ \dot{\widetilde{\boldsymbol{v}}} \\ \dot{\widetilde{\boldsymbol{\mu}}}_{\mathrm{aug}} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{Z}_{\mathrm{lin}}^{\mathrm{y}} & \boldsymbol{Z}_{\mathrm{lin}} & \boldsymbol{0} \\ -\boldsymbol{K}_{\mathrm{lin}} & -\boldsymbol{D}_{\mathrm{lin}} & \boldsymbol{G}_{\mathrm{aug,lin}}^{\mathsf{T}} \\ \boldsymbol{C}_{\mathrm{aug,lin}}^{\mathrm{y}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{A}^\star} \begin{bmatrix} \widetilde{\boldsymbol{y}} \\ \widetilde{\boldsymbol{v}} \\ \widetilde{\boldsymbol{\mu}}_{\mathrm{aug}} \end{bmatrix} \tag{3.59}$$

with $\boldsymbol{E}^\star, \boldsymbol{A}^\star \in \mathbb{R}^{(2n_{\mathrm{f}}+3n_{\mathrm{c}}+n_{\mathrm{u}}) \times (2n_{\mathrm{f}}+3n_{\mathrm{c}}+n_{\mathrm{u}})}$. Note the difference compared to the linearized forward dynamics in the constraint equation, since for the inverse model it is $\boldsymbol{G}_{\mathrm{aug}}(\boldsymbol{y},t) \neq \frac{\partial \boldsymbol{c}_{\mathrm{aug}}(\boldsymbol{y},t)}{\partial \boldsymbol{y}} \boldsymbol{Z}(\boldsymbol{y}) = \boldsymbol{C}_{\mathrm{aug}}(\boldsymbol{y},t)$. The generalized eigenvalue problem is

$$(\boldsymbol{A}^\star - \lambda^\star \boldsymbol{E}^\star)\, \boldsymbol{c}^\star = \boldsymbol{0} \tag{3.60}$$

with $2n_{\mathrm{f}} + 3n_{\mathrm{c}} + n_{\mathrm{u}}$ generalized eigenvalues $\lambda^\star \in \mathbb{C}$ and generalized eigenvectors $\boldsymbol{c}^\star \in \mathbb{C}^{2n_{\mathrm{f}}+3n_{\mathrm{c}}+n_{\mathrm{u}}}$. There exist infinitely large eigenvalues $\lambda^\star$, which are related to the algebraic constraints [Géradin01]. They represent the algebraic constraints with infinitely fast motion related to zero mass. The finite eigenvalues $\lambda^\star$ of the generalized eigenvalue problem for equation (3.59) represent the invariant zeros of the forward system. Therefore, they determine if the system is minimum phase or non-minimum phase.

Note that in an equivalent manner, the generalized eigenvalues of the forward dynamics (2.4)–(2.6) can be determined by a generalized eigenvalue problem. For the forward dynamics, the matrices $\boldsymbol{E}^\star$ and $\boldsymbol{A}^\star$ are given as $\boldsymbol{E}^\star = \boldsymbol{E}_{\mathrm{lin}}$ and $\boldsymbol{A}^\star = \boldsymbol{A}_{\mathrm{lin}}$ of equation (2.32). The finite eigenvalues represent the poles of the forward dynamics. Using this approach, the relative degree r can be determined as difference between the finite poles and zeros of the system. Note that the derived properties are local results around the chosen linearization point.

3.2.4 Relative Degree and DAE Index

The differentiation index of the inverse model DAEs (3.36)–(3.39) is closely related to the relative degree r defined in Sec. 3.1.3. It is shown in [Campbell95] that the differentiation index is $r+1$ for the case that the internal dynamics are described by ordinary differential equations. This rule of thumb usually holds for multibody systems. However, there are few special cases for which the differentiation index can be larger than $r+1$. The concept of the differentiation index concerns the complete set of DAEs describing the dynamics, while the relative degree only concerns the input-output behavior. Thus, it is theoretically possible that the equations of motion are DAEs of index 3, while the relative degree is smaller than 2. The following example illustrates such a special case.

Example 3.4 The undamped torsional oscillator shown in Fig. 3.4 consists of three disks. In contrast to the model introduced in Sec. 2.4.2, the system input is modeled as a position-based actuator mounted at the second disk. It is enforced by the Lagrange multiplier λ and the constraint $c(\boldsymbol{y})$. Thus, the equations of motion are given using redundant coordinates as

$$\begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \\ \ddot{\varphi}_3 \end{bmatrix} + \begin{bmatrix} k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^{\mathsf{T}} \lambda \tag{3.61}$$

$$\varphi_2 - u = 0 \tag{3.62}$$

The system output is defined as the velocity of the third mass with $z = \dot{\varphi}_3$. Thus, the servo-constraint is

$$\dot{\varphi}_3 - z_{\mathrm{d}}(t) = 0\,. \tag{3.63}$$

The differentiation index of the inverse model DAEs (3.61)–(3.63) can be verified by computing the derivatives of the two constraints (3.62) and (3.63) until the algebraic variables λ and u show up. The highest derivative arises for differentiating the actuator constraint (3.62) with

$$\dot{c} = \dot{\varphi}_2 - \dot{u} = 0\,, \tag{3.64}$$

$$\ddot{c} = \ddot{\varphi}_2 - \ddot{u} = \frac{1}{I}\left(k\varphi_1 - 2k\varphi_2 + k\varphi_3 + \lambda\right) - \ddot{u} = 0\,. \tag{3.65}$$

Since the Lagrange multiplier λ appears in the second derivative of the geometric constraint, the DAEs describing the inverse model have differentiation index 3. The relative degree is verified by differentiating the system output until the input arises. This yields

$$\dot{z} = \ddot{\varphi}_3 = \frac{1}{I}\left(-k\varphi_3 + k\varphi_2\right) = \frac{1}{I}\left(-k\varphi_3 + ku\right)\,. \tag{3.66}$$

The system input u shows up in the first derivative of the system output and thus the relative degree is $r = 1$, while the DAE index is 3. The general

rule of thumb described above does not hold for this example. It can be interpreted as follows. The system output is chosen on velocity level to yield small relative degree of $r = 1$. At the same time, the dynamics of the first mass is not related to the input-output behavior. The dynamics can be interpreted as internal dynamics and it is described by index 3 DAEs.

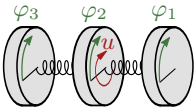


Figure 3.4: Torsional oscillator with 3 disks.

The above example is designed to be simple and to show the relevant properties at the same time. For the above system, the dynamics should not be described by the DAEs (3.61)–(3.62), but the constraint (3.62) should be substituted to obtain ODEs. However, it shows a simple example for which the rule of thumb collapses. More complex systems can also show this behavior.

3.2.5 Index Reduction Approaches

The inverse model DAEs (3.36)–(3.39) can be of high index even for simple mechanical systems. For example, the cable robot introduced in Sec. 2.4.1 has index 5 and the index of the torsional oscillator introduced in Sec. 2.4.2 increases by two for each additional disk. However, numerical methods become ill-conditioned for high index DAEs. Thus, index reduction methods are developed to reduce the differentiation index [Hairer02]. In the context of servo-constraints, popular choices are Baumgarte stabilization [BencsikKovács-Zelei17], projection [BlajerKołodziejczyk04] and minimal extension [Altmann-BetschYang16, BetschAltmannYang16]. Moreover, a reformulation of the mechanical model using redundant coordinates can help to reduce the index in special cases [OttoSeifried18c, BlajerKołodziejczyk11].
In the following, Baumgarte stabilization, the projection approach and a reformulation using redundant coordinates are briefly reviewed. Their application and effect on the numerical solution are shown in simulations in Chap. 4.

Baumgarte Stabilization

In multibody system dynamics, the geometric constraints usually constrain the position of coordinates, e.g. a constraint representing a joint. The analytical solution to the DAEs also fulfills the constraints on velocity and acceleration level [Hairer02]. This property gives rise to the idea to substitute the original constraint $\boldsymbol{c}_{\mathrm{aug}}(\boldsymbol{y}, t)$ by its first derivative $\dot{\boldsymbol{c}}_{\mathrm{aug}}(\boldsymbol{y}, \boldsymbol{v}, t)$ or second derivative $\ddot{\boldsymbol{c}}_{\mathrm{aug}}(\boldsymbol{y}, \boldsymbol{v}, \dot{\boldsymbol{v}}, t)$. It follows from the definition of the index, that this substitution reduces the index.

While the original analytical solution is also a solution to the new set of equations, there will be a drift in the numerical computation [Hairer02]. Constraining only the velocity level of the constraints yields a numerical error which grows linear in time. This is due to small rounding and truncation errors in each time step which accumulate on position level. Accordingly, fulfilling the constraints on acceleration level yields an error which grows quadratically in time. A stabilization is proposed in [Baumgarte72] to reduce the drift problem by substituting the original constraint $\boldsymbol{c}_{\mathrm{aug}}(\boldsymbol{y}, t)$ with

$$\widetilde{\boldsymbol{c}}_{\mathrm{aug}}(\boldsymbol{y}, \boldsymbol{v}, \dot{\boldsymbol{v}}, t) = \ddot{\boldsymbol{c}}_{\mathrm{aug}}(\boldsymbol{y}, \boldsymbol{v}, \dot{\boldsymbol{v}}, t) + 2\boldsymbol{\alpha}_{\mathrm{bg}}\dot{\boldsymbol{c}}_{\mathrm{aug}}(\boldsymbol{y}, \boldsymbol{v}, t) + \boldsymbol{\beta}_{\mathrm{bg}}\boldsymbol{c}_{\mathrm{aug}}(\boldsymbol{y}, t) = \boldsymbol{0}\,. \quad (3.67)$$

The diagonal matrices $\boldsymbol{\alpha}_{\mathrm{bg}} \in \mathbb{R}^{(n_{\mathrm{c}}+n_{\mathrm{u}})\times(n_{\mathrm{c}}+n_{\mathrm{u}})}$ and $\boldsymbol{\beta}_{\mathrm{bg}} \in \mathbb{R}^{(n_{\mathrm{c}}+n_{\mathrm{u}})\times(n_{\mathrm{c}}+n_{\mathrm{u}})}$ contain the Baumgarte parameters on the diagonal, which are selected such that equation (3.67) has asymptotically stable dynamics [Hairer02]. Equation (3.67) acts as a PD controller to enforce the constraints on position and velocity level. The selection of Baumgarte parameters $\boldsymbol{\alpha}_{\mathrm{bg}}$ and $\boldsymbol{\beta}_{\mathrm{bg}}$ is problem-specific and therefore not straightforward. If the parameters are chosen too large, the differential equations become stiff and more difficult to solve. If they are chosen too small, the constraints are enforced with slow transient dynamics.

Projection Approach

The projection approach in the context of servo-constraints is presented in [BlajerKołodziejczyk04] for systems modeled by ODEs and in [BlajerKołodziejczyk11] for systems modeled by DAEs. Here, it is briefly reviewed for systems modeled by the ODEs (2.1)–(2.2).
Suitable projection matrices are defined to project the equations onto the constrained and unconstrained subspaces. The constrained subspace is orthogonal to the tangent of the constraint manifold. The unconstrained subspace is the complementary subspace. In the case analyzed here, it is assumed that the servo-constraints are the only constraints on the dynamics. Thus, the projection matrix $\boldsymbol{H} : \mathbb{R}^{n_{\mathrm{f}}} \to \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{f}}}$ on the constrained subspace is the Jacobian of the servo-constraints, see equation (3.47). The projection matrix $\boldsymbol{P}^{\mathrm{u}} : \mathbb{R}^{n_{\mathrm{f}}} \to \mathbb{R}^{n_{\mathrm{f}}\times(n_{\mathrm{f}}-n_{\mathrm{u}})}$ onto the unconstrained subspace is constructed with the conditions

$$\boldsymbol{H}(\boldsymbol{y})\boldsymbol{P}^{\mathrm{u}}(\boldsymbol{y}) = \boldsymbol{0}\,, \quad \boldsymbol{P}^{\mathrm{u}}(\boldsymbol{y})^{\mathsf{T}}\boldsymbol{H}(\boldsymbol{y})^{\mathsf{T}} = \boldsymbol{0}\,. \quad (3.68)$$

Applying the projection onto the set of equations yields

$$\begin{bmatrix} \boldsymbol{P}^{\mathrm{u}}(\boldsymbol{y})^{\mathsf{T}} \\ \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y}, t)^{-1} \end{bmatrix} \left(\boldsymbol{M}(\boldsymbol{y}, t)\,\dot{\boldsymbol{v}} + \boldsymbol{k}(\boldsymbol{y}, \boldsymbol{v}, t) = \boldsymbol{q}(\boldsymbol{y}, \boldsymbol{v}, t) + \boldsymbol{B}(\boldsymbol{y})\,\boldsymbol{u}\right). \quad (3.69)$$

In these equations, it is possible to reduce the number of differential equations. From the output equation (3.48) follows $\boldsymbol{H}(\boldsymbol{y})\dot{\boldsymbol{v}} = \ddot{\boldsymbol{z}}_{\mathrm{d}}(t) - \dot{\boldsymbol{H}}(\boldsymbol{y}, \dot{\boldsymbol{y}})\boldsymbol{v}$, where the second derivative of the system output is replaced by the desired trajectory $\ddot{\boldsymbol{z}}_{\mathrm{d}}(t)$.

Substituting this information into the second projected equation (3.69) yields the algebraic equation

$$\boldsymbol{H}(\boldsymbol{y})\dot{\boldsymbol{v}} = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y},t)^{-1}\left(\boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y})\,\boldsymbol{u}\right), \tag{3.70}$$

$$\ddot{\boldsymbol{z}}_{\mathrm{d}}(t) - \dot{\boldsymbol{H}}(\boldsymbol{y},\dot{\boldsymbol{y}})\boldsymbol{v} = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y},t)^{-1}\left(\boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y})\,\boldsymbol{u}\right). \tag{3.71}$$

The complete arising DAEs are reordered to distinguish between the remaining differential and algebraic equations, such that

$$\dot{\boldsymbol{y}} = \boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} \tag{3.72}$$

$$\boldsymbol{P}^{\mathrm{u}}(\boldsymbol{y})^{\mathsf{T}}\boldsymbol{M}(\boldsymbol{y},t)\dot{\boldsymbol{v}} = \boldsymbol{P}^{\mathrm{u}}(\boldsymbol{y})^{\mathsf{T}}\left(\boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y})\,\boldsymbol{u}\right) \tag{3.73}$$

$$\begin{aligned}\boldsymbol{0} = \boldsymbol{H}(\boldsymbol{y})\boldsymbol{M}(\boldsymbol{y},t)^{-1}&\left(\boldsymbol{q}(\boldsymbol{y},\boldsymbol{v},t) - \boldsymbol{k}(\boldsymbol{y},\boldsymbol{v},t) + \boldsymbol{B}(\boldsymbol{y})\boldsymbol{u}\right) - \\ &\ddot{\boldsymbol{z}}_{\mathrm{d}}(t) + \dot{\boldsymbol{H}}(\boldsymbol{y},\dot{\boldsymbol{y}})\boldsymbol{v}\end{aligned} \tag{3.74}$$

$$\boldsymbol{0} = \boldsymbol{z}(\boldsymbol{y}) - \boldsymbol{z}_{\mathrm{d}}(t)\,. \tag{3.75}$$

The differential equations (3.72) and (3.73) are of dimensions n_{f} and $n_{\mathrm{f}} - n_{\mathrm{u}}$ respectively, while the algebraic equations (3.74) and (3.75) are each of dimension n_{u}. Using this projection, the index is reduced by 2.

Direct Substitution

It is shown in [OttoSeifried18c] that it might be convenient to describe the equations of motion using partially redundant coordinates. In the special case when the system output $\boldsymbol{z}(\boldsymbol{y})$ is directly part of the redundant coordinate vector $\boldsymbol{y}$, the above index reduction process can be simplified, since the Jacobian $\boldsymbol{H}(\boldsymbol{y})$ reduces to two submatrices, which are the identity and the null matrix. The redundant coordinate vector $\boldsymbol{y}$ can be reordered such that the output is represented as

$$\boldsymbol{z} = \begin{bmatrix}\boldsymbol{I}_{n_{\mathrm{u}}} & \boldsymbol{0}\end{bmatrix}\boldsymbol{y}\,, \tag{3.76}$$

with identity $\boldsymbol{I}_{n_{\mathrm{u}}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{u}}}$ and null matrix $\boldsymbol{0} \in \mathbb{R}^{n_{\mathrm{u}}\times(n_{\mathrm{f}}+n_{\mathrm{c}}-n_{\mathrm{u}})}$. Then, the servo-constraints and their derivatives can be formulated as

$$\boldsymbol{s}(\boldsymbol{y}) = \begin{bmatrix}\boldsymbol{I}_{n_{\mathrm{u}}} & \boldsymbol{0}\end{bmatrix}\boldsymbol{y} - \boldsymbol{z}_{\mathrm{d}}(t) = \boldsymbol{0}\,, \tag{3.77}$$

$$\dot{\boldsymbol{s}}(\boldsymbol{y},\boldsymbol{v}) = \begin{bmatrix}\boldsymbol{I}_{n_{\mathrm{u}}} & \boldsymbol{0}\end{bmatrix}\boldsymbol{Z}(\boldsymbol{y})\boldsymbol{v} - \dot{\boldsymbol{z}}_{\mathrm{d}}(t) = \boldsymbol{0}\,, \tag{3.78}$$

$$\ddot{\boldsymbol{s}}(\boldsymbol{y},\boldsymbol{v},\dot{\boldsymbol{v}}) = \begin{bmatrix}\boldsymbol{I}_{n_{\mathrm{u}}} & \boldsymbol{0}\end{bmatrix}\left(\boldsymbol{Z}(\boldsymbol{y})\dot{\boldsymbol{v}} + \dot{\boldsymbol{Z}}(\boldsymbol{y},\boldsymbol{v})\boldsymbol{v}\right) - \ddot{\boldsymbol{z}}_{\mathrm{d}}(t) = \boldsymbol{0}\,. \tag{3.79}$$

These equations represent the output and therefore some of the redundant states are given directly in terms of the desired output and its derivatives. Thus, they can be substituted into the equations of motion (3.36)–(3.38). Therefore, the first n_{u} equations of equation (3.37) reduce to algebraic equations instead of differential equations and the index is reduced by 2.

Example 3.5 The undamped torsional oscillator with 2 disks is considered for demonstration of the direct substitution approach. The system output z can be represented directly by the state vector with

$$z = \varphi_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \widetilde{\boldsymbol{y}} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \varphi_2 \\ \varphi_1 \end{bmatrix} . \tag{3.80}$$

Substituting the servo-constraint $s(\boldsymbol{y}, t) = z - z_\mathrm{d}(t) = 0$ and its first two derivatives yields the inverse model equations

$$\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \ddot{z}_\mathrm{d}(t) \\ \ddot{\varphi}_1 \end{bmatrix} + \begin{bmatrix} -k & k \\ k & -k \end{bmatrix} \begin{bmatrix} z_\mathrm{d}(t) \\ \varphi_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \, . \tag{3.81}$$

The servo-constraints can be dropped due to the substitution into the equations of motion. The resulting DAEs have a differentiation index 3 compared to original index 5.

3.3 Numerical Methods

The computation of an analytical solution for the inverse model DAEs is usually tedious or impossible. Thus, the equations are solved using numerical methods. After a brief discussion about the existence of solutions, the numerical methods which are applied in the scope of this work are presented. First, Newton's method is introduced to solve nonlinear equations, which arise within the time integration schemes for the DAEs. Afterwards, several integration schemes are discussed to solve DAEs numerically. These include implicit Runge-Kutta methods and the backwards differentiation formulas. Considerations for real-time applicability of the presented schemes are discussed and approaches to reduce the computational effort are presented.

3.3.1 Considerations for Existence of Solutions

Theoretical results on the existence of solutions to DAE problems are generally more involved compared to equivalent results for ODEs [Brenan89]. Most theoretical results exist for constant coefficient linear DAEs [Brenan89, Hairer08], but results are also stated for general DAEs [Griepentrog86]. A solution to the inverse model problem stated by the DAEs (3.36)–(3.39) does not necessarily exist or there might be infinitely many solutions. For the existence of a solution, the initial conditions $\boldsymbol{y}_0, \boldsymbol{v}_0, \boldsymbol{\lambda}_0, \boldsymbol{u}_0$ must be consistent with the algebraic constraints [Hairer02]. For index 1 problems, the algebraic constraints must be fulfilled at initial time t_0 with

$$\boldsymbol{c}_\mathrm{aug}(\boldsymbol{y}_0, t_0) = \boldsymbol{0} \, . \tag{3.82}$$

For higher index problems, the respective derivatives of the constraints must be fulfilled at initial time as well. Moreover, for the existence of at least one

solution, some conditions on the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ must be fulfilled. The desired output trajectory $\boldsymbol{z}_\mathrm{d}(t)$ must be compatible with the geometric constraints on the system and cannot impose conflicting constraints on the motion. Moreover, the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ must be at least r times continuously differentiable [Brenan89]. Note that this is also true for the inverse model based on differential flatness and the inverse model based on the Byrnes-Isidori normal form, see Sec 3.1.2 and Sec. 3.1.3 respectively. The smoothness assumption can be quite restricting on the choice of trajectories.
Approaches for feasibility conditions of servo-constraints are presented in [FumagalliEtAl10]. The generation of feasible trajectories itself is topic of active research. For example, the generation of feasible trajectories for differentially flat systems is based on the solutions of optimization problems in [ArnoldEtAl08, FloresMilam06, SchultzMurphey12, Aschemann09]. An algorithm which does not need to solve an iterative optimization problem is presented in [KnierimSawodny12].
Servo-constraints can also be applied to redundantly actuated multibody systems with more system inputs than degrees of freedom. In that case, the solution to the inverse problem is not unique. Additional requirements, such as minimizing the energy can be enforced to choose one of the solutions, see e.g. [BajodahHodgesChen05].
In the following, it is assumed that the desired trajectories are sufficiently smooth and compatible with the possible motion of the system. Moreover, consistent initial conditions are assumed.

3.3.2 Method for Solving Nonlinear Equations

Usually, time discretization schemes for initial value problems result in a set of nonlinear equations. Here, Newton's method is applied to solve the arising set of nonlinear equations [Strang10, Strang12]. For notational simplicity, the nonlinear equations of dimension n_e are collected in

$$\boldsymbol{f}_\mathrm{e}(\boldsymbol{\xi}) = \boldsymbol{0} \tag{3.83}$$

with $\boldsymbol{f}_\mathrm{e} : \mathbb{R}^{n_\mathrm{e}} \to \mathbb{R}^{n_\mathrm{e}}$ and the unknown vector $\boldsymbol{\xi} \in \mathbb{R}^{n_\mathrm{e}}$. The scheme solves for the unknown vector iteratively with

$$\boldsymbol{\chi}^i = -\left(\boldsymbol{J}^i(\boldsymbol{\xi}^i)\right)^{-1} \boldsymbol{f}_\mathrm{e}(\boldsymbol{\xi}^i) \tag{3.84}$$

$$\boldsymbol{\xi}^{i+1} = \boldsymbol{\xi}^i + \boldsymbol{\chi}^i \,. \tag{3.85}$$

Thereby, $\boldsymbol{\chi}^i \in \mathbb{R}^{n_\mathrm{e}}$ represents the update of the current guess $\boldsymbol{\xi}^i$ and the Jacobian $\boldsymbol{J}^i : \mathbb{R}^{n_\mathrm{e}} \to \mathbb{R}^{n_\mathrm{e} \times n_\mathrm{e}}$ denotes the gradient evaluated at the current guess $\boldsymbol{\xi}^i$ with

$$\boldsymbol{J}^i(\boldsymbol{\xi}^i) = \left.\frac{\partial \boldsymbol{f}_\mathrm{e}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}\right|_{\boldsymbol{\xi}=\boldsymbol{\xi}^i} \,. \tag{3.86}$$

The scheme describes a linearization of the nonlinear equation $\boldsymbol{f}_\mathrm{e}(\boldsymbol{\xi})$ at the current guess $\boldsymbol{\xi}^i$, which is then solved for its new root $\boldsymbol{\xi}^{i+1}$. An initial guess $\boldsymbol{\xi}^0$ must be supplied to start the iterative scheme. The method converges locally quadratic for a sufficiently accurate initial guess [Hermann11, Strang10]. The iterative scheme terminates when the condition

$$\left\| \boldsymbol{f}_\mathrm{e}\left(\boldsymbol{\xi}^i\right) \right\|_\mathrm{max} < \varepsilon_\mathrm{tol} \quad \text{for } i < n_\mathrm{N,max} \tag{3.87}$$

is fulfilled for some positive finite value ε_tol or if the maximum admissible number of iterations $n_\mathrm{N,max}$ is reached.

3.3.3 Methods for Solving Initial Value Problems

The inverse model DAEs (3.36)–(3.39) must be solved preferably in real-time. The solution includes the desired state trajectory and the desired system input which can be applied directly as the feedforward input $\boldsymbol{u}_\mathrm{ffw}$. Generally, it is more difficult to solve DAEs than ODEs and a solution cannot be guaranteed. Here, it is assumed that a solution exists, see Sec. 3.3.1. Since some of the considered examples possess internal dynamics, a suitable DAE solver must be selected carefully and the solution should be monitored closely. For example, this is possible by applying the computed input $\boldsymbol{u}_\mathrm{ffw}$ to the model in a forward simulation and comparing the simulated system output with the desired output. Note that the system must be minimum phase for solving the inverse model as an initial value problem. For non-minimum phase systems, stable system inversion is presented in Chap. 6.
Common DAE solvers are presented in [Brenan89, Hairer02]. Here, the implicit Runge-Kutta schemes and backwards differentiation formulas (BDF) are applied and compared in the context of servo-constraints. Both schemes can be applied to DAEs as well as ODEs. To simplify notation, the ODE/DAE system of dimension n_x is summarized with the unknown state vector $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{n_\mathrm{x}}$ in the form

$$\widetilde{\boldsymbol{f}}(\widetilde{\boldsymbol{x}}, \dot{\widetilde{\boldsymbol{x}}}, t) = \boldsymbol{0}\,, \tag{3.88}$$

with $\widetilde{\boldsymbol{f}} : \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{x}}$. For an ODE system, the Jacobian $\frac{\partial \widetilde{\boldsymbol{f}}}{\partial \dot{\widetilde{\boldsymbol{x}}}}$ is regular, while it is singular for a DAE system.

Implicit Runge-Kutta methods

The implicit Runge-Kutta schemes belong to the class of single-step integration methods. The approximated solution $\widetilde{\boldsymbol{x}}_n$ at time t_n is computed based on the solution $\widetilde{\boldsymbol{x}}_{n-1}$ at the previous time step t_{n-1}. Note that in this work, the step size is constant with $t_n - t_{n-1} = \Delta t$. This is important for implementation on experimental setups which are running with a fixed control loop frequency.

Following [Brenan89], the s_{rk}-stage Runge-Kutta scheme for ODEs/DAEs given by equation (3.88) is

$$\widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}_{n-1}+\Delta t\sum_{j=1}^{s_{\mathrm{rk}}}a'_{ij}\widetilde{\boldsymbol{X}}'_j,\ \widetilde{\boldsymbol{X}}'_i,\ t_{n-1}+c'_i\,\Delta t\right)=\boldsymbol{0}\qquad i=1,2,\ldots,s_{\mathrm{rk}}\quad(3.89)$$

$$\widetilde{\boldsymbol{x}}_n=\widetilde{\boldsymbol{x}}_{n-1}+\Delta t\sum_{i=1}^{s_{\mathrm{rk}}}b'_i\,\widetilde{\boldsymbol{X}}'_i\,.\qquad(3.90)$$

Thereby, the values $\widetilde{\boldsymbol{X}}'_i\in\mathbb{R}^{n_{\mathrm{x}}}$ denote the approximated derivatives at the time instances $t_{n-1}+c'_i\,\Delta t$ between the times t_{n-1} and t_n. The values $\widetilde{\boldsymbol{X}}'_i$ are computed with Newton's method as solutions of the nonlinear equation (3.89), which is of dimension $n_{\mathrm{e}}=s_{\mathrm{rk}}\,n_{\mathrm{x}}$. A specific Runge-Kutta method is defined by the coefficients a'_{ij}, b'_i and c'_i. They are collected in the matrices $\boldsymbol{A}'$, $\boldsymbol{b}'$, $\boldsymbol{c}'$ and summarized in the Butcher-tableau

$$\begin{array}{c|c}\boldsymbol{c}'&\boldsymbol{A}'\\\hline&\boldsymbol{b}'^{\mathrm{T}}\end{array}.$$

The implicit Euler scheme is a Runge-Kutta method of stage $s_{\mathrm{rk}}=1$ with the coefficients given in Tab. 3.1. This scheme is usually applied in the context of servo-constraints. It is here compared to the 3-stage Runge-Kutta method Radau IIA. The 3-stage Runge-Kutta method Radau IIA is of order 5 and is therefore called Radau5 in the following. Its coefficients are given in Tab. 3.1. The convergence rates of the considered schemes are derived in [Hairer02] for ODEs. For general higher index DAEs, convergence results are stated in [HairerRocheLubich89]. Thereby, the convergence rate of the algebraic variables might be lower than the convergence rate of the differential variables. Since the algebraic variable $\boldsymbol{u}$ is used for the control, its convergence rate is of interest here. The convergence rate for algebraic variables for an s_{rk}-stage scheme with $s_{\mathrm{rk}}\geq 2$ is given as $s_{\mathrm{rk}}-1$ in [HairerRocheLubich89].

Table 3.1: Coefficients of the considered Runge-Kutta schemes.

Implicit Euler ($s_{\mathrm{rk}}=1$)	Radau IIA of order 5 ($s_{\mathrm{rk}}=3$)			
	$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$1 \mid 1$	$\frac{4+\sqrt{6}}{10}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
$\hline\ \ 1$	1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
		$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

Backwards Differentiation Formulas

The backwards differentiation formulas belong to the class of multi-step solvers. The approximated solution $\widetilde{\boldsymbol{x}}_n$ at time t_n is computed based on the last k_{bdf} solutions $\widetilde{\boldsymbol{x}}_{n-1}, \widetilde{\boldsymbol{x}}_{n-2}, \ldots, \widetilde{\boldsymbol{x}}_{n-k_{\text{bdf}}}$. Following [Brenan89], the k_{bdf}-step BDF scheme for the implicit ODEs/DAEs (3.88) is

$$\widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}_n,\ \frac{1}{\Delta t}\sum_{i=0}^{k_{\text{bdf}}}\alpha_i'\ \widetilde{\boldsymbol{x}}_{n-i},\ t_n\right) = \boldsymbol{0}\,. \tag{3.91}$$

In order to obtain the next approximation $\widetilde{\boldsymbol{x}}_n$, the nonlinear equation (3.91) of dimension $n_{\text{e}} = n_{\text{x}}$ is solved with Newton's method. The coefficients α_i' define a particular BDF scheme. Table 3.2 summarizes the coefficients for schemes with $k_{\text{bdf}} = 1, 2, \ldots, 6$. Note that the implicit Euler scheme can again be obtained for $k_{\text{bdf}} = 1$. For a k_{bdf}-step scheme, the first k_{bdf} solutions $\widetilde{\boldsymbol{x}}_{k_{\text{bdf}}-1}, \widetilde{\boldsymbol{x}}_{k_{\text{bdf}}-2}, \ldots, \widetilde{\boldsymbol{x}}_0$ must be provided for example by a single-step scheme. Convergence results for BDF methods are given in [Brenan89] for index 3 DAEs. In case the first k_{bdf} values are consistent of order $k_{\text{bdf}}+1$ and the algebraic equations are solved at each step with accuracy $O\left(\Delta t^{k+3}\right)$ for $k_{\text{bdf}} = 1$ or accuracy $O\left(\Delta t^{k_{\text{bdf}}+2}\right)$ for $k_{\text{bdf}} \geq 2$ respectively, then the constant step size k_{bdf}-step BDF method converges with order k_{bdf} for $k_{\text{bdf}} < 7$.

Table 3.2: Coefficients α_i' of k_{bdf}-step BDF schemes.

k_{bdf}	α_0'	α_1'	α_2'	α_3'	α_4'	α_5'	α_6'
1	1	-1					
2	$\frac{3}{2}$	-2	$\frac{1}{2}$				
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$			
4	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$		
5	$\frac{137}{60}$	-5	5	$-\frac{10}{3}$	$\frac{5}{4}$	$-\frac{1}{5}$	
6	$\frac{49}{20}$	-6	$\frac{15}{2}$	$-\frac{20}{3}$	$\frac{15}{4}$	$-\frac{6}{5}$	$\frac{1}{6}$

3.3.4 Considerations for Real-time Applicability

Most control applications run with a fixed control loop frequency $f = \frac{1}{\Delta t}$. For example, the frequency of the experimental setup introduced in Sec. 2.4.1 is 100 Hz. An algorithm with execution time t_{ex} is considered to be real-time capable, if the condition

$$t_{\text{ex}} < \frac{1}{f} = \Delta t = t_{n+1} - t_n \tag{3.92}$$

holds in every iteration n of the control loop [BurgermeisterArnoldEsterl06]. Figure 3.5 visualizes this condition for two consecutive time steps t_n and t_{n+1}. Detailed descriptions of real-time systems can be found in [Laplante04]. Due to the constant control loop frequency, it is not possible to apply automatic step size control methods, which are often used to increase efficiency [BurgermeisterArnoldEsterl06]. Note that requirements on the numerical accuracy are not as strict compared to off-line simulations. Since the controlled system is designed to be stable, small integration errors can be compensated in future time steps [BurgermeisterArnoldEsterl06]. Moreover, there are other effects, such as friction in the actuators that limit the control accuracy.

The complete execution time t_{ex} includes reading sensor data, computing control inputs and sending the control input to the actuators. Thus, computation of the feedforward control should only take a part of the available time Δt. The feedforward control is computed by integrating the DAEs (3.36)–(3.39) numerically. During numerical time integration, most computational effort is due to solving nonlinear equations with Newton's method. Different approaches are considered to speed up the Newton iterations. They are presented in the following.

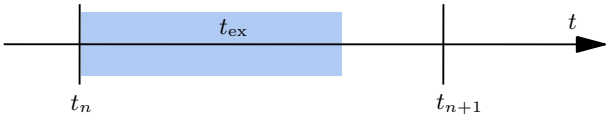


Figure 3.5: Visualization of the real-time condition.

Limitation of Maximum Number of Iterations

The number of Newton iterations is limited to a maximum of $n_{\mathrm{N,max}}$. In case the maximum number of iterations is reached before the abort criterion (3.92) is met, the Newton algorithm (3.84)–(3.85) terminates and uses the current guess as solution. Usually, the Newton method converges in few steps due to its quadratic convergence for sufficiently accurate initial guesses [Hermann11]. The number of iterations can be decreased by providing an accurate first guess. The first guess of one time step is usually the solution of the previous time step. Since the considered trajectories are smooth by definition and discretization is sufficiently small, the solution is expected to change smoothly.

A more restricted version of the approach is setting the maximum allowed number of iterations to $n_{\mathrm{N,max}} = 1$. For the implicit Euler integration scheme, this results in the linear implicit Euler [BurgermeisterArnoldEsterl06]. It is shown in experiments for the cable robot that the linear implicit Euler scheme yields a sufficiently accurate solution for trajectory tracking, see Sec. 4.2.4.

Analytical Jacobian

The Newton algorithm of equations (3.84)–(3.85) relies on the Jacobian $\boldsymbol{J}(\boldsymbol{\xi})$ of the nonlinear equations $\boldsymbol{f}_\mathrm{e}(\boldsymbol{\xi})$ at each current guess. The Jacobian is commonly calculated with finite differences [Hermann11]. This involves $n_\mathrm{e}+1$ function evaluations of the underlying n_e equations for forward and backwards finite differences. This computational effort can be eliminated by providing the analytical Jacobian $\boldsymbol{J}_\mathrm{ana}(\boldsymbol{\xi})$ wherever possible. Analytical Jacobians can be generated e.g. with the SYMBOLIC MATH toolbox or based on analytical derivations before the simulation. Then, only one function evaluation of the nonlinear equation (3.83) and of the analytical Jacobian $\boldsymbol{J}_\mathrm{ana}(\boldsymbol{\xi})$ is necessary. However, the generation of an analytical Jacobian might be difficult or impossible for complex systems.

Broyden's Method

Broyden's method is another approach to circumvent the costly computation of the Jacobian $\boldsymbol{J}(\boldsymbol{\xi})$ with finite differences in each time step. An approximated Jacobian $\widetilde{\boldsymbol{J}}(\boldsymbol{\xi})$ is computed iteratively in each Newton iteration [Broyden65, Hermann11]. The iterative scheme for the approximated Jacobian $\widetilde{\boldsymbol{J}}(\boldsymbol{\xi})$ is

$$\widetilde{\boldsymbol{J}}^{i+1}(\boldsymbol{\xi}^{i+1}) = \widetilde{\boldsymbol{J}}^{i}(\boldsymbol{\xi}^{i}) + \frac{\boldsymbol{f}_\mathrm{e}\left(\boldsymbol{\xi}^{i+1}\right)(\boldsymbol{\chi}^{i})^\mathsf{T}}{\|\boldsymbol{\chi}^{i}\|^2}\,. \tag{3.93}$$

Thereby, $\boldsymbol{\chi}^i$ is the update from equation (3.84) and $\boldsymbol{\xi}^{i+1}$ the updated guess of the Newton scheme. No additional function evaluations are required and thus n_e function evaluations are saved compared to computing the Jacobian with finite differences. Theoretically, Broyden's method (3.93) can be initialized with an identity matrix as first guess $\widetilde{\boldsymbol{J}}^0$. However, faster convergence is obtained by calculating the initial Jacobian $\widetilde{\boldsymbol{J}}^0$ with finite differences in the very first time step. Afterwards, the scheme is initialized with the approximated Jacobian of the last time step. Note that generally speaking, applying Broyden's method results in more Newton iterations, since the Jacobian is not as accurate. However, the overall computation time can be significantly faster, since each individual Newton iteration is considerably faster. The trade-off must be monitored for each individual problem. Note that there exist hybrid methods for which the approximated Jacobian $\widetilde{\boldsymbol{J}}(\boldsymbol{\xi})$ is reinitialized after a fixed number of time steps or based on some other criteria. Simulation results are discussed in Sec. 4.2.5 to demonstrate the trade-off.

Scaling the Algebraic Constraints

The Jacobian matrix $\boldsymbol{J}(\boldsymbol{\xi})$ might be ill-conditioned for higher index DAEs. Ill-conditioning causes rounding errors and less accurate solutions [Hermann11]. The condition can be improved by scaling the algebraic constraints $\boldsymbol{c}_\mathrm{aug}(\boldsymbol{y}, t)$

with a scalar constant ϱ in the order of magnitude of the time step Δt [Brenan89, HairerRocheLubich89], such that the new constraint is

$$\bar{\boldsymbol{c}}_{\mathrm{aug}}(\boldsymbol{y}, t) = \varrho\, \boldsymbol{c}_{\mathrm{aug}}(\boldsymbol{y}, t) = \boldsymbol{0}\,. \tag{3.94}$$

Simulation results demonstrate the effect of scaling in Sec. 4.2.5.

3.4 Combination with Feedback Control

Regarding the two design degree of freedom control structure in Fig. 3.1, the feedforward control input $\boldsymbol{u}_{\mathrm{ffw}}$ is mainly responsible for trajectory tracking. In case the inverse model reflects the system dynamics accurately, the tracking errors are expected to be small. However, a model never captures all effects of the real system. Five factors of uncertainty are identified in [ThrunBurgardFox06].

1. The environment of the system cannot be modeled completely. There remain unknown external forces which can act on the system. These include for example process forces or wind loads.

2. There are parameter uncertainties due to imperfect sensor measurements.

3. System actuation is never perfect and there remain unmodeled effects in the motors, such as dead zones or friction.

4. The system model itself is not perfect and there remains unmodeled dynamics, such as friction effects.

5. Computation time is limited in a real-time system, see Sec. 3.3.4. Therefore, there are numerical uncertainties and rounding errors in the calculation of the control law.

For all these reasons, a feedback controller must be designed to reject those disturbances and to minimize the tracking error. Note that the formulation of the inverse model is closely related to the control strategy called feedback linearization [Isidori96]. However, feedback linearization is known to have robustness issues, see [SlotineLi91], and is not further pursued in the context of this work. There are several advantages of using the two degree of freedom control structure in Fig. 3.1. In that case, the inverse model can be designed independently from the feedback controller. Both control components can be designed, tuned and tested separately. The existence of a feedforward control module poses some advantages compared to using a feedback controller by itself. For example, since most of the motion is generated by the feedforward control input $\boldsymbol{u}_{\mathrm{ffw}}$, the absolute values of the feedback control input $\boldsymbol{u}_{\mathrm{fb}}$ are considerably smaller than the feedforward control input $\boldsymbol{u}_{\mathrm{ffw}}$. Therefore, sensor noise is not amplified as much compared to only using a feedback controller. This is briefly demonstrated in the following example.

Example 3.6 The mass-on-car system introduced in Sec. 2.4.3 is considered in reduced form without the first mass. The adaptive funnel controller is applied in the feedback path, see [BergerEtAl18, BergerHoàngReis18] and the inverse model is computed with servo-constraints. In order to show the effect of the feedback controller, the model is simulated with disturbed parameters that differ from the nominal parameters used for controller design. Figure 3.6 shows the control input for different setups. First, the feedback control law $\boldsymbol{u}_{\mathrm{fb}}$ is applied without additional feedforward control. Then, the combination $\boldsymbol{u} = \boldsymbol{u}_{\mathrm{ffw}} + \boldsymbol{u}_{\mathrm{fb}}$ of feedforward and feedback control is applied. As reference, the feedforward control law $\boldsymbol{u}_{\mathrm{ffw}}$ is shown. It is evident that the combination of feedforward and feedback part reduces spikes in the feedback control input. This reduces loads on the mechanical parts and oscillations of the structure. Refer to [BergerEtAl18] for more details on this example.

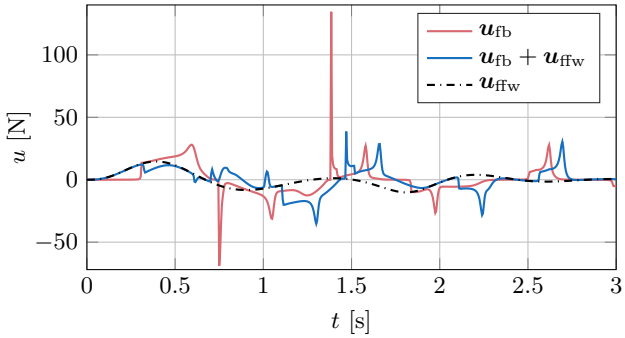
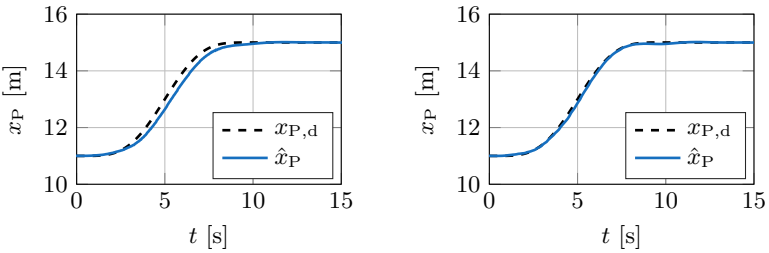


Figure 3.6: System input for trajectory tracking of the mass-on-car system with disturbed parameters [BergerEtAl18].

Another advantage of the two design degree of freedom control structure is that tracking of a trajectory is possible without a time lag. If the complete tracking motion is generated by the feedback controller, a time lag always exists. The feedback controller can only react to the reference motion, since it does not know about it beforehand. This is demonstrated in the following example.

Example 3.7 For the cable robot introduced in Sec. 2.4.1, trajectory control is performed with a combination of feedforward and feedback control and compared to pure feedback control. Figure 3.7 shows experimental results for the experimental setup described in Sec. 4.2.2. For the pure feedback control $\boldsymbol{u}_{\mathrm{fb}}$ in Fig. 3.7(a), a time lag between the desired platform trajectory $x_{\mathrm{P,d}}$ and the measured position $\hat{x}_{\mathrm{P}}$ is evident. The time lag is not present for the combined control $\boldsymbol{u}_{\mathrm{ffw}} + \boldsymbol{u}_{\mathrm{fb}}$ in Fig. 3.7(b) because of the feedforward control part.



(a) Control $\boldsymbol{u}_\mathrm{fb}$.

(b) Control $\boldsymbol{u}_\mathrm{ffw} + \boldsymbol{u}_\mathrm{fb}$.

Figure 3.7: Experimental results for trajectory tracking of the cable robot.

Feedforward control based on servo-constraints can be combined with many different feedback strategies. In the following, simple feedback controllers which are applied in the context of this work are briefly presented.

3.4.1 Linear Quadratic Regulator

A classical model-based state feedback controller is the linear quadratic regulator (LQR) [Brogan91, Bryson75, Föllinger88]. The feedback gains are calculated by solving an optimal control problem subject to either the linear time-invariant (LTI) or time-variant equations of motion. Here, the LTI case is considered for ODEs in the form (2.19) with $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{y}^\mathsf{T} & \boldsymbol{v}^\mathsf{T}\end{bmatrix}^\mathsf{T}$. The control goal is to stabilize the system around the equilibrium $\boldsymbol{x}_\mathrm{eq} = \begin{bmatrix}\boldsymbol{y}_\mathrm{eq}^\mathsf{T} & \boldsymbol{v}_\mathrm{eq}^\mathsf{T}\end{bmatrix}^\mathsf{T}$. For an LTI system, the control is optimal with respect to the cost function

$$J(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{2} \int_{t_0}^{\infty} (\boldsymbol{x}(t) - \boldsymbol{x}_\mathrm{eq})^\mathsf{T} \boldsymbol{Q}_\mathrm{lqr} (\boldsymbol{x}(t) - \boldsymbol{x}_\mathrm{eq}) + \boldsymbol{u}(t)^\mathsf{T} \boldsymbol{R}_\mathrm{lqr} \boldsymbol{u}(t) \, \mathrm{d}t \tag{3.95}$$

with $J : \mathbb{R}^{2n_\mathrm{f}} \times \mathbb{R}^{n_\mathrm{u}} \to \mathbb{R}$. Thereby, $\boldsymbol{Q}_\mathrm{lqr} \in \mathbb{R}^{2n_\mathrm{f} \times 2n_\mathrm{f}}$ and $\boldsymbol{R}_\mathrm{lqr} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{u}}$ are weighting matrices used for tuning [Brogan91]. The weighting matrix $\boldsymbol{Q}_\mathrm{lqr}$ penalizes errors in the state $\boldsymbol{x}$ and the weighting matrix $\boldsymbol{R}_\mathrm{lqr}$ penalizes the system input. Assuming a linear ansatz function for the system input $\boldsymbol{u}$ and setting up the optimization of the cost function $J(\boldsymbol{x}, \boldsymbol{u})$ subject to the LTI dynamics (2.19) results in the algebraic Riccati equation

$$\boldsymbol{P}_\mathrm{lqr} \boldsymbol{A}_\mathrm{lin} + \boldsymbol{A}_\mathrm{lin}^\mathsf{T} \boldsymbol{P}_\mathrm{lqr} - \boldsymbol{P}_\mathrm{lqr} \boldsymbol{B}_\mathrm{lin} \boldsymbol{R}_\mathrm{lqr}^{-1} \boldsymbol{B}_\mathrm{lin}^\mathsf{T} \boldsymbol{P}_\mathrm{lqr} + \boldsymbol{Q}_\mathrm{lqr} = \boldsymbol{0} \tag{3.96}$$

with unknown matrix $\boldsymbol{P}_\mathrm{lqr} \in \mathbb{R}^{2n_\mathrm{f} \times 2n_\mathrm{f}}$, see e.g. [Föllinger88]. Solving the algebraic Riccati equation (3.96) for the matrix $\boldsymbol{P}_\mathrm{lqr}$ yields the constant gain matrix $\boldsymbol{F}_\mathrm{lqr} \in \mathbb{R}^{n_\mathrm{u} \times 2n_\mathrm{f}}$, which is computed with

$$\boldsymbol{F}_\mathrm{lqr} = \boldsymbol{R}_\mathrm{lqr}^{-1} \boldsymbol{B}_\mathrm{lin}^\mathsf{T} \boldsymbol{P}_\mathrm{lqr} \, . \tag{3.97}$$

The gain matrix can be implemented in the linear feedback law

$$\boldsymbol{u}_{\mathrm{lqr}}(t) = -\boldsymbol{F}_{\mathrm{lqr}} \left(\boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{eq}}\right) . \tag{3.98}$$

Note that the control strategy only guarantees stability for the linearized system. For a nonlinear system, stability cannot be guaranteed and the controller might only work well in a region in which the linearized system approximates the nonlinear system accurately. The presented controller is implemented on the experimental setup of the cable robot and results are presented in Sec. 4.2.7 and in [OttoSeifried18c].

Note that the control strategy described above can be extended for models described by DAEs. This is for example shown in [Heiland16] for flow control problems. Here, the general nonlinear DAEs are first linearized. Then, a projection approach based on a QR decomposition is applied to transform the DAEs into a set of equivalent ODEs, see e.g. [Bauchau11]. Note that by taking this projection approach, the dynamic equations are still formulated in the original redundant coordinates. Therefore, the tuning matrices $\boldsymbol{Q}_{\mathrm{lqr}}$ and $\boldsymbol{R}_{\mathrm{lqr}}$ act on the original redundant coordinates in the cost function. This makes tuning the weighting matrices more intuitive compared to introducing virtual states. Details of the proposed strategy are presented in [OttoRückwaldSeifried19] and experimental results are shown in Sec. 4.2.8.

3.4.2 Sliding Mode Control

Sliding mode control is a model-based state feedback controller developed in the field of nonlinear control theory [Isidori96, SlotineLi91]. It is designed to achieve accurate tracking in the presence of model uncertainties. Thereby, the tracking problem of a system is transformed to a stabilization problem of smaller dimension. Perfect tracking comes a the price of high controller activity, namely fast switching of the control input [SlotineLi91].

Design of a sliding mode controller (SMC) for underactuated multibody systems is presented according to [AshrafiuonErwin08] for systems in the ODE form (2.1)–(2.2). The equations of motion are partitioned into actuated and unactuated parts

$$\begin{bmatrix} \dot{\boldsymbol{y}}_{\mathrm{a}} \\ \dot{\boldsymbol{y}}_{\mathrm{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z}_{\mathrm{aa}} & \boldsymbol{Z}_{\mathrm{au}} \\ \boldsymbol{Z}_{\mathrm{ua}} & \boldsymbol{Z}_{\mathrm{uu}} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{\mathrm{a}} \\ \boldsymbol{v}_{\mathrm{u}} \end{bmatrix} \tag{3.99}$$

$$\begin{bmatrix} \boldsymbol{M}_{\mathrm{aa}} & \boldsymbol{M}_{\mathrm{au}} \\ \boldsymbol{M}_{\mathrm{ua}} & \boldsymbol{M}_{\mathrm{uu}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}_{\mathrm{a}} \\ \dot{\boldsymbol{v}}_{\mathrm{u}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{k}_{\mathrm{a}} \\ \boldsymbol{k}_{\mathrm{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{\mathrm{a}} \\ \boldsymbol{q}_{\mathrm{u}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_{\mathrm{a}} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u} \tag{3.100}$$

with the subvectors $\boldsymbol{y}_{\mathrm{a}}, \boldsymbol{v}_{\mathrm{a}}, \boldsymbol{k}_{\mathrm{a}}, \boldsymbol{q}_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}}}$, $\boldsymbol{y}_{\mathrm{u}}, \boldsymbol{v}_{\mathrm{u}}, \boldsymbol{k}_{\mathrm{u}}, \boldsymbol{q}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{f}}-n_{\mathrm{u}}}$ and submatrices $\boldsymbol{M}_{\mathrm{aa}}, \boldsymbol{Z}_{\mathrm{aa}}, \boldsymbol{B}_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\mathrm{u}}}$, $\boldsymbol{M}_{\mathrm{au}}, \boldsymbol{Z}_{\mathrm{au}} \in \mathbb{R}^{n_{\mathrm{u}} \times (n_{\mathrm{f}}-n_{\mathrm{u}})}$, $\boldsymbol{M}_{\mathrm{ua}}, \boldsymbol{Z}_{\mathrm{ua}} \in \mathbb{R}^{(n_{\mathrm{f}}-n_{\mathrm{u}}) \times n_{\mathrm{u}}}$,

$\boldsymbol{M}_{\mathrm{uu}}, \boldsymbol{Z}_{\mathrm{uu}} \in \mathbb{R}^{(n_{\mathrm{f}}-n_{\mathrm{u}})\times(n_{\mathrm{f}}-n_{\mathrm{u}})}$. The tracking errors are defined as

$$\widetilde{\boldsymbol{y}}(\boldsymbol{y},t) = \begin{bmatrix} \widetilde{\boldsymbol{y}}_{\mathrm{a}}(\boldsymbol{y}_{\mathrm{a}},t) \\ \widetilde{\boldsymbol{y}}_{\mathrm{u}}(\boldsymbol{y}_{\mathrm{u}},t) \end{bmatrix} = \boldsymbol{y} - \boldsymbol{y}_{\mathrm{d}}(t)\,, \tag{3.101}$$

$$\dot{\widetilde{\boldsymbol{y}}}(\dot{\boldsymbol{y}},t) = \begin{bmatrix} \dot{\widetilde{\boldsymbol{y}}}_{\mathrm{a}}(\dot{\boldsymbol{y}}_{\mathrm{a}},t) \\ \dot{\widetilde{\boldsymbol{y}}}_{\mathrm{u}}(\dot{\boldsymbol{y}}_{\mathrm{u}},t) \end{bmatrix} = \dot{\boldsymbol{y}} - \dot{\boldsymbol{y}}_{\mathrm{d}}(t) \tag{3.102}$$

with the tracking errors $\widetilde{\boldsymbol{y}}_{\mathrm{a}}, \dot{\widetilde{\boldsymbol{y}}}_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}}}$ of the actuated coordinates and the tracking errors $\widetilde{\boldsymbol{y}}_{\mathrm{u}}, \dot{\widetilde{\boldsymbol{y}}}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{f}}-n_{\mathrm{u}}}$ of the unactuated coordinates. The tracking problem of the coordinates $\boldsymbol{y}, \dot{\boldsymbol{y}}$ is transformed to a stabilization problem of a n_{u}-dimensional sliding surface

$$\begin{aligned} \boldsymbol{\sigma}_{\mathrm{smc}}(\widetilde{\boldsymbol{y}}(\boldsymbol{y},t), \dot{\widetilde{\boldsymbol{y}}}(\boldsymbol{v},t)) = \boldsymbol{\Theta}_{\mathrm{a}}\, \dot{\widetilde{\boldsymbol{y}}}_{\mathrm{a}}(\dot{\boldsymbol{y}}_{\mathrm{a}},t) + \boldsymbol{\Lambda}_{\mathrm{a}}\, \widetilde{\boldsymbol{y}}_{\mathrm{a}}(\boldsymbol{y}_{\mathrm{a}},t) + \\ \boldsymbol{\Theta}_{\mathrm{u}}\, \dot{\widetilde{\boldsymbol{y}}}_{\mathrm{u}}(\dot{\boldsymbol{y}}_{\mathrm{u}},t) + \boldsymbol{\Lambda}_{\mathrm{u}}\, \widetilde{\boldsymbol{y}}_{\mathrm{u}}(\boldsymbol{y}_{\mathrm{u}},t) \end{aligned} \tag{3.103}$$

with $\boldsymbol{\sigma}_{\mathrm{smc}} : \mathbb{R}^{n_{\mathrm{f}}} \times \mathbb{R}^{n_{\mathrm{f}}} \to \mathbb{R}^{n_{\mathrm{u}}}$. Thereby, the matrices $\boldsymbol{\Theta}_{\mathrm{a}}\,, \boldsymbol{\Lambda}_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{u}}}$ and $\boldsymbol{\Theta}_{\mathrm{u}}\,, \boldsymbol{\Lambda}_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{u}}\times(n_{\mathrm{f}}-n_{\mathrm{u}})}$ are chosen to yield stable dynamics on the sliding surface. The parameters of the matrices are for example selected based on a linearization of the system in [AshrafiuonErwin08] or with a model-predictive control scheme in [MuskeEtAl12]. The sliding surface is rewritten in terms of the velocities $\dot{\boldsymbol{y}}_{\mathrm{a}}, \dot{\boldsymbol{y}}_{\mathrm{u}}$ such that

$$\begin{aligned} \boldsymbol{\sigma}_{\mathrm{smc}}(\widetilde{\boldsymbol{y}}(\boldsymbol{y},t), \dot{\widetilde{\boldsymbol{y}}}(\boldsymbol{v},t)) = \boldsymbol{\Theta}_{\mathrm{a}}\, \dot{\boldsymbol{y}}_{\mathrm{a}} + \boldsymbol{\Theta}_{\mathrm{u}}\, \dot{\boldsymbol{y}}_{\mathrm{u}} + \\ \underbrace{\left(-\boldsymbol{\Theta}_{\mathrm{a}}\, \dot{\boldsymbol{y}}_{\mathrm{d,a}}(t) - \boldsymbol{\Theta}_{\mathrm{u}}\, \dot{\boldsymbol{y}}_{\mathrm{d,u}}(t) + \boldsymbol{\Lambda}_{\mathrm{a}}\, \widetilde{\boldsymbol{y}}_{\mathrm{a}}(\boldsymbol{y}_{\mathrm{a}},t) + \boldsymbol{\Lambda}_{\mathrm{u}}\, \widetilde{\boldsymbol{y}}_{\mathrm{u}}(\boldsymbol{y}_{\mathrm{u}},t)\right)}_{\boldsymbol{s}_{\mathrm{p}}(\widetilde{\boldsymbol{y}}(\boldsymbol{y},t),t)} \end{aligned} \tag{3.104}$$

with the function $\boldsymbol{s}_{\mathrm{p}} : \mathbb{R}^{n_{\mathrm{f}}} \times \mathbb{R} \to \mathbb{R}^{n_{\mathrm{u}}}$. The stabilization of the sliding surface $\boldsymbol{\sigma}_{\mathrm{smc}}(\widetilde{\boldsymbol{y}}(\boldsymbol{y},t), \dot{\widetilde{\boldsymbol{y}}}(\boldsymbol{v},t))$ and therefore tracking of the original coordinates is obtained by enforcing $\dot{\boldsymbol{\sigma}}_{\mathrm{smc}}(\widetilde{\boldsymbol{y}}(\boldsymbol{y},t), \dot{\widetilde{\boldsymbol{y}}}(\boldsymbol{v},t)) = \boldsymbol{0}$. This condition results in the control law

$$\boldsymbol{u}_{\mathrm{smc}} = -\boldsymbol{M}_{\mathrm{s}}^{-1} \left(\hat{\boldsymbol{f}}_{\mathrm{s}} + \dot{\boldsymbol{s}}_{\mathrm{p}} + \boldsymbol{\kappa}_{\mathrm{smc}}\, \mathrm{sat}\left(\boldsymbol{s}\, \boldsymbol{\Phi}_{\mathrm{smc}}^{-1}\right) \right) \tag{3.105}$$

with the tuning parameters $\boldsymbol{\kappa}_{\mathrm{smc}} \in \mathbb{R}^{n_{\mathrm{u}}}$ and $\boldsymbol{\Phi}_{\mathrm{smc}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{u}}}$ and $\hat{\boldsymbol{f}}_{\mathrm{s}}$ denoting the best estimate for $\boldsymbol{f}_{\mathrm{s}}$. The terms $\boldsymbol{M}_{\mathrm{s}} \in \mathbb{R}^{n_{\mathrm{u}}\times n_{\mathrm{u}}}$ and $\boldsymbol{f}_{\mathrm{s}} \in \mathbb{R}^{n_{\mathrm{u}}}$ are defined as

$$\begin{aligned} \boldsymbol{M}_{\mathrm{s}} = (\boldsymbol{\Theta}_{\mathrm{a}}\boldsymbol{Z}_{\mathrm{aa}} + \boldsymbol{\Theta}_{\mathrm{u}}\boldsymbol{Z}_{\mathrm{ua}})\; \boldsymbol{M}_{\mathrm{aa}}'^{-1}\boldsymbol{B}_{\mathrm{a}} - \\ (\boldsymbol{\Theta}_{\mathrm{a}}\boldsymbol{Z}_{\mathrm{au}} + \boldsymbol{\Theta}_{\mathrm{u}}\boldsymbol{Z}_{\mathrm{uu}})\; \boldsymbol{M}_{\mathrm{uu}}'^{-1}\, \boldsymbol{M}_{\mathrm{ua}}\, \boldsymbol{M}_{\mathrm{aa}}'^{-1}\boldsymbol{B}_{\mathrm{a}}\,, \end{aligned} \tag{3.106}$$

$$\begin{aligned} \boldsymbol{f}_{\mathrm{s}} = (\boldsymbol{\Theta}_{\mathrm{a}}\boldsymbol{Z}_{\mathrm{aa}} + \boldsymbol{\Theta}_{\mathrm{u}}\boldsymbol{Z}_{\mathrm{ua}})\; \boldsymbol{M}_{\mathrm{aa}}'^{-1}\, \boldsymbol{f}_{\mathrm{a}}' + (\boldsymbol{\Theta}_{\mathrm{a}}\boldsymbol{Z}_{\mathrm{au}} + \boldsymbol{\Theta}_{\mathrm{u}}\boldsymbol{Z}_{\mathrm{uu}})\; \boldsymbol{M}_{\mathrm{uu}}'^{-1}\, \boldsymbol{f}_{\mathrm{u}}' + \\ \left(\boldsymbol{\Theta}_{\mathrm{a}}\dot{\boldsymbol{Z}}_{\mathrm{aa}} + \boldsymbol{\Theta}_{\mathrm{u}}\dot{\boldsymbol{Z}}_{\mathrm{ua}}\right)\boldsymbol{v}_{\mathrm{a}} + \left(\boldsymbol{\Theta}_{\mathrm{a}}\dot{\boldsymbol{Z}}_{\mathrm{au}} + \boldsymbol{\Theta}_{\mathrm{u}}\dot{\boldsymbol{Z}}_{\mathrm{uu}}\right)\boldsymbol{v}_{\mathrm{u}}\,. \end{aligned} \tag{3.107}$$

Thereby, the auxiliary matrices are

$$\boldsymbol{M}'_{\mathrm{aa}} = \boldsymbol{M}_{\mathrm{aa}} - \boldsymbol{M}_{\mathrm{au}}\,\boldsymbol{M}_{\mathrm{uu}}^{-1}\,\boldsymbol{M}_{\mathrm{ua}}\,, \tag{3.108}$$

$$\boldsymbol{f}'_{\mathrm{a}} = (\boldsymbol{q}_{\mathrm{a}} - \boldsymbol{k}_{\mathrm{a}}) - \boldsymbol{M}_{\mathrm{au}}\,\boldsymbol{M}_{\mathrm{uu}}^{-1}\,(\boldsymbol{q}_{\mathrm{u}} - \boldsymbol{k}_{\mathrm{u}})\,, \tag{3.109}$$

$$\boldsymbol{M}'_{\mathrm{uu}} = \boldsymbol{M}_{\mathrm{uu}} - \boldsymbol{M}_{\mathrm{ua}}\,\boldsymbol{M}_{\mathrm{aa}}^{-1}\,\boldsymbol{M}_{\mathrm{au}}\,, \tag{3.110}$$

$$\boldsymbol{f}'_{\mathrm{u}} = (\boldsymbol{q}_{\mathrm{u}} - \boldsymbol{k}_{\mathrm{u}}) - \boldsymbol{M}_{\mathrm{ua}}\,\boldsymbol{M}_{\mathrm{aa}}^{-1}\,(\boldsymbol{q}_{\mathrm{a}} - \boldsymbol{k}_{\mathrm{a}}) \tag{3.111}$$

with $\boldsymbol{M}'_{\mathrm{aa}} \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\mathrm{u}}}$, $\boldsymbol{f}'_{\mathrm{a}} \in \mathbb{R}^{n_{\mathrm{u}}}$, $\boldsymbol{M}'_{\mathrm{uu}} \in \mathbb{R}^{(n_{\mathrm{f}}-n_{\mathrm{u}}) \times (n_{\mathrm{f}}-n_{\mathrm{u}})}$, $\boldsymbol{f}'_{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{f}}-n_{\mathrm{u}}}$. Two parameters are available to tune the performance of the controller. The controller gain $\boldsymbol{\kappa}_{\mathrm{smc}}$ is

$$\boldsymbol{\kappa}_{\mathrm{smc}} = \boldsymbol{F}_{\mathrm{s}} + \boldsymbol{\eta}_{\mathrm{smc}} \tag{3.112}$$

with the upper bound $\boldsymbol{F}_{\mathrm{s}} \geq |\boldsymbol{f}_{\mathrm{s}} - \hat{\boldsymbol{f}}_{\mathrm{s}}|$ on the estimation of the dynamic forces $\boldsymbol{f}_{\mathrm{s}}$. This term ensures robustness against unmodeled dynamics and parameter uncertainties. The gain $\boldsymbol{\eta}_{\mathrm{smc}} \in \mathbb{R}^{n_{\mathrm{u}}}$ is tuned such that the sliding surface is reached in finite time [AshrafiuonErwin08, SlotineLi91]. The width $\boldsymbol{\Phi}_{\mathrm{smc}}$ of the saturation function determines the trade-off between high control action and minimizing the tracking error. For very small width $\boldsymbol{\Phi}_{\mathrm{smc}}$, the saturation function reduces to a sign-function and there is high frequent switching of the control input $\boldsymbol{u}_{\mathrm{smc}}$ for exact tracking. For large width $\boldsymbol{\Phi}_{\mathrm{smc}}$, the control action is smooth, but the tracking error increases. It is referred to [AshrafiuonErwin08] for more details on the control strategy. From the control law (3.105) follows, that the desired state trajectories $\boldsymbol{y}_{\mathrm{d}}(t), \dot{\boldsymbol{y}}_{\mathrm{d}}(t)$ must be available for evaluation of the control law. Note that these trajectories are usually not known for a defined output trajectory $\boldsymbol{z}_{\mathrm{d}}(t)$. Here, they are obtained by solving the inverse model DAEs (3.36)–(3.39).
Application of the sliding mode controller on the cable robot is shown in Sec. 4.2.7 and in [OttoSeifried18c]. Moreover, classical sliding mode control is compared to an adaptive sliding mode control strategy in [BessaEtAl19] with supporting experimental results for the cable robot.

3.5 Classification of the Application Examples

The application examples presented in Sec. 2.4 are chosen due to their specific properties in the context of servo-constraints. The systems are divided into three categories. Differentially flat systems do not have internal dynamics and their inverse models are algebraic systems. Systems with internal dynamics are subdivided into minimum phase and non-minimum phase systems. In order to give an overview of the considered application examples, their properties are briefly summarized in the following. The derivations of the properties and detailed analyses are given in the subsequent chapters. Differentially flat systems are analyzed in Chap. 4, minimum phase systems are considered in Chap. 5 and non-minimum phase systems are considered in Chap. 6.

The undamped torsional oscillator and the mass-on-car system in configuration $\alpha = 0$ are examples for differentially flat SISO systems, while the cable robot is an example for differentially flat MIMO systems. The damped torsional oscillator and the mass-on-car system in configuration $\alpha > 0$ are examples for minimum phase SISO systems. The examples modeling elastic bodies are non-minimum phase systems. Thus, the manipulator with one passive joint and the flexible manipulator with one link are examples for non-minimum phase SISO systems. The three-dimensional manipulator with one passive joint and the flexible two-link manipulator are examples for non-minimum phase MIMO systems. It might be possible to transform non-minimum phase systems into minimum phase systems by performing parameter optimization or output redefinition, see Sec. 5.2 and Sec. 6.5 respectively.

Models of all system categories can have high relative degree r and therefore the inverse model DAEs can have high differentiation index. Of the considered models, the torsional oscillator with n_d dampers has the highest relative degree which increases for N disks with $r = 2N - n_\mathrm{d}$. Moreover, the mass-on-car system can have a relative degree as high as $r = 6$ depending on the system configuration α. The cable robot and the flexible manipulator with one link have relative degree $r = 4$ in at least one input-output channel. This shows that even simple mechanical systems result in inverse model DAEs of higher differentiation index. The remaining systems, such as the two- and three-dimensional manipulators with one passive joint have relative degree $r = 2$ and their inverse models have differentiation index 3.

An overview of the chosen application examples and their properties regarding system classification and relative degree is shown in Tab. 3.3. The application examples for each system class are analyzed in one of the three following chapters.

Table 3.3: Properties of the application examples.

Example	Model	Application	System category	(Vector) relative degree
Cable robot	Sec. 2.4.1	Ex. 3.7, Sec. 4.2	differentially flat	$\boldsymbol{r} = \{4, 2, 2\}$ for the complex model and $\boldsymbol{r} = \{4, 2\}$ for the reduced model
Torsional oscillator	Sec. 2.4.2	Ex. 3.1, Ex. 3.4, Ex. 3.5, Sec. 4.1	differentially flat for the undamped case, otherwise minimum phase	relative degree changes for N disks and n_d dampers with $r = 2N - n_\mathrm{d}$
Mass-on-car	Sec. 2.4.3	Ex. 3.3, Ex. 3.6, Sec. 5.1	differentially flat for $\alpha = 0\,^\circ$, otherwise minimum phase	relative degree depends on system configuration and varies from $r = 3$ to $r = 6$
2D manipulator with one passive joint	Sec. 2.4.4	Sec. 6.3	non-minimum phase	$r = 2$
3D manipulator with one passive joint	Sec. 2.4.4	Sec. 5.2	non-minimum phase	$\boldsymbol{r} = \{2, 2, 2\}$
Flexible one-link manipulator	Sec. 2.4.5	Sec. 6.4	non-minimum phase	relative degree depends on the system configuration and varies from $r = 1$ to $r = 4$
Flexible two-link manipulator	Sec. 2.4.5	Sec. 6.5.3	non-minimum phase	relative degree undefined due to singular decoupling matrix

Chapter 4

Differentially Flat Systems

Differentially flat systems are introduced in Sec. 3.1.2 and have favorable properties for model inversion. By definition, the inverse model can be computed analytically for differentially flat systems. Therefore, this property is often used in controller design. In order to name few examples, model inversion based on differential flatness is performed for cable robots in [HeydenWoernle06], for three-dimensional overhead cranes in [KnierimKriegerSawodny10], for vehicles in [FuchshumerSchlacherRittenschober05, RouchonEtAl93], for a complex drive train model in [Pham19] and for quadrotors in [FaesslerFranchiScaramuzza18]. A catalog of differentially flat systems is given in [MurrayRathinamSluis95].

For complex systems, it is often not straightforward to derive the analytical input-output relationship. Also, there does not exist a simple systematic approach to prove differential flatness [Knoll16]. In contrast to the analytical derivation of the inverse model, the concept of servo-constraints is applicable for systems with unknown differential flatness property, see Sec. 3.2. Nonetheless, differentially flat systems are popular application examples in the context of servo-constraints. Most literature on servo-constraints is published for differentially flat systems, for example overhead cranes [BlajerKołodziejczyk04, FumagalliEtAl10, LeitzLeyendecker14, OttoSeifried18c], three-dimensional rotary cranes [BetschAltmannYang16, BetschQuasemUhlar09, YangBetschZhang20] as well as undamped torsional or translational oscillators [AltmannHeiland17, FumagalliEtAl10, LiuEtAl15, OttoSeifried18b]. Recently, the approach is extended to systems with internal dynamics, which are covered in Chap. 5.

In the context of servo-constraints, differentially flat systems are popular examples due to the existence of an analytical solution for the inverse model, which can pose as a reference to the numerical computation [OttoSeifried18c]. Moreover, inverse models of differentially flat systems do not contain internal dynamics, which simplifies the numerical solution of the inverse model DAEs. Numerical damping of the selected solvers only plays a minor role in contrast to the model inversion of systems with internal dynamics [OttoSeifried18a, Seifried14]. Thus, the implicit Euler scheme is often chosen to solve the inverse model DAEs of differentially flat systems [BetschQuasemUhlar09, BlajerKołodziejczyk04]. On the other hand, inverse model DAEs of differentially flat systems are often of higher differentiation index. This complicates the numerical solution and index reduction strategies introduced in Sec. 3.2.5 must be applied. First experimental results for servo-constraints are published in [BlajerEtAl09] for a mass-spring

system and in [Bestle05] for a small scale crane system. However, the approach is not implemented to run in real-time. A thorough analysis of the real-time applicability and combination with feedback laws is shown in [OttoSeifried18c]. In the following, both the undamped torsional oscillator as well as the cable robot are analyzed for model inversion based on servo-constraints. Results for the undamped torsional oscillator demonstrate and compare the application of various index reduction methods. Experimental results for the cable robot analyze the application of servo-constraints in a real-time environment as well as its combination with feedback controllers.

4.1 Undamped Torsional Oscillator

The torsional oscillator introduced in Sec. 2.4.2 is a popular example in the context of model inversion [AltmannHeiland17, FumagalliEtAl10, SeifriedBlajer13]. The system properties are helpful for analyzing the servo-constraints approach. The system is linear for linear spring-damper models and therefore enables a linear reference analysis. For neglectable damping, the model is differentially flat and enables an analytical reference analysis. Moreover, its relative degree increases with the number of considered disks. Thus, the limits of the numerical schemes for increasing relative degree and differentiation index can be analyzed. In the following, the system properties are analyzed. Afterwards, different index reduction methods are compared in simulations for increasing relative degree. The simulation results are presented according to [OttoSeifried18b].

4.1.1 System Properties

For the torsional oscillator introduced in Sec. 2.4.2, the system output is chosen as the rotation φ_N of the last disk N. The respective servo-constraint is

$$s(\boldsymbol{y}, t) = \varphi_\mathrm{N} - z_\mathrm{d}(t) = 0\,. \tag{4.1}$$

The equations of motion given by the matrices (2.79)–(2.81) together with equation (4.1) form the inverse model. Since the equations of motion are linear, the relative degree can be derived with various methods, such as analysis of the transfer function $G(s)$, see Sec. 3.2.3, analysis of the generalized eigenvalue problem, see Sec. 3.2.3 or a fully analytical analysis by differentiating the system output, see Sec. 3.1.3.

Example 4.1 The analysis of the relative degree is demonstrated for $N = 2$ disks with the equations of motion following from (2.79)–(2.81) as

$$\begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix} + \begin{bmatrix} d_1 & -d_1 \\ -d_1 & d_1 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} + \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{4.2}$$

and system output $z = \varphi_2$. In frequency domain, the transfer function is

$$G(s) = \frac{k_1 + d_1 s}{s^2 \left(I_1 \, I_2 \, s^2 + (I_1 + I_2) \, d_1 \, s + (I_1 + I_2) \, k_1 \right)} \, . \tag{4.3}$$

The transfer function has one transmission zero at $-\frac{k_1}{d_1}$ and 4 poles. This yields a relative degree $r = 4 - 1 = 3$. For neglectable damping with $d_1 = 0$, the transfer function reduces to

$$G(s) = \frac{k_1}{I_1 \, I_2 \, s^4 + (I_1 + I_2) \, k_1 s^2} \, . \tag{4.4}$$

There is no transmission zero and the relative degree is $r = 4$, which corresponds to the dimension of the state space. Thus, the undamped system is differentially flat and the inverse model is given by $G(s)^{-1}$. The analytical solution of the inverse model in time domain is derived in Example 3.1.

In general, for the undamped oscillator with $d_i = 0$ for $i = 1\,, 2\,, N - 1$, the analysis yields a relative degree of $r = 2N$ for N disks. Therefore, there is no internal dynamics. For each additional damper with $d_i \neq 0$, the numerator polynomial of the transfer function includes a term $(k_i + d_i s)$. Thus, the oscillator with n_d dampers for $n_\mathrm{d} = 1, 2, \ldots, N - 1$, has relative degree $r = 2N - n_\mathrm{d}$. The fully damped system is therefore of relative degree $r = N + 1$. The internal dynamics of the system consists of n_d first order differential equations. The differentiation index of the DAEs describing the inverse model changes according to the relative degree r and is $2N - n_\mathrm{d} + 1$. Thus, requirements on numerical schemes and the smoothness of the desired trajectory increase with increasing number of disks.

4.1.2 Comparison of Problem Formulations

In the following, the inverse model DAEs are solved for the undamped torsional oscillator. The objective of this section is the comparison of index reduction strategies and to determine a maximum possible number of disks to be considered using this approach.

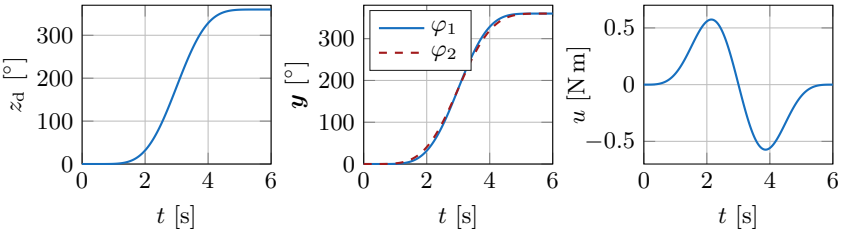
The analytical solution of the inverse model is given in equation (3.11) for $N = 2$ disks and respective analytical solutions for $N > 2$ disks can be derived in a similar manner. The analytical solution is used as reference for the inverse model computed with servo-constraints. The simulation parameters are shown in Tab. 4.1. All parameters $I_i = I$, $k_i = k$ and $d_i = d$ with $i = 1, 2, \ldots, N - 1$ are assumed to be equal. The desired trajectory $z_\mathrm{d}(t)$ is defined by the smooth polynomial

$$\begin{aligned} z_\mathrm{d}(t) = z_0 + (z_\mathrm{f} - z_0) \big(& -3432 \, \tau^{15} + 25740 \, \tau^{14} - 83160 \, \tau^{13} \\ & + 150150 \, \tau^{12} - 163800 \, \tau^{11} + 108108 \, \tau^{10} - 40040 \, \tau^9 + 6435 \, \tau^8 \big) \end{aligned} \tag{4.5}$$

with $\tau = \frac{t}{t_\mathrm{f}}$. Here, the initial point is chosen as $z_0 = 0\,^\circ$ and the final position as $z_\mathrm{f} = 360\,^\circ$ with transition time $t_\mathrm{f} = 6\,\mathrm{s}$. Figure 4.1(a) shows the desired trajectory. The analytical solution of the inverse model given by equation (3.11) for $N = 2$ disks is shown in Fig. 4.1(b) in terms of the states $\boldsymbol{y}$ and in Fig. 4.1(c) in terms of the desired system input u_ffw.

Table 4.1: Simulation parameters of the torsional oscillator.

Parameter	I	k	d
Value	$0.12\,\mathrm{kg\,m^2}$	$2\,\frac{\mathrm{N\,m}}{\mathrm{rad}}$	$0\,\frac{\mathrm{N\,m\,s}}{\mathrm{rad}}$



(a) Desired trajectory. (b) Desired states. (c) Desired system input.

Figure 4.1: Simulation results for a 2-disk torsional oscillator.

In order to compare the index reduction methods introduced in Sec. 3.2.5, the servo-constraint inverse model problem is now formulated using the different index reduction schemes and increasing number of disks N. The inverse model is integrated with the implicit Euler scheme. Since the system output z is directly part of the coordinate vector $\boldsymbol{y}$, the direct coordinate substitution presented in Sec. 3.2.5 can be applied and is denoted by the index *subs*. The projection approach presented in Sec. 3.2.5 is denoted by the index *proj*. Both index reduction approaches are compared to the original higher index formulation denoted by *orig*.
The maximum numerical error e_max is defined as

$$e_\mathrm{max} = \max_t \left| u_\mathrm{ref}(t) - u_\mathrm{ffw}(t) \right| \tag{4.6}$$

with the analytical reference solution $u_\mathrm{ref}(t)$ and the numerical solution $u_\mathrm{ffw}(t)$. The error is shown in the convergence plot in Fig. 4.2 for the different formulations. Both the projection approach as well as the direct substitution approach yield similar numerical results. Moreover, they yield numerically more stable results than the original formulation *orig* due to index reduction. This is true for all tested number of disks.

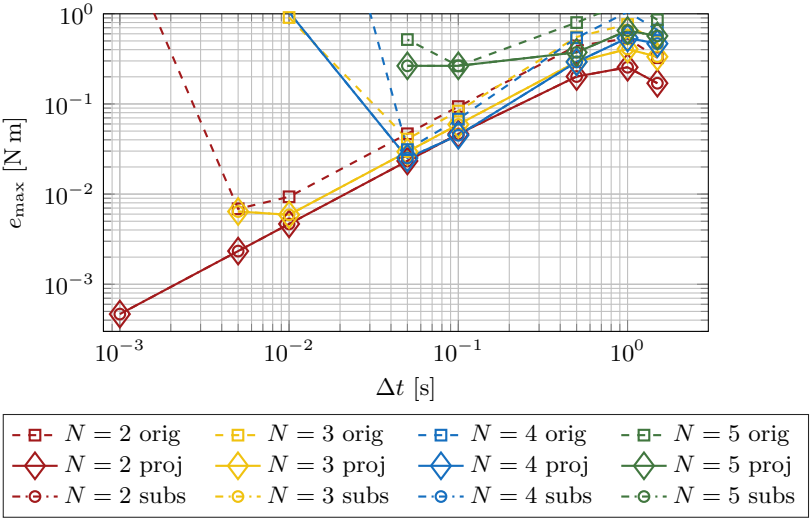


Figure 4.2: Convergence of the error $e_{\max}$ for the N-disk torsional oscillator for various inverse model formulations.

Using a larger number of disks in the mechanical model yields a numerically less stable solution. This is due to a higher differentiation index. For $N = 5$ disks, convergence is slow and the smallest error can be reached at $\Delta t = 0.1\,\mathrm{s}$ with $e_{\max} = 0.26\,\mathrm{N\,m}$. Index reduction only influences the numerical stability slightly and the results are not satisfactory for accurate feedforward control. However, acceptable errors can be obtained for $N = 2, 3$ and 4 disks. For $N = 2$ and 3 disks, index reduction helps to stabilize the numerical solution and the error can be reduced by decreasing the step size Δt. The smallest error for $N = 2$ is $e_{\max} = 5 \times 10^{-4}\,\mathrm{N\,m}$ at $\Delta t = 1\,\mathrm{ms}$, while it is $e_{\max} = 6 \times 10^{-3}\,\mathrm{N\,m}$ at $\Delta t = 10\,\mathrm{ms}$ for $N = 3$ and $e_{\max} = 2.5 \times 10^{-2}\,\mathrm{N\,m}$ at $\Delta t = 50\,\mathrm{ms}$ for $N = 4$. These solutions are expected to be sufficiently accurate for application of the inverse model as a feedforward controller on an experimental setup.

The Jacobian $\boldsymbol{J}$ of the set of nonlinear equations to be solved in every time step is analyzed in the following. The Jacobian can be derived analytically for all three approaches and is used to speed up the computations. Its condition number κ serves as an indicator of the stability of the numerical scheme. Large condition numbers indicate an ill-conditioned set of equations. Figure 4.3 shows the condition number κ for the analyzed formulations. The condition number increases for larger numbers N and for higher differentiation index. This supports the numerical problems shown in Fig. 4.2. At a given step size, the condition number κ is larger for the original formulation *orig* compared to the formulations

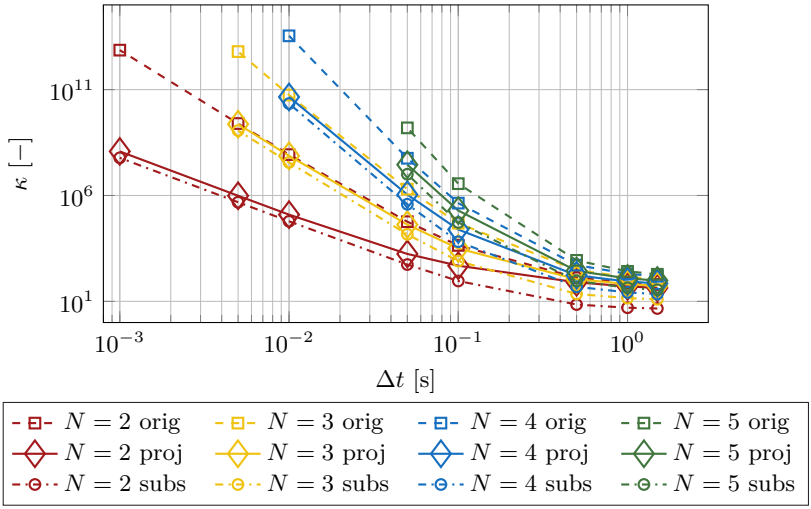


Figure 4.3: Condition number κ of the respective Jacobians $\boldsymbol{J}$ of the N-disk torsional oscillator.

proj as well as *subs*. The condition numbers for the formulations *proj* and *subs* behave similar with a small different constant.

In conclusion, with the currently available tools, the undamped torsional oscillator can be accurately inverted for up to $N = 4$ disks which corresponds to a differentiation index of 9. However, numerically more stable results are obtained for $N = 2$ and $N = 3$ with index reduction. Application of repetitive index reductions can be further investigated. Moreover, very light damping might be introduced to reduce the differentiation index of the model directly at the cost of introducing stable internal dynamics.

4.2 Cable Robot

Cable robots are typical differentially flat examples in the context of servo-constraints. They are for example considered in terms of overhead cranes in [BlajerKołodziejczyk04, FumagalliEtAl10, LeitzLeyendecker14] and considered in terms of three-dimensional rotary cranes in [BetschAltmannYang16, BetschQuasemUhlar09, YangBetschZhang20]. Thorough experimental validation of the proposed scheme is given in [OttoSeifried18c]. The following results for the cable robot demonstrate that servo-constraints can be implemented on an experimental setup to compute the feedforward control in real-time. Moreover, combinations with feedback controllers are evaluated.

First, the system properties are analyzed for the complex and reduced cable robot models introduced in Sec. 2.4.1. Afterwards, the experimental setup and a state observer are briefly introduced. Simulation results for the reduced cable robot model are shown first to analyze the implicit Euler scheme for different problem formulations. Afterwards, higher order integration schemes and different approaches for improving the numerical computation are compared. A solver most suitable for experiments is selected based on this analysis. Then, experiments demonstrate the real-time capabilities of the selected solvers as well as the combination of the feedforward control with feedback controllers. Moreover, experimental results are presented for the complex cable robot model with variable platform orientation. This demonstrates the use of implicit servo-constraints as well as the applicability of the approach to systems of increased complexity.

4.2.1 System Properties

Two cable robot configurations are considered in the following, which are both introduced in Sec. 2.4.1. The complex cable robot model has $n_\mathrm{f} = 4$ degrees of freedom and $n_\mathrm{u} = 3$ system inputs. It allows for individual cable lengths and therefore models a variable platform orientation. The reduced cable robot model with $n_\mathrm{f} = 3$ degrees of freedom and $n_\mathrm{u} = 2$ system inputs assumes equal cable length and therefore a horizontal platform. For both systems, the vector relative degree is determined according to Sec. 3.1.3 and Sec. 3.2.3.

For the complex cable robot model, the generalized eigenvalue problem according to Sec. 3.2.3 shows that the sum of the vector relative degree is $r = 8 = \Sigma r_i$. Since this corresponds to $2n_\mathrm{f} = 8$, the system is differentially flat and it is theoretically possible to describe the system input as a function of the output and its derivatives. A symbolic analysis of the system dynamics shows that one of the inputs u_1,u_2,u_3 arises for the first time at the output derivatives $z_1^{(4)}$, $\ddot{z}_2$ and $\ddot{z}_3$. However, the decoupling matrix is singular and strictly speaking, the vector relative degree is therefore undefined. Further analytical derivations including dynamic extension are quite burdensome and are not performed here. Therefore, the inverse model DAEs of differentiation index 5 are solved.

For the reduced cable robot model, the sum of the vector relative is $r = 2n_\mathrm{f} = 6$ and the system is differentially flat. Even though the model is differentially flat, model inversion based on the Byrnes-Isidori normal form according to Sec. 3.1.3 is quite involved. Differentiating the system output shows that one of the inputs u_1,u_2 first appears at the output derivatives $z_1^{(4)}$ and $\ddot{z}_2$. However, the decoupling matrix between input and output channels is again singular. Strictly speaking, the vector relative degree is undefined and dynamic extension must be performed. This involves further analytical derivations, see [Isidori96] for a general introduction and [BoustanyD'Andrea-Novel92] for application of the overhead crane. This can be avoided by applying the presented servo-constraints and directly solving the inverse model DAEs of differentiation index 5.

4.2.2 Experimental Setup

The experimental cable robot is setup at the Institute for Mechanics and Ocean Engineering at Hamburg University of Technology and is shown in Fig. 2.4(a). The trolley is actuated by two servomotors running synchronously. Each of the four cables can be actuated individually by a servomotor connected to a winch. All actuators are controlled by a SIEMENS SIMOTION D435 control unit with cascaded internal control loops. As introduced in Sec. 2.3, the actuators are operated in velocity-controlled mode. All actuators are equipped with absolute angle encoders and force sensors measure the cable forces. Figure 4.4 shows a top view of one half of the trolley with the described components. Physical dimensions and the identified time constants of the test bench are listed in Tab. 4.2. Kinematic parameters, such as cable velocities are listed in Tab. 4.3. For a detailed description of the experimental setup refer to [Rapp12].

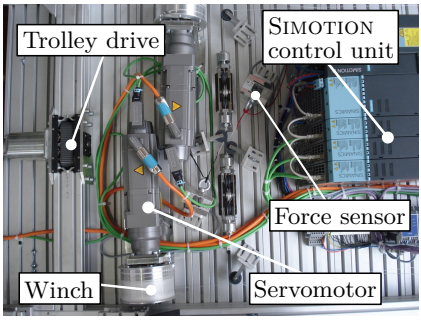


Figure 4.4: Top view of one half of the cable robot trolley.

The hardware setup is supplemented by a real-time computer and a desktop computer. The desktop computer offers a user-interface for programming the control code which is then employed on the real-time computer. The control of the test bench is implemented in LABVIEW [LABVIEW15]. The main program architecture is developed in [Oltmann11].
The real-time computer is connected by a Process Field Bus (Profibus) connection to the SIMOTION control unit. Sensor data and control commands are exchanged via the Profibus. The Profibus loop is running at 500 Hz. A COMPACTRIO unit pre-filters the force sensor measurements and passes the information to the Profibus.
The control loop is executed at a frequency of 100 Hz and is running in parallel to the Profibus loop on the real-time computer. The developed control strategies are implemented in MATLAB for simulations and testing. For application on the test bench, the MATLAB code is exported to C code, which is more efficient than MATLAB code. The generated C code is compiled in an executable dynamic link library, which is called by LABVIEW during runtime.

Table 4.2: Physical parameters of the cable robot test bench [KreuzerRadisch14].

Parameter	Symbol	Value
Platform size	$L_{\mathrm{P},x} \times L_{\mathrm{P},y} \times L_{\mathrm{P},z}$	$0.35\,\mathrm{m} \times 0.86\,\mathrm{m} \times 0.37\,\mathrm{m}$
Platform mass 1	$m_{\mathrm{P},1}$	$12.9\,\mathrm{kg}$
Platform mass 2	$m_{\mathrm{P},2}$	$18.9\,\mathrm{kg}$
Trolley mass	m_{T}	$270\,\mathrm{kg}$
Time constant trolley	τ_{T}	$0.03\,\mathrm{s}$
Time constant winch	τ_{L}	$0.02\,\mathrm{s}$

Table 4.3: Admissible ranges of the cable robot states [Rapp12, Radisch18].

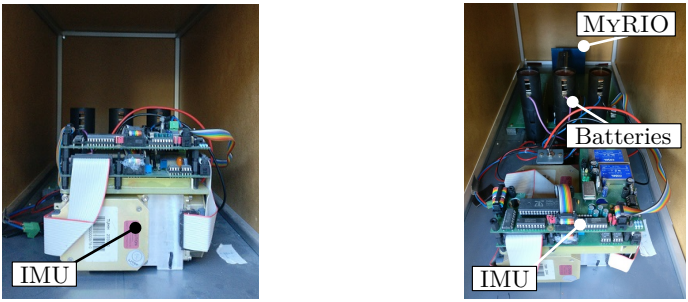
Parameter	Admissible range
Sway angle	$\varphi \in [-12, 12]\ ^\circ$
Cables	$L \in [3, 12]\,\mathrm{m}\ ,\ \dot{L} \in [-2, 2]\,\frac{\mathrm{m}}{\mathrm{s}}\ ,\ \ddot{L} \in [-3, 3]\,\frac{\mathrm{m}}{\mathrm{s}^2}$
Trolley position	$x_{\mathrm{T}} \in [0, 13]\,\mathrm{m}\ ,\ \dot{x}_{\mathrm{T}} \in [-3, 3]\,\frac{\mathrm{m}}{\mathrm{s}}\ ,\ \ddot{x}_{\mathrm{T}} \in [-4, 4]\,\frac{\mathrm{m}}{\mathrm{s}^2}$

4.2.3 Observer Design

While absolute angle encoders measure the trolley position x_{T} and all cables lengths L_i, the sway angle φ as well as the platform orientation δ_{P} cannot be measured directly. Therefore, an observer is implemented to estimate these values. An unscented Kalman filter (UKF) for observing the sway angle φ is designed in [Theis10, Rapp12] for the reduced cable robot model. It is based on measuring the cable forces.

For the complex cable robot model with variable platform orientation, the previously designed UKF is not applicable because the motion of platform orientation δ_{P} and sway angle φ is coupled. Their dynamics is not observable from the force sensor measurements and additional measurements of the platform motion are necessary. The measurements are obtained by an inertial measurement unit (IMU), which is placed inside the load platform, see Fig. 4.5. Thus, the platform mass increases to platform mass $m_{\mathrm{P},2}$, see Tab. 4.2. The IMU measurements are evaluated by a MyRIO control unit, which is placed inside the platform. The MyRIO control unit sends the sensor data to the real-time computer by a wireless connection.

Note that the complex cable robot model is given in DAE form. Therefore, the classical UKF algorithm for ODEs, e.g. described in [ThrunBurgardFox06], cannot be applied directly. Several approaches extend the UKF for systems in DAE form, see e.g. the survey in [PatwardhanEtAl12]. It is noted in [MandelaEtAl10] that the observer design is more complex due to two reasons. First, some approaches do not allow for measurements of the algebraic variables, because they



(a) Front view. (b) Top view.

Figure 4.5: Pictures of the hardware inside the load platform.

are usually eliminated in the process of transforming a DAE into an ODE. Secondly, the estimated states might not be consistent with the algebraic constraints of the system. In this thesis, the DAEs are transformed to ODEs by a QR decomposition, see e.g. [Bauchau11]. Transforming the equations of motion in this fashion preserves the original redundant coordinates. Thus, it is possible to tune the observer gains with the physical quantities in mind. This can facilitate observer tuning. Moreover, measurements of the algebraic variables, such as cable forces, can be directly considered in the observer. See [OttoRückwaldSeifried19] for details on the proposed UKF algorithm.

4.2.4 Comparison of Generalized and Redundant Coordinates

Considering the reduced cable robot model with platform mass $m_{\mathrm{P},1}$, its equations of motion can be written in DAE or ODE form, see Sec. 2.4.1. Both formulations are compared in the following with respect to the solution of the inverse model problem. For both formulations, the inverse model problem is solved and the error is compared to the analytical solution $\boldsymbol{u}_{\mathrm{ref}}$, e.g. given in [FliessEtAl95, Rudolph03]. The following results are presented according to [OttoSeifried18c].
Computing the maximum difference between the numerical feedforward control input $\boldsymbol{u}_{\mathrm{ffw}}$ and the analytical reference $\boldsymbol{u}_{\mathrm{ref}}$ yields the error $e(t)$ with

$$e(t) = \max_i \| u_{\mathrm{ffw},i}(t) - u_{\mathrm{ref},i}(t) \| \qquad i = 1, 2\,. \tag{4.7}$$

Note that according to Sec. 2.4.1, both system inputs u_1 and u_2 are reference velocities of the actuators given in the dimension of trolley and cable velocity respectively. The maximum numerical error is defined as

$$e_{\max} = \max_t e(t)\,. \tag{4.8}$$

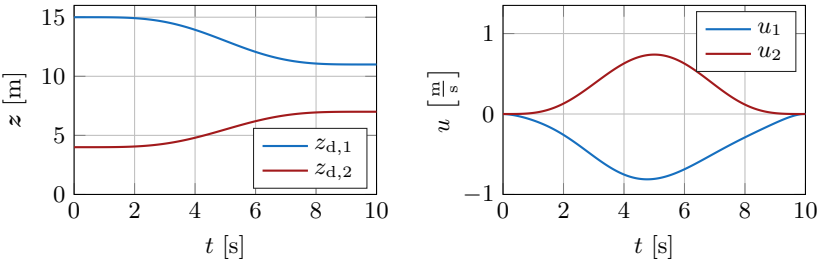
For definition of the reference trajectory $\boldsymbol{z}_\mathrm{d}(t)$, the system output of the cable robot is the platform position, see equation (2.76). Each point $\boldsymbol{p}(\sigma)$ on the trajectory is given by the straight line

$$\boldsymbol{p}(\sigma(t)) = \boldsymbol{p}_0 + \frac{\sigma(t)}{\sigma_\mathrm{t}} (\boldsymbol{p}_\mathrm{f} - \boldsymbol{p}_0) , \quad \sigma(t) \in \begin{bmatrix} 0 , & \sigma_\mathrm{t} \end{bmatrix} , \tag{4.9}$$

in the coordinate system K: $\{\mathrm{O}, x, y, z\}$, see Fig. 2.4(b). Thus, the initial point is $\boldsymbol{p}_0 = \begin{bmatrix} x_{\mathrm{T},0} & L_0 \end{bmatrix}^\mathsf{T}$ and the final point is $\boldsymbol{p}_\mathrm{f} = \begin{bmatrix} x_{\mathrm{T,f}} & L_\mathrm{f} \end{bmatrix}^\mathsf{T}$. The path is parameterized by the scalar value $\sigma(t)$, which defines the temporal sequence for the trajectory. Due to vector relative degree $\boldsymbol{r} = \{4, 2\}$, the timing law $\sigma(t)$ must be at least four times continuously differentiable, see Sec. 3.3.1. Here, it is defined as the polynomial

$$\sigma(t) = \left(126 \left(\frac{t}{t_\mathrm{f}}\right)^5 - 420 \left(\frac{t}{t_\mathrm{f}}\right)^6 + 540 \left(\frac{t}{t_\mathrm{f}}\right)^7 - 315 \left(\frac{t}{t_\mathrm{f}}\right)^8 + 70 \left(\frac{t}{t_\mathrm{f}}\right)^9 \right) \sigma_\mathrm{t} , \tag{4.10}$$

with total arc length $\sigma_\mathrm{t} = \|\boldsymbol{p}_\mathrm{f} - \boldsymbol{p}_0\|_2$. The following results are provided for a reference trajectory from the initial position $\boldsymbol{y}_0 = \begin{bmatrix} x_{\mathrm{T},0} & L_0 & \varphi_0 \end{bmatrix}^\mathsf{T} = \begin{bmatrix} 15\,\mathrm{m} & 4\,\mathrm{m} & 0^\circ \end{bmatrix}^\mathsf{T}$ to the final position $\boldsymbol{y}_\mathrm{f} = \begin{bmatrix} 11\,\mathrm{m} & 7\,\mathrm{m} & 0^\circ \end{bmatrix}^\mathsf{T}$ with transition time $t_\mathrm{f} = 10\,\mathrm{s}$. The desired trajectory and the desired system input $\boldsymbol{u}_\mathrm{ffw}$ are shown in Fig. 4.6, while Fig. 4.7 visualizes the trajectory in the x, y plane. The simulation results are shown for the projected inverse model equations for the system model described by generalized coordinates. The inverse model is solved using the BDF solver with $k_\mathrm{bdf} = 4$ and step size $\Delta t = 10\,\mathrm{ms}$.



(a) Desired output $\boldsymbol{z}_\mathrm{d}(t)$.

(b) Computed system input $\boldsymbol{u}_\mathrm{ffw}$.

Figure 4.6: Desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ and numerical results for the reduced cable robot model.

In the following, various formulations of the reduced cable robot model are compared. Thereby, the focus lies on the formulations of the equations of motion in either redundant or generalized coordinates, which are denoted by the indices *dae* and *ode* respectively. Both formulations are presented in Sec. 2.4.1.

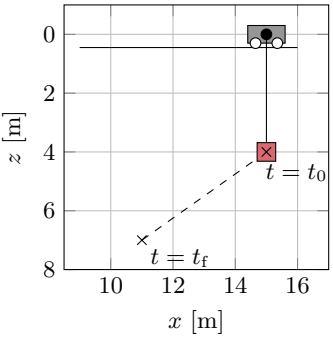


Figure 4.7: Visualization of the desired trajectory.

The original inverse model problem has differentiation index 5, see Sec. 4.2.1. The formulation of the cable robot using redundant coordinates is given by the matrices (2.73)–(2.74) and the respective servo-constraint for the system output (2.76). The index is reduced to 3 by direct substitution, see Sec. 3.2.5. The formulations are denoted as $\boldsymbol{y}_{\mathrm{dae,i5}}$ and $\boldsymbol{y}_{\mathrm{dae,i3}}$ for the index 5 and index 3 formulation respectively. The formulation of the cable robot with generalized coordinates is given by the matrices (2.69)–(2.70) and the respective servo-constraint for the system output (2.76). The index is reduced to 3 by the projection method introduced in Sec. 3.2.5. The formulations are denoted as $\boldsymbol{y}_{\mathrm{ode,i5}}$ and $\boldsymbol{y}_{\mathrm{ode,i3}}$ for the index 5 and index 3 formulation respectively. Different reduction schemes are applied here, since the projection approach is formulated for equations of motion in ODE form, while the direct substitution approach can be easily applied to systems in DAE form due to the special selection of redundant coordinates.
All formulations are solved with the implicit Euler scheme and maximum number of $n_{\mathrm{N,max}} = 10$ Newton iterations. Figure 4.8 shows the error $e_{\max}$ for different step sizes Δt. The time discretization $\Delta t = 10\,\mathrm{ms}$ available on the test bench is highlighted by the vertical line. Both formulations in redundant coordinates and generalized coordinates show a similar behavior for the index 3 and index 5 case, respectively. Due to improved numerical condition, the index 3 formulation shows smaller maximum errors. Moreover, the index 3 formulation runs into numerical rounding errors for much smaller steps sizes and is therefore more robust. Note also that the maximum error $e_{\max} \approx 1.4 \times 10^{-3}\,\frac{\mathrm{m}}{\mathrm{s}}$ for the discretization $\Delta t = 10\,\mathrm{ms}$ of the test bench is considerably smaller than other disturbances in the actuators, such as friction. For practical application, the numerical accuracy should be at least in the range of other disturbances on the test bench. More accurate numerical solutions do not increase the tracking accuracy while the numerical effort increases with more advanced numerical schemes. Here, nearly similar results are generated for the linear implicit Euler for which the maximum number of Newton iterations is set to $n_{\mathrm{N,max}} = 1$, see Sec. 3.3.4.

The convergence diagram is shown in Fig. 4.9. The only difference is present in the redundant coordinates in index 5 formulation, which does not converge at all.

In conclusion, the implicit Euler as well as the linear implicit Euler are applicable to solve the inverse model of the cable robot with the desired step size. The solution is considered to be sufficiently accurate for application on the test bench in comparison to other disturbances.

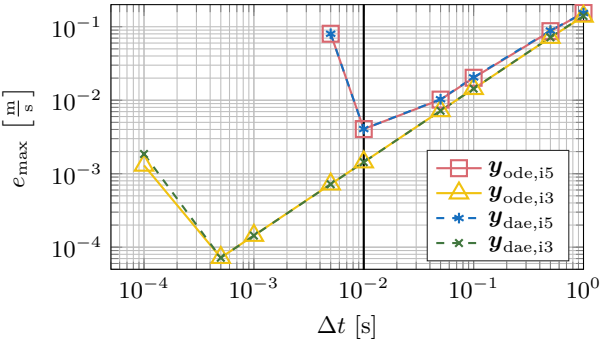


Figure 4.8: Maximum error $e_{\max}$ for the implicit Euler scheme.

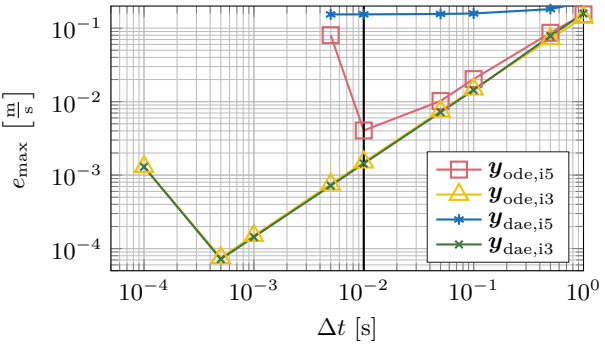


Figure 4.9: Maximum error $e_{\max}$ for the linear implicit Euler scheme.

4.2.5 Application of Higher Order Integration Schemes

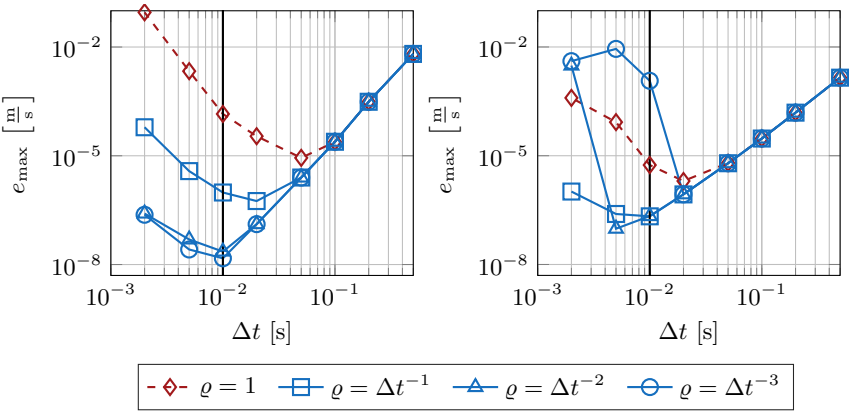
Even though the (linear) implicit Euler scheme proves to be sufficient in experiments, the higher order schemes of the 4-step BDF method and 3-stage Runge-Kutta method are applied in order to demonstrate their capabilities in real-time applications. They might also be applied to other systems, for which the implicit Euler scheme is not sufficient, such as systems with internal dynamics. The following results are presented according to [OttoSeifried18a].

The results for the higher order integration schemes are computed for the reduced cable robot model with platform mass $m_{\mathrm{P},2}$. The inverse model formulation based on redundant coordinates with index reduction based on direct substitution (configuration $\boldsymbol{y}_{\mathrm{dae,i3}}$) is considered for this analysis. The desired trajectory is again given by equations (4.9) and (4.10). Various approaches for increasing computational speed are introduced in Sec. 3.3.4. These include scaling of the algebraic constraints and Broyden's method. These methods are applied and compared in the following. Moreover, the BDF scheme is compared to the Radau5 scheme, see Sec. 3.3.3 for an overview of the solvers. As a result of the analysis, a solver most suitable for experiments is proposed.

Scaling of Algebraic Constraints

For the BDF as well as the Runge-Kutta methods, the influence of scaling the algebraic constraints is analyzed according to Sec. 3.3.4. The BDF scheme is considered with $k_{\mathrm{bdf}} = 4$ and is compared to the Radau5 integration scheme. The Jacobian is approximated with Broyden's method and reinitialized every 10 steps, see Sec. 3.3.4. The maximum error $e_{\max}$ defined in equation (4.8) is computed for the dimensionless scaling factors $\varrho \in \left[1, \Delta t^{-1}, \Delta t^{-2}, \Delta t^{-3}\right]$. Figure 4.10 shows the results of the inverse model computation. For both the BDF scheme in Fig. 4.10(a) as well as the Radau5 scheme in Fig. 4.10(b), the original unscaled inverse system with $\varrho = 1$ results in an ill-conditioned set of DAEs. The maximum error $e_{\max}$ increases due to rounding errors for step sizes $\Delta t > 0.1\,\mathrm{s}$ and $\Delta t > 0.05\,\mathrm{s}$ for the BDF and Radau5 scheme respectively.

Regarding the BDF scheme, Fig. 4.10(a) shows that scaling improves the numerical accuracy. Scalings of $\varrho = \Delta t^{-2}$ and $\varrho = \Delta t^{-3}$ yield almost identical results, which are more stable compared to the unscaled solution with $\varrho = 1$. The maximum error $e_{\max}$ reduces up to a step size $\Delta t = 10\,\mathrm{ms}$ before rounding errors start to influence the accuracy. Concerning the Radau5 scheme, a scaling of $\varrho = \Delta t^{-2}$ results in the most stable solution. The maximum error $e_{\max}$ starts to increase for step sizes $\Delta t < 5\,\mathrm{ms}$. Note that in contrast to the results for the BDF scheme, a scaling $\varrho = \Delta t^{-3}$ reduces the solution accuracy and yields an ill-conditioned set of equations for this system. Due to the presented results, the scaling $\varrho = \Delta t^{-2}$ is chosen for the following simulations for both integration methods.



(a) BDF $k_{\text{bdf}} = 4$. (b) Radau5.

Figure 4.10: Variation of scaling factor ϱ of the algebraic constraints.

Analysis of Radau5 Scheme with Broyden's Method

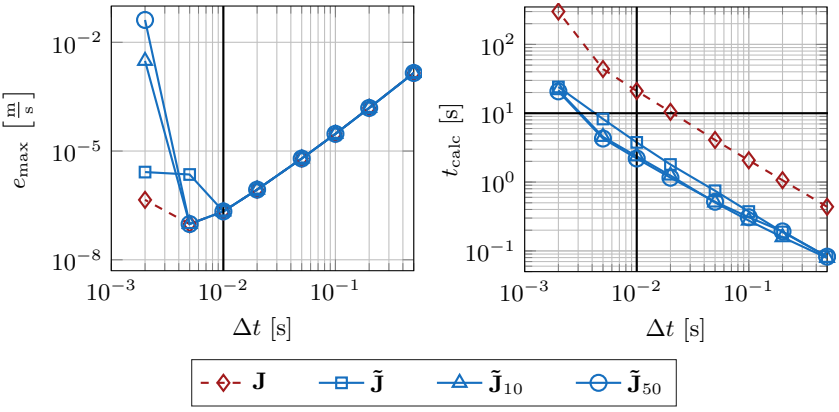
In the following, Broyden's method is analyzed in more detail for its application within the Newton scheme of the Radau5 solver, see Sec. 3.3.4. Broyden's method is initialized by calculating the Jacobian $\widetilde{\boldsymbol{J}}^0$ with backwards finite differences. The results for applying Broyden's method continuously without reinitialization are denoted by $\widetilde{\boldsymbol{J}}$. A reinitialization every 10 and 50 steps is denoted by $\widetilde{\boldsymbol{J}}_{10}$ and $\widetilde{\boldsymbol{J}}_{50}$, respectively. The results are compared to Newton iterations with a newly computed numerical Jacobian in each iteration, denoted by $\boldsymbol{J}$. Convergence of the approximated schemes might be improved by allowing more Newton iterations in each time step. However, this will eventually slow down computation time and is not further considered here. The maximum number of Newton iterations is set to $n_{\text{N,max}} = 20$.

The results for the various Broyden schemes are shown in Fig. 4.11(a) in terms of the maximum error $e_{\max}$. It can be seen that the Radau5 scheme converges approximately with order 2. This corresponds to the theoretical results stated in Sec. 3.3.3. The convergence order is independent of the chosen Broyden update method for step sizes $\Delta t \geq 10\,\text{ms}$. For step sizes $\Delta t < 10\,\text{ms}$, recalculating the Jacobian $\boldsymbol{J}$ in each time step yields the most accurate results since it is the most accurate approximation of the correct Jacobian. The maximum error $e_{\max}$ increases due to rounding errors for step sizes $\Delta t \leq 2\,\text{ms}$.

Broyden's method is supposed to speed up calculation of the inverse model. The total computation time t_{calc} for a complete computation of the inverse model on a standard desktop computer is shown in Fig. 4.11(b). Note that the total computation time t_{calc} only gives an estimate of the real-time applicability since

the computation time $t_{\mathrm{calc},i}$ of each time step must be within the respective limit Δt, see Sec. 3.3.4. Moreover, the computation time is usually slower on the desktop computer compared to the experimental setup, since faster C code is running on the real-time setup, see Sec. 4.2.2. Nonetheless, the comparison yields an insight and relative comparison of the various schemes. In Fig. 4.11(b), the total simulation time of 10 s and the available step size $\Delta t = 10\,\mathrm{ms}$ are highlighted.

Recomputing the Jacobian $\boldsymbol{J}$ with finite differences in each time step yields the largest computation times t_{calc}. This is not real-time capable for this system. Applying Broyden's method considerably speeds up the computation due to less function evaluations. However, inaccuracies are introduced if the Jacobian is not reinitialized at all, as can be seen in the convergence diagram in Fig. 4.11(a). The approximated Jacobian $\widetilde{\boldsymbol{J}}$ becomes inaccurate after too many iterations. Therefore, the number of Newton iterations increases, which eventually slows down the complete computation. Therefore, the computation time with the Jacobian $\widetilde{\boldsymbol{J}}$ is slower than using the Jacobians $\widetilde{\boldsymbol{J}}_{10}$ and $\widetilde{\boldsymbol{J}}_{50}$. Reinitializing the Jacobian after 10 or 50 steps yields accurate results for step sizes $\Delta t \geq 5\,\mathrm{ms}$. At the step size $\Delta t = 10\,\mathrm{ms}$, the solution with the Jacobian $\widetilde{\boldsymbol{J}}_{10}$ is approximately 9 times faster compared to recalculating the Jacobian in each time step. In absolute values, the computation times are $t_{\mathrm{calc}}(\widetilde{\boldsymbol{J}}_{10}) = 2.4\,\mathrm{s}$ and $t_{\mathrm{calc}}(\boldsymbol{J}) = 20.5\,\mathrm{s}$ respectively. As a consequence of these analyses, experiments will be performed with Radau5 with Broyden's method and a reinitialization every 10 steps.



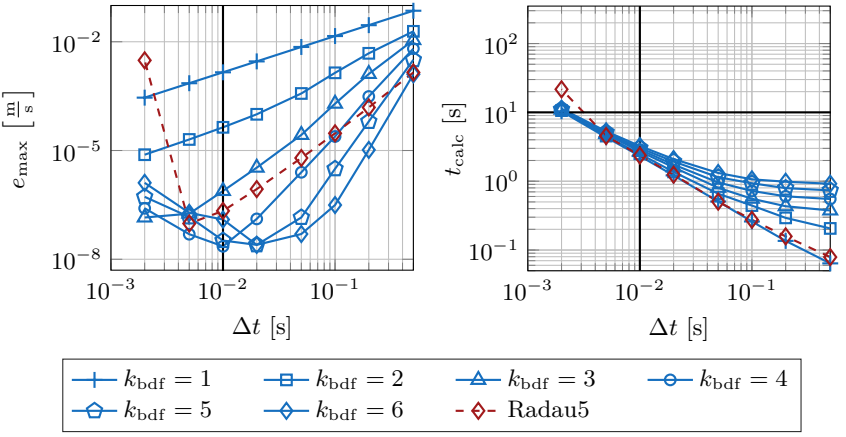
(a) Convergence. (b) Total computation time.

Figure 4.11: Variation of Broyden's method within Radau5 scheme.

Analysis of BDF Scheme

The BDF schemes are analyzed for different steps k_{bdf}. The BDF results are compared to the Radau5 schemes with reinitialized Jacobian $\widetilde{\boldsymbol{J}}_{10}$ every 10 steps. In the Newton iterations within the BDF method, the Jacobian $\boldsymbol{J}$ is computed numerically.

Figure 4.12(a) shows the convergence of the BDF method and verifies the theoretical results for the convergence orders stated in Sec. 3.3.3 for $k_{\mathrm{bdf}} = 1, 2, \ldots, 6$. The maximum error $e_{\max}$ for $k_{\mathrm{bdf}} > 2$ is orders of magnitude smaller than the error of the implicit Euler scheme corresponding to $k_{\mathrm{bdf}} = 1$, which is usually applied in the context of servo-constraints. However, note that the problem becomes ill-conditioned for $k \geq 5$. The maximum error increases due to numerical rounding errors for step sizes $\Delta t < 20\,\mathrm{ms}$ and $\Delta t < 50\,\mathrm{ms}$ for $k_{\mathrm{bdf}} = 5$ and $k_{\mathrm{bdf}} = 6$, respectively. A choice of $k_{\mathrm{bdf}} = 4$ yields convergence of order 4 and thus faster convergence compared to Radau5, while the results are still stable for the step size $\Delta t = 10\,\mathrm{ms}$ of the experimental test bench. Concerning the computation times in Fig. 4.12(b), the BDF and Radau5 scheme yield comparable computation times. They are real-time capable at the test bench frequency. As a result of the numerical analyses, the BDF scheme with $k_{\mathrm{bdf}} = 4$ is chosen for experiments.

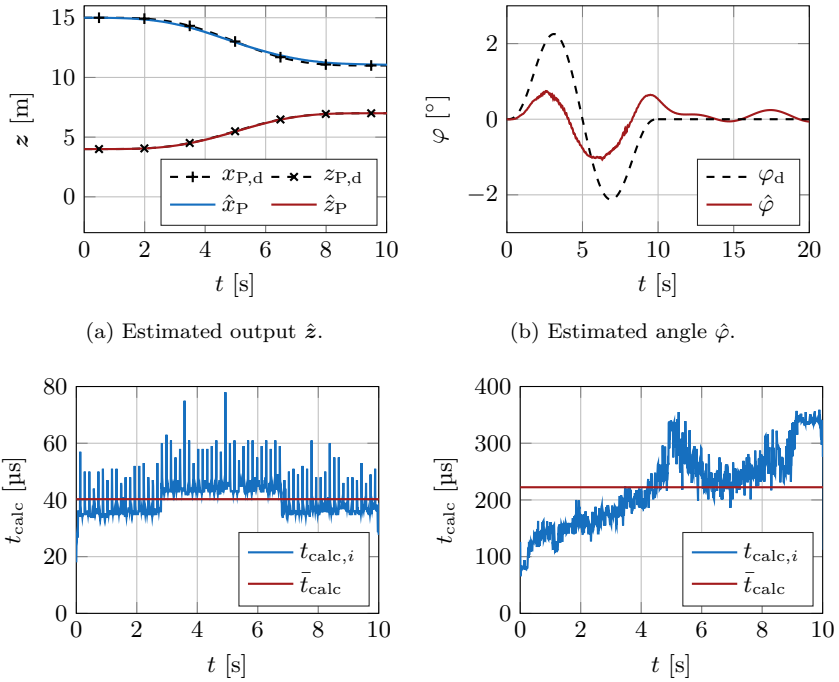


(a) Convergence. (b) Total computation time.

Figure 4.12: Variation of k_{bdf}-step BDF scheme.

4.2.6 Experimental Results for Higher Order Integration Schemes

The higher order solvers analyzed above are now applied in real-time on the experimental setup. Solver settings for both the BDF as well as Radau5 are selected based on the above discussion. Note that the numerical errors are generally very small compared to other uncertainties and all solvers are expected to yield similar experimental results. The desired trajectory is chosen according to equations (4.9)–(4.10). Experimental results are shown in Fig. 4.13 for pure feedforward control $\boldsymbol{u}_{\mathrm{ffw}}$ computed by the inverse model. Estimated and measured values are denoted by $\hat{\square}$. The results demonstrate that the feedforward controller is able to move the system on the desired trajectory quite accurately. A small sway angle of $\hat{\varphi} \approx 0.3°$ remains, see Fig. 4.13(b). A distinction between observer error and tracking error is not possible due to lack of a direct measurement.



(a) Estimated output $\hat{\boldsymbol{z}}$.

(b) Estimated angle $\hat{\varphi}$.

(c) Calculation time $k_{\mathrm{bdf}} = 4$ BDF.

(d) Calculation time Radau5.

Figure 4.13: Experimental results for the reduced cable robot model with higher order integration schemes.

The inverse model is computed on the experimental setup in real-time. The computation time $t_{\text{calc},i}$ to compute the desired system input $\boldsymbol{u}_{\text{ffw}}(t_i)$ at each time step t_i is measured on the experimental setup on a microsecond clock. Figure 4.13(c) shows the measured computation times for the BDF scheme with a mean of $\bar{t}_{\text{calc}} = 40.28\,\mu\text{s}$ over the course of the trajectory. Figure 4.13(d) shows the measured computation times for the Radau5 scheme with a mean of $\bar{t}_{\text{calc}} = 222.38\,\mu\text{s}$.

The timing results are validated with a series over 5 experiments each, see Fig. 4.14. The mean computation time of the BDF scheme is $\bar{t}_{\text{calc}} = 40.29\,\mu\text{s}$ with standard deviation $\sigma_{\text{calc}} = 0.08\,\mu\text{s}$. For the Radau5 scheme, the overall mean calculation time is $\bar{t}_{\text{calc}} = 235.87\,\mu\text{s}$ with standard deviation $\sigma_{\text{calc}} = 10.85\,\mu\text{s}$. In comparison, the evaluation of the analytical solution is faster with a mean computation time $\bar{t}_{\text{calc}} = 5.04\,\mu\text{s}$ and standard deviation $\sigma_{\text{calc}} = 0.006\,\mu\text{s}$.

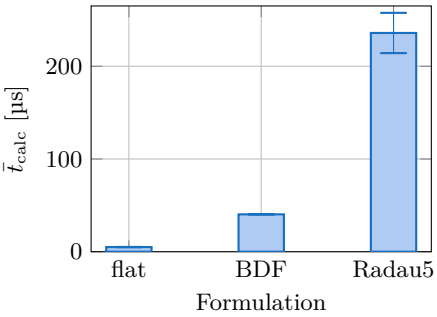


Figure 4.14: Mean and two times of the standard deviation of 5 measurements each of the mean calculation time $\bar{t}_{\text{calc}}$.

Compared to the simulation results presented in Sec. 4.2.5, the computation time t_{calc} is considerably shorter on the experimental setup. This is due to the C code export and faster code evaluation on the real-time operating system. On the experimental setup, the BDF method is approximately 6 times faster compared to Radau5. The computation time highly depends on the chosen implementation of the integration scheme, the optimization during automatic C code generation and the hardware. Nonetheless, as a main result, both higher order integration methods are applicable on the experimental setup and are able to solve the index 3 DAEs in real-time. Compared to the available time Δt, there is potential to solve for larger mechanical models, such as the cable robot model including the platform orientation. This is demonstrated in Sec. 4.2.8.

4.2.7 Experimental Results for Combining Feedforward and Feedback Control

The combination of feedforward and feedback control is demonstrated in the following for the reduced cable robot with platform mass $m_{\mathrm{P},1}$. Even though the numerical results using the implicit Euler are not as accurate as using higher order integration schemes, the error is sufficiently small compared to other uncertainties and disturbances on the experimental setup. Therefore, the implicit Euler scheme is used to compute the feedforward control law in the following experiments. The pure feedforward control $\boldsymbol{u}_{\mathrm{ffw}}$ is compared to a combination of feedforward control and LQR feedback control according to Fig. 3.1. The LQR control is implemented according to Sec. 3.4.1 for the model linearized around the final position. Moreover, the sliding mode controller introduced in Sec. 3.4.2 is applied without the feedforward part. Tables 4.4 and 4.5 list the chosen control parameters. Both controllers require information of the desired state trajectory $\boldsymbol{y}_{\mathrm{d}}(t)\,, \boldsymbol{v}_{\mathrm{d}}(t)$, which is calculated as part of the inverse model. Thus, solving the inverse model is necessary for controller application. The following results are presented according to [OttoSeifried18c].

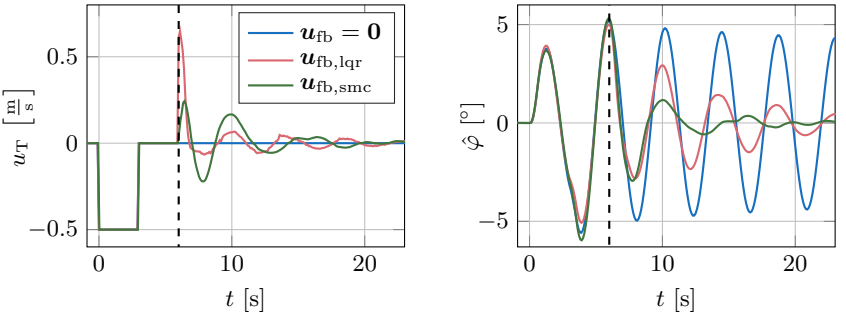
First, the feedback controllers are evaluated separately for stabilizing the sway angle φ. The experimental results are shown in Fig. 4.15. A maximum platform sway angle of $\hat{\varphi} \approx 5.2\,^{\circ}$ is generated by imposing a step input of $u_{\mathrm{T}} = -0.5\,\frac{\mathrm{m}}{\mathrm{s}}$ on the trolley for 3 s, as can be seen in Fig. 4.15(a). Then, there is a waiting period of another 3 s before the controller is activated at time $t = 6\,\mathrm{s}$. The dashed lines in Fig. 4.15 denote the controller activation. Figure 4.15(b) shows that both controllers are able to dampen the sway angle φ compared to the free system with $\boldsymbol{u}_{\mathrm{fb}} = \boldsymbol{0}$. The free system itself contains very little damping. While the LQR reduces the sway angle in approximately 17 s, the SMC shows

Table 4.4: Control parameters of the LQR [OttoSeifried18c].

Parameter	$\boldsymbol{R}_{\mathrm{lqr}}$	$\boldsymbol{Q}_{\mathrm{lqr}}$	
Value	$100\,\boldsymbol{I}_2$	$\begin{bmatrix} 1000\,\boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & 1000\,\boldsymbol{I}_3 \end{bmatrix}$	

Table 4.5: Control parameters of the sliding mode control [OttoSeifried18c].

Parameter	$\boldsymbol{\Theta}_{\mathrm{a}}$	$\boldsymbol{\Lambda}_{\mathrm{a}}$	$\boldsymbol{\Theta}_{\mathrm{u}}$	$\boldsymbol{\Lambda}_{\mathrm{u}}$	$\boldsymbol{\kappa}_{\mathrm{smc}}$	$\boldsymbol{\Phi}_{\mathrm{smc}}$
Value	$\boldsymbol{I}_2$	$0.5\,\boldsymbol{I}_2$	$\begin{bmatrix} 1.5 \\ 0 \end{bmatrix}$	$\begin{bmatrix} -5.0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$\boldsymbol{I}_2$



(a) System input u_{T}.

(b) Estimated sway angle $\hat{\varphi}$.

Figure 4.15: Experimental results for platform sway reduction.

superior damping properties. Further tuning of the control parameters change the performance on a Pareto optimal curve, since there is always a trade-off between control accuracy and control input.

Both feedback controllers are now applied in combination with feedforward control $\boldsymbol{u}_{\mathrm{ffw}}$. The trajectory is again given by equations (4.9)–(4.10). The experimental results are shown in Fig. 4.16. The pure feedforward control $\boldsymbol{u}_{\mathrm{ffw}}$ as well as the results with feedback control show accurate tracking. Most accurate tracking is here achieved with the combination of feedforward control and LQR. Accurate tracking of the pure feedforward control $\boldsymbol{u}_{\mathrm{ffw}}$ indicates an accurate inverted model of the experimental setup.

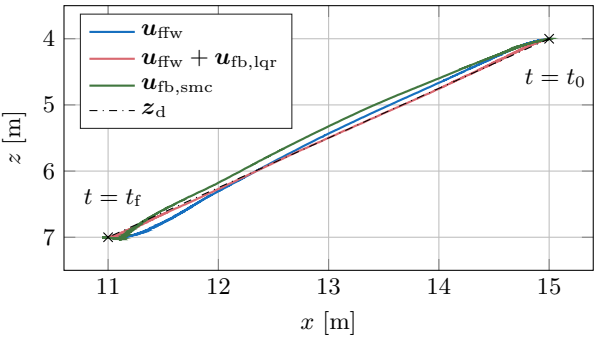


Figure 4.16: Experimental results for the straight trajectory.

The influence and robustness of the feedback controllers is now demonstrated by introducing an initial position error of $\Delta x_\mathrm{T} = 0.5\,\mathrm{m}$. For this experiment, the real initial position is chosen as $\boldsymbol{y}_0 = \begin{bmatrix} 11\,\mathrm{m} & 4.5\,\mathrm{m} & 0^\circ \end{bmatrix}^\mathsf{T}$ and the final position is $\boldsymbol{y}_\mathrm{f} = \begin{bmatrix} 15\,\mathrm{m} & 6\,\mathrm{m} & 0^\circ \end{bmatrix}^\mathsf{T}$, while the transition time is $t_\mathrm{f} = 15\,\mathrm{s}$. The timing law is again chosen according to equation (4.9). The trajectory is visualized in Fig. 4.17 and the experimental results are shown in Fig. 4.18. Pure feedforward control does not detect the initial position error and the error Δx_T is present over the complete course of the trajectory, see Fig. 4.18(a). Both controllers with feedback reduce the initial position error Δx_T. The LQR computes a large correcting control input u_T at the start of the trajectory to quickly reduce the position error, see Fig. 4.18(b) and Fig. 4.18(c). However, this jerk motion results in a sway motion, which is slowly reduced during the trajectory, see Fig. 4.18(e). On the other hand, the SMC shows a slower response to the initial error, see Fig. 4.18(b) and Fig. 4.18(c). Therefore, less sway motion is induced compared to the LQR, see Fig. 4.18(e). However, the overall convergence to the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ is slower and the trajectory error $e(t)$ is larger compared to the LQR.

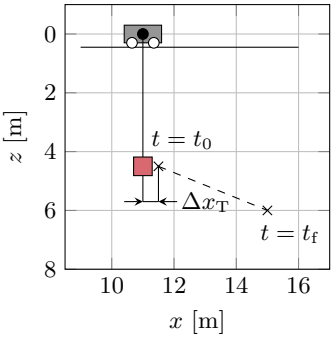
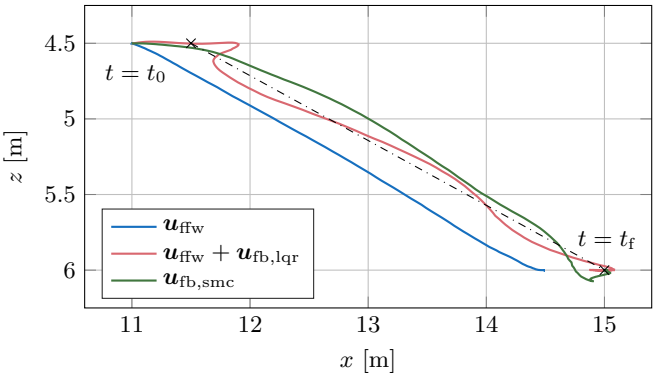
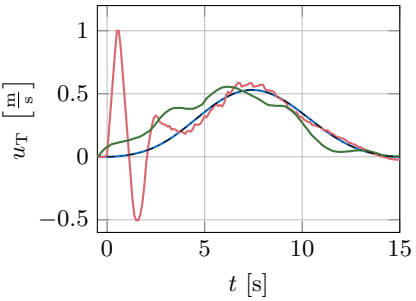


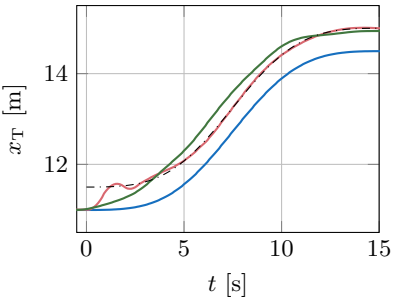
Figure 4.17: Visualization of the desired straight trajectory with initial error.



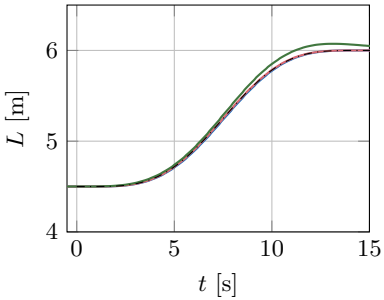
(a) System output $\boldsymbol{z}$.



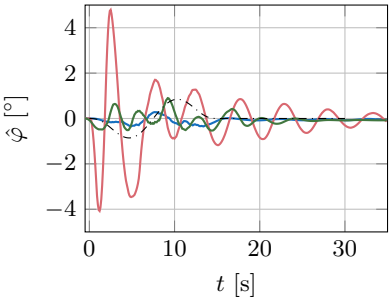
(b) System input u_T.



(c) Trolley position x_T.



(d) Cable length L.



(e) Estimated sway angle $\hat{\varphi}$.

Figure 4.18: Experimental results with initial position error $\Delta x_\mathrm{T} = 0.5\,\mathrm{m}$.

4.2.8 Experimental Results for Control of the Platform Orientation

Experimental results are presented for control of the complex cable robot model with variable platform orientation and with platform mass $m_{\mathrm{P},2}$. The respective cable robot model has four degrees of freedom and is described by equations (2.58), (2.59) and (2.61). First, the feedforward control is analyzed. Afterwards, the DAE-based LQR controller, as presented in Sec. 3.4.1 is applied to control the platform orientation.

Feedforward Control of the Platform Orientation

For the cable robot model with orientation control, the system output $\boldsymbol{z}$ is given by equation (2.65) and it contains the platform orientation δ_{P}. It might be unclear how to choose a certain reference trajectory $\boldsymbol{z}_{\mathrm{d}}(t)$ for the platform orientation δ_{P}. Therefore, it is demonstrated how servo-constraints can also represent an implicit condition. Instead of prescribing the platform orientation directly, an implicit condition is derived to enforce zero relative acceleration for an object lying on top of the platform. The following results are presented according to [OttoSeifried17].
Figure 4.19 shows an object lying on the platform. The inertial coordinate system is K: $\{\mathrm{O}, x, y, z\}$, while the coordinate system $\mathrm{K}' : \{\mathrm{O}', x', y', z'\}$ is fixed in the center of gravity of the platform and the coordinate system $\mathrm{K}'' : \{\mathrm{O}'', x'', y'', z''\}$ is fixed in the center of gravity of the small object.

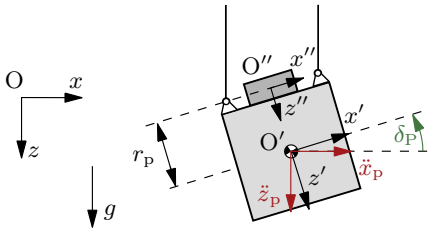


Figure 4.19: Geometry for object lying on the platform.

Enforcing the relative acceleration $\boldsymbol{a}_{\mathrm{O}''}$ of the object to be zero and deriving the acceleration in the body-fixed coordinate system $\mathrm{K}'' : \{\mathrm{O}'', x'', y'', z''\}$ yields

$$\boldsymbol{a}_{\mathrm{O}''}(\boldsymbol{y}) = \frac{\boldsymbol{F}_{\mathrm{a}}(\boldsymbol{y})}{m} - \boldsymbol{a}_{\mathrm{F}}(\boldsymbol{y}) - \boldsymbol{a}_{\mathrm{C}}(\boldsymbol{y}) = \begin{bmatrix} \ddot{x}_{\mathrm{O}''} \\ \ddot{y}_{\mathrm{O}''} \\ \ddot{z}_{\mathrm{O}''} \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.11}$$

with the global acceleration $\boldsymbol{a}_{\mathrm{F}} : \mathbb{R}^3 \to \mathbb{R}^3$, the Coriolis acceleration $\boldsymbol{a}_{\mathrm{C}} : \mathbb{R}^3 \to \mathbb{R}^3$ and the applied forces $\boldsymbol{F}_{\mathrm{a}} : \mathbb{R}^3 \to \mathbb{R}^3$. Substituting all the terms and assuming a small distance r_{P} between O' and O'' results in the implicit condition

$$\ddot{x}_{\mathrm{P}} = \tan\left(\delta_{\mathrm{P}}\right) \left(-g + \ddot{z}_{\mathrm{P}}\right) \tag{4.12}$$

for the platform orientation δ_P. Thereby, g denotes the gravitational constant. The orientation angle is adjusted such that an object lying on the platform is not accelerated relative to the platform. The implicit condition (4.12) is added to the two original servo-constraints enforcing the position of the center of gravity of the platform. Experimental results for enforcing condition (4.12) are presented in the following. The initial position is

$$\boldsymbol{y}_0 = \begin{bmatrix} x_{\mathrm{T},0} & L_{1,0} & L_{2,0} & x_{\mathrm{P},0} & z_{\mathrm{P},0} & \delta_{\mathrm{P},0} \end{bmatrix}^\mathsf{T} \tag{4.13}$$

$$= \begin{bmatrix} 16\,\mathrm{m} & 5.5\,\mathrm{m} & 5.5\,\mathrm{m} & 16\,\mathrm{m} & 5.68\,\mathrm{m} & 0\,^\circ \end{bmatrix}^\mathsf{T} . \tag{4.14}$$

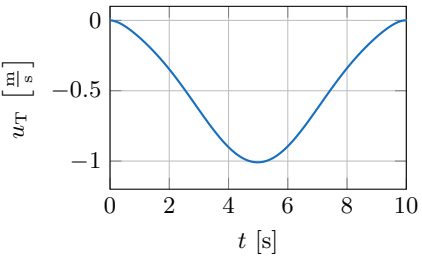
The reference trajectory $\boldsymbol{z}_\mathrm{d}(t)$ for the center of gravity of the platform is a horizontal transition with transition time $t_\mathrm{f} = 10\,\mathrm{s}$, while the timing law is defined by equation (4.10). The final position is

$$\boldsymbol{y}_\mathrm{f} = \begin{bmatrix} x_{\mathrm{T,f}} & L_{1,f} & L_{2,f} & x_{\mathrm{P,f}} & z_{\mathrm{P,f}} & \delta_{\mathrm{P,f}} \end{bmatrix}^\mathsf{T} \tag{4.15}$$

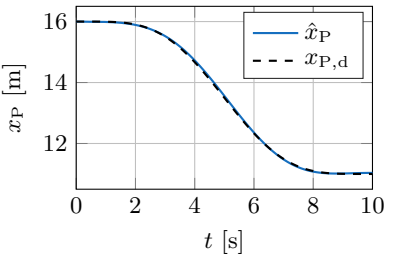
$$= \begin{bmatrix} 11\,\mathrm{m} & 5.5\,\mathrm{m} & 5.5\,\mathrm{m} & 11\,\mathrm{m} & 5.68\,\mathrm{m} & 0\,^\circ \end{bmatrix}^\mathsf{T} . \tag{4.16}$$

The platform orientation δ_P is adjusted implicitly by condition (4.12). In total, $n_\mathrm{u} = 3$ servo-constraints constrain the motion of the system to compute the $n_\mathrm{u} = 3$ system inputs. Experimental results are shown in Fig. 4.20 for applying feedforward control based on solving the inverse model DAEs. The system input u_T of the trolley is shown in Fig. 4.20(a) and the respective motion x_P of the platform is shown in Fig. 4.20(b). The smooth transition between the initial position $x_{\mathrm{P},0} = 16\,\mathrm{m}$ to the final position $x_{\mathrm{P,f}} = 11\,\mathrm{m}$ is evident. The system inputs $u_{\mathrm{L},i}$ for the cable lengths are shown in Fig. 4.20(d). The cable lengths are adjusted such that the orientation angle reaches $\delta_\mathrm{P} \approx 3\,^\circ$ during the acceleration phase and $\delta_\mathrm{P} \approx -3\,^\circ$ in the deceleration phase. The platform is tilted such that an object stays on top of the platform. The desired zero relative motion of the virtual object is enforced. The desired platform orientation $\delta_{\mathrm{P,d}}$ is compared to the measured IMU data in Fig. 4.20(c). There is a good agreement between both values. The desired horizontal acceleration $\ddot{x}_{\mathrm{P,d}}$ is also compared to the measured IMU data in Fig. 4.20(e). There is again a good agreement between both values. Figure 4.21 shows snapshots of a video taken during the experiment. Thereby, a negative rotation of the platform is visible during the acceleration phase on the left hand side and a position rotation is visible during the deceleration on the right hand side of the figure.

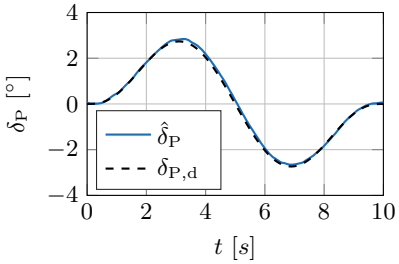
In conclusion, the experimental results show the effective application of servo-constraints to an underactuated multibody system described by DAEs as well as the application of an implicit servo-constraint.



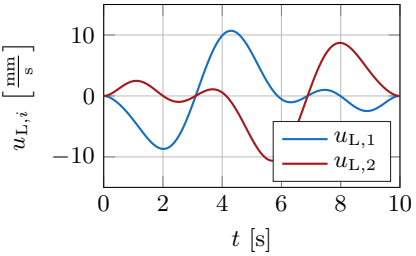
(a) System input u_{T}.



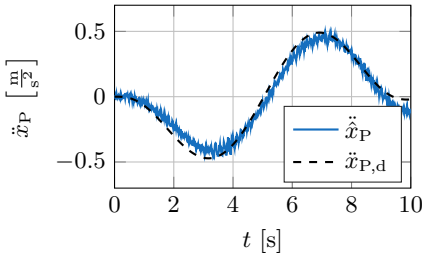
(b) Platform position x_{P}.



(c) Platform orientation δ_{P}.



(d) System input $u_{\mathrm{L},i}$.



(e) Platform acceleration $\ddot{x}_{\mathrm{P}}$.

Figure 4.20: Experimental results for a horizontal trajectory with platform orientation control.

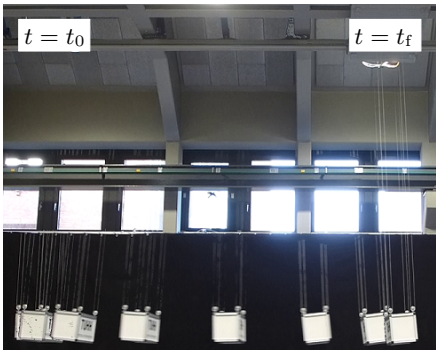


Figure 4.21: Video snapshots of feedforward control of the platform orientation.

Feedback Control of the Platform Orientation

Application of a feedback controller for platform orientation control is demonstrated in the following. Since the model is only available in DAE form, a DAE-based controller and observer are implemented, see Sec. 3.4.1 and Sec. 4.2.3 respectively. For details of the implementation refer to [OttoRückwaldSeifried19]. For feedback control of the platform orientation, the trolley position is held constant, while a constant desired angle $\delta_{\mathrm{P,d}} = 5^\circ$ is prescribed for the platform orientation. Initially, the platform is horizontal with $\delta_{\mathrm{P,d}} = 0^\circ$. The complete vector of initial conditions is

$$\boldsymbol{y}_0 = \begin{bmatrix} x_{\mathrm{T},0} & L_{1,0} & L_{2,0} & x_{\mathrm{P},0} & z_{\mathrm{P},0} & \delta_{\mathrm{P},0} \end{bmatrix}^{\mathsf{T}} \tag{4.17}$$

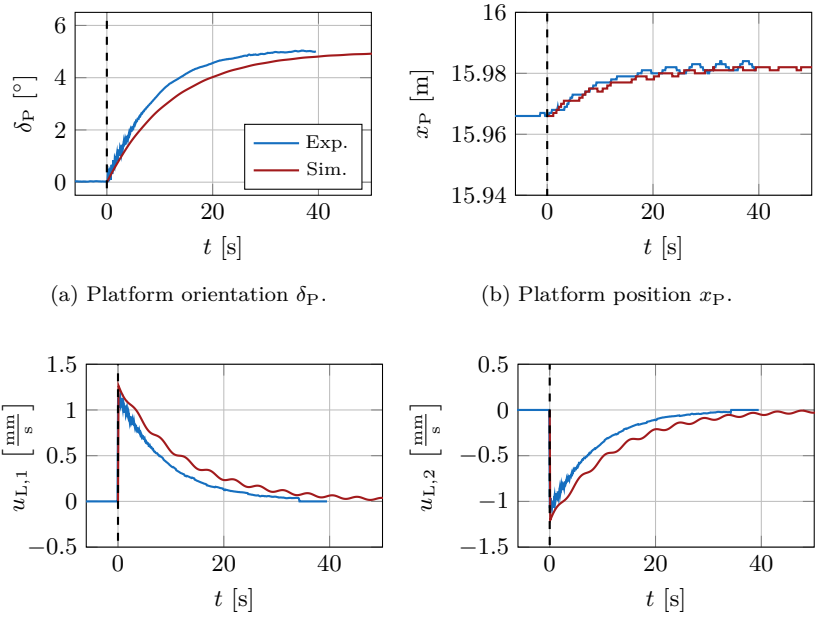
$$= \begin{bmatrix} 16\,\mathrm{m} & 5.35\,\mathrm{m} & 5.35\,\mathrm{m} & 16\,\mathrm{m} & 5.53\,\mathrm{m} & 0\,^\circ \end{bmatrix}^{\mathsf{T}} . \tag{4.18}$$

The weighting matrices of the LQR for the linearization point $\boldsymbol{y}_0$ are chosen as

$$\boldsymbol{Q}_{\mathrm{lqr}} = \mathrm{diag}(1.87, 7.95, 6.03, 2.58, 2.03, 0.20, 1.59, 6.15, 2.68, 1.23) , \tag{4.19}$$

$$\boldsymbol{R}_{\mathrm{lqr}} = \mathrm{diag}(8000, 8000) . \tag{4.20}$$

Note that the weight on the control input is considerably high, such that a slow motion on the experimental setup is achieved. This avoids dangerous oscillations in the cables during the experiments. Simulation and experimental results are shown in Fig. 4.22. The desired orientation $\delta_{\mathrm{P,d}}$ is reached after approximately 45 s in the simulation, see Fig. 4.22(a). This is achieved by actuating the cables in opposite directions, see Fig. 4.22(c) and 4.22(d). The variation in cable length changes the horizontal position x_{P} of the platform, see Fig. 4.22(b). In the experiments, the reference angle is reached after approximately 30 s and therefore faster than in simulation. However, the general behavior is reproduced in the simulation. Figure 4.23 shows snapshots of the video during the experiments.



(a) Platform orientation δ_P.

(b) Platform position x_P.

(c) System input $u_{L,1}$.

(d) System input $u_{L,2}$.

Figure 4.22: Experimental and simulation results for platform orientation control of the cable robot [OttoRückwaldSeifried19].

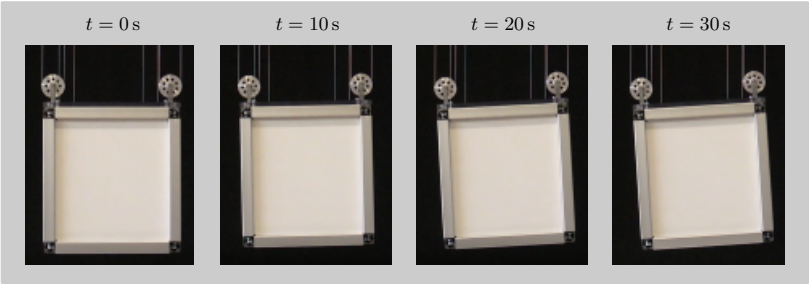
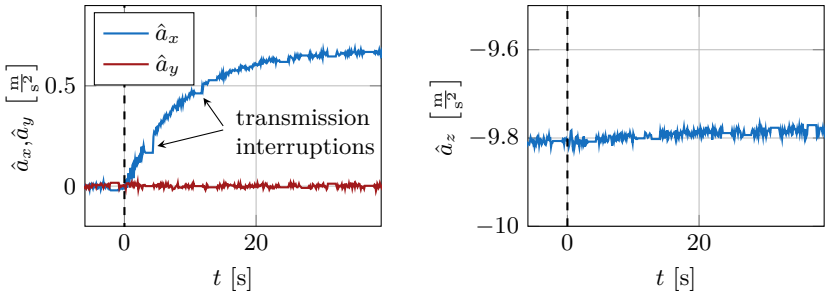


Figure 4.23: Video snapshots of platform orientation control [OttoRückwaldSeifried19].

Measurements of the IMU form the basis for estimation of the platform orientation δ_{P}. Its acceleration measurements are displayed in Fig. 4.24. The acceleration measurements in static equilibrium are evaluated to compute the true final orientation $\delta_{\mathrm{P,true}}$ with

$$\delta_{\mathrm{P,true}} = \arctan\left(-\frac{\hat{a}_x}{\hat{a}_z}\right) \tag{4.21}$$

with $\hat{a}_x$ and $\hat{a}_z$ denoting the measured acceleration in x- and z-direction respectively. An evaluation shows that the true final orientation angle is $\delta_{\mathrm{P,true}} = 4°$ in contrast to the desired angle $\delta_{\mathrm{P,d}} = 5°$. Further tuning of the UKF weighting matrices is necessary to obtain a more accurate estimation of the platform orientation. Note also that the IMU transmits the data by a wireless connection. Transmission interruptions occur sometimes, as can be seen by the constant lines at $t = 3.5\,\mathrm{s}$ or $t = 11\,\mathrm{s}$ in Fig. 4.24. These defects can also decrease the filter accuracy. See [OttoRückwaldSeifried19] for additional results including a comparison of a DAE-based and ODE-based controller and simulations for trajectory control including feedback control of the platform orientation.



(a) Accelerations $\hat{a}_x$ and $\hat{a}_y$.

(b) Acceleration $\hat{a}_z$.

Figure 4.24: IMU measurements for platform orientation control.

Chapter 5

Minimum Phase Systems

While some underactuated mechanical systems have a differentially flat output, there are also systems with internal dynamics. Here, the notion of systems with internal dynamics implies that the system is not differentially flat with respect to the chosen system output. There might be other outputs that render the system differentially flat with respect to the new output. However, this is not considered here. Typical mechanical systems with internal dynamics are flexible link and flexible joint manipulators used for end-effector tracking [Seifried14, deLucaBook08]. Usually, flexible link manipulators are non-minimum phase systems for homogeneous materials, while flexible joint manipulators are minimum phase in case damping effects are taken into account. The joints are usually modeled as torsional oscillators which are introduced in Sec. 2.4.2. For neglectable damping, the flexible joint manipulators are differentially flat, such as the torsional oscillator analyzed in Sec. 4.1. Physical interpretation of the internal dynamics of linear elastic systems is given in [Miu91] for translational mass-spring systems and simple continuous elastic beams. It is shown that the system zeros relate to energy propagation properties of the flexible systems.

From the point of view of nonlinear control theory, input-output linearization is an efficient control strategy for systems with internal dynamics [Isidori96, Sastry99]. The ideas are specified and extended for underactuated mechanical systems in [Spong94]. Input-output linearization is for example applied to viscoelastic joint manipulators in [deLucaFarinaLucibello05] or for two-link walking robots in [CelikovskyZikmundMoog08]. However, input-output linearization is known to have robustness issues because of the combined feedforward and feedback structure [SlotineLi91]. Moreover, input-output linearization is based on burdensome analytical derivations introduced in Sec. 3.1.3. Thus, the method is difficult to apply to complex multibody systems with many input and output channels.

Servo-constraints are a suitable alternative, since less analytical derivations are necessary to obtain a feedforward controller. Most literature on servo-constraints focuses on differentially flat systems as discussed in Chap. 4. However, recent results are reported for minimum phase systems. For example, the mass distribution of flexible link manipulators is optimized to yield stable internal dynamics in [Seifried14]. These minimum phase systems are then inverted using servo-constraints. Flexible joint and flexible link manipulators are also considered in [MobergHanssen10]. A comparison of integration methods for minimum

phase systems is presented in [OttoSeifried18a]. A minimum phase design of a flexible link manipulator is inverted and the feedforward control is combined with funnel feedback control in [BergerEtAl18]. This combination shows that control input peaks in the feedback channel can be effectively reduced by adding feedforward control. Switching between two different servo-constraints periodically is proposed in [BencsikKovácsZelei17] in order to control the motion of the internal dynamics for a certain type of cable robot. A stability analysis of a control strategy directly combining servo-constraints with error feedback is performed in [KovácsBencsik12] for the same robot.
In this chapter, minimum phase systems are considered, i.e. their inverse models can be solved by forward time integration. Due to the arising internal dynamics, the solvers of the inverse model DAEs need to be chosen more careful compared to differentially flat systems. This is shown in the following and in [OttoSeifried18a, Seifried14]. Damping properties of the solvers must be taken into account, since an accurate representation of the internal dynamics is important for accurate feedforward control. In the following, the internal dynamics of the mass-on-car system is analyzed analytically for different possible configurations and solvers are compared with respect to accuracy. Moreover, a three-dimensional manipulator with one passive joint is considered in minimum phase configuration. This example demonstrates the application of servo-constraints for complex three-dimensional systems with internal dynamics.

5.1 Mass-on-car System

The mass-on-car system introduced in Sec. 2.4.3 is considered in the following. Due to the simple equations of motion, the internal dynamics can be derived explicitly. It is shown that the internal dynamics can be a combination of first order and second order dynamics and that the relative degree changes based on the system configuration. The explicit internal dynamics can be integrated directly to obtain the feedforward control input, see Sec. 3.1.3. Therefore, the system can pose as a benchmark problem for the inversion of minimum phase systems using servo-constraints. Simulation results are shown to demonstrate the necessity of using higher order integration schemes for minimum phase systems.

5.1.1 System Properties

In the following, the system properties are analyzed and the internal dynamics is given explicitly for the mass-on-car system. The system properties vary with the inclination angle α of the third mass and with damping d_i for $i = 1, 2$. Details of the derivations can be found in [OttoSeifried18b].
The system has $n_\mathrm{f} = 3$ degrees of freedom and $n_\mathrm{u} = 1$ system input. It is therefore underactuated with two unactuated coordinates s_1 and s_2 which describe the motion of the second and third mass respectively. The system output is chosen as the horizontal position of the third mass and is given by equation (2.87).

The system properties are evaluated according to Sec. 3.2.2. Evaluation of the coupling matrix $\boldsymbol{P}(\boldsymbol{y}, t)$ shows that it is $P = 0$ for all inclination angles α. This result is in contrast to the original system without the added mass m_1, which is analyzed in [SeifriedBlajer13]. Thus, the inverse model is always in tangential realization. For stability analysis, the internal dynamics is derived explicitly based on a coordinate transformation similar to the one described in Sec. 3.1.3. The coordinate transformation is here directly stated for the second order dynamics, see [Seifried14] for details. The output z is chosen as part of the new coordinate vector $\bar{\boldsymbol{y}}$ and the positions s_1 and s_2 complete the vector, such that

$$\bar{\boldsymbol{y}} = \begin{bmatrix} z \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_1 + s_1 + s_2 \cos\alpha \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cos\alpha \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{y}\,. \tag{5.1}$$

Note that in this case, the coordinate transformation is linear and can be applied in a straightforward manner. The system dynamics in the new coordinates $\bar{\boldsymbol{y}}$, $\bar{\boldsymbol{v}}$ follows as

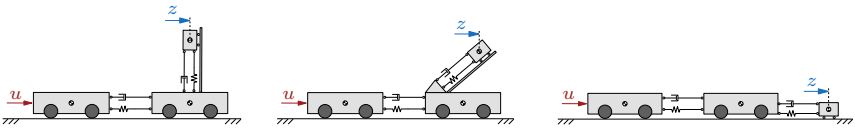
$$\dot{\bar{\boldsymbol{y}}} = \begin{bmatrix} \dot{z} \\ \dot{s}_1 \\ \dot{s}_2 \end{bmatrix} = \bar{\boldsymbol{v}} \tag{5.2}$$

$$\dot{\bar{\boldsymbol{v}}} = \begin{bmatrix} -\dfrac{m_3\, F_{\mathrm{sd},1}\, \sin^2(\alpha) + m_2\, F_{\mathrm{sd},2}\, \cos(\alpha)}{m_3\big(m_2 + m_3\, \sin^2(\alpha)\big)} \\ \dfrac{F_{\mathrm{sd},2}\, \cos(\alpha)}{m_2 + m_3 \sin^2(\alpha)} - \dfrac{F_{\mathrm{sd},1}\,(m_1 + m_2 + m_3 \sin^2(\alpha))}{m_1\,\big(m_2 + m_3 \sin^2(\alpha)\big)} \\ \dfrac{F_{\mathrm{sd},1}\, \cos(\alpha)}{m_2 + m_3 \sin^2(\alpha)} - \dfrac{(m_2 + m_3)\, F_{\mathrm{sd},2}}{m_3\big(m_2 + m_3 \sin^2(\alpha)\big)} \end{bmatrix} + \begin{bmatrix} 0 \\ -\dfrac{1}{m_1} \\ 0 \end{bmatrix} u\,, \tag{5.3}$$

with the spring-damper forces

$$F_{\mathrm{sd},1} = d_1\, \dot{s}_1 + k_1\, s_1 \quad \text{and} \quad F_{\mathrm{sd},2} = d_2\, \dot{s}_2 + k_2\, s_2\,. \tag{5.4}$$

The transformed equations (5.2)–(5.3) enable the derivation of the internal dynamics and the relative degree. The system has different properties depending on the system parameters and the following cases are distinguished. The cases 1.1 and 1.2 describe configurations with angle $\alpha = 90\,^\circ$, the cases 2.1 and 2.2 describe configurations with angle $0\,^\circ < \alpha < 90\,^\circ$ and the cases 3.1 and 3.2 describe configurations with angle $\alpha = 90\,^\circ$. Thereby, the cases $\square.1$ denote finite damping $d_i > 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$, while the cases $\square.2$ denote neglectable damping $d_i = 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$. Figure 5.1 visualizes the considered configurations.



case 1.1: $d_i > 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$ case 2.1: $d_i > 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$ case 3.1: $d_i > 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$

case 1.2: $d_i = 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$ case 2.2: $d_i = 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$ case 3.2: $d_i = 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$

Figure 5.1: Considered configurations of the mass-on-car system.

Case 1.1 The inclination angle is set to $\alpha = 90\,^\circ$ and damping $d_i > 0 \frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$ is considered. Setting the angle to $\alpha = 90\,^\circ$ decouples the motion of body 3 from the first two bodies and the system output reduces to $z = x_1 + s_1$. Thus, the motion of body 3 cannot be observed from the output. Analyzing the second part of equation (5.3) gives the relation

$$u = -m_1 \ddot{s}_1 - \frac{m_1 + m_2 + m_3}{m_2 + m_3} F_{\mathrm{sd},1}\,, \tag{5.5}$$

with input u depending on $\ddot{s}_1$. From the first part of equation (5.3) follows that $\dot{s}_1$ is a function of the output acceleration $\ddot{z}$, which can be differentiated once. Thus, the third derivative of the output z influences the input u and the system is of relative degree $r = 3$ and internal dynamics of dimension $2n_\mathrm{f} - r = 3$ remains. The states representing the internal dynamics can be chosen as $\boldsymbol{\eta} = \begin{bmatrix} s_1 & s_2 & \dot{s}_2 \end{bmatrix}^\mathsf{T}$. The internal dynamics is determined from the first and third part of equation (5.3) as

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} -\frac{k_1}{d_1} s_1 - \frac{m_2 + m_3}{d_1} \ddot{z} \\ \dot{s}_2 \\ -\frac{d_2 \dot{s}_2 + k_2 s_2}{m_3} \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{k_1}{d_1} & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{k_2}{m_3} & -\frac{d_2}{m_3} \end{bmatrix}}_{\boldsymbol{A}_{1.1}} \boldsymbol{\eta} + \boldsymbol{f}_{1.1}(\ddot{z})\,. \tag{5.6}$$

The internal dynamics consists of first order dynamics for the state s_1 and second order dynamics for the state s_2. The internal dynamics is driven by the output acceleration $\ddot{z}$, which is collected in the function $\boldsymbol{f}_{1.1} : \mathbb{R} \to \mathbb{R}^3$. Stability of the internal dynamics is analyzed as zero dynamics with $z = \dot{z} = \ddot{z} = 0 \;\forall\; t$ and therefore $\boldsymbol{f}_{1.1}(0) = \boldsymbol{0}$. Thus, stability of the linear zero dynamics can be

determined by the eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{1.1}}$ of the matrix $\boldsymbol{A}_{1.1}$. The eigenvalues are

$$\boldsymbol{\lambda}_{\mathrm{A}_{1.1}} = \begin{bmatrix} -\dfrac{k_1}{d_1} \\ -\dfrac{d_2 + \sqrt{d_2^2 - 4k_2m_3}}{2\,m_3} \\ -\dfrac{d_2 - \sqrt{d_2^2 - 4k_2m_3}}{2\,m_3} \end{bmatrix} \tag{5.7}$$

and have negative real parts for all $d_i > 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$. Thus, the zero dynamics is asymptotically stable for case 1.1.

Case 1.2 In this case, the angle is again set to $\alpha = 90\,^\circ$, but damping is neglected with $d_i = 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$. Therefore, the first part of equation (5.3) reduces to an algebraic relationship between the output acceleration $\ddot{z}$ and the coordinate s_1, while equation (5.5) still holds. Thus, the relative degree is $r = 4$. The internal dynamics is now second order dynamics with $\boldsymbol{\eta} = \begin{bmatrix} s_2 & \dot{s}_2 \end{bmatrix}^\mathsf{T}$ and is given by

$$\dot{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\dfrac{k_2}{m_3} & 0 \end{bmatrix}}_{\boldsymbol{A}_{1.2}} \boldsymbol{\eta} + \boldsymbol{f}_{1.2}(\ddot{z})\,. \tag{5.8}$$

The eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{1.2}}$ of the matrix $\boldsymbol{A}_{1.2}$ are

$$\boldsymbol{\lambda}_{\mathrm{A}_{1.2}} = \begin{bmatrix} \sqrt{\dfrac{-k_2}{m_3}} \\ -\sqrt{\dfrac{-k_2}{m_3}} \end{bmatrix} \tag{5.9}$$

and lie completely on the imaginary axis. Therefore, the internal dynamics is marginally stable for case 1.2.

Case 2.1 For case 2.1 with angle $0\,^\circ < \alpha < 90\,^\circ$ and damping $d_i > 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$, the motion of body 3 is coupled to the motion of bodies 1 and 2. From the second part of equation (5.3), the relation

$$u = -m_1\ddot{s}_1 + \frac{m_1 F_{\mathrm{sd},2}\cos(\alpha) - F_{\mathrm{sd},1}\left(m_1 + m_2 + m_3\sin^2(\alpha)\right)}{m_2 + m_3\sin^2(\alpha)} \tag{5.10}$$

can be obtained, which shows that the input u is again a function of $\ddot{s}_1$. Moreover, from the first part of equation (5.3) follows that $\dot{s}_1$ depends on $\ddot{z}$. Thus, the relative degree is $r = 3$. Choosing again the states of the internal dynamics as $\boldsymbol{\eta} = \begin{bmatrix} s_1 & s_2 & \dot{s}_2 \end{bmatrix}^\mathsf{T}$, yields the internal dynamics from the first and third part of equation (5.3) as

$$\dot{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} -\dfrac{k_1}{d_1} & -\dfrac{k_2\, m_2 \cos(\alpha)}{d_1\, m_3 \sin^2(\alpha)} & -\dfrac{d_2\, m_2 \cos(\alpha)}{d_1\, m_3 \sin^2(\alpha)} \\ 0 & 0 & 1 \\ 0 & -\dfrac{k_2}{m_3 \sin^2(\alpha)} & -\dfrac{d_2}{m_3 \sin^2(\alpha)} \end{bmatrix}}_{\boldsymbol{A}_{2.1}} \boldsymbol{\eta} + \boldsymbol{f}_{2.1}(\ddot{z})\,. \tag{5.11}$$

The internal dynamics consists of first order dynamics in s_1 and second order dynamics in s_2. In contrast to case 1.1, the motion of both coordinates is coupled. Stability of the zero dynamics (5.11) is determined by the eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{2.1}}$ of the state space matrix $\boldsymbol{A}_{2.1}$ with

$$\boldsymbol{\lambda}_{\mathrm{A}_{2.1}} = \begin{bmatrix} -\dfrac{k_1}{d_1} \\ -\dfrac{d_2 + \sqrt{d_2^2 - 4\, k_2 m_3 \sin^2(\alpha)}}{2\, m_3 \sin^2(\alpha)} \\ -\dfrac{d_2 - \sqrt{d_2^2 - 4\, k_2 m_3 \sin^2(\alpha)}}{2\, m_3 \sin^2(\alpha)} \end{bmatrix}\,. \tag{5.12}$$

The eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{2.1}}$ have negative real parts for all $d_i > 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$. Therefore, the internal dynamics is asymptotically stable for case 2.1.

Case 2.2 In this case, the angle is set to $0\,^\circ < \alpha < 90\,^\circ$ and damping is neglected with $d_i = 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$. Equation (5.10) still holds, but the first part of equation (5.3) shows that s_1 depends directly on the output acceleration $\ddot{z}$. Thus, the system has relative degree $r = 4$. Choosing the states $\boldsymbol{\eta} = \begin{bmatrix} s_2 & \dot{s}_2 \end{bmatrix}^\mathsf{T}$ yields the internal dynamics as

$$\dot{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\dfrac{k_2}{m_3 \sin^2(\alpha)} & 0 \end{bmatrix}}_{\boldsymbol{A}_{2.2}} \boldsymbol{\eta} + \boldsymbol{f}_{2.2}(\ddot{z})\,. \tag{5.13}$$

The internal dynamics consists of second order dynamics in the coordinate s_2. The eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{2.2}}$ of the matrix $\boldsymbol{A}_{2.2}$ are

$$\boldsymbol{\lambda}_{\mathrm{A}_{2.2}} = \begin{bmatrix} -\dfrac{\sqrt{-k_2}}{\sqrt{m_3}\sin(\alpha)} \\ \\ \dfrac{\sqrt{-k_2}}{\sqrt{m_3}\sin(\alpha)} \end{bmatrix} \tag{5.14}$$

and lie on the imaginary axis. Thus, the zero dynamics is marginally stable for case 2.2.

Case 3.1 For the angle $\alpha = 0\,^\circ$ and damping $d_i > 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ with $i = 1, 2$, the system reduces to a three-mass-damper chain, which is similar to the one introduced in Sec. 2.4.2 for rotational degrees of freedom. Thus, the results for the internal dynamics are similar to Sec. 2.4.2 and are briefly summarized here. For a three-mass chain, the relative degree is $r = 4$ and internal dynamics of dimension 2 remains. The coordinates of the internal dynamics are chosen as $\boldsymbol{\eta} = \begin{bmatrix} s_1 & s_2 \end{bmatrix}^\mathsf{T}$ and the internal dynamics is

$$\dot{\boldsymbol{\eta}} = \underbrace{\begin{bmatrix} -\dfrac{k_1}{d_1} & \dfrac{k_2^2\, m_2}{d_1\, d_2^2} \\ \\ 0 & -\dfrac{k_2}{d_2} \end{bmatrix}}_{\boldsymbol{A}_{3.1}} \boldsymbol{\eta} + \boldsymbol{f}_{3.1}(\dddot{z})\,. \tag{5.15}$$

The internal dynamics consists of coupled first order dynamics in the coordinates s_1 and s_2. Analyzing stability in terms of the zero dynamics, the eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{3.1}}$ of the matrix $\boldsymbol{A}_{3.1}$ are

$$\boldsymbol{\lambda}_{\mathrm{A}_{3.1}} = \begin{bmatrix} -\dfrac{k_1}{d_1} \\ \\ -\dfrac{k_2}{d_2} \end{bmatrix}\,. \tag{5.16}$$

The eigenvalues have negative real parts for all $d_i > 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$ and the internal dynamics is asymptotically stable for case 3.1.

Case 3.2 For angle $\alpha = 0\,^\circ$ and neglectable damping $d_i = 0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$ for $i = 1, 2$, there is no internal dynamics. The system has relative degree $r = 6$ and is therefore differentially flat. This case is analyzed and simulated in Sec. 4.1 for the torsional oscillator.

Summary The preceding analysis demonstrates that the same system can have fundamentally different characteristics in terms of the internal dynamics. The relative degree of the mass-on-car system ranges from $r = 3$ to $r = 6$. The internal dynamics consists of either first order or second order dynamics or a coupling of both. The different cases are summarized in Table 5.1. Such a detailed analysis can pose as a benchmark case, since the explicit derivation of the internal dynamics enables a direct integration of the internal dynamics and therefore the inverse model, see Sec. 3.1.3. The solution can then be compared to the solution of the inverse model stated with servo-constraints.

Table 5.1: Overview of different configurations of the mass-on-car system.

Case	Angle	Damping ($i = 1, 2$)	Relative degree r	DAE index	Dimension int. dyn.
1.1	$\alpha = 90°$	$d_i > 0$	3	4	3
1.2	$\alpha = 90°$	$d_i = 0$	4	5	2
2.2	$0° < \alpha < 90°$	$d_i > 0$	3	4	3
2.2	$0° < \alpha < 90°$	$d_i = 0$	4	5	2
3.1	$\alpha = 0°$	$d_i > 0$	4	5	2
3.2	$\alpha = 0°$	$d_i = 0$	6	7	0

5.1.2 Simulation Results

The mass-on-car system is simulated in the following. First, the root locus plot visualizes the eigenvalues of the zero dynamics derived in Sec. 5.1.1. Afterwards, simulations demonstrate numerical damping of the implicit Euler scheme for minimum phase systems. The simulation parameters of the mass-on-car system are given in Tab. 5.2. The system is simulated for case 2.
For the root locus plot, the eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}_{2.1}}$ and $\boldsymbol{\lambda}_{\mathrm{A}_{2.2}}$ given by equations (5.12) and (5.14) are shown in Fig. 5.2. For case 2.1 with damping $d_1 = d_2 = 1\,\frac{\mathrm{N\,s}}{\mathrm{m}}$, they have negative real parts. While the first eigenvalue is constant at $\lambda_{\mathrm{A}_{2.1},1} = -\frac{k_1}{d_1}$, the other two eigenvalues $\lambda_{\mathrm{A}_{2.1},2}$ and $\lambda_{\mathrm{A}_{2.1},3}$ depend on the damping d_2 and the angle α. The system is lightly damped for $d_2 < 2\sqrt{k_2\, m_3}\sin\alpha = \hat{d}_2$ and the eigenvalues are complex conjugates. For larger damping $d_2 > \hat{d}_2$, the system is strongly damped and the eigenvalues are real numbers. For a given value d_2,

Table 5.2: Simulation parameters for the mass-on-car system.

Parameter	m_1	m_2	m_3	$k_1 = k_2$	$d_1 = d_2$	α
Value	1 kg	1 kg	3 kg	$3\,\frac{\mathrm{N}}{\mathrm{m}}$	$0\,\frac{\mathrm{N\,s}}{\mathrm{m}}$	15 °

the value $\hat{\alpha} = \arcsin\left(\frac{d_2}{2\sqrt{k_2 m_3}}\right)$ denotes this fundamental change of behavior. For the chosen parameters, it is $\hat{\alpha} = 9.6\,^\circ$. As the angle α approaches $0\,^\circ$, the eigenvalues tend to

$$\lim_{\alpha\to 0} \lambda_{\mathrm{A}_{2.1},2} = -\infty \quad \text{and} \quad \lim_{\alpha\to 0} \lambda_{\mathrm{A}_{2.1},3} = -\frac{k_2}{d_2} \tag{5.17}$$

and as the angle α approaches $90\,^\circ$, the eigenvalues tend to

$$\lim_{\alpha\to 90^\circ} \lambda_{\mathrm{A}_{2.1},2} = -\frac{d_2 + \sqrt{d_2^2 - 4k_2 m_3}}{2m_3}\,, \tag{5.18}$$

$$\lim_{\alpha\to 90^\circ} \lambda_{\mathrm{A}_{2.1},3} = -\frac{d_2 - \sqrt{d_2^2 - 4k_2 m_3}}{2m_3}\,. \tag{5.19}$$

For case 2.2 without damping, the two eigenvalues $\lambda_{\mathrm{A}_{2.2},1}$ and $\lambda_{\mathrm{A}_{2.2},2}$ lie on the imaginary axis. As the angle α approaches $0\,^\circ$, the eigenvalues tend to

$$\lim_{\alpha\to 0} \lambda_{\mathrm{A}_{2.2},1} = -\infty\mathrm{i} \quad \text{and} \quad \lim_{\alpha\to 0} \lambda_{\mathrm{A}_{2.2},2} = \infty\mathrm{i} \tag{5.20}$$

and as the angle α approaches $90\,^\circ$, the eigenvalues tend to

$$\lim_{\alpha\to 90^\circ} \lambda_{\mathrm{A}_{2.2},1} = -\sqrt{\frac{-k_2}{m_3}} \quad \text{and} \quad \lim_{\alpha\to 90^\circ} \lambda_{\mathrm{A}_{2.2},2} = \sqrt{\frac{-k_2}{m_3}}\,. \tag{5.21}$$

The root locus plot visualizes the eigenvalues and thus the different properties of the zero dynamics. For the chosen parameters in Tab. 5.2, the eigenvalues of the undamped internal dynamics are $\lambda_{\mathrm{A}_{2.2},1} = -3.9\mathrm{i}\,\frac{1}{\mathrm{s}}$ and $\lambda_{\mathrm{A}_{2.2},2} = 3.9\mathrm{i}\,\frac{1}{\mathrm{s}}$.

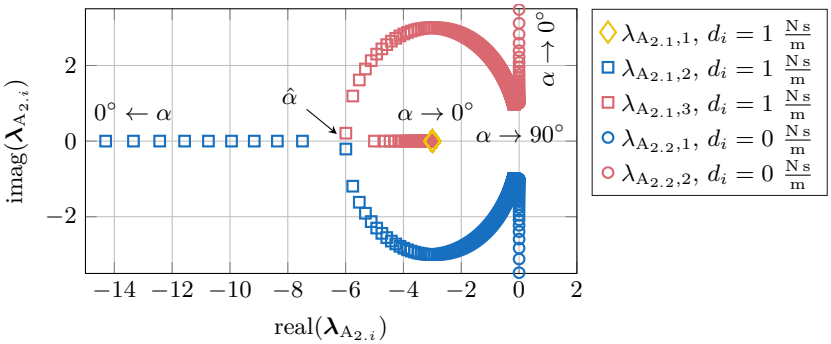
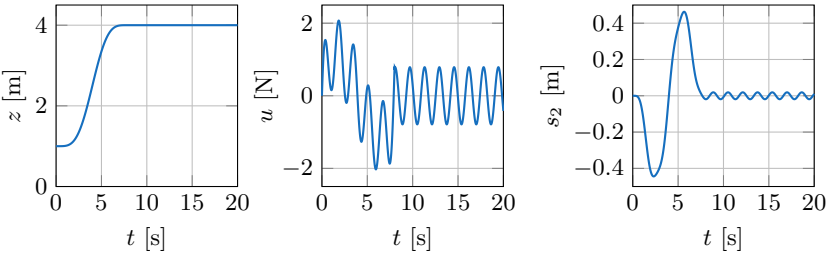


Figure 5.2: Eigenvalues $\boldsymbol{\lambda}_{\mathrm{A}2.i}$ of the internal dynamics for cases 2.1 and 2.2.

In the following, the influence of the integrator on the feedforward control based on servo-constraints is investigated using case 2.2. In this case, the system has internal dynamics but no physical damping. Thus, numerical damping properties of the solvers can be analyzed. The trajectory is defined as a smooth transition from the initial output $z_0 = 1\,\mathrm{m}$ to the final position $z_\mathrm{f} = 4\,\mathrm{m}$ with transition time $t_\mathrm{f} = 8\,\mathrm{s}$. The timing law is chosen as the polynomial given by equation (4.10). The desired trajectory and a solution of the inverse model problem is shown in Fig. 5.3. Since the inverse model DAEs are of differentiation index 5, the index is reduced to 3 using the projection approach introduced in Sec. 3.2.5. The system is solved using the BDF scheme with $k_\mathrm{bdf} = 3$, which is introduced in Sec. 3.3.3. The undamped internal dynamics in the coordinate s_2 is given by equation (5.13). It shows an oscillation after the end of the trajectory at time $t_\mathrm{f} = 8\,\mathrm{s}$, see Fig. 5.3(c). This oscillation is counteracted by the input u in order to keep the output z constant, see Figs. 5.3(b) and 5.3(a).



(a) Desired output $z_\mathrm{d}(t)$. (b) Desired input u_ffw. (c) Internal dynamics s_2.

Figure 5.3: Simulation results for model inversion of the mass-on-car system for case 2.2.

Accurate integration of the inverse model DAEs is necessary for accurate tracking. However, numerical solvers with strong numerical damping in the low frequency range are not able to compute the dynamics accurately. This is demonstrated by solving the inverse model in this form with the implicit Euler and the Radau5 scheme with step size $\Delta t = 10\,\mathrm{ms}$. Figure 5.4 visualizes the computed system input. The implicit Euler strongly damps the low-frequency internal dynamics and therefore cannot accurately compute the desired system input necessary to keep the output constant. In contrast, the Radau5 scheme has much less numerical damping in this frequency range and correctly reflects the undamped internal dynamics.

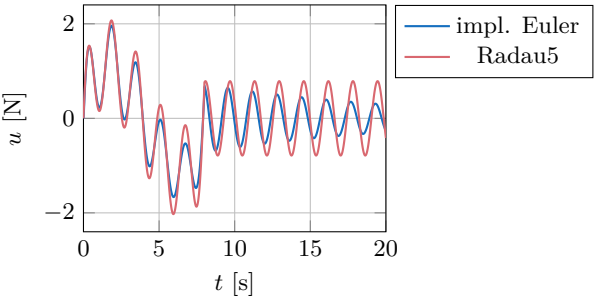


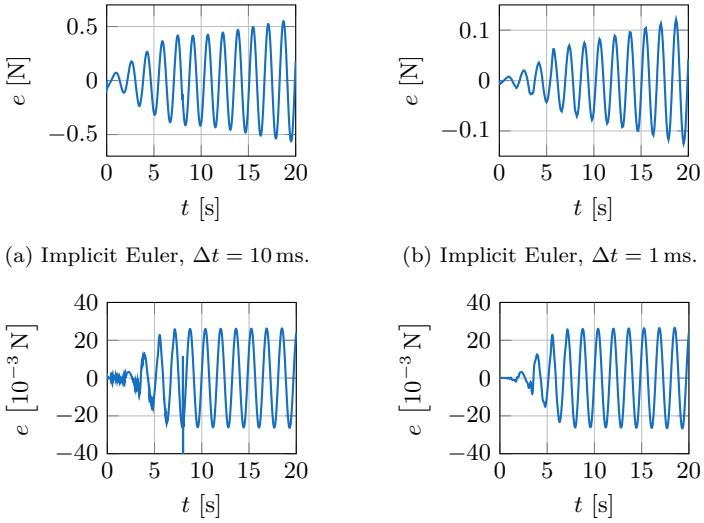
Figure 5.4: Desired input u_{ffw} computed with $\Delta t = 10\,\mathrm{ms}$ for case 2.2.

In the following, the numerical error $e(t)$ is computed by

$$e(t) = u_{\mathrm{ffw}}(t) - u_{\mathrm{ref}}(t) \tag{5.22}$$

with u_{ffw} denoting the solution from the inverse model DAEs and u_{ref} denoting the solution obtained from directly integrating the internal dynamics of equation (5.13) with the MATLAB solver *ode45*. The numerical error $e(t)$ is shown for different integration schemes in Fig. 5.5. The implicit Euler scheme strongly damps out the low-frequency dynamics and the error $e(t)$ grows linearly in time. Reducing the step size to $\Delta t = 1\,\mathrm{ms}$ reduces the error but not the general behavior, see Fig. 5.5(b). The maximum error is $e_{\max} \approx 0.5\,\mathrm{N}$ and $e_{\max} \approx 0.11\,\mathrm{N}$ for $\Delta t = 10\,\mathrm{ms}$ and $\Delta t = 1\,\mathrm{ms}$, see Fig. 5.5(a) and 5.5(b) respectively. In contrast, both the 3-step BDF scheme as well as the Radau5 scheme with step size $\Delta t = 10\,\mathrm{ms}$ yield similar numerical errors $e(t)$ with a maximum error of approximately $e_{\max} \approx 0.02\,\mathrm{N}$. The errors are considerably smaller compared to the error obtained by the implicit Euler scheme. Thus, the oscillation of the internal dynamics is not damped as strongly, see Fig. 5.5(c) and 5.5(d) respectively.

In conclusion, integration methods must be chosen carefully for the inversion of minimum phase systems based on servo-constraints. A system analysis and stability analysis of the internal dynamics helps to determine the system properties, such as the relative degree, the dimension and order of the internal dynamics and its eigenvalues. This can facilitate solver selection. For minimum phase systems, integration schemes with low numerical damping in the low frequency range should be preferred over the simple implicit Euler scheme. Then, the internal dynamics can be reflected accurately and tracking errors can be avoided.



(a) Implicit Euler, $\Delta t = 10\,\text{ms}$.

(b) Implicit Euler, $\Delta t = 1\,\text{ms}$.

(c) BDF with $k_{\text{bdf}} = 3$, $\Delta t = 10\,\text{ms}$.

(d) Radau5, $\Delta t = 10\,\text{ms}$.

Figure 5.5: Numerical error $e(t)$ for different integration schemes.

5.2 Three-dimensional Manipulator with one Passive Joint

The three-dimensional manipulator introduced in Sec. 2.4.4 is considered in the following. Based on this example, the application of servo-constraints to a rather complex MIMO system with internal dynamics is demonstrated. The considered manipulator has $n_{\text{u}} = 3$ system inputs. Thus, a number of three coordinates can be controlled, such as the position of the end-effector in three-dimensional space. After analyzing the system properties, simulation results demonstrate the real-time capabilities of the presented approach.

5.2.1 System Properties

The three-dimensional manipulator with one passive joint has $n_{\text{f}} = 4$ degrees of freedom and $n_{\text{u}} = 3$ system inputs. Thus, it is underactuated and the angle γ of the passive joint describes the unactuated degree of freedom.
The system properties are analyzed symbolically according to Sec. 3.1.3. The vector relative degree of the system is computed as $\boldsymbol{r} = \{2, 2, 2\}$. Due to $\Sigma r_i = 6 < 2n_{\text{f}} = 8$, internal dynamics of dimension 2 remain. The second order internal dynamics can be derived in terms of the angle γ for a linearized system output. This is shown in [Seifried11, Seifried12] for a planer manipulator with one passive joint. Stability analysis of the internal dynamics is here

performed based on the generalized eigenvalue problem presented in Sec. 3.2.3. Thus, no explicit derivation of the internal dynamics is required. When choosing model parameters which correspond to a homogeneous mass distribution of link 3 and 4, one system zero is in the right half plane. The system is therefore non-minimum phase. A particle swarm optimization of the model parameters is performed in [Seifried11, Seifried12] to obtain stable internal dynamics. This involves adding counter-weights to link 4, which shift the center of gravity of the unactuated link and alter its moment of inertia. The optimized mass distribution alters the dynamic behavior and shifts both system zeros to the left half plane. The system zeros $\bar{z}$ of the homogeneous mass distribution and the optimized mass distribution are visualized in Fig. 5.6 for the parameters listed in Tab. 5.3. Thereby, the stiffness c of the passive joint is varied from $c = 5\ \frac{\mathrm{N\,m}}{\mathrm{rad}}$ to $c = 100\ \frac{\mathrm{N\,m}}{\mathrm{rad}}$. A variation of the stiffness does not influence the stability of the internal dynamics, but its transient behavior. In the following, simulations are carried out with the optimized parameters, yielding a minimum phase system.

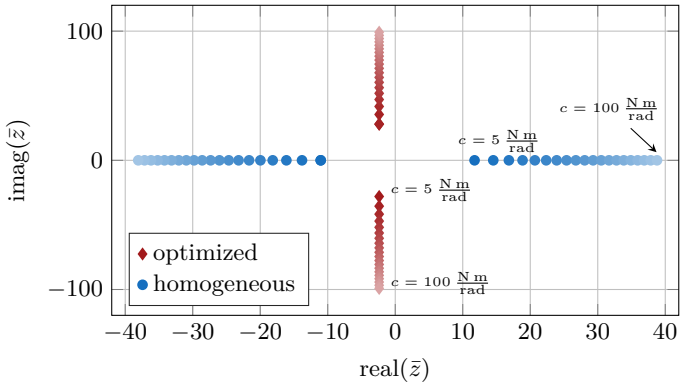


Figure 5.6: System zeros $\bar{z}$ of the three-dimensional manipulator.

Table 5.3: Optimized simulation parameters of the three-dimensional manipulator according to [Seifried11, Seifried14].

body	m_i [kg]	L_i [m]	I_{ix}, I_{iy} [kgm^2]	I_{iz} [kgm^2]
1	6.876	1	0.5773	0.0086
2	6.876	1	0.5773	0.0086
3	3.438	0.5	0.0738	0.0043
4	3.969	0.5	0.2449	0.005
spring-damper		$c = 50\ \frac{\mathrm{N\,m}}{\mathrm{rad}}$	$d = 0.05\ \frac{\mathrm{N\,m\,s}}{\mathrm{rad}}$	

5.2.2 Simulation Results

Simulations are performed in the following to show the real-time capabilities of the servo-constraints approach for a complex three-dimensional MIMO system. The optimized model parameters yielding a minimum phase system are taken from [Seifried11] and are listed in Tab. 5.3. The system output $\boldsymbol{z}$ is the end-effector position in the inertial coordinate system K: $\{\mathrm{O}, x, y, z\}$ and is given by equation (2.106). The desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ is defined as a circle in a plane parallel to the x, y-plane with center $\boldsymbol{p}_\mathrm{c}$ and initial point $\boldsymbol{p}_0$ on the circle. The circular path is parameterized by the scalar arc length $\sigma(t)$ and is given by

$$\boldsymbol{z}_\mathrm{d}(\sigma(t)) = \boldsymbol{p}_\mathrm{c} + \rho\,\sigma_\mathrm{t}\,\boldsymbol{S}_{\mathrm{KK^c}} \begin{bmatrix} \cos\left(\frac{\sigma(t)}{\rho}\right) \\ \sin\left(\frac{\sigma(t)}{\rho}\right) \\ 0 \end{bmatrix}, \quad \sigma(t) \in \left[0, \quad \sigma_\mathrm{t}\right], \tag{5.23}$$

with radius $\rho = ||\boldsymbol{p}_\mathrm{c} - \boldsymbol{p}_0||$ and the rotation matrix $\boldsymbol{S}_{\mathrm{KK^c}}$ transforming the local coordinate system $\mathrm{K^c} : \{\mathrm{O^c}, x^\mathrm{c}, y^\mathrm{c}, z^\mathrm{c}\}$ to the inertial coordinate system K: $\{\mathrm{O}, x, y, z\}$. The local coordinate system is defined by the axis x^c pointing from the center $\boldsymbol{p}_\mathrm{c}$ to the initial point $\boldsymbol{p}_0$, the axis z^c parallel to the inertial axis z and the axis y^c is computed to yield a right hand coordinate system, see e.g. [SicilianoEtAl09]. The total arc length σ_t is the diameter of the circle with $\sigma_\mathrm{t} = 2\pi\rho$. The timing law for the scalar arc length $\sigma(t)$ is given by equation (4.10).
For the simulations, the initial state vector $\boldsymbol{y}_0$ of the manipulator is chosen as $\boldsymbol{y}_0 = \begin{bmatrix}\Omega_0 & \alpha_0 & \beta_0 & \gamma_0\end{bmatrix}^\mathsf{T} = \begin{bmatrix}0 & 10 & 160 & 0\end{bmatrix}^\mathsf{T}\,^\circ$. This yields an initial end-effector position $\boldsymbol{p}_0 = \begin{bmatrix}0.35 & 0 & 1\end{bmatrix}^\mathsf{T}$ m. The center of the circle is chosen as $\boldsymbol{p}_\mathrm{c} = \begin{bmatrix}0.65 & 0.7 & 1\end{bmatrix}^\mathsf{T}$ m and the transition time is $t_\mathrm{f} = 2\,\mathrm{s}$ to yield an aggressive maneuver. Figure 5.7 shows the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$ and initial position $\boldsymbol{y}_0$.

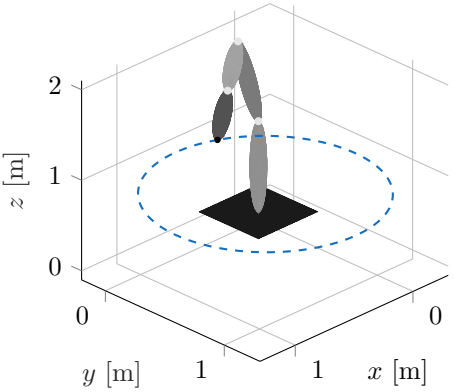
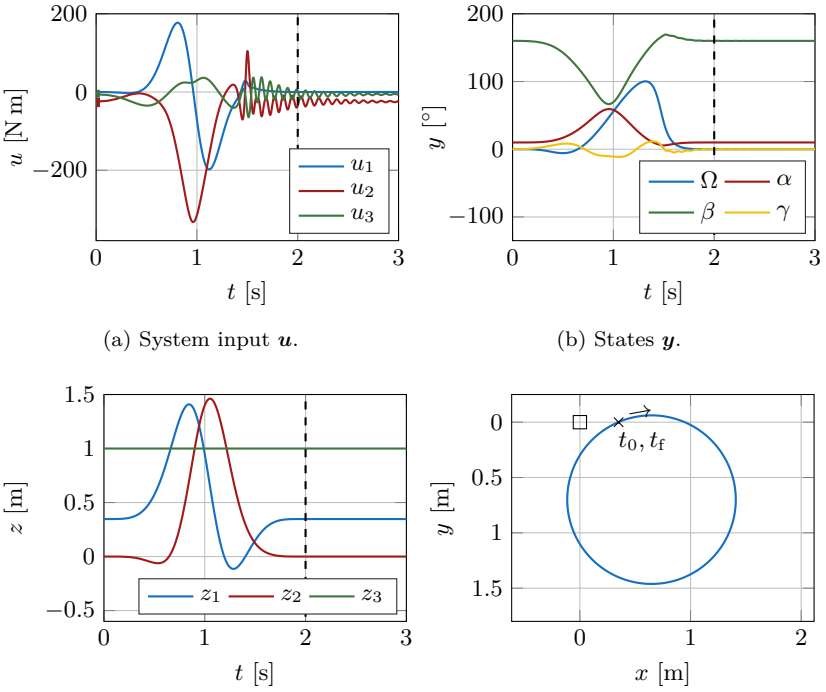


Figure 5.7: Visualization of the three-dimensional manipulator in initial position $\boldsymbol{y}_0$ and the desired trajectory $\boldsymbol{z}_\mathrm{d}(t)$.

The servo-constraints problem is set up and solved with the BDF solvers introduced in Sec. 3.3.3. The results of the model inversion are shown in Fig. 5.8 for a step size $\Delta t = 1\,\mathrm{ms}$ and $k_{\mathrm{bdf}} = 2$. Note that a small step size of $\Delta t = 1\,\mathrm{ms}$ is necessary due to stiff equations and high-frequency oscillations in the solution. Figure 5.8(a) shows the computed system input $\boldsymbol{u}_{\mathrm{ffw}}$ over time. Oscillations of the system input are present at the end of the trajectory at $t = 2\,\mathrm{s}$. These oscillations compensate the motion of the internal dynamics. The corresponding system states $\boldsymbol{y}$ are shown in Fig. 5.8(b). The oscillations are also present in the states and are shown in more detail below. Figure 5.8(c) shows the evolution of the system output $\boldsymbol{z}$ over time, while the trajectory is shown in the x, y-plane in Fig. 5.8(d). Thereby, the symbol □ marks the base position of the manipulator. The desired trajectory is tracked perfectly and the system output is completely at rest at the end of the trajectory, since the motion of the internal dynamics is not observable from the output.



(a) System input $\boldsymbol{u}$.

(b) States $\boldsymbol{y}$.

(c) System output $\boldsymbol{z}$ over time.

(d) System output $\boldsymbol{z}$ in the x, y-plane.

Figure 5.8: Inversion results of the three-dimensional manipulator for a circular trajectory.

In order to show the motion of the internal dynamics in more detail, Fig. 5.9 shows the unactuated coordinate γ, which describes the motion of the internal dynamics. The solution is shown for the BDF integrator with $k_{\mathrm{bdf}} = 2$ and the implicit Euler scheme with $k_{\mathrm{bdf}} = 1$. The implicit Euler strongly damps the low frequency oscillation of the internal dynamics. The oscillation can not be reflected accurately and the corresponding system input computed with the implicit Euler cannot guarantee perfect tracking. Therefore, higher order integration schemes with less numerical damping in the low frequency range should be preferred to compute the inverse model of systems with internal dynamics.

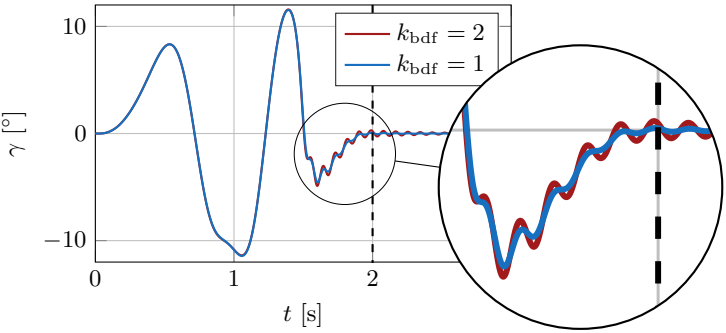
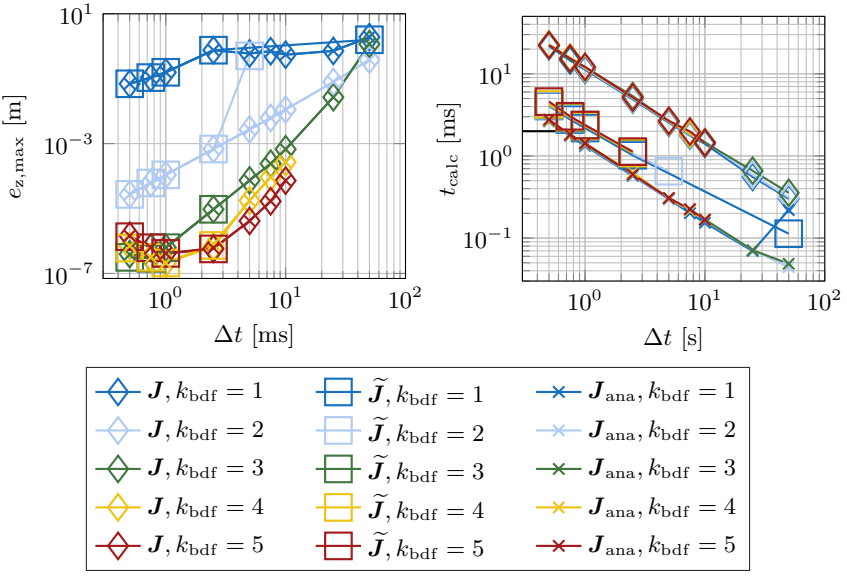


Figure 5.9: Numerical results of the internal dynamics γ.

In order to analyze real-time capabilities, the inverse model problem is solved using different BDF schemes with $k_{\mathrm{bdf}} = 1, 2, \ldots 5$ and the total computation time t_{calc} of the inversion is measured. Various schemes to evaluate the Jacobian of the problem are applied. These include the analytic Jacobian $\boldsymbol{J}_{\mathrm{ana}}$, the approximated Jacobian $\widetilde{\boldsymbol{J}}$ using Broyden's method and the numerical Jacobian $\boldsymbol{J}$ computed with finite differences, see Sec. 3.3.4. After computing the inverse model, the resulting feedforward control input $\boldsymbol{u}_{\mathrm{ffw}}$ is applied to the model in a forward simulation, using the MATLAB *ode15s* integrator. The maximum tracking error $e_{\mathrm{z,max}}$ of the output is computed with

$$e_{\mathrm{z,max}} = \max_{t} \|\boldsymbol{z}_{\mathrm{sim}}(t) - \boldsymbol{z}_{\mathrm{d}}(t)\| \tag{5.24}$$

based on the simulated system output $\boldsymbol{z}_{\mathrm{sim}}$. The convergence results for the error $e_{\mathrm{z,max}}$ are presented in Fig. 5.10(a) and the respective total computation time t_{calc} is presented in Fig. 5.10(b). The total simulation time of $t_{\mathrm{f}} = 2\,\mathrm{s}$ is denoted by the horizontal line to denote the real-time barrier. The total computation time t_{calc} must at least be lower than 2 s to enable a computation in real-time. Then, the computation time of each time step must be verified to fulfill the real-time criterion (3.92), see Sec. 3.3.4.
The convergence of the error $e_{\mathrm{z,max}}$ in Fig. 5.10(a) reflects the theoretical convergence order results for the BDF integrators, see Sec. 3.3.3. The implicit Euler



(a) Convergence. (b) Total computation time.

Figure 5.10: Convergence of the BDF solvers for the three-dimensional manipulator.

with $k_{\mathrm{bdf}} = 1$ converges with first order and the tracking errors $e_{\mathrm{z,max}}$ are rather large due to the numerical damping demonstrated above. The BDF schemes with $k_{\mathrm{bdf}} = 2$ and $k_{\mathrm{bdf}} = 3$ show decreasing errors for step sizes $\Delta t \geq 1\,\mathrm{ms}$. For smaller step sizes, numerical errors start to influence the solution. For $k_{\mathrm{bdf}} > 3$, the problem starts to get ill-conditioned. A solution can only be calculated for step sizes $\Delta t \leq 10\,\mathrm{ms}$, while rounding errors start to influence the solution at $\Delta t \approx 1\,\mathrm{ms}$.

All Jacobian schemes yield nearly similar errors $e_{\mathrm{z,max}}$ for each BDF scheme respectively. However, note that Jacobians based on Broyden's method do not result in a solution for the larger time steps. This is reflected in the absence of the respective markers for nearly all $\Delta t \geq 5\,\mathrm{ms}$. For these cases, the method does not converge and no results are given. This is due to an inaccurate estimation of the Jacobian and the maximum Newton iterations of $n_{\mathrm{N,max}} = 20$.

Considering the timing results in Fig. 5.10(b) shows that for a selected Jacobian method, all BDF schemes result in similar computation times t_{calc}. Thereby, the analytic Jacobian yields the smallest computation time and Broyden's method yields slightly larger computation times. The numerical Jacobian results in the longest computation times, which are approximately 8 times larger compared

to the analytical Jacobian. Therefore, the BDF schemes using the numerical Jacobian $\boldsymbol{J}$ are only real-time capable for step sizes $\Delta t \geq 10\,\mathrm{ms}$. The BDF schemes using Broyden's Jacobian $\widetilde{\boldsymbol{J}}$ are real-time capable for step sizes $\Delta t \geq 2.5\,\mathrm{ms}$ and the analytical Jacobian $\boldsymbol{J}_{\mathrm{ana}}$ is real-time capable for step sizes $\Delta t \geq 0.75\,\mathrm{ms}$. Thus, Broyden's method can be a convenient alternative if the analytical Jacobian is not available due to system complexity. Depending on the desired accuracy, it might be sufficient to compute the Jacobian numerically and use larger step sizes. But if higher accuracy is required, Broyden's method or the analytical Jacobian are required to ensure real-time computation of the inverse model for the three-dimensional manipulator.

In conclusion, the servo-constraints approach may be applied to rather complex MIMO multibody systems with internal dynamics. Real-time computation of the inverse model is possible if the numerical computation of the Jacobian can be avoided by applying e.g. Broyden's method or an analytical derivation. These simulation results hint that the model inversion can be implemented on a test bench of a minimum phase three-dimensional robot. However, experimental data is necessary for further validation of the real-time applicability.

Non-minimum Phase Systems

Non-minimum phase systems possess unstable internal dynamics. Therefore, integration forward in time of the inverse model yields unbounded states and system inputs.

Different approaches exist for the inversion of non-minimum phase systems and they are summarized in the following. The system output can be redefined, such that the new system output yields stable internal dynamics, while approximate tracking of the original output is possible [SlotineLi91]. Such output relocation is applied to flexible manipulators in [MorlockBurkhardtSeifried16, RaoufMohamadMaarouf13]. Alternatively, it is proposed to optimize system parameters such as mass, inertia and the center of gravity, to obtain stable internal dynamics [Seifried12]. The optimized parameters can be realized e.g. by adding counterweights. Deriving an exact inverse model for the original non-minimum phase system is proposed by [ChenPaden96, DevasiaChenPaden96] in terms of stable inversion. For this purpose, a bounded solution to the inverse model problem is obtained by solving a boundary value problem (BVP). The originally proposed scheme formulates the BVP for the internal dynamics which have to be derived explicitly in ODE form, such as shown in Sec. 3.1.3. It is shown in [BrülsBastosSeifried13] that stable inversion can also be formulated in terms of the inverse model described by servo-constraints. Stable inversion yields a non-causal system input, which includes pre- and postactuation phases before and after the time horizon of the output trajectory. Alternatively, it is proposed in [GraichenHagenmeyerZeitz05] to modify the BVP, such that the desired output trajectory is slightly varied instead of extending the time horizon.

Stable inversion involves the definition of boundary conditions based on the stable and unstable eigenspaces of the linearized zero dynamics. It is burdensome to derive the boundary conditions, since a state transformation must be found to transform the system into input-output normal form, see Sec. 3.1.3. Therefore, systems with few degrees of freedom are usually considered in literature in the context of stable inversion. The manipulator in [BastosSeifriedBrüls17] has one passive joint, while the manipulators in [BastosSeifriedBrüls13, deLucaPanzieriUlivi98, Seifried12] have two passive joints. However, these simple models might not be sufficient to accurately reflect the behavior of flexible systems. Recently, first results are presented in [BurkhardtSeifriedEberhard15] and [Seifried14] for manipulators with 6 and 18 unactuated degrees of freedom respectively. Thereby, the floating frame of reference approach is applied to model the flexibility.

It is shown in [BastosSeifriedBrüls13] that solving the boundary value problem can be avoided by formulating an equivalent optimization problem. In that case, there is no need to derive the boundary conditions explicitly. Both approaches are compared in [BastosSeifriedBrüls17]. It is shown that the solution of the optimization problem converges to the solution of the boundary value problem if the considered time horizon goes to infinity.
In the following, stable inversion is introduced and the various solving approaches are discussed. Moreover, an approximation of the boundary conditions is proposed to simplify the setup of the BVP. Convergence of the simplification is analyzed for systems with linear scalar second order internal dynamics. Afterwards, stable inversion is applied to a manipulator with one passive joint and the different approaches are compared with respect to accuracy and efficiency. With the newly introduced simplified boundary conditions, it is possible to apply stable inversion to highly flexible manipulators. As an application example, this is demonstrated for a manipulator modeled by the ANCF. Despite the simplification, the setup of the BVP still yields some disadvantages, such as the definition on an initial guess and long computation time. Hence, a systematic output redefinition procedure is proposed for the flexible manipulator to render the internal dynamics stable. This enables integration forward in time of the inverse model DAEs. Thus, no initial guess is necessary and real-time computation is possible.

6.1 Stable Inversion

In the following, stable inversion is introduced. First, the original formulation is presented, which depends on deriving the stable and unstable eigenspaces of the linearized zero dynamics for the definition of the boundary conditions. Afterwards, simplifying boundary conditions are proposed, which do not depend on the derivation of the internal dynamics. Finally, the reformulation as an optimization problem is introduced, which completely omits the definition of boundary conditions.

6.1.1 Original Formulation

In case the internal dynamics (3.25) are unstable, it is not possible to integrate the inverse model forward in time. This applies to the analytical inverse model given by equations (3.24) and (3.25) as well as the inverse model given by the servo-constraints DAEs (3.36)–(3.39). Stable inversion is proposed in [ChenPaden96] in order compute a bounded solution of the inversion problem by solving a BVP. It is applicable for hyperbolic equilibria of the internal dynamics. Stable inversion is originally stated for the explicit internal dynamics (3.25) in [ChenPaden96] and is extended for the inverse model DAEs (3.36)–(3.39) in [BrülsBastosSeifried13].
Concerning the original formulation based on the explicit internal dynamics in ODE form (3.25), the boundary conditions are defined, such that the initial state

starts on the unstable manifold W_0^{U} of the initial equilibrium point $\boldsymbol{\eta}_{\mathrm{eq},0}$. The final state reaches the final equilibrium point $\boldsymbol{\eta}_{\mathrm{eq,f}}$ on its stable manifold $W_{\mathrm{f}}^{\mathrm{S}}$. The stable and unstable manifolds are locally approximated by stable and unstable eigenspaces E^{S} and E^{U} of the linearized zero dynamics. Thus, the boundary conditions are given by

$$\boldsymbol{B}_{\mathrm{u}}^{\mathrm{ode}} \left(\boldsymbol{\eta}(T_0) - \boldsymbol{\eta}_{\mathrm{eq},0} \right) = 0 \tag{6.1}$$

$$\boldsymbol{B}_{\mathrm{s}}^{\mathrm{ode}} \left(\boldsymbol{\eta}(T_{\mathrm{f}}) - \boldsymbol{\eta}_{\mathrm{eq,f}} \right) = 0 \,. \tag{6.2}$$

Thereby, T_0 and T_{f} denote the initial and final simulation time. The matrices $\boldsymbol{B}_{\mathrm{s}}^{\mathrm{ode}} \in \mathbb{R}^{n^{\mathrm{s}} \times (2n_{\mathrm{f}} - r)}$ and $\boldsymbol{B}_{\mathrm{u}}^{\mathrm{ode}} \in \mathbb{R}^{n^{\mathrm{u}} \times (2n_{\mathrm{f}} - r)}$ contain the n^{s} stable and n^{u} unstable eigenvectors of the zeros dynamics respectively. With n_{f} degrees of freedom and the relative degree r, the dimension of the internal dynamics is $2n_{\mathrm{f}} - r$ and it holds $n^{\mathrm{s}} + n^{\mathrm{u}} = 2n_{\mathrm{f}} - r$. The boundary value problem yields a non-causal solution in the sense that the system input $\boldsymbol{u}_{\mathrm{ffw}}$ induces motion into the system before the start t_0 of the trajectory. Moreover, there is a system input after the end t_{f} of the trajectory to bring the internal dynamics to its equilibrium. In order to capture these pre- and postactuation phases correctly, the simulation time interval $[T_0, T_{\mathrm{f}}]$ is chosen larger than the interval $[t_0, t_{\mathrm{f}}]$ of the trajectory. Therefore holds $T_0 \leq t_0$ and $T_{\mathrm{f}} \geq t_{\mathrm{f}}$. The stable inversion BVP consists of the dynamics (3.25) subject to the constraints (6.1)–(6.2).
Equivalently, the problem can be formulated for the inverse model based on servo-constraints [BrülsBastosSeifried13]. The boundary conditions are given by

$$\boldsymbol{B}_{\mathrm{u}}^{\mathrm{dae}} \left(\boldsymbol{x}(T_0) - \boldsymbol{x}_{\mathrm{eq},0} \right) = 0 \tag{6.3}$$

$$\boldsymbol{B}_{\mathrm{s}}^{\mathrm{dae}} \left(\boldsymbol{x}(T_{\mathrm{f}}) - \boldsymbol{x}_{\mathrm{eq,f}} \right) = 0 \tag{6.4}$$

with the matrices $\boldsymbol{B}_{\mathrm{s}}^{\mathrm{dae}} \in \mathbb{R}^{n^{\mathrm{s}} \times (2n_{\mathrm{f}} + 3n_{\mathrm{c}} + n_{\mathrm{u}})}$ and $\boldsymbol{B}_{\mathrm{u}}^{\mathrm{dae}} \in \mathbb{R}^{n^{\mathrm{u}} \times (2n_{\mathrm{f}} + 3n_{\mathrm{c}} + n_{\mathrm{u}})}$ and with $\boldsymbol{x}$ collecting all unknown variables $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{y}^{\mathsf{T}} & \boldsymbol{v}^{\mathsf{T}} & \boldsymbol{\lambda}^{\mathsf{T}} & \boldsymbol{u}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$ of the inverse model DAEs (3.36)–(3.39). With n_{c} constraints and n_{u} system inputs, the number of boundary conditions matches again the number of unkowns of the BVP and it holds $n^{\mathrm{s}} + n^{\mathrm{u}} = 2n_{\mathrm{f}} + 3n_{\mathrm{c}} + n_{\mathrm{u}}$. The stable inversion BVP consists of the dynamics (3.36)–(3.39) subject to the constraints (6.3)–(6.4). Deriving the boundary conditions (6.1)–(6.2) or (6.3)–(6.4) is not straightforward in practice since they depend on the explicit derivation of the internal dynamics.

6.1.2 Simplified Boundary Conditions

It is proposed in [DrückerSeifried20] to simplify the above boundary conditions (6.1)–(6.2) or (6.3)–(6.4). Instead of enforcing the states to lie on the stable and unstable eigenspaces of the equilibrium point, it is proposed to constrain part of the state vector to the initial and final equilibrium point respectively.

For the internal dynamics described in ODE form (3.25), the simplified boundary conditions are

$$\boldsymbol{L}_0^{\mathrm{ode}}\,\boldsymbol{\eta}\,(T_0) = \boldsymbol{L}_0^{\mathrm{ode}}\,\boldsymbol{\eta}_{\mathrm{eq},0} \tag{6.5}$$

$$\boldsymbol{L}_{\mathrm{f}}^{\mathrm{ode}}\,\boldsymbol{\eta}\,(T_{\mathrm{f}}) = \boldsymbol{L}_{\mathrm{f}}^{\mathrm{ode}}\,\boldsymbol{\eta}_{\mathrm{eq,f}}\,. \tag{6.6}$$

The matrices $\boldsymbol{L}_0^{\mathrm{ode}} \in \mathbb{R}^{n^{\mathrm{a}}\times(2n_{\mathrm{f}}-r)}$ and $\boldsymbol{L}_{\mathrm{f}}^{\mathrm{ode}} \in \mathbb{R}^{n^{\mathrm{b}}\times(2n_{\mathrm{f}}-r)}$ select n^{a} states to be equal to the initial equilibrium $\boldsymbol{\eta}_{\mathrm{eq},0}$ and n^{b} states to be equal to the final equilibrium $\boldsymbol{\eta}_{\mathrm{eq,f}}$. Thereby it is $n^{\mathrm{b}} > 0$ and $n^{\mathrm{a}} > 0$. In total, a number of $n^{\mathrm{a}} + n^{\mathrm{b}} = 2n_{\mathrm{f}} - r$ conditions are given, which equals the number of unknowns. Equivalently for the inverse model described by the DAEs (3.36)–(3.39), the simplified boundary conditions are proposed as

$$\boldsymbol{L}_0^{\mathrm{dae}}\,\boldsymbol{x}\,(T_0) = \boldsymbol{L}_0^{\mathrm{dae}}\,\boldsymbol{x}_{\mathrm{eq},0} \tag{6.7}$$

$$\boldsymbol{L}_{\mathrm{f}}^{\mathrm{dae}}\,\boldsymbol{x}\,(T_{\mathrm{f}}) = \boldsymbol{L}_{\mathrm{f}}^{\mathrm{dae}}\,\boldsymbol{x}_{\mathrm{eq,f}}\,. \tag{6.8}$$

with the matrices $\boldsymbol{L}_0^{\mathrm{dae}} \in \mathbb{R}^{n^{\mathrm{a}}\times(2n_{\mathrm{f}}+3n_{\mathrm{c}}+n_{\mathrm{u}})}$ and $\boldsymbol{L}_{\mathrm{f}}^{\mathrm{dae}} \in \mathbb{R}^{n^{\mathrm{b}}\times(2n_{\mathrm{f}}+3n_{\mathrm{c}}+n_{\mathrm{u}})}$. The number of boundary conditions is again equal to the number of unknowns, such that it holds $n^{\mathrm{a}} + n^{\mathrm{b}} = 2n_{\mathrm{f}} + 3n_{\mathrm{c}} + n_{\mathrm{u}}$.
The simplified boundary conditions (6.5)–(6.6) or (6.7)–(6.8) reduce the effort for system analysis. Especially in the form of the conditions (6.7)–(6.8), there is no need to derive the internal dynamics. For small values of the states $\boldsymbol{\eta}$ or $\boldsymbol{x}$ respectively, the simplified boundary conditions approximate the correct boundary conditions.

Convergence Analysis of Simplified Boundary Conditions

In the following, it is analyzed that the solution η^{simp} with simplified boundary conditions (6.5)–(6.6) converges to the solution η^{orig} with correct boundary conditions (6.1)–(6.2) for increasing the pre- and postactuation intervals. Convergence is shown for a system with scalar linear internal dynamics of the form

$$M\ddot{\eta} + D\dot{\eta} + K\eta = f(z_{\mathrm{d}}(t))\,. \tag{6.9}$$

The internal dynamics is driven by the desired trajectory $z_{\mathrm{d}}(t)$. The linearization holds for small values η, $\dot{\eta}$, e.g. when the desired trajectory does not excite the internal dynamics too far from the equilibrium. The driving term $f(z_{\mathrm{d}}(t))$ is a piecewise polynomial of the form

$$f(z_{\mathrm{d}}(t)) = \begin{cases} 0 & T_0 \le t < t_0 \\ p(t) & t_0 \le t \le t_{\mathrm{f}} \\ 0 & t_{\mathrm{f}} < t \le T_{\mathrm{f}} \end{cases} \tag{6.10}$$

since the output trajectory $z_{\mathrm{d}}(t)$ is constant in the pre- and postactuation phase. Thereby, $p(t)$ describes a smooth polynomial of usually higher order due to the

desired trajectory $z_\mathrm{d}(t)$. The analytical solution of the linear internal dynamics (6.9) has the piecewise form

$$\eta(t) = \begin{cases} \eta_1(t) = C_1\,\mathrm{e}^{\lambda_\mathrm{u} t} + C_2\,\mathrm{e}^{\lambda_\mathrm{s} t} & T_0 \le t \le t_0 \\ \eta_2(t) = C_3\,\mathrm{e}^{\lambda_\mathrm{u} t} + C_4\,\mathrm{e}^{\lambda_\mathrm{s} t} + \tilde{p}(t) & t_0 \le t \le t_\mathrm{f} \\ \eta_3(t) = C_5\,\mathrm{e}^{\lambda_\mathrm{u} t} + C_6\,\mathrm{e}^{\lambda_\mathrm{s} t} & t_\mathrm{f} \le t \le T_\mathrm{f} \end{cases} \tag{6.11}$$

with the two eigenvalues λ_s and λ_u of the internal dynamics (6.9) and with $\tilde{p}(t)$ denoting a particular solution of the inhomogeneous ODE. There are 6 unknown constants C_i, which are determined by boundary conditions. The general solution (6.11) holds for the solution η^orig using correct boundary conditions with coefficients C_i^orig and for the solution η^simp using simplified boundary conditions with coefficients C_i^simp. The particular solution $\tilde{p}(t)$ is the same in both cases, since the right hand side $f(z_\mathrm{d}(t))$ is the same. Two boundary conditions are given by either the correct boundary conditions or the simplified boundary conditions. The correct boundary conditions (6.1)–(6.2) can be written as

$$-\eta^\mathrm{orig}(T_0) + \Psi^U \dot{\eta}^\mathrm{orig}(T_0) = 0 \quad \text{and} \quad -\eta^\mathrm{orig}(T_\mathrm{f}) + \Psi^S \dot{\eta}^\mathrm{orig}(T_\mathrm{f}) = 0 \tag{6.12}$$

with the coefficients Ψ^S and Ψ^U describing the directions of the stable and unstable eigenspaces respectively. The proposed simplified boundary conditions (6.5)–(6.6) are chosen in this example as

$$\eta^\mathrm{simp}(T_0) = 0 \quad \text{and} \quad \eta^\mathrm{simp}(T_\mathrm{f}) = 0\,. \tag{6.13}$$

Other possible choices are e.g. $\dot{\eta}^\mathrm{simp}(T_0) = 0$ and $\dot{\eta}^\mathrm{simp}(T_\mathrm{f}) = 0$, which lead to similar convergence results. The remaining four conditions are continuity conditions given at the beginning and end of the trajectory. The continuity conditions hold for both solutions η^orig and η^simp and are

$$\eta_1(t_0) = \eta_2(t_0)\,, \tag{6.14}$$
$$\dot{\eta}_1(t_0) = \dot{\eta}_2(t_0)\,, \tag{6.15}$$
$$\eta_2(t_\mathrm{f}) = \eta_3(t_\mathrm{f})\,, \tag{6.16}$$
$$\dot{\eta}_2(t_\mathrm{f}) = \dot{\eta}_3(t_\mathrm{f})\,. \tag{6.17}$$

For the convergence analysis, the error

$$e(t) = \left\| \eta^\mathrm{simp}(t) - \eta^\mathrm{orig}(t) \right\| \tag{6.18}$$

is of interest in the limit $T_0 \to -\infty$ and $T_\mathrm{f} \to \infty$. To evaluate the error $e(t)$, the difference between the coefficients C_i^simp and C_i^orig is computed. The six conditions each for C_i^simp and C_i^orig are collected in the set of linear equations

$$\boldsymbol{A}\boldsymbol{X} = \boldsymbol{B} \tag{6.19}$$

with the matrices

$$\boldsymbol{A} = \begin{bmatrix} \mathrm{e}^{\lambda_\mathrm{u} T_0} & \mathrm{e}^{\lambda_\mathrm{s} T_0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{e}^{\lambda_\mathrm{u} T_\mathrm{f}} & \mathrm{e}^{\lambda_\mathrm{s} T_\mathrm{f}} \\ \mathrm{e}^{\lambda_\mathrm{u} t_0} & \mathrm{e}^{\lambda_\mathrm{s} t_0} & -\mathrm{e}^{\lambda_\mathrm{u} t_0} & -\mathrm{e}^{\lambda_\mathrm{s} t_0} & 0 & 0 \\ \lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u} t_0} & \lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s} t_0} & -\lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u} t_0} & -\lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s} t_0} & 0 & 0 \\ 0 & 0 & \mathrm{e}^{\lambda_\mathrm{u} t_\mathrm{f}} & \mathrm{e}^{\lambda_\mathrm{s} t_\mathrm{f}} & -\mathrm{e}^{\lambda_\mathrm{u} t_\mathrm{f}} & -\mathrm{e}^{\lambda_\mathrm{s} t_\mathrm{f}} \\ 0 & 0 & \lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u} t_\mathrm{f}} & \lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s} t_\mathrm{f}} & -\lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u} t_\mathrm{f}} & -\lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s} t_\mathrm{f}} \end{bmatrix}, \tag{6.20}$$

$$\boldsymbol{X} = \begin{bmatrix} C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}} \\ C_2^{\mathrm{simp}} - C_2^{\mathrm{orig}} \\ C_3^{\mathrm{simp}} - C_3^{\mathrm{orig}} \\ C_4^{\mathrm{simp}} - C_4^{\mathrm{orig}} \\ C_5^{\mathrm{simp}} - C_5^{\mathrm{orig}} \\ C_6^{\mathrm{simp}} - C_6^{\mathrm{orig}} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} -C_1^{\mathrm{orig}}\Psi^U\lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u} T_0} - C_2^{\mathrm{orig}}\Psi^U\lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s} T_0} \\ -C_5^{\mathrm{orig}}\Psi^S\lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u} T_\mathrm{f}} - C_6^{\mathrm{orig}}\Psi^S\lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s} T_\mathrm{f}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.21}$$

Now, let λ_u have positive real part denoting unstable dynamics, while λ_s has negative real part denoting stable dynamics. Therefore, two coefficients of the solution η^{orig} are known with

$$C_2^{\mathrm{orig}} = 0 \qquad \text{and} \qquad C_5^{\mathrm{orig}} = 0\,. \tag{6.22}$$

This information is substituted into equation (6.19) to simplify the right hand side $\boldsymbol{B}$. Moreover, some matrix manipulations are performed on equation (6.19). The first row is multiplied by $\mathrm{e}^{-\lambda_\mathrm{u} T_0}$ and the second row is multiplied by $\mathrm{e}^{-\lambda_\mathrm{s} T_\mathrm{f}}$. This yields

$$\begin{bmatrix} 1 & \mathrm{e}^{(\lambda_\mathrm{s}-\lambda_\mathrm{u})\,T_0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{e}^{(\lambda_\mathrm{u}-\lambda_\mathrm{s})\,T_\mathrm{f}} & 1 \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} -C_1^{\mathrm{orig}}\Psi^U\lambda_\mathrm{u} \\ -C_6^{\mathrm{orig}}\Psi^S\lambda_\mathrm{s} \\ * \\ * \\ * \\ * \end{bmatrix}. \tag{6.23}$$

The symbol $*$ in the last four equations denotes that the entries are equal to the entries of the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ in equations (6.20)–(6.21) and are not displayed for readability. Afterwards, the second and fifth column of matrix $\boldsymbol{A}$ are manipulated to obtain a set of equations in terms of the vector

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} \hat{C}_1^{\mathrm{simp}} - \hat{C}_1^{\mathrm{orig}} \\ \hat{C}_2^{\mathrm{simp}} - \hat{C}_2^{\mathrm{orig}} \\ \hat{C}_3^{\mathrm{simp}} - \hat{C}_3^{\mathrm{orig}} \\ \hat{C}_4^{\mathrm{simp}} - \hat{C}_4^{\mathrm{orig}} \\ \hat{C}_5^{\mathrm{simp}} - \hat{C}_5^{\mathrm{orig}} \\ \hat{C}_6^{\mathrm{simp}} - \hat{C}_6^{\mathrm{orig}} \end{bmatrix} = \begin{bmatrix} C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}} \\ \mathrm{e}^{(\lambda_\mathrm{s}-\lambda_\mathrm{u})T_0}(C_2^{\mathrm{simp}} - C_2^{\mathrm{orig}}) \\ C_3^{\mathrm{simp}} - C_3^{\mathrm{orig}} \\ C_4^{\mathrm{simp}} - C_4^{\mathrm{orig}} \\ \mathrm{e}^{(\lambda_\mathrm{u}-\lambda_\mathrm{s})T_\mathrm{f}}(C_5^{\mathrm{simp}} - C_5^{\mathrm{orig}}) \\ C_6^{\mathrm{simp}} - C_6^{\mathrm{orig}} \end{bmatrix}. \tag{6.24}$$

This is achieved by multiplying the second column by $\mathrm{e}^{-(\lambda_\mathrm{s}-\lambda_\mathrm{u})T_0}$ and multiplying the fifth column by $\mathrm{e}^{-(\lambda_\mathrm{u}-\lambda_\mathrm{s})T_\mathrm{f}}$. The set of equations is then

$$\widehat{\boldsymbol{A}}\widehat{\boldsymbol{X}} = \widehat{\boldsymbol{B}} \tag{6.25}$$

with the matrices

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ \mathrm{e}^{\lambda_\mathrm{u}t_0} & A_1 & -\mathrm{e}^{\lambda_\mathrm{u}t_0} & -\mathrm{e}^{\lambda_\mathrm{s}t_0} & 0 & 0 \\ \lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u}t_0} & \lambda_\mathrm{s}A_1 & -\lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u}t_0} & -\lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s}t_0} & 0 & 0 \\ 0 & 0 & \mathrm{e}^{\lambda_\mathrm{u}t_\mathrm{f}} & \mathrm{e}^{\lambda_\mathrm{s}t_\mathrm{f}} & A_2 & -\mathrm{e}^{\lambda_\mathrm{s}t_\mathrm{f}} \\ 0 & 0 & \lambda_\mathrm{u}\mathrm{e}^{\lambda_\mathrm{u}t_\mathrm{f}} & \lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s}t_\mathrm{f}} & \lambda_\mathrm{u}A_2 & -\lambda_\mathrm{s}\mathrm{e}^{\lambda_\mathrm{s}t_\mathrm{f}} \end{bmatrix}, \tag{6.26}$$

$$\widehat{\boldsymbol{B}} = \begin{bmatrix} -C_1^{\mathrm{orig}}\Psi^U\lambda_\mathrm{u} \\ -C_6^{\mathrm{orig}}\Psi^S\lambda_\mathrm{s} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.27}$$

and the abbreviations

$$A_1 = \mathrm{e}^{\lambda_\mathrm{s}t_0-(\lambda_\mathrm{s}-\lambda_\mathrm{u})T_0}, \tag{6.28}$$

$$A_2 = -\mathrm{e}^{\lambda_\mathrm{u}t_\mathrm{f}-(\lambda_\mathrm{u}-\lambda_\mathrm{s})T_\mathrm{f}}. \tag{6.29}$$

As the limit $T_0 \to -\infty$ and $T_\mathrm{f} \to \infty$ is taken, the terms A_1 and A_2 tend to 0 and the matrices $\widehat{\boldsymbol{A}}$ and $\widehat{\boldsymbol{B}}$ stay finite. Therefore, the vector $\widehat{\boldsymbol{X}}$ must be finite as well. Thus, it follows in the limit

$$\lim_{T_0\to-\infty}\left(C_2^{\mathrm{simp}} - C_2^{\mathrm{orig}}\right) = \lim_{T_0\to-\infty}\left(\hat{C}_2^{\mathrm{simp}} - \hat{C}_2^{\mathrm{orig}}\right)\mathrm{e}^{-(\lambda_\mathrm{s}-\lambda_\mathrm{u})T_0} = 0. \tag{6.30}$$

This difference can be substituted in the error (6.18) between the solution η^{simp} and the solution η^{orig} at time $t = T_0$. For the error holds

$$\lim_{T_0\to-\infty}\left\|\eta^{\mathrm{simp}}(T_0) - \eta^{\mathrm{orig}}(T_0)\right\| \tag{6.31}$$

$$= \lim_{T_0\to-\infty}\left\|(C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}})\mathrm{e}^{\lambda_\mathrm{u}T_0} + (C_2^{\mathrm{simp}} - C_2^{\mathrm{orig}})\mathrm{e}^{\lambda_\mathrm{s}T_0}\right\| \tag{6.32}$$

$$= \lim_{T_0\to-\infty}\left\|(C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}} + \hat{C}_2^{\mathrm{simp}} - \hat{C}_2^{\mathrm{orig}})\mathrm{e}^{\lambda_\mathrm{u}T_0}\right\| = 0. \tag{6.33}$$

Since it is $\mathrm{Re}(\lambda_\mathrm{u}) > 0$, the error approaches zero at the speed of λ_u in the limit $T_0 \to -\infty$. A similar reasoning is performed for the right hand bound T_f. Equivalently follows for the difference $\hat{C}_5^{\mathrm{simp}} - \hat{C}_5^{\mathrm{orig}}$ in the limit

$$\lim_{T_\mathrm{f}\to\infty}\left(C_5^{\mathrm{simp}} - C_5^{\mathrm{orig}}\right) = \lim_{T_\mathrm{f}\to\infty}\left(\hat{C}_5^{\mathrm{simp}} - \hat{C}_5^{\mathrm{orig}}\right)\mathrm{e}^{-(\lambda_\mathrm{u}-\lambda_\mathrm{s})T_\mathrm{f}} = 0. \tag{6.34}$$

Therefore, for the error between both solutions at time T_f holds

$$\lim_{T_\mathrm{f}\to\infty} \left\| \eta^{\mathrm{simp}}(T_\mathrm{f}) - \eta^{\mathrm{orig}}(T_\mathrm{f}) \right\| \tag{6.35}$$

$$= \lim_{T_\mathrm{f}\to\infty} \left\| (C_5^{\mathrm{simp}} - C_5^{\mathrm{orig}}) \mathrm{e}^{\lambda_\mathrm{u} T_\mathrm{f}} + (C_6^{\mathrm{simp}} - C_6^{\mathrm{orig}}) \mathrm{e}^{\lambda_\mathrm{s} T_\mathrm{f}} \right\| \tag{6.36}$$

$$= \lim_{T_\mathrm{f}\to\infty} \left\| (\hat{C}_5^{\mathrm{simp}} - \hat{C}_5^{\mathrm{orig}} + C_6^{\mathrm{simp}} - C_6^{\mathrm{orig}}) \mathrm{e}^{\lambda_\mathrm{s} T_\mathrm{f}} \right\| = 0\,. \tag{6.37}$$

Since it is $\mathrm{Re}(\lambda_\mathrm{s}) < 0$, the error approaches zero at the speed of λ_s as $T_\mathrm{f} \to \infty$. For obtaining all other differences $C_i^{\mathrm{simp}} - C_i^{\mathrm{orig}}$ with $i = 1, 3, 4, 6$, the matrix $\widehat{\boldsymbol{A}}$ of equation (6.25) is inverted symbolically and the solution is manipulated to show the upper bound between the differences $C_i^{\mathrm{simp}} - C_i^{\mathrm{orig}}$. This is here shown exemplary for $C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}}$ and equivalent computations are possible for the other terms. The solution to equation (6.25) yields

$$\lim_{\substack{T_0\to-\infty \\ T_\mathrm{f}\to\infty}} \left(\hat{C}_1^{\mathrm{simp}} - \hat{C}_1^{\mathrm{orig}} \right) = \lim_{\substack{T_0\to-\infty \\ T_\mathrm{f}\to\infty}} \left(C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}} \right) \tag{6.38}$$

$$= \lim_{\substack{T_0\to-\infty \\ T_\mathrm{f}\to\infty}} \frac{-\mathrm{e}^{\lambda_\mathrm{s} T_\mathrm{f}} \left(C_1^{\mathrm{orig}} \Psi^U \lambda_\mathrm{u} \mathrm{e}^{\lambda_\mathrm{u} T_0} - C_6^{\mathrm{orig}} \Psi^S \lambda_\mathrm{s} \mathrm{e}^{\lambda_\mathrm{s} T_0} \right)}{\mathrm{e}^{\lambda_\mathrm{u} T_0} \mathrm{e}^{\lambda_\mathrm{s} T_\mathrm{f}} - \mathrm{e}^{\lambda_\mathrm{s} T_0} \mathrm{e}^{\lambda_\mathrm{u} T_\mathrm{f}}} \tag{6.39}$$

$$= \lim_{\substack{T_0\to-\infty \\ T_\mathrm{f}\to\infty}} \frac{-\mathrm{e}^{(\lambda_\mathrm{s}-\lambda_\mathrm{u}) T_\mathrm{f}} \left(C_1^{\mathrm{orig}} \Psi^U \lambda_\mathrm{u} \mathrm{e}^{(\lambda_\mathrm{u}-\lambda_\mathrm{s}) T_0} - C_6^{\mathrm{orig}} \Psi^S \lambda_\mathrm{s} \right)}{\mathrm{e}^{(\lambda_\mathrm{u}-\lambda_\mathrm{s}) T_0} \mathrm{e}^{(\lambda_\mathrm{s}-\lambda_\mathrm{u}) T_\mathrm{f}} - 1} = 0\,. \tag{6.40}$$

The blue terms approach 0 in the limit $T_0 \to -\infty$ and $T_\mathrm{f} \to \infty$. Thus, the difference $C_1^{\mathrm{simp}} - C_1^{\mathrm{orig}}$ approaches 0. With equivalent results for $C_i^{\mathrm{simp}} - C_i^{\mathrm{orig}}$ with $i = 3, 4, 6$, the error $\left\| \eta^{\mathrm{simp}}(t) - \eta^{\mathrm{orig}}(t) \right\|$ approaches 0 at all times t for increasing pre- and postactuation phases. This convergence analysis is verified in simulations with the manipulator with one passive joint in Sec. 6.3.4.

6.1.3 Reformulation as an Optimization Problem

In order to avoid derivation of the boundary conditions (6.1)–(6.2) or (6.3)–(6.4) respectively, a reformulation as an optimization problem is proposed in [BastosSeifriedBrüls13, BastosSeifriedBrüls17]. It is briefly summarized in the following. The optimization problem can be stated as

$$\min_{\boldsymbol{\eta}} \int_{T_0}^{T_\mathrm{f}} J^{\mathrm{ode}}(\boldsymbol{\eta}, t)\, \mathrm{d}t\,, \tag{6.41}$$

subject to the internal dynamics (3.25) as constraints. It can also be written as

$$\min_{\boldsymbol{x}} \int_{T_0}^{T_\mathrm{f}} J^{\mathrm{dae}}(\boldsymbol{x}, t)\, \mathrm{d}t\,, \tag{6.42}$$

subject to the inverse model DAEs (3.36)–(3.39) as constraints. The cost functionals $J^{\mathrm{ode}} : \mathbb{R}^{2n_\mathrm{f}-r} \times \mathbb{R} \to \mathbb{R}$ and $J^{\mathrm{dae}} : \mathbb{R}^{2n_\mathrm{f}+3n_\mathrm{c}+n_\mathrm{u}} \times \mathbb{R} \to \mathbb{R}$ are chosen in order to obtain a bounded solution. A common choice is for example

$$J^{\mathrm{ode}}(\boldsymbol{\eta}, t) = \left\| \boldsymbol{\eta} - \boldsymbol{\eta}_{\mathrm{eq}} \right\|_2^2 , \tag{6.43}$$

$$J^{\mathrm{dae}}(\boldsymbol{x}, t) = \left\| \boldsymbol{y} - \boldsymbol{y}_{\mathrm{eq}} \right\|_2^2 . \tag{6.44}$$

The cost functionals prevent the internal dynamics from becoming unbounded. For the DAE case, the cost functional must be chosen such that the internal dynamics states are kept bounded implicitly by preventing the complete state vector from becoming unbounded. It is shown in [BastosSeifriedBrüls17] that the solution of the optimization problem converges to the solution of the original stable inversion problem as the time interval $[T_0, T_\mathrm{f}]$ increases. The optimization problems are infinite dimensional. In general, there exist direct and indirect methods to solve such optimization problems [Betts10]. Direct methods are so far applied in the context of stable inversion, such as direct transcription [BastosSeifriedBrüls13, LismondeSonnevilleBrüls19] or multiple shooting [BastosSeifriedBrüls17]. For the direct approach, the cost functional as well as the constraints are discretized using a suitable integration scheme. The resulting finite dimensional optimization problem is then solved using standard optimization algorithms [Betts10]. In the context of this work, the MATLAB solver *fmincon* is used.

Alternatively, indirect methods derive necessary optimality conditions for such optimization problems by introducing adjoint variables [Betts10, Gerdts12]. The optimality conditions result in a BVP, which is solved for the unknown variables as well as the adjoint variables. The BVP has some similarities to the original stable inversion BVP, but it is of larger dimension due to the adjoint variables. The indirect approach is rather burdensome, since the optimality conditions must be derived for the optimization problem and the derivation of an initial guess of the adjoint variables is not straightforward.

6.1.4 Overview of Approaches for Stable Inversion

In total, the presented approaches result in six different problem formulations for stable inversion. For the inverse model described by the ODEs (3.25), stable inversion is stated as a boundary value problem (*bvp-ode*), as an optimization problem solved directly (*opt-ode*) and an optimization problem solved indirectly (*idopt-ode*). Similarly, for the inverse model described by the DAEs (3.36)–(3.39), stable inversion is stated as a boundary value problem (*bvp-dae*), as an optimization problem solved directly (*opt-dae*) and an optimization problem solved indirectly (*idopt-dae*). Figure 6.1 visualizes the six problem formulations. The approaches are compared in a numerical study for the manipulator with one passive joint in Sec. 6.3.

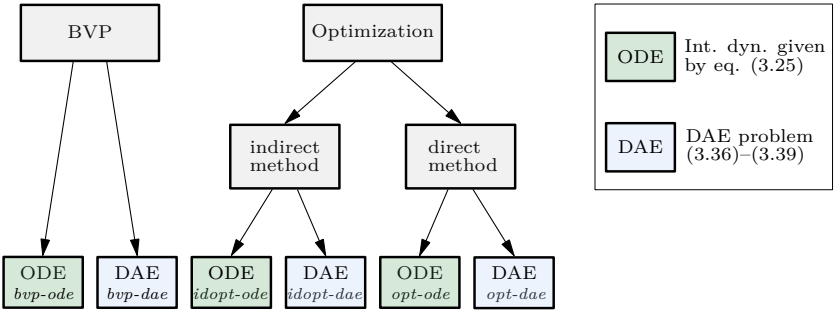


Figure 6.1: Overview of problem formulations for stable inversion.

6.2 Numerical Method for Boundary Value Problems

The original stable inversion and the indirect method for the optimization problem result in BVPs. For notational simplicity, the BVPs are summarized as

$$\boldsymbol{E}\,\dot{\widetilde{\boldsymbol{x}}} = \widetilde{\boldsymbol{f}}(\widetilde{\boldsymbol{x}}, t) \tag{6.45}$$

$$\widetilde{\boldsymbol{b}}\left(\widetilde{\boldsymbol{x}}(T_0), \widetilde{\boldsymbol{x}}(T_\mathrm{f})\right)) = \boldsymbol{0} \tag{6.46}$$

with the vector of unknowns $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{n_\mathrm{x}}$, the boundary conditions $\widetilde{\boldsymbol{b}} : \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{x}} \to \mathbb{R}^{n_\mathrm{x}}$ and the function $\widetilde{\boldsymbol{f}} : \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R} \to \mathbb{R}^{n_\mathrm{x}}$ containing the DAEs or ODEs. For ODEs, the matrix $\boldsymbol{E} \in \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}}$ is identity and for DAEs, it is a singular matrix separating differential and algebraic equations.
Generally, boundary value problems can be solved by single shooting, multiple shooting or finite differences [AscherMattheijRussell95]. Here, the problem is solved using finite differences with Simpson discretization. The time interval $T_0 = t^0 < t^1 < \cdots < t^N = T_\mathrm{f}$ is divided into $N+1$ grid points and the values $\widetilde{\boldsymbol{x}}(T_0) = \widetilde{\boldsymbol{x}}^0, \widetilde{\boldsymbol{x}}^1, \ldots, \widetilde{\boldsymbol{x}}^N = \widetilde{\boldsymbol{x}}(T_\mathrm{f})$ approximate the solution at these grid points. Then, Simpson discretization is given by

$$\boldsymbol{E}\,\widetilde{\boldsymbol{x}}^{i+1/2} = \boldsymbol{E}\,\widetilde{\boldsymbol{x}}^{i} + \Delta t \left(\frac{5}{24} \widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}^{i}, t^{i}\right) + \frac{1}{3} \widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}^{i+1/2}, t^{i+1/2}\right) - \frac{1}{24} \widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}^{i+1}, t^{i+1}\right) \right) \tag{6.47}$$

$$\boldsymbol{E}\,\widetilde{\boldsymbol{x}}^{i+1} = \boldsymbol{E}\,\widetilde{\boldsymbol{x}}^{i} + \Delta t \left(\frac{1}{6} \widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}^{i}, t^{i}\right) + \frac{2}{3} \widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}^{i+1/2}, t^{i+1/2}\right) + \frac{1}{6} \widetilde{\boldsymbol{f}}\left(\widetilde{\boldsymbol{x}}^{i+1}, t^{i+1}\right) \right). \tag{6.48}$$

Thereby, $\widetilde{\boldsymbol{x}}^{i+1/2}$ denotes the approximation on the solution at the midpoint between two grid points i and $i+1$ and Δt denotes the constant step size [ShampineGladwellThompson03]. Equations (6.47)–(6.48) add up to a large but sparse set of nonlinear equations. They are solved using Newton's method described in Sec. 3.3.2. The Simpson discretization yields convergence of order 4 for solving ODEs or index 1 DAEs [Hairer02, ShampineGladwellThompson03].

6.3 Manipulator with one Passive Joint

The manipulator with one passive joint, which is introduced in Sec. 2.4.4, is a typical testing example for stable inversion. The simplicity of the model makes it possible to derive the internal dynamics explicitly, allowing a detailed analysis. The system is non-minimum phase for end-effector tracking when a homogeneous mass distribution is chosen. This is a typical property for flexible manipulators. After analyzing the system properties, stable inversion is solved with the various approaches presented in Sec. 6.1 for a comprehensive comparison. Afterwards, application of the proposed simplified boundary conditions is evaluated.

6.3.1 System Properties

In the following, the internal dynamics of the manipulator with one passive joint is analyzed analytically according to [Seifried12, Seifried14]. The system output z is defined in equation (2.96) to depend linearly on the actuated and unactuated coordinates and it approximates the angle of the end-effector, see Fig. 2.9(b). The input-output normal form is derived according to Sec. 3.1.3. The relative degree of the system is $r = 2$ and internal dynamics of dimension 2 remains. Since the system output z depends on the actuated coordinates, a diffeomorphic state transformation $\boldsymbol{\Psi}(\boldsymbol{x})$ can be found. The unactuated states $\beta\,, \dot{\beta}$ are chosen to represent the internal dynamics $\boldsymbol{\eta}$ and therefore the new states are

$$\bar{\boldsymbol{x}} = \boldsymbol{\Psi}(\boldsymbol{x}) = \begin{bmatrix} z & \dot{z} & \beta & \dot{\beta} \end{bmatrix}^{\mathsf{T}} . \tag{6.49}$$

Zero dynamics according to equation (3.26) arises as

$$\frac{L_1 \left(4\, I_2 + L_2^2\, m_2 - 2\, L_2^2\, m_2 \cos(\beta)\right)}{4\,(L_1 + L_2)}\, \ddot{\beta} + d\dot{\beta} + k\beta + \frac{L_1 L_2^3\, \dot{\beta}^2\, m_2 \sin(\beta)}{2\,(L_1^2 + 2\, L_1 L_2 + L_2^2)} = 0\,. \tag{6.50}$$

Linearizing the zero dynamics around its equilibrium $\beta_{\mathrm{eq}} = \dot{\beta}_{\mathrm{eq}} = 0$ yields

$$\frac{L_1 \left(4\, I_2 - L_2^2\, m_2\right)}{4\,(L_1 + L_2)}\, \ddot{\beta} + d\dot{\beta} + k\beta = 0\,. \tag{6.51}$$

A stability analysis is performed by analyzing the eigenvalues of equation (6.51). Assuming a homogeneous mass distribution with $I_2 = mL^2/12$ and setting $L_1 = L_2 = L$, the eigenvalues λ_i are

$$\lambda_{1,2} = \frac{1}{L^3\, m_2} \left(6\, L\, d \pm \sqrt{6\, L\, (2\, k\, m_2\, L^3 + 6\, L\, d^2)}\right) . \tag{6.52}$$

Since the system parameters m_2, L, k, d are positive for a physical system, the eigenvalues are real numbers with $\lambda_1 > 0$ and $\lambda_2 < 0$. Thus, the manipulator with one passive joint is a non-minimum phase system.

6.3.2 Simulation Results for Stable Inversion

In the following, the manipulator with one passive joint is inverted using stable inversion. The simulation parameters are given in Tab. 6.1. The desired trajectory $z_\mathrm{d}(t)$ is chosen as a smooth transition from the initial output $z_0 = p_0 = 0\,^\circ$ to the final output $z_\mathrm{f} = p_\mathrm{f} = 30\,^\circ$. The trajectory and timing law are defined by equations (4.9) and (4.10). Beginning and end of the trajectory are chosen as $t_0 = 0\,\mathrm{s}$ and $t_\mathrm{f} = 1\,\mathrm{s}$ respectively. Figure 6.2(b) visualizes the desired trajectory with the dashed lines denoting t_0 and t_f.

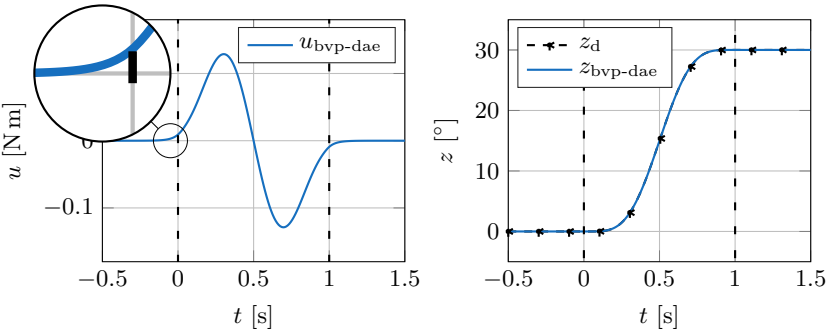
Stable inversion results are shown for the formulation *bvp-dae*, which denotes the BVP subject to the inverse model DAEs, see Sec. 6.1.4. The boundary value problem is solved using Simpson discretization with step size $\Delta t = 0.01\,\mathrm{s}$, see Sec. 6.2. Numerical results of this stable inversion problem are shown in Fig. 6.2. The preactuation phase has a length of approximately 0.2 s, see Fig. 6.2(a). During the preactuation, an input is applied on the system, while the output is constant at $z_0 = 0\,^\circ$. This is necessary to induce motion to the internal dynamics $\boldsymbol{\eta}$. The inverse model is verified in a forward simulation with the desired system input u_ffw. Figure 6.2(b) shows that perfect tracking of the system output z is possible for known parameters. The phase space of the internal dynamics $\boldsymbol{\eta}$ is shown in Fig. 6.3. As enforced by the boundary conditions (6.3)–(6.4), the coordinates leave the equilibrium $\beta_\mathrm{eq} = 0\,^\circ$ in the direction of the unstable eigenspace E_0^U and converge to the equilibrium in the direction of the stable eigenspace E_f^S.

Table 6.1: Simulation parameters of the manipulator with one passive joint.

Parameter	Value
$L_1 = L_2$	$0.5\,\mathrm{m}$
$m_1 = m_2$	$0.05\,\mathrm{kg}$
d	$2.5 \times 10^{-5}\,\frac{\mathrm{N\,m\,s}}{\mathrm{rad}}$
k	$0.5\,\frac{\mathrm{N\,m}}{\mathrm{rad}}$

6.3.3 Numerical Comparison of Stable Inversion Approaches

In the following, all six stable inversion formulations summarized in Sec. 6.1.4 are compared numerically using the manipulator with one passive joint as a testing system. The desired trajectory and the simulation parameters are the same as in Sec. 6.3.2. For the convergence analysis, the reference solution u_ref is calculated with the MATLAB solver *bvp4c* for the case *bvp-ode* and very small step size. Note that the MATLAB solver is only capable of solving BVPs for ODEs and is therefore not applicable for solving the DAE-based boundary value problems.



(a) Desired system input u_{ffw}.

(b) Simulated system output z.

Figure 6.2: Simulation results for the manipulator with one passive joint.

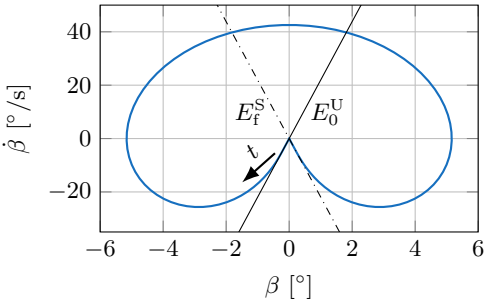


Figure 6.3: Phase space of the internal dynamics $\boldsymbol{\eta}$ for the manipulator with one passive joint.

The maximum error of the system input is computed with

$$e_{\max} = \max_t \left| u_{\text{ref}}(t) - u_{\text{ffw}}(t) \right| . \tag{6.53}$$

Figure 6.4 shows the accuracy of all six problem formulations for different step sizes Δt. All solutions converge in a similar behavior and the order of convergence is approximately 4 as stated in [Hairer02]. The accuracy of the solution does not improve for step sizes $\Delta t < 0.03\,\text{s}$ for this example due to rounding errors. The smallest achievable error is approximately $e_{\max} = 4 \times 10^{-6}\,\text{N m}$, which is small enough to apply the control input as feedforward control on a real system.
The comparison shows that the various formulations result in solutions with comparable accuracy. However, there are some differences in the computation time t_{calc}. Both direct optimization problems result in comparably long computation times compared to the other formulations. The formulation *opt-dae* is ap-

proximately 70 times slower and the formulation *opt-ode* is approximately 4 times slower compared to the other approaches. The indirect optimization approaches and the original boundary value problems result in similar fast computation times. Since these four approaches have comparable accuracy and computation times, it is proposed to apply the most simple approach for stable inversion. The most simple approach is given by the formulation *bvp-dae*, since it is not necessary to solve for any adjoint variables compared to the indirect optimization approach and it is not necessary to explicitly derive the internal dynamics. Thus, the effort in the equation setup is comparably small.

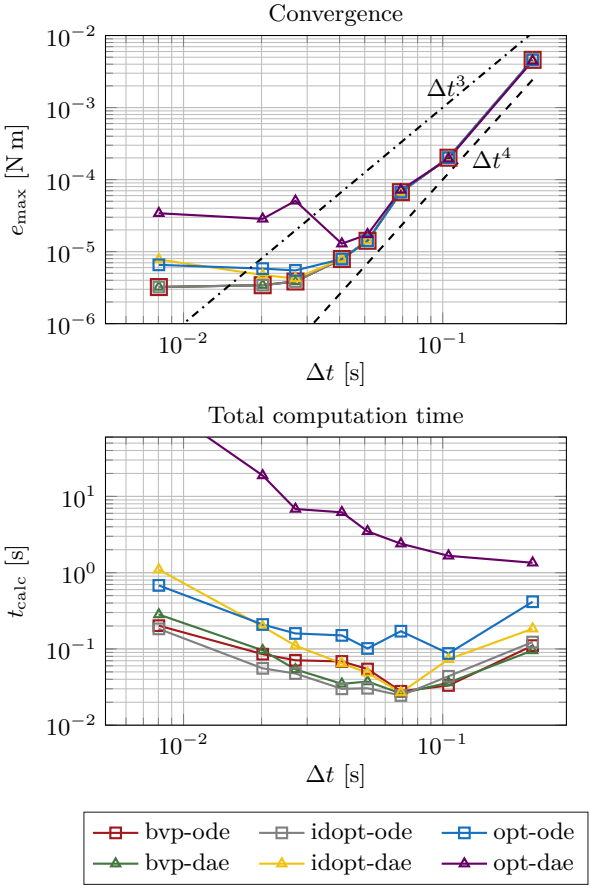


Figure 6.4: Accuracy and computation time of the stable inversion formulations.

6.3.4 Application of Simplified Boundary Conditions

Since it might not be possible to derive the internal dynamics (3.25) explicitly, it is proposed in Sec. 6.1.2 to apply simplified boundary conditions (6.5)–(6.6) or (6.7)–(6.8) for the boundary value problem. They approximate the correct boundary conditions (6.1)–(6.2) or (6.3)–(6.4), respectively, and their influence on the solution is analyzed numerically in the following. Thereby, the formulation *bvp-ode* is applied in order to compare the numerical results to the analytical convergence analysis in Sec. 6.1.2. Similar numerical results are obtained for the formulation *bvp-dae*. The simplified boundary conditions are chosen in this example as

$$\boldsymbol{\eta}(T_0) = \begin{bmatrix} 0 & \star \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}(T_\mathrm{f}) = \begin{bmatrix} 0 & \star \end{bmatrix}, \tag{6.54}$$

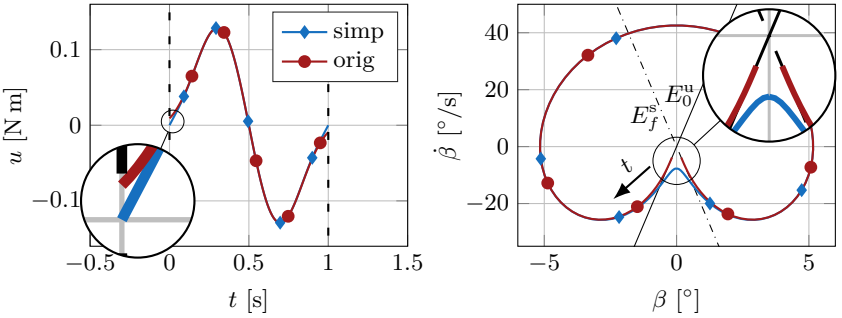
with the symbol $\star$ denoting free bounds. Note that the other possible boundary conditions

$$\boldsymbol{\eta}(T_0) = \begin{bmatrix} 0 & \star \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}(T_\mathrm{f}) = \begin{bmatrix} \star & 0 \end{bmatrix}, \tag{6.55}$$

$$\boldsymbol{\eta}(T_0) = \begin{bmatrix} \star & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}(T_\mathrm{f}) = \begin{bmatrix} 0 & \star \end{bmatrix}, \tag{6.56}$$

$$\boldsymbol{\eta}(T_0) = \begin{bmatrix} \star & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\eta}(T_\mathrm{f}) = \begin{bmatrix} \star & 0 \end{bmatrix} \tag{6.57}$$

yield similar solutions. Figure 6.5 shows the numerical results of the stable inversion for the interval $T_0 = t_0$ and $T_\mathrm{f} = t_\mathrm{f}$. Thereby, *simp* denotes the solution computed with the simplified boundary conditions (6.7)–(6.8) and *orig* denotes the solution computed with the correct boundary conditions (6.3)–(6.4). Due to the simplified boundary conditions, there is a small error of the system input compared to the solution with correct boundary conditions, see Fig. 6.5(a). This is also visualized in the phase diagram in Fig. 6.5(b).



(a) System input u.

(b) Phase space of the coordinates $\boldsymbol{\eta}$.

Figure 6.5: Simulation results for the interval $T_0 = t_0$ and $T_\mathrm{f} = t_\mathrm{f}$.

Increasing the pre- and postactuation phases

$$\Delta T = t_0 - T_0 = T_\mathrm{f} - t_\mathrm{f} \tag{6.58}$$

with $T_0 < t_0$ and $T_\mathrm{f} > t_\mathrm{f}$, shows the convergence of the solution using simplified boundary conditions to the solution using correct boundary conditions. This is reflected in the system input shown in Fig. 6.6.
For further visualization of the convergence of the two solutions, the error

$$e(t) = \left\| \boldsymbol{\eta}^{\mathrm{simp}}(t) - \boldsymbol{\eta}^{\mathrm{orig}}(t) \right\| \tag{6.59}$$

is computed between the internal dynamics $\boldsymbol{\eta}^{\mathrm{simp}}$ using simplified boundary conditions and the internal dynamics $\boldsymbol{\eta}^{\mathrm{orig}}$ using correct boundary conditions. The error is evaluated at the times $t = T_0$ and $t = T_\mathrm{f}$ and is shown in Fig. 6.7 for different pre- and postactuation phases ΔT. Note that in contrast to the previous simulations, the plot is generated with damping $d = 0.05\,\frac{\mathrm{N\,m\,s}}{\mathrm{rad}}$ for better visualization of the convergence speed. For this damping coefficient, the eigenvalues (6.52) of the zero dynamics have very different absolute values and thus, the different convergence speeds at time $t = T_0$ and $t = T_\mathrm{f}$ are clearly visible. Figure 6.7 shows that the error $e(T_0)$ converges with the speed determined by the unstable eigenvalue λ_u, denoted by the line $\mathrm{e}^{-\lambda_\mathrm{u}\Delta T}$. The error $e(T_\mathrm{f})$ converges with the speed determined by the stable eigenvalue λ_s, denoted by the line $\mathrm{e}^{\lambda_\mathrm{s}\Delta T}$. The results support the convergence analysis given in Sec. 6.1.2. Especially the convergence speed of equations (6.33) and (6.37) for the beginning and end of the trajectory can be verified.
For this example, the numerical solution using simplified boundary conditions (6.7)–(6.8) converges to the solution using correct boundary conditions (6.3)–(6.4) for increasing pre- and postactuation phases ΔT. Thus, it it proposed that the simplified boundary conditions enable the application of stable inversion to more complex systems, such as flexible manipulators.

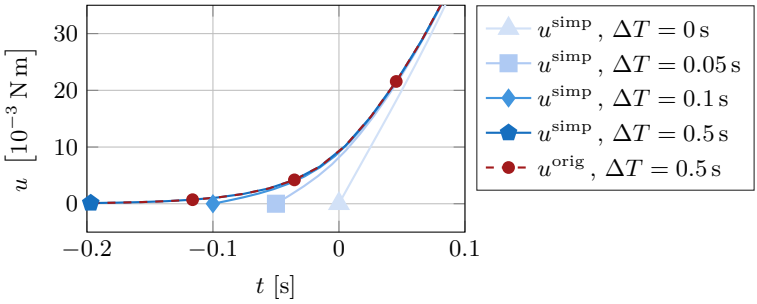


Figure 6.6: System input u for different simulation intervals ΔT.

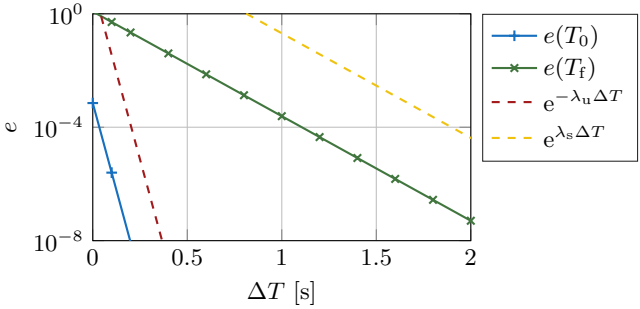


Figure 6.7: Convergence of the solution $\boldsymbol{\eta}^{\text{simp}}$ to the solution $\boldsymbol{\eta}^{\text{orig}}$.

6.4 Flexible Manipulator with one Link

The manipulator with one passive joint might not be sufficiently accurate for simulating the behavior of highly flexible systems. Therefore, an ANCF model of a flexible manipulator with one link is considered for model inversion in the following. Solving stable inversion with the original boundary conditions is not straightforward for this complex flexible system, because it is not straightforward to derive its internal dynamics. Therefore, the simplified boundary conditions proposed in Sec. 6.1.2 are applied. Note that the model based on the ANCF serves as an application example of the simplified boundary conditions. Other modeling approaches for flexible systems might also be applicable. The considered flexible manipulator is introduced in Sec. 2.4.5.
In the following, system properties with respect to the inverse model are discussed. Afterwards, stable inversion is solved with the simplified boundary conditions proposed in Sec. 6.1.2, see also [DrückerSeifried20].

6.4.1 Discussion of the System Properties

The system output z is chosen in Sec. 2.4.5 as the angle between end-effector and joint and is given by equation (2.112). The complex system dynamics make a complete analytical derivation of the internal dynamics very difficult. Nevertheless, it is possible to derive the relative degree using the approaches presented in Sec. 3.1.3 or Sec. 3.2.3.
As introduced in Sec. 2.4.5, four different manipulator configurations are considered. The cross-section at the left node of the beam element is either free or constrained to be orthogonal to the neutral axis. Moreover, the system input is either modeled as force-controlled or velocity-controlled. For the ANCF model with unconstrained cross-section and force-controlled actuator, the relative degree is computed as $r = 4$. For the same model with velocity-controlled actuator, the relative degree is $r = 3$. The decrease of the relative degree due to

a velocity-controlled actuator is in accordance with results published in [Otto-Seifried18b]. This reduction is due to a more direct actuation on velocity level. For a constrained cross-section and force-controlled actuator, the relative degree is $r = 2$, while it is $r = 1$ with velocity-controlled actuator respectively. Thus, the relative degree of the system varies between $r = 1$ and $r = 4$ depending on the considered configuration. Table 6.2 summarizes the results. The analysis of the relative degree is performed for $N = 1, 2, 3$ elements. For the tested elements, the relative degree is independent of the discretization. It is assumed that it is true for any N elements, but not proven.

Since the constrained cross-section enforces Euler-Bernoulli assumptions at the left node, the relative degree of $r = 2$ is consistent with literature in the sense that flexible manipulators modeled by Euler-Bernoulli beams with force control are reported to have relative degree $r = 2$ [BurkhardtSeifriedEberhard15, WangVidyasagar91]. If the constraint is relaxed in the ANCF model, there is a dynamic coupling of second order between the rotation of the cross-section and the rotation of the neutral axis. Since the input is applied at the cross-section and the system output at the right end of the manipulator depends on the motion of the neutral axis, an unconstraint cross-section increases the relative degree by 2. This is in accordance with a relative degree $r = 4$ in configuration 1. Nevertheless, there remain some open questions regarding the analysis of the system properties, which are discussed in the following.

The above results for the relative degree are computed symbolically based on the definitions in Sec. 3.1.3 or the generalized eigenvalue problem in Sec. 3.2.3 respectively. Both approaches result in ill-conditioned equations for the computation of the system zeros, in the sense that small disturbances yield large changes in the computed system zeros. The problem of highly sensitive system zeros is documented in literature for other flexible beam models [SpectorFlashner90, Williams92].

Thus, significance of the presented results must be further analyzed. For example, the question remains, if the relative degree of the model corresponds to the relative degree of the real system. It is relevant to compute the relative degree of the model in order to know the differentiation index of the inverse model DAEs, to choose a suitable solver, to perform index reduction and to choose sufficiently

Table 6.2: Relative degree of considered ANCF configurations.

Case	Cross-section at joint	System input type	Relative degree r
1	free	force-controlled	4
2	free	velocity-controlled	3
3	constrained	force-controlled	2
4	constrained	velocity-controlled	1

smooth trajectories. However, the relative degree of the underlying real system must be concluded by convergence analysis for increasing number of elements and comparison with the underlying partial differential equations. Results of the system zeros of flexible manipulators modeled by partial differential equations are for example documented in [Maslen95, Miu91, SpectorFlashner89] and convergence results are documented in [Williams92]. In this respect, it is still an open question, if the ANCF model is suitable to model the real system properties. It is also unclear if some system zeros and poles are introduced as artefacts of the system discretization, since finite element methods are known to introduce artificial high frequency modes [Bauchau11]. Moreover, the relative degree is closely connected to wave propagation in flexible elements [WangVidyasagar91] and it is unclear if the ANCF is suitable to model wave propagation accurately. Note that some other finite element approaches are known to introduce numerical dispersion and therefore model wave propagation incorrectly [Krenk01].

6.4.2 Simulation Results

The proposed simplified boundary conditions (6.7)–(6.8) enable the application of stable inversion to flexible manipulators which are modeled by the ANCF. Simulation results are presented in the following. First, convergence is shown for increasing number of ANCF beam elements to determine a suitable number of beam elements. Afterwards, the accuracy of the Jacobian of the ANCF equations of motion (2.44), (2.45) and (2.51) is discussed. Then, the BVP is solved with the simplified boundary conditions (6.7)–(6.8).

Simulation results are shown in the following for a flexible manipulator with quadratic cross-section and with system parameters given in Tab. 6.3. Configuration 3 with velocity-controlled actuator and free cross-section is considered at first. Note that the other cases yield nearly similar numerical results, since the chosen trajectory does not excite relevant shear deformation.

Convergence of the number of beam elements is shown in the following. For this purpose, an equivalent rigid manipulator of same geometry and mass is inverted and the computed system input is applied to the flexible manipulator in a forward simulation. This yields oscillations of the flexible manipulator, since the flexible modes are not considered during inversion. The forward simulation is

Table 6.3: Simulation parameters of the flexible manipulator with one link.

Material parameter	Value	Geometry parameter	Value
ρ	$910\ \frac{\mathrm{kg}}{\mathrm{m}^3}$	L	$1\,\mathrm{m}$
ν	0	A	$0.0081\,\mathrm{m}^2$
E	$1.2 \times 10^7\ \mathrm{Pa}$		

performed with an increasing number of ANCF beam elements and the simulated output z_N for N elements is compared to the output with a maximum number of $N = 15$ elements. The percentage value $z'(N) = 100\,\frac{\max(z_N)}{\max(z_{15})}$ is taken as a measure of convergence and is shown in Fig. 6.8. As reported in [García-VallejoMikkolaEscalona07], the ANCF converges slowly and a converged solution is reached for approximately $N = 8$ elements. Since each additional ANCF element increases the number of unknowns of the stable inversion BVP by 6 unknowns per grid point in time, $N = 4$ elements are chosen for system inversion. This seems an appropriate trade-off between accuracy and number of unknowns. The percentage of convergence for $N = 4$ elements is $z' = 99.3\,\%$.

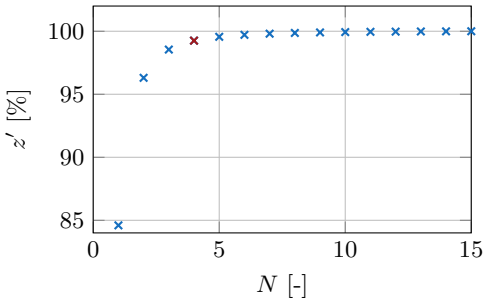


Figure 6.8: Convergence of ANCF beam elements.

In order to speed up the computation of the stable inversion BVP, accurate computation of the Jacobian of the inverse model DAEs is important in order to quickly solve the resulting set of nonlinear equations (6.47)–(6.48). The Jacobian of the complete nonlinear equations is composed of the Jacobian of the ANCF equations of motion (2.44), (2.45) and (2.51), which is analyzed in the following. For the chosen ANCF implementation, the analytical Jacobian is given in [García-VallejoEtAl04]. Figure 6.9 shows the maximum relative error e_{rel} between the analytical Jacobian and the numerical Jacobian computed with finite differences of step size h. The relative error converges with first order for forward and backwards differences and with second order for central differences. However, the computation is ill-conditioned. The error between analytical and numerical computation is quite large for forward and backwards differences and the inaccurate Jacobian will cause many Newton iterations. While central differences yield an accurate Jacobian, they are time-consuming due to many function evaluations. Therefore, for stable inversion of a highly flexible manipulator, existence of the analytical Jacobian is an important aspect.
The stable inversion BVP is solved in the following using Simpson discretization with step size $\Delta t = 0.01\,\text{s}$, see Sec. 6.2. The desired trajectory is chosen as in Sec. 6.3, see Fig. 6.2(b). The pre- and postactuation phase is chosen as 1 s each, such that the BVP is solved on the interval from $T_0 = -1\,\text{s}$ to $T_\text{f} = 2\,\text{s}$. The

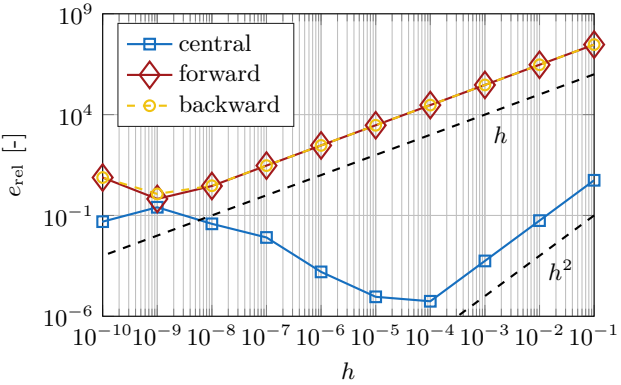
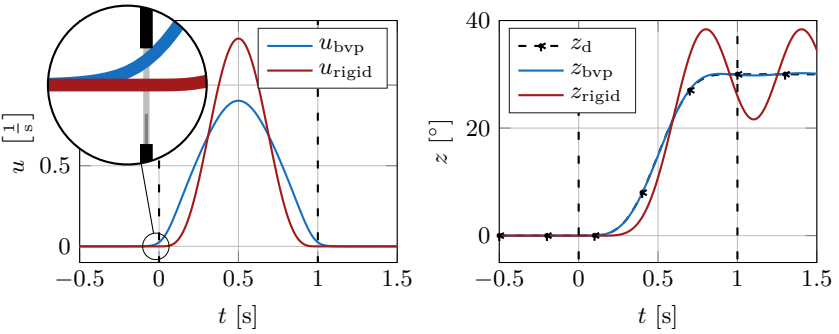


Figure 6.9: Relative error of the Jacobian compared to the analytical Jacobian.

initial guess for the BVP solver is obtained as follows. An equivalent rigid system with similar system properties is inverted to obtain the equivalent rigid system input u_{rigid}. The input u_{rigid} is then applied on the highly flexible manipulator with stiffness $E = 1.2 \times 10^9$ Pa in a forward simulation. These values serve as an initial guess for the BVP. Alternatively, it is possible to obtain the initial guess from a forward simulation with a redefined system output, which renders the internal dynamics stable, see Sec. 6.5.

Figure 6.10 shows the model inversion results. The control input u_{bvp} from stable inversion is shown in Fig. 6.10(a). It is compared to the control input u_{rigid}, which is obtained from inverting the equivalent rigid beam. The system input u_{bvp} features a short preactuation phase before the start of the trajectory. Both system inputs are applied on the flexible manipulator in a forward simulation. Note that for the forward simulation, $N = 10$ elements are chosen, since convergence is reached for this number of elements, see Fig. 6.8. The simulated system output is shown in Fig. 6.10(b) with z_{bvp} and z_{rigid} denoting the output obtained from applying the input u_{bvp} and u_{rigid} respectively. Applying the system input u_{rigid} induces oscillations of approximately $8.4\,^\circ$ around the final value of the trajectory. Inversion of the rigid system is therefore not sufficient. The input u_{bvp} results in almost perfect tracking, since it captures most dynamic modes of the model. Due to considering only $N = 4$ elements in the inversion, but $N = 10$ numbers in the forward simulation, a residual oscillation of approximately $0.2\,^\circ$ remains. This shows that $N = 4$ elements are sufficient for the inversion as the computational effort can be reduced while accurate results are still possible. For visualization of the results, the flexible manipulator is plotted in the x, y-plane for different time instances in Fig. 6.11.



(a) System input.

(b) Simulated system output z.

Figure 6.10: Model inversion results for the flexible manipulator.

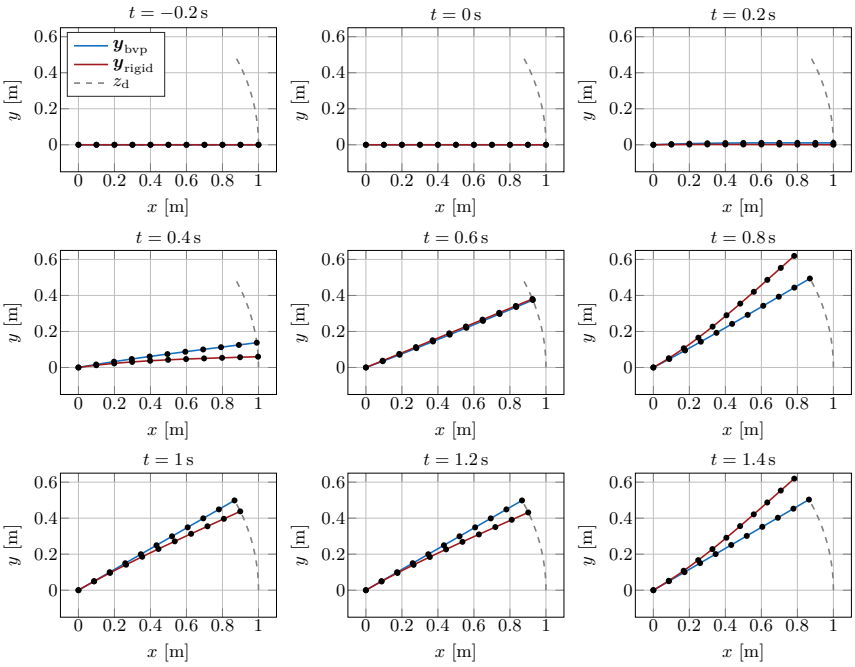


Figure 6.11: Simulation results for $N = 10$ beam elements at different times.

Since the considered trajectory does not induce shear motion at the left node, all considered beam configurations listed in Tab. 6.2 yield similar numerical results, while the computation time differs. Solving the inverse model problem for a force-controlled actuator (case 1) and velocity-controlled actuator (case 2) with free cross-section takes approximately $t_{\mathrm{calc,c1}} = 55.5\,\mathrm{s}$ and $t_{\mathrm{calc,c2}} = 29.5\,\mathrm{s}$ respectively. The cases 3 and 4 result in computation times $t_{\mathrm{calc,c3}} = 51.5\,\mathrm{s}$ and $t_{\mathrm{calc,c4}} = 76.5\,\mathrm{s}$ respectively. Overall, the computation of case 2 is fastest, even though the DAE index is higher compared to cases 3 and 4. The cross-section constraint, which reduces the index, probably introduces a very stiff constraint of the dynamics and therefore, the computation times do not reduce compared to the higher index configurations.

In conclusion, the results show that flexibility of a model must be considered during model inversion to yield accurate tracking results. It is possible to solve the stable inversion BVP efficiently with the proposed simplified boundary conditions. This extends the area of application of stable inversion to more complex systems, for which the internal dynamics cannot be easily derived. Application and selection of the simplified boundary conditions must be further analyzed for other flexible systems. Nevertheless, model inversion based on stable inversion is far from real-time capable. Alternatively, output redefinition can be considered for faster model inversion.

6.5 Output Redefinition

The above discussion about the inversion of non-minimum phase systems shows that stable inversion is quite involved. Solving the boundary value problem is not straightforward, since a sufficiently accurate initial guess is required. It is also generally not possible to obtain a solution in real-time. Moreover, stable inversion causes a non-causal preactuation phase, which might not be desired. Alternatively, the system output can be redefined to an output which yields stable internal dynamics. Tracking of the new output should still enable approximate tracking of the original output [SlotineLi91]. Perfect tracking is not possible and a small error remains, which can be decreased by a feedback controller. Due to the minimum phase characteristic for the new output, the inverse model can be integrated forward in time. This is straightforward and potentially real-time capable, see Chap. 5.
First results on output redefinition are published in [GopalswamyHedrick93] in combination with a sliding mode controller. A flexible manipulator is considered in [MoallemPatelKhorasani97, MoallemPatelKhorasani01] for output redefinition. The output is the end-effector of the flexible manipulator described by rigid and few elastic modes. The weighting of the elastic modes is varied until the system becomes minimum phase. A similar output redefinition is proposed in [TalebiPatelKhorasani04] in combination with a flexible manipulator modeled by neural networks. The system output of a flexible manipulator is redefined

in [YangKrishnanAng99] based on the tracking errors to yield a minimum phase system of relative degree 1. An optimization approach is taken in [MorlockBurkhardtSeifried16, Seifried14] to redefine the output for flexible manipulators modeled by the floating frame of reference approach.
In the following, a systematic approach is presented for flexible manipulators modeled by the ANCF to redefine the system output, such that the internal dynamics becomes stable. Simulation results are shown for one- and two-link manipulators modeled by ANCF beam elements.

6.5.1 Systematic Approach for Output Redefinition

The new system output $\widetilde{\boldsymbol{z}}$ is defined as a linear combination of the original system output $\boldsymbol{z}_{\text{flex}}$ and an equivalent rigid system output $\boldsymbol{z}_{\text{rigid}}$ with

$$\widetilde{\boldsymbol{z}} = \Gamma\, \boldsymbol{z}_{\text{flex}} + (1-\Gamma)\, \boldsymbol{z}_{\text{rigid}}\,. \tag{6.60}$$

The scalar weighting factor $\Gamma \in [0,\, 1]$ determines the influence of the original output $\boldsymbol{z}_{\text{flex}}$. For $\Gamma = 1$, the original flexible output remains and the system is non-minimum phase. For $\Gamma = 0$, the equivalent rigid output $\boldsymbol{z}_{\text{rigid}}$ remains and the system is minimum phase. For some value $0 < \Gamma < 1$, the stability property switches. The equivalent rigid output $\boldsymbol{z}_{\text{rigid}}$ describes the system output of a rigid beam with similar properties as the flexible beam. It corresponds to the collocated output of the system. It is shown below that the maximum value of Γ for which the internal dynamics is stable yields best tracking results. The proposed approach is applied to flexible manipulators with one and two links.

6.5.2 Simulation Results for the Flexible Manipulator with one Link

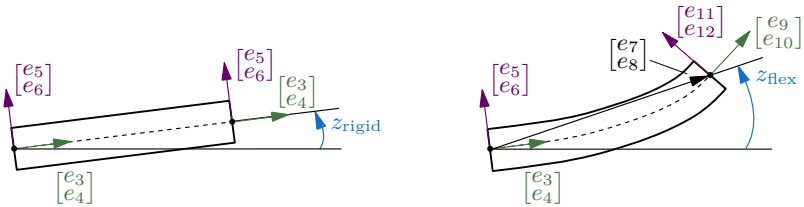
The flexible manipulator with one link introduced in Sec. 2.4.5 is considered for an analysis of the proposed redefined system output (6.60). The original system output is defined in equation (2.112) as the angle between the end-effector and the horizontal with

$$z = z_{\text{flex}} = \arctan\left(\frac{e_{6(N+1)-4}}{e_{6(N+1)-5}}\right)\,. \tag{6.61}$$

The motion of an equivalent rigid beam is completely described by the motion of the cross-section vector $\begin{bmatrix} e_5 & e_6 \end{bmatrix}^{\mathsf{T}}$, since the actuator is applied on this vector, see Fig. 6.12 and equation (2.111). The neutral axis of an equivalent rigid beam moves perpendicular to the cross-section vector. Hence, the system output z_{rigid} of a rigid beam, which describes the angle between the end-effector and the horizontal can be expressed as

$$z_{\text{rigid}} = \arctan\left(\frac{-e_5}{e_6}\right)\,. \tag{6.62}$$

Both system outputs z_{rigid} and z_{flex} are visualized in Fig. 6.12.



(a) Equivalent rigid motion. (b) Deformed ANCF element.

Figure 6.12: Geometry of one ANCF element.

In order to analyze stability of the internal dynamics, the generalized eigenvalue problem (3.60) is solved according to Sec. 3.2.3 for the output $\widetilde{\boldsymbol{z}}$ of equation (6.60). The system becomes minimum phase for values $\Gamma \lessapprox 0.75$ and the inverse model can be integrated forward in time for the redefined system output $\widetilde{\boldsymbol{z}}$. This lifts the described restrictions of the BVP and more ANCF beam elements can be considered. The following simulation results are shown for $N = 10$ beam elements, since this yields a converged solution of the ANCF model, see Fig. 6.8. For solving the inverse model DAEs, the BDF integrator with $k_{\text{bdf}} = 2$ and $\Delta t = 0.01\,\text{s}$ is chosen and the Jacobian is given analytically, see [García-VallejoEtAl04]. The desired trajectory is chosen as before, see Fig. 6.2(b). The total tracking error e_{t} of the original output z_{flex} is computed with

$$e_{\text{t}} = \frac{1}{t_{\text{f}}} \int_0^{t_{\text{f}}} \left(z_{\text{flex}} - z_{\text{d}}(t)\right)^2 \text{d}t\,. \tag{6.63}$$

Figure 6.13 shows the tracking error e_{t} for different values of the weighting factor $0 < \Gamma \leq 0.75$. The tracking error for inverting the equivalent rigid system is $e_{\text{t},\Gamma=0} \approx 5.7 \times 10^{-3}$. The error decreases for increasing values Γ. For the maximum possible value, the error is $e_{\text{t},\Gamma=0.75} \approx 6 \times 10^{-5}$. Thus, the maximum possible value Γ should be applied for model inversion to minimize the tracking error and therefore minimize the control effort of a feedback controller.
The results for model inversion are shown in Fig. 6.14 for $\Gamma = \{0, 0.2, 0.75\}$. Setting $\Gamma = 0$ is equivalent to inverting the rigid system without any internal dynamics. Therefore, the computed desired system input is zero at the end of the trajectory, see Fig. 6.14(a). It does not compensate for the internal dynamics of the fully flexible system. Applying this system input to the flexible system in a forward simulation results in oscillations of approximately $8.3\,^\circ$ around the desired final value $z_{\text{d}}(t_{\text{f}}) = 30\,^\circ$, see Fig. 6.14(b). Setting $\Gamma = 0.2$ results in small oscillations of the desired system input after the end of the trajectory. Some motion of the internal dynamics can be compensated and the residual oscillations are reduced to approximately $5\,^\circ$ around the desired value. Choosing the maximum possible weighting factor $\Gamma = 0.75$ yields nearly perfect tracking and the residual oscillation is less than $1\,^\circ$.

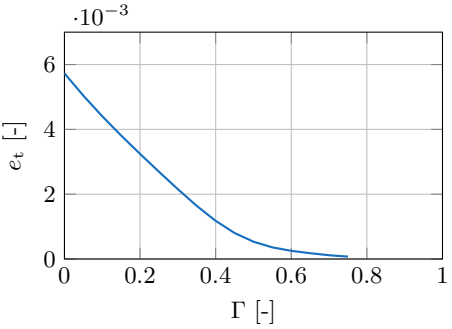
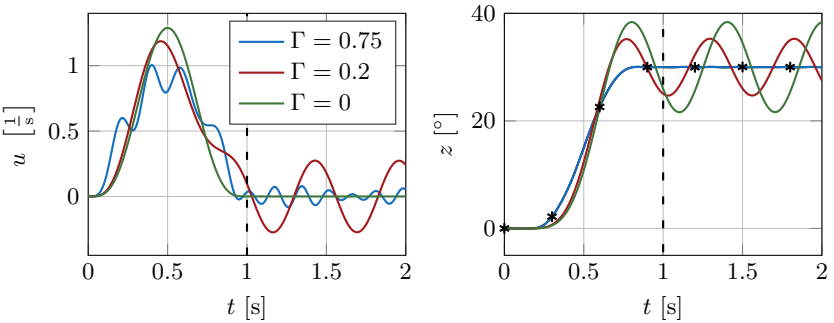


Figure 6.13: Tracking error e_t for different values Γ.



(a) System input u.

(b) System output z_{flex}.

Figure 6.14: Simulation results for different Γ for $N = 10$ beam elements.

Regarding the computation time, solving the inverse model for 10 beam elements for 2 s simulation time takes approximately $t_{\text{calc}} = 8.5\,\text{s}$. Therefore, the computation is not real-time capable. Reducing the complexity to $N = 1$ and $N = 2$ beam elements results in computation times of $t_{\text{calc}} = 1\,\text{s}$ and $t_{\text{calc}} = 1.8\,\text{s}$. Thus, real-time inversion of the flexible manipulator with redefined system output $\widetilde{z}$ can be possible for at least one and two elements. Further real-time capabilities seem possible for optimized code implementation in a real-time environment.

6.5.3 Simulation Results for the Flexible Manipulator with two Links

The proposed redefined system output enables the incorporation of ANCF beam elements into more complex multibody systems. Thus, the flexible manipulator with two links, which is introduced in Sec. 2.4.5, is considered in the following for model inversion. The system output is defined as the end-effector position in

the x, y-plane and can be expressed directly by the generalized coordinates with

$$\boldsymbol{z} = \boldsymbol{z}_{\text{flex}} = \begin{bmatrix} e_{6(N+1)-5} \\ e_{6(N+1)-4} \end{bmatrix} . \tag{6.64}$$

The equivalent rigid output follows from the geometry in Fig. 6.12 and the above considerations for the motion of an equivalent rigid beam as

$$\boldsymbol{z}_{\text{rigid}} = \begin{bmatrix} L_1 \cos(\alpha) + L_2 \dfrac{e_6}{\sqrt{e_5^2 + e_6^2}} \\ L_1 \sin(\alpha) - L_2 \dfrac{e_5}{\sqrt{e_5^2 + e_6^2}} \end{bmatrix} . \tag{6.65}$$

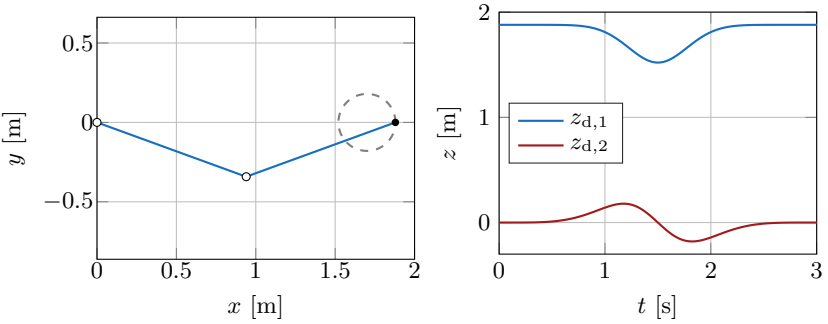
The model parameters of the flexible ANCF link are equal to the above simulations, see Tab. 6.3, while the model parameters of the rigid first link are given in Tab. 6.4. The initial state vector is chosen as

$$\boldsymbol{y}_0 = \begin{bmatrix} \alpha_0 \\ L_1 \cos(\alpha_0) \\ L_1 \sin(\alpha_0) \\ \cos(\alpha_0) \\ -\sin(\alpha_0) \\ \sin(\alpha_0) \\ \cos(\alpha_0) \\ (L_1 + L_2)\cos(\alpha_0) \\ (L_1 - L_2)\sin(\alpha_0) \\ \cos(\alpha_0) \\ -\sin(\alpha_0) \\ \sin(\alpha_0) \\ \cos(\alpha_0) \end{bmatrix} \tag{6.66}$$

with $\alpha_0 = -20°$. For model inversion, the desired trajectory is chosen as a circle, similar to the trajectory defined by equation (5.23). The center of the circle is $\boldsymbol{p}_{\text{c}} = \begin{bmatrix} 1.7 & 0 \end{bmatrix}^{\text{T}}$ m and the transition time is $t_{\text{f}} = 3\,\text{s}$. The initial position $\boldsymbol{y}_0$ is visualized in Fig. 6.15(a), while Fig. 6.15(b) shows the desired output trajectory over time. The inverse model is computed with the BDF scheme with $k_{\text{bdf}} = 2$ for step size $\Delta t = 0.01\,\text{s}$.

Table 6.4: Parameters of the rigid link for the flexible two-link manipulator.

Parameter	Value
m_1	0.1 kg
L_1	1 m



(a) Initial position and desired trajectory.

(b) Desired trajectory.

Figure 6.15: Initial position $\boldsymbol{y}_0$ and desired trajectory $z_\mathrm{d}(t)$ of the flexible two-link manipulator.

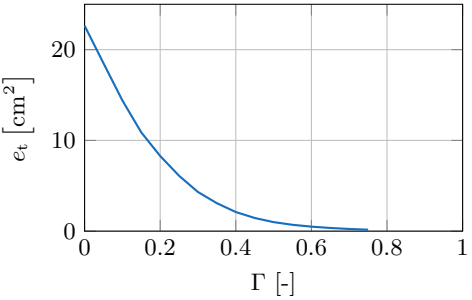
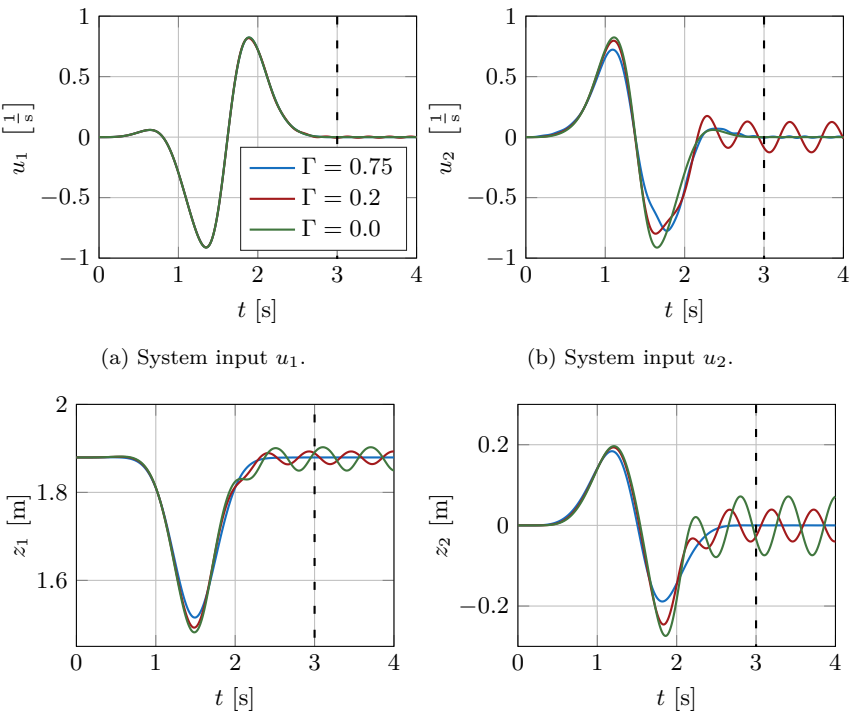


Figure 6.16: Tracking error e_t for different values Γ.

The tracking error e_t defined in equation (6.63) is shown in Fig. 6.16 for different values Γ. Similar to the single link, the two-link manipulator is minimum phase for $\Gamma \lessapprox 0.75$ and the tracking error e_t reduces exponentially for increasing Γ.

Detailed simulation results are shown for values $\Gamma = \{0, 0.2, 0.75\}$ in Fig. 6.17. The first and second system input are shown in Fig. 6.17(a) and Fig. 6.17(b) respectively. While the first system input, which is applied to the rigid body, is nearly similar for all values Γ, there are differences in the second system input. This yields different behavior in both system outputs, which are shown in Fig. 6.17(c) and Fig. 6.17(d). There remain large oscillations of the system output around the desired trajectory for $\Gamma = 0$ and $\Gamma = 0.2$, since only little flexibility is taken into account. Considering more flexibility in the system output by setting $\Gamma = 0.75$, almost perfect tracking is achieved.



(a) System input u_1.

(b) System input u_2.

(c) System output z_1.

(d) System output z_2.

Figure 6.17: Simulation results for the two-link manipulator for different Γ.

In conclusion, the redefined system output (6.60) yields a new system output with stable internal dynamics for small values Γ. The inverse model can therefore be integrated forward in time. A certain amount of flexibility can be considered during the model inversion. The maximum value yields almost perfect tracking in a forward simulation for the considered flexible manipulator. The integration forward in time lifts the limitations of stable inversion, such as computation time, computation of an initial guess and the definition of boundary conditions. Therefore, more complex systems can be considered compared to stable inversion.

Chapter 7

Conclusion and Outlook

Technological trends such as the design of compliant manipulators or light and slender machines often yield underactuated multibody systems. These developments in hardware design necessitate the simultaneous development of suitable control concepts. However, control of underactuated multibody systems is not straightforward and is therefore subject to current research. Two degree of freedom control is often used for trajectory control of such systems. A feedforward controller is responsible for trajectory tracking, while a feedback controller rejects disturbances. Within this framework, model inversion is an important tool for the design of accurate feedforward control. An accurate inverse model yields many advantages. For example, it moves the system approximately on the desired trajectory and tracking errors are expected to be small. Therefore, simple feedback controllers can be applied. Moreover, control effort in the feedback loop is reduced, which reduces noise effects.

This work contributes to the efficient design of accurate inverse models for complex underactuated multibody systems and to the efficient inversion of multibody systems in real-time. To this end, the framework of servo-constraints is analyzed in a comprehensive manner. The individual chapters are summarized in the following and contributions are pointed out.

First, Chap. 2 introduces multibody system dynamics and the absolute nodal coordinate formulation to model rigid and flexible multibody systems, respectively. Numerous application examples are introduced, such as a torsional oscillator, two cable robot models, a mass-on-car system, manipulators with one passive joint and highly flexible manipulators.

Afterwards, model inversion based on servo-constraints is analyzed comprehensively in Chap. 3. In contrast to classical inversion approaches, servo-constraints can be applied straightforwardly to more complex multibody systems. An analytical manipulation of the equations of motion is not necessary, since the inverse model is represented by a set of DAEs. Within this work, all possible system categories, such as differentially flat systems, minimum phase systems and non-minimum phase systems are considered. Analysis tools and solving methods are presented for all system categories. This gives a comprehensive overview of the application of servo-constraints and points out important inverse model properties.

Differentially flat systems are considered in Chap. 4. These systems do not contain internal dynamics. Solving methods are analyzed for an undamped tor-

sional oscillator as well as a cable robot. Due to the lack of internal dynamics, simple integration methods, such as the implicit Euler scheme are sufficient for computing the inverse model. Real-time capabilities of the servo-constraints approach are shown experimentally for the cable robot. Moreover, short online computation times show that there is potential to solve systems with more degrees of freedom within the available control loop step size. Application of implicit servo-constraints is demonstrated experimentally for the cable robot with platform orientation control. Furthermore, feedforward control based on servo-constraints is combined with feedback control in experiments. Both the linear quadratic regulator and sliding mode control result in good performance for disturbance rejection.

Minimum phase systems possess internal dynamics and are considered in Chap. 5. Due to the internal dynamics, special care must be taken in the selection of solving methods. The implicit Euler scheme is not sufficient because it quickly dampens out the internal dynamics and cannot represent the system accurately. Therefore, higher order integration schemes should be applied for minimum phase systems. Comparative studies are shown for a mass-on-car system and a three-dimensional manipulator with one passive joint. The mass-on-car system is designed to possess first order as well as second order dynamics to serve as a benchmark problem for model inversion based on servo-constraints. Simulations for the three-dimensional manipulator show that real-time inversion can be possible for complex minimum phase systems.

Non-minimum phase systems are considered in Chap. 6. They contain unstable internal dynamics and the inverse models cannot be integrated forward in time. For stable inversion, a boundary value problem or an optimization problem must be solved instead. A simplification of the boundary value problem is proposed in this work to extend stable inversion to systems with more degrees of freedom. First, convergence of the solution with simplified boundary conditions to the correct solution is shown for a simple manipulator with one passive joint. Afterwards, the simplified methodology is applied to a highly flexible manipulator modeled by the ANCF. Using the simplified boundary conditions now enables the application of stable inversion to more complex systems than considered so far. However, solving the boundary value problem is still burdensome. Therefore, a new systematic output redefinition strategy is proposed for manipulators modeled by the ANCF, which renders the internal dynamics stable. This enables integration forward in time of the inverse model, while the original output is tracked approximately. Due to less computational effort of the forward time integration, it is possible to incorporate the considered highly flexible links into more complex systems. This is demonstrated for a two-link manipulator with a rigid first link.

From the above contributions, the following conclusions can be drawn. Servo-constraints are a suitable method for the efficient inversion of complex underactuated multibody systems independently of the underlying system class. However, the system class must be determined in order to choose suitable solving methods.

The real-time capabilities of servo-constraints are validated experimentally for differentially flat systems. The experimental results show that there is potential to solve systems with more degrees of freedom within the available control loop step size. Moreover, simulations with a three-dimensional minimum phase system show that real-time inversion is possible for this system class as well. This result should be validated in experiments. In order to enable real-time computation, requirements on the accuracy can be lowered, since there exist other disturbances such as friction on the real system. Furthermore, feedback controllers are able to reduce the arising tracking errors. With real-time inversion of differentially flat systems and minimum phase systems, a large range of typical mechanical systems is covered. This makes the approach suitable for many real world industrial applications.

For non-minimum phase systems, stable inversion in real-time does not seem possible in the near future. This is due to the boundary value problem which has to be solved within one time step. However, the proposed simplifications of the boundary value problem enable the consideration of systems with more flexible elements than considered so far. This makes servo-constraints applicable to highly flexible systems, such as machines for human-robot interaction or soft robots. A suitable alternative to stable inversion is the proposed output redefinition strategy for flexible systems modeled by the ANCF. Based on the output redefinition, flexible systems can be treated as minimum phase systems. The arising tracking errors of the original output stay small and can be further reduced by adding feedback control.

With these aspects, steps are taken towards real-time inversion of standard industrial machines and real world applications. Moreover, the present work raises interesting questions to be considered in future research.

In general, the servo-constraints approach can be extended to models with parametric uncertainty in order to represent the real model more accurately. Preliminary studies with uncertain parameters modeled by fuzzy logic show promising results. A system with fuzzy-valued parameters is inverted in order to yield a system input which in average shows the best performance for the uncertain system. Further investigation with different uncertainty models and experimental studies should be performed for an extended insight into this research direction.

Regarding the real-time capabilities of servo-constraints, three developments are of interest. First of all, application to standard industrial machines can be investigated in order to transfer current research results to industry standard. Secondly, with computation in real-time, it is possible to consider not only planned trajectories, but also adaptable trajectories, which are modified in real-time. This is interesting for collision avoidance or learning new tasks, which can be specified by the user. Thirdly, it is possible to consider mechanical models which are adapted on-line. Then, the models can be learned over time to approximate the real system more accurately. Machine learning strategies or classical adaptive algorithms can be applied for the model update. Moreover, the models could be adapted to account for arising contact forces. Considering models which are

adapted on-line is strictly speaking not a feedforward strategy, since measurements are necessary for the model update. However, preliminary studies show that the overall tracking performance can be improved because the real system is represented more accurately.

In terms of the inversion of minimum phase systems, simulation results presented in this work show that real-time integration of the inverse model DAEs is possible. Experimental data is necessary to support this claim. The damped torsional oscillator is a suitable application example for experimental studies.

Regarding non-minimum phase systems, several aspects can be investigated. The proposed simplified boundary conditions for stable inversion should be applied to other highly flexible multibody systems. A systematic procedure for the selection of boundary conditions must be further explored. This can further reduce the effort in the setup of the boundary value problem. Moreover, the question remains if the considered flexible multibody systems are applicable to reflect the relative degree of the real system. To this end, convergence of the inverse model to the underlying continuous system must be further investigated. Additionally, the application of the proposed output redefinition strategy to manipulators with multiple highly flexible links should be analyzed. This is necessary for an incorporation of flexible links into general flexible multibody systems. Finally, feedback controllers for non-minimum phase systems must be further improved for a complete implementation of the two design degree of freedom control structure.

Bibliography

[AltmannBetschYang16] Altmann, R.; Betsch, P.; Yang, Y.: Index reduction by minimal extension for the inverse dynamics simulation of cranes. Multibody System Dynamics, Vol. 36, No. 3, pp. 295–321, 2016.

[AltmannHeiland17] Altmann, R.; Heiland, J.: Simulation of multibody systems with servo constraints through optimal control. Multibody System Dynamics, Vol. 40, pp. 75–98, 2017.

[Ambrosio13] Ambrosio, J.A.C.: Multibody dynamics approaches to biomechanical applications to human motion tasks. In H. Gattringer; J. Gerstmayr (Eds.) Multibody System Dynamics, Robotics and Control, pp. 259–289, Vienna: Springer Vienna, 2013.

[ArnoldEtAl08] Arnold, E.; Neupert, J.; Sawodny, O.; Schneider, K.: Modellprädiktive Trajektoriengenerierung für flachheitsbasierte Folgeregelungen am Beispiel eines Hafenmobilkrans (model-predictive trajectory generation for flatness-based nonlinear tracking control with application to boom cranes). at - Automatisierungstechnik, Vol. 56, No. 8, pp. 395–405, 2008.

[Aschemann09] Aschemann, H.: Passivity-based trajectory control of an overhead crane by interconnection and damping assignment. In H. Ulbrich; L. Ginzinger (Eds.) Motion and Vibration Control, pp. 21–30. Springer Netherlands, 2009.

[AscherMattheijRussell95] Ascher, U.; Mattheij, R.; Russell, R.: Numerical Solution of Boundary Value Problems for Ordinary Differential Equations. Society for Industrial and Applied Mathematics, 1995.

[AshrafiuonErwin08] Ashrafiuon, H.; Erwin, R.S.: Sliding mode control of underactuated multibody systems and its application to shape change control. International Journal of Control, Vol. 81, No. 12, pp. 1849–1858, 2008.

[BajodahHodgesChen05] Bajodah, A.H.; Hodges, D.H.; Chen, Y.H.: Inverse dynamics of servo-constraints based on the generalized inverse. Nonlinear Dynamics, Vol. 39, No. 1, pp. 179–196, 2005.

[BastosSeifriedBrüls17] Bastos, G.; Seifried, R.; Brüls, O.: Analysis of stable model inversion methods for constrained underactuated mechanical systems. Mechanism and Machine Theory, Vol. 111, No. Supplement C, pp. 99 – 117, 2017.

[BastosSeifriedBrüls13] Bastos, G.; Seifried, R.; Brüls, O.: Inverse dynamics of serial and parallel underactuated multibody systems using a DAE optimal control approach. Multibody System Dynamics, Vol. 30, No. 3, pp. 359–376, 2013.

[Bauchau11] Bauchau, O.A.: Flexible Multibody Dynamics. Solid Mechanics and Its Applications. Dordrecht, 2011.

[BauchauEtAl15] Bauchau, O.A.; Han, S.; Mikkola, A.; Matikainen, M.K.; Gruber, P.: Experimental validation of flexible multibody dynamics beam formulations. Multibody System Dynamics, Vol. 34, No. 4, pp. 373–389, 2015.

[Baumgarte72] Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Computer Methods in Applied Mechanics and Engineering, Vol. 1, No. 1, pp. 1–16, 1972.

[BencsikKovácsZelei17] Bencsik, L.; Kovács, L.L.; Zelei, A.: Stabilization of internal dynamics of underactuated systems by periodic servo-constraints. International Journal of Structural Stability and Dynamics, Vol. 17, No. 05, pp. 1740004–1 – 1740004–14, 2017.

[Berger17] Berger, T.: The zero dynamics form for nonlinear differential-algebraic systems. Automatic Control, IEEE Transactions on, Vol. 62, No. 8, pp. 4131–4137, 2017.

[BergerEtAl18] Berger, T.; Otto, S.; Reis, T.; Seifried, R.: Combined open-loop and funnel control for underactuated multibody systems. Nonlinear Dynamics, Vol. 95, p. 1977–1998, 2018.

[BergerHoàngReis18] Berger, T.; Hoàng, L.H.; Reis, T.: Funnel control for nonlinear systems with known strict relative degree. Vol. 87, pp. 345–357, 2018.

[BessaEtAl19] Bessa, W.M.; Otto, S.; Kreuzer, E.; Seifried, R.: An adaptive fuzzy sliding mode controller for uncertain underactuated mechanical systems. Journal of Vibration and Control, Vol. 25, No. 9, pp. 1521–1535, 2019.

[Bestle05] Bestle, D.: Design of a laboratory crane for testing control approaches. In H. Ulbrich; W. Günthner (Eds.) IUTAM Symposium on Vibration Control of Nonlinear Mechanisms and Structures, Vol. 130 of

Solid Mechanics and its Applications, pp. 101–110. Springer Netherlands, 2005.

[BetschAltmannYang16] Betsch, P.; Altmann, R.; Yang, Y.: Numerical Integration of Underactuated Mechanical Systems Subjected to Mixed Holonomic and Servo Constraints. pp. 1–18. Cham: Multibody Dynamics, 2016.

[BetschQuasemUhlar09] Betsch, P.; Quasem, M.; Uhlar, S.: Numerical integration of discrete mechanical systems with mixed holonomic and control constraints. Journal of Mechanical Science and Technology, Vol. 23, No. 4, pp. 1012–1018, 2009.

[Betts10] Betts, J.: Practical Methods for Optimal Control and Estimation Using Nonlinear Programming. Society for Industrial and Applied Mathematics, 2. Edn., 2010.

[BicchiPeshkinColgate08] Bicchi, A.; Peshkin, M.A.; Colgate, J.E.: Safety for physical human–robot interaction. In M.A. Peshkin; J.E. Colgate (Eds.) Handbook of Robotics, pp. 1335–1348. Springer Berlin Heidelberg, 2008.

[Bishop08] Bishop, R.H. (Ed.): Mechatronic systems, sensors, and actuators : fundamentals and modeling. The mechatronics handbook. Boca Raton, Fla. u.a.: CRC Press, 2. Edn., 2008.

[BlajerEtAl09] Blajer, W.; Dziewiecki, K.; Kołodziejczyk, K.; Mazur, Z.: Dynamics and control of underactuated mechanical systems: analysis and simple experimental verification. PAMM, Vol. 9, No. 1, pp. 107–108, 2009.

[BlajerKołodziejczyk04] Blajer, W.; Kołodziejczyk, K.: A geometric approach to solving problems of control constraints: Theory and a DAE framework. Multibody System Dynamics, Vol. 11, No. 4, pp. 343–364, 2004.

[BlajerKołodziejczyk07] Blajer, W.; Kołodziejczyk, K.: Control of underactuated mechanical systems with servo-constraints. Nonlinear Dynamics, Vol. 50, No. 4, pp. 781–791, 2007.

[BlajerKołodziejczyk11] Blajer, W.; Kołodziejczyk, K.: Improved DAE formulation for inverse dynamics simulation of cranes. Multibody System Dynamics, Vol. 25, No. 2, pp. 131–143, 2011.

[BoustanyD'Andrea-Novel92] Boustany, F.; D'Andrea-Novel, B.: Adaptive control of an overhead crane using dynamic feedback linearization and estimation design. Proceedings IEEE International Conference on Robotics and Automation, pp. 1963–1968, 1992.

[Brenan89] Brenan, K.E.: Numerical solution of initial-value problems in differential-algebraic equations. New York: North-Holland, 1989.

[Brogan91] Brogan, W.L.: Modern Control Theory. Prentice Hall, 1991.

[Broyden65] Broyden, C.G.: A class of methods for solving nonlinear simultaneous equations. Mathematics of Computation, Vol. 19, No. 92, pp. 577–593, 1965.

[BrülsBastosSeifried13] Brüls, O.; Bastos, G.J.; Seifried, R.: A stable inversion method for feedforward control of constrained flexible multibody systems. Journal of Computational and Nonlinear Dynamics, Vol. 9, No. 1, pp. 011014–1 – 011014–9, 2013.

[Bryson75] Bryson, A.E.: Applied optimal control : optimization, estimation, and control. New York u.a.: Hemisphere Publ. Corp., 5. Edn., 1975.

[BurgermeisterArnoldEsterl06] Burgermeister, B.; Arnold, M.; Esterl, B.: DAE time integration for real-time applications in multi-body dynamics. ZAMM Journal of Applied Mathematics and Mechanics, Vol. 86, No. 10, pp. 759–771, 2006.

[BurkhardtSeifriedEberhard15] Burkhardt, M.; Seifried, R.; Eberhard, P.: Experimental studies of control concepts for a parallel manipulator with flexible links. Journal of Mechanical Science and Technology, Vol. 29, No. 7, pp. 2685–2691, 2015.

[CamarilloEtAl08] Camarillo, D.B.; Milne, C.F.; Carlson, C.R.; Zinn, M.R.; Salisbury, J.K.: Mechanics modeling of tendon-driven continuum manipulators. IEEE Transactions on Robotics, Vol. 24, No. 6, pp. 1262–1273, 2008.

[Campbell95] Campbell, S.L.: High-index differential algebraic equations. Mechanics of Structures and Machines, Vol. 23, No. 2, pp. 199–222, 1995.

[CampbellGear95] Campbell, S.L.; Gear, C.W.: The index of general nonlinear DAEs. Numerische Mathematik, Vol. 72, No. 2, pp. 173–196, 1995.

[CelikovskyZikmundMoog08] Celikovsky, S.; Zikmund, J.; Moog, C.: Partial exact linearization design for the acrobot walking. In 2008 American Control Conference, pp. 874–879, 2008.

[Chen08] Chen, Y.H.: Equations of motion of mechanical systems under servo constraints: The maggi approach. Mechatronics, Vol. 18, No. 4, pp. 208 – 217, 2008.

[ChenPaden96] Chen, D.; Paden, B.: Stable inversion of nonlinear non-minimum phase systems. International Journal of Control, Vol. 64, No. 1, pp. 81–97, 1996.

[DelaleauRudolph98] Delaleau, E.; Rudolph, J.: Control of flat systems by quasi-static feedback of generalized states. International Journal of Control, Vol. 71, No. 5, pp. 745–765, 1998.

[deLucaBook08] deLuca, A.; Book, W.: Robots with flexible elements. In B. Siciliano; O. Khatib (Eds.) Springer Handbook of Robotics, pp. 287–319. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[deLucaFarinaLucibello05] deLuca, A.; Farina, R.; Lucibello, P.: On the control of robots with visco-elastic joints. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 4297–4302, 2005.

[deLucaPanzieriUlivi98] deLuca, A.; Panzieri, S.; Ulivi, G.: Stable inversion control for flexible link manipulators. pp. 799–805. Handbook of Robotics, 1998.

[DevasiaChenPaden96] Devasia, S.; Chen, D.; Paden, B.: Nonlinear inversion-based output tracking. IEEE Transactions on Automatic Control, Vol. 41, No. 7, pp. 930–942, 1996.

[DiboldGerstmayrIrschik09] Dibold, M.; Gerstmayr, J.; Irschik, H.: A Detailed Comparison of the Absolute Nodal Coordinate and the Floating Frame of Reference Formulation in Deformable Multibody Systems. Journal of Computational and Nonlinear Dynamics, Vol. 4, No. 2, 2009.

[DrückerSeifried20] Drücker, S.; Seifried, R.: Stable inversion for flexible multibody systems using the ANCF. Accepted to Proc. in Applied Mathematics and Mechanics, 2020.

[DufvaSopanenMikkola06] Dufva, K.E.; Sopanen, J.T.; Mikkola, A.M.: Three-dimensional beam element based on a cross-sectional coordinate system approach. Nonlinear Dynamics, Vol. 43, No. 4, pp. 311–327, 2006.

[DwivedyEberhard06] Dwivedy, S.K.; Eberhard, P.: Dynamic analysis of flexible manipulators, a literature review. Mechanism and Machine Theory, Vol. 41, No. 7, pp. 749 – 777, 2006.

[EscalonaHussienShabana98] Escalona, J.; Hussien, H.; Shabana, A.: Application of the absolute nodal co-ordinate formulation to multibody system dynamics. Journal of Sound and Vibration, Vol. 214, No. 5, pp. 833 – 851, 1998.

[FaesslerFranchiScaramuzza18] Faessler, M.; Franchi, A.; Scaramuzza, D.: Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. IEEE Robotics and Automation Letters, Vol. 3, No. 2, pp. 620–626, 2018.

[FliessEtAl95] Fliess, M.; Lévine, J.; Martin, P.; Rouchon, P.: Flatness and defect of non-linear systems: introductory theory and examples. International Journal of Control, Vol. 61, No. 6, pp. 1327–1361, 1995.

[FloresMilam06] Flores, M.E.; Milam, M.B.: Trajectory generation for differentially flat systems via nurbs basis functions with obstacle avoidance. In American Control Conference, 2006, pp. 5769–5775, 2006.

[Föllinger88] Föllinger, O.: Optimierung dynamischer Systeme: eine Einführung für Ingenieure. München [u.a.]: Oldenbourg, 2. Edn., 1988. Ab 3. Aufl.: Föllinger, Otto: Optimale Regelung und Steuerung.

[FuchshumerSchlacherRittenschober05] Fuchshumer, S.; Schlacher, K.; Rittenschober, T.: Nonlinear vehicle dynamics control - a flatness based approach. In Proceedings of the 44th IEEE Conference on Decision and Control, pp. 6492–6497, 2005.

[FumagalliEtAl10] Fumagalli, A.; Masarati, P.; Morandini, M.; Mantegazza, P.: Control constraint realization for multibody systems. Journal of Computational and Nonlinear Dynamics, Vol. 6, No. 1, pp. 011002–1 – 011002–8, 2010.

[García-VallejoMikkolaEscalona07] García-Vallejo, D.; Mikkola, A.M.; Escalona, J.L.: A new locking-free shear deformable finite element based on absolute nodal coordinates. Nonlinear Dynamics, Vol. 50, No. 1, pp. 249–264, 2007.

[García-VallejoEtAl04] García-Vallejo, D.; Mayo, J.; Escalona, J.L.; Domínguez, J.: Efficient evaluation of the elastic forces and the jacobian in the absolute nodal coordinate formulation. Nonlinear Dynamics, Vol. 35, No. 4, pp. 313–329, 2004.

[Gerdts12] Gerdts, M.: Optimal Control of ODEs and DAEs. De Gruyter textbook. s.l.: Walter de Gruyter GmbH Co.KG, 1. Edn., 2012.

[GerstmayrShabana06] Gerstmayr, J.; Shabana, A.: Analysis of thin beams and cables using the absolute nodal co-ordinate formulation. Nonlinear Dynamics, Vol. 45, No. 1-2, pp. 109–130, 2006.

[GerstmayrSugiyamaMikkola13] Gerstmayr, J.; Sugiyama, H.; Mikkola, A.: Review on the absolute nodal coordinate formulation for large deformation analysis of multibody systems. Journal of Computational And Nonlinear Dynamics, Vol. 8, No. 3, 2013.

[GonzálezEtAl17] González, F.; Masarati, P.; Cuadrado, J.; Naya, M.A.: Assessment of Linearization Approaches for Multibody Dynamics Formulations. Journal of Computational and Nonlinear Dynamics, Vol. 12, No. 4, 2017.

[GopalswamyHedrick93] Gopalswamy, S.; Hedrick, J.K.: Tracking nonlinear non-minimum phase systems using sliding control. International Journal of Control, Vol. 57, No. 5, pp. 1141–1158, 1993.

[Géradin01] Géradin, M.: Flexible multibody dynamics : a finite element approach. Chichester u.a.: Wiley, 2001.

[GraichenHagenmeyerZeitz05] Graichen, K.; Hagenmeyer, V.; Zeitz, M.: A new approach to inversion-based feedforward control design for nonlinear systems. Automatica, Vol. 41, No. 12, pp. 2033 – 2041, 2005.

[Griepentrog86] Griepentrog, E.: Differential-algebraic equations and their numerical treatment. Teubner-Texte zur Mathematik. Leipzig: Teubner, 1. Edn., 1986.

[HagenmeyerZeitz04] Hagenmeyer, V.; Zeitz, M.: Flachheitsbasierter Entwurf von linearen und nichtlinearen Vorsteuerungen (flatness-based design of linear and nonlinear feedforward controls). at – Automatisierungstechnik/Methoden und Anwendungen der Steuerungs-, Regelungs- und Informationstechnik, Vol. 52, No. 1, pp. 3–12, 2004.

[Hairer02] Hairer, E.: Stiff and differential-algebraic problems. Hairer, Ernst Solving ordinary differential equations / E. Hairer; G. Wanner. Berlin u.a.: Springer, 2. Edn., 2002.

[Hairer08] Hairer, E.: Nonstiff problems. Hairer, Ernst Solving ordinary differential equations / E. Hairer. Berlin u.a.: Springer, 2., rev. ed., 3. printing. Edn., 2008.

[HairerRocheLubich89] Hairer, E.; Roche, M.; Lubich, C.: The numerical solution of differential-algebraic systems by runge-kutta methods, 1989.

[Heiland16] Heiland, J.: A differential-algebraic riccati equation for applications in flow control. SIAM Journal on Control and Optimization, Vol. 54, No. 2, pp. 718–739, 2016.

[Hermann11] Hermann, M.: Numerische Mathematik. Berlin, Boston: de Gruyter, 2011.

[HeydenWoernle06] Heyden, T.; Woernle, C.: Dynamics and flatness-based control of a kinematically undetermined cable suspension manipulator. Multibody System Dynamics, Vol. 16, No. 2, pp. 155–177, 2006.

[HoaggBernstein07] Hoagg, J.B.; Bernstein, D.S.: Nonminimum-phase zeros - much to do about nothing - classical control - revisited part ii. IEEE Control Systems Magazine, Vol. 27, No. 3, pp. 45–57, 2007.

[Hollburg07] Hollburg, U.: Maschinendynamik. Naturwissenschaft und Technik 1/2010. München: Oldenbourg Wissenschaftsverlag, 2. Edn., 2007.

[Isidori96] Isidori, A.: Nonlinear control systems. Berlin [u.a.]: Springer, 3. Edn., 1996.

[KnierimKriegerSawodny10] Knierim, K.L.; Krieger, K.; Sawodny, O.: Flatness based control of a 3-dof overhead crane with velocity controlled drives. IFAC Proceedings Volumes, Vol. 43, No. 18, pp. 363–368, 2010.

[KnierimSawodny12] Knierim, K.L.; Sawodny, O.: Real-time trajectory generation for three-times continuous trajectories. 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1462–1467, 2012.

[Knoll16] Knoll, C.: Regelungstheoretische Analyse- und Entwurfsansätze für unteraktuierte mechanische Systeme. Dresden: Verlag Carsten Knoll, 2016.

[KovácsBencsik12] Kovács, L.L.; Bencsik, L.: Stability case study of the acroboter underactuated service robot. Theoretical and Applied Mechanics Letters, Vol. 2, No. 4, 2012.

[Krenk01] Krenk, S.: Dispersion-corrected explicit integration of the wave equation. Computer methods in applied mechanics and engineering, Vol. 191, No. 8-10, pp. 975–987, 2001.

[KreuzerRadisch14] Kreuzer, E.; Radisch, C.: Sliding mode control of underactuated mechanical systems by means of nonlinear sliding surfaces. In Proceedings of 8th European Nonlinear Dynamics Conference, 2014.

[Labisch13] Labisch, D.: Verkopplungsbasierte Methoden zum Regler- und Beobachterentwurf für nichtlineare Deskriptorsysteme, Vol. 1227 of 8. Düsseldorf: VDI-Verlag, 2013.

[Laplante04] Laplante, P.A.: Real-time systems design and analysis. Hoboken, NJ: Wiley-Interscience, 3. Edn., 2004.

[LeitzLeyendecker14] Leitz, T.; Leyendecker, S.: Simulating underactuated multibody dynamics using servo constraints and variational integrators. PAMM, Vol. 14, No. 1, pp. 61–62, 2014.

[LismondeSonnevilleBrüls19] Lismonde, A.; Sonneville, V.; Brüls, O.: A geometric optimization method for the trajectory planning of flexible manipulators. Multibody System Dynamics, Vol. 47, p. 347–362, 2019.

[LiuEtAl15] Liu, X.; Zhen, S.; Huang, K.; Zhao, H.; Chen, Y.H.; Shao, K.: A systematic approach for designing analytical dynamics and servo control of constrained mechanical systems. IEEE/CAA Journal of Automatica Sinica, Vol. 2, No. 4, pp. 382–393, 2015.

[LiuYu13] Liu, Y.; Yu, H.: A survey of underactuated mechanical systems. IET Control Theory Applications, Vol. 7, No. 7, pp. 921–935, 2013.

[MandelaEtAl10] Mandela, R.K.; Rengaswamy, R.; Narasimhan, S.; Sridhar, L.N.: Recursive state estimation techniques for nonlinear differential algebraic systems. Chemical Engineering Science, Vol. 65, No. 16, pp. 4548 – 4556, 2010.

[MaquedaMohamedShabana10] Maqueda, L.G.; Mohamed, A.N.A.; Shabana, A.A.: Use of general nonlinear material models in beam problems: Application to belts and rubber chains. Journal of Computational and Nonlinear Dynamics, Vol. 5, No. 2, pp. 021003–1 – 021003–10, 2010.

[Maslen95] Maslen, E.H.: Positive real zeros in flexible beams. Shock and Vibration, Vol. 2, pp. 429–435, 1995.

[Miu91] Miu, D.K.: Physical Interpretation of Transfer Function Zeros for Simple Control Systems With Mechanical Flexibilities. Journal of Dynamic Systems, Measurement, and Control, Vol. 113, No. 3, pp. 419–424, 1991.

[MoallemPatelKhorasani97] Moallem, M.; Patel, R.V.; Khorasani, K.: An inverse dynamics control strategy for tip position tracking of flexible multilink manipulators. Journal of Robotic Systems, Vol. 14, No. 9, pp. 649–658, 1997.

[MoallemPatelKhorasani01] Moallem, M.; Patel, R.; Khorasani, K.: Nonlinear tip-position tracking control of a flexible-link manipulator: theory and experiments. Automatica, Vol. 37, No. 11, pp. 1825 – 1834, 2001.

[MobergHanssen10] Moberg, S.; Hanssen, S.: Inverse Dynamics of Flexible Manipulators. Technical report from Automatic Control at Linköpings universitet. 2010. Report no.: LiTH-ISY-R-2939.

[MorlockBurkhardtSeifried16] Morlock, M.; Burkhardt, M.; Seifried, R.: Control concepts for a parallel manipulator with flexible links. Proceedings of Applied Mathematics and Mechanics, Vol. 16, No. 1, pp. 819–820, 2016.

[MurrayRathinamSluis95] Murray, R.M.; Rathinam, M.; Sluis, W.: Differential flatness of mechanical control systems: A catalog of prototype systems. In Proceedings of the 1995 ASME International Congress and Exposition, 1995.

[MuskeEtAl12] Muske, K.R.; Ashrafiuon, H.; Nersesov, S.; Nikkhah, M.: Optimal sliding mode cascade control for stabilization of underactuated nonlinear systems. Journal of Dynamic Systems, Measurement, and Control, Vol. 134, No. 2, pp. 021020–1 – 021020–11, 2012.

[NachbagauerZehetnerGerstmayr12] Nachbagauer, K.; Zehetner, C.; Gerstmayr, J.: Nonlinear Finite Element Modelling of Moving Beam Vibrations Controlled by Distributed Actuators. pp. 167–174. Vienna: Springer Vienna, 2012.

[NeupertEtAl06] Neupert, J.; Hildebrandt, A.; Sawodny, O.; Schneider, K.: Trajectory tracking for boom cranes using a flatness based approach. In SICE-ICASE, 2006. International Joint Conference, pp. 1812–1816, 2006.

[Oltmann11] Oltmann, J.: Analyse und Applikation eines neuartigen Regelkonzepts zur Lastschwingungsdämpfung am Containerkran. Diplomarbeit DIPL - 073, Institut für Mechanik und Meerestechnik, Technische Universität Hamburg-Harburg, 2011.

[OmarShabana01] Omar, M.; Shabana, A.: A two-dimensional shear deformable beam for large rotation and deformation problems. Journal of Sound and Vibration, Vol. 243, No. 3, pp. 565–576, 2001.

[OttoRückwaldSeifried19] Otto, S.; Rückwald, T.; Seifried, R.: Control of multibody systems in DAE form applied to a cable robot. In Proceedings of the XVIII International Symposium on Dynamic Problems of Mechanics, DINAME 2019, 2019.

[OttoSeifried17] Otto, S.; Seifried, R.: Real-time trajectory tracking of a cable-driven parallel robot using servo-constraints. PAMM, Vol. 17, No. 1, pp. 161–162, 2017.

[OttoSeifried18a] Otto, S.; Seifried, R.: Analysis of servo-constraints solution approaches for underactuated multibody systems. In Proceedings of the 5^{th} Joint International Conference on Multibody System Dynamics, Lisbon, Portugal, 2018.

[OttoSeifried18b] Otto, S.; Seifried, R.: Open-loop control of underactuated mechanical systems using servo-constraints: Analysis and some examples. In S. Campbell; A. Ilchmann; V. Mehrmann; T. Reis (Eds.) Applications of Differential-Algebraic Equations: Examples and Benchmarks, pp. 81–122. Springer, Cham, 2018.

[OttoSeifried18c] Otto, S.; Seifried, R.: Real-time trajectory control of an overhead crane using servo-constraints. Multibody System Dynamics, Vol. 42, No. 1, pp. 1–17, 2018.

[PatwardhanEtAl12] Patwardhan, S.C.; Narasimhan, S.; Jagadeesan, P.; Gopaluni, B.; Shah, S.L.: Nonlinear bayesian state estimation: A review of recent developments. Control Engineering Practice, Vol. 20, No. 10, pp. 933 – 953, 2012. 4th Symposium on Advanced Control of Industrial Processes (ADCONIP).

[Pham19] Pham, H.T.: Control methods of powertrains with backlash and time delay. Ph.D. thesis, Technische Universität Hamburg-Harburg, Hamburg, 2019.

[Popp10] Popp, K.: Ground Vehicle Dynamics. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2010.

[Radisch11] Radisch, C.: Analysis, improvement and implementation of a linear parameter-varying controller design for the active damping of container crane oscillations. Master Thesis, Technische Universität Hamburg-Harburg, 2011.

[Radisch18] Radisch, C.: Dynamikuntersuchungen und Regelungskonzepte zum Betrieb von Containerkranen. Mechanik. München: Dr. Hut, 1. Edn., 2018.

[RaoufMohamadMaarouf13] Raouf, F.; Mohamad, S.; Maarouf, S.: Workspace tracking control of two-flexible-link manipulator using distributed control strategy. Journal of Control Science and Engineering, Vol. 2013, p. 17, 2013.

[Rapp12] Rapp, C.: Aktive Dämpfung der Lastschwingungen bei Containerkranen. Ph.D. thesis, Technische Universität Hamburg-Harburg, 2012.

[Romero04] Romero, I.: The interpolation of rotations and its application to finite element models of geometrically exact rods. Computational Mechanics, Vol. 34, No. 2, pp. 121–133, 2004.

[Romero08] Romero, I.: A comparison of finite elements for nonlinear beams: the absolute nodal coordinate and geometrically exact formulations. Multibody System Dynamics, Vol. 20, No. 1, pp. 51–68, 2008.

[RothfußRudolphZeitz97] Rothfuß, R.; Rudolph, J.; Zeitz, M.: Flachheit: Ein neuer Zugang zur Steuerung und Regelung nichtlinearer Systeme. at - Automatisierungstechnik, Vol. 45, No. 11, 1997.

[RouchonEtAl93] Rouchon, P.; Fliess, M.; Évine, J.L.; Martin, P.: Flatness and motion planning: the car with n trailers. In Proc. European Control Conference, pp. 1518–1522, 1993.

[Rudolph03] Rudolph, J.: Flatness Based Control of Distributed Parameter Systems. Shaker, 2003.

[Sastry99] Sastry, S.: Nonlinear systems analysis, stability, and control. New York, NY [u.a.]: Springer, 1999.

[Schiehlen14] Schiehlen, W.: Applied Dynamics. Cham s.l.: Springer International Publishing, 2014.

[SchraderSain89] Schrader, C.B.; Sain, M.K.: Research on system zeros: a survey. International Journal of Control, Vol. 50, No. 4, pp. 1407–1433, 1989.

[SchultzMurphey12] Schultz, J.; Murphey, T.: Trajectory generation for underactuated control of a suspended mass. Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 123–129, 2012.

[Schwertassek99] Schwertassek, R.: Dynamik flexibler Mehrkörpersysteme : Methoden der Mechanik zum rechnergestützten Entwurf und zur Analyse mechatronischer Systeme. Grundlagen und Fortschritte der Ingenieurwissenschaften. Braunschweig u.a.: Vieweg, 1999.

[Seifried11] Seifried, R.: Optimization-Based Design of Minimum Phase Underactuated Multibody Systems. pp. 261–282. Dordrecht: Springer Netherlands, 2011.

[Seifried12] Seifried, R.: Integrated mechanical and control design of underactuated multibody systems. Nonlinear Dynamics, Vol. 67, No. 2, pp. 1539–1557, 2012.

[Seifried14] Seifried, R.: Dynamics of Underactuated Multibody Systems Modeling, Control and Optimal Design. Cham s.l.: Springer International Publishing, 2014.

[SeifriedBlajer13] Seifried, R.; Blajer, W.: Analysis of servo-constraint problems for underactuated multibody systems. Mechanical Sciences, Vol. 4, No. 1, pp. 113–129, 2013.

[Shabana98] Shabana, A.A.: Computer implementation of the absolute nodal coordinate formulation for flexible multibody dynamics. Nonlinear Dynamics, Vol. 16, No. 3, pp. 293–306, 1998.

[Shabana11] Shabana, A.A.: Computational Continuum Mechanics. 2011.

[ShampineGladwellThompson03] Shampine, L.F.; Gladwell, I.; Thompson, S.: Solving ODEs with MATLAB. Cambridge University Press, 2003.

[Siciliano08] Siciliano, B.; Khatib, O. (Eds.): Springer handbook of robotics. Berlin u.a.: Springer, 2008.

[SicilianoEtAl09] Siciliano, B.; Oriolo, G.; Sciavicco, L.; Villani, L.: Robotics Modelling, Planning and Control. London: Springer London, 2009.

[Simo85] Simo, J.: A finite strain beam formulation. the three-dimensional dynamic problem. part i. Computer methods in applied mechanics and engineering, Vol. 49, No. 1, pp. 55–70, 1985.

[Skogestad04] Skogestad, S.: Multivariable feedback control : analysis and design. Chichester u.a.: Wiley, reprinted. Edn., 2004.

[SlotineLi91] Slotine, J.J.E.; Li, W.: Applied nonlinear control. Englewood Cliffs, NJ: Prentice Hall, 1991.

[SpectorFlashner89] Spector, V.A.; Flashner, H.: Sensitivity of Structural Models for Noncollocated Control Systems. Journal of Dynamic Systems, Measurement, and Control, Vol. 111, No. 4, pp. 646–655, 1989.

[SpectorFlashner90] Spector, V.A.; Flashner, H.: Modeling and Design Implications of Noncollocated Control in Flexible Systems. Journal of Dynamic Systems, Measurement, and Control, Vol. 112, No. 2, pp. 186–193, 1990.

[Spong87] Spong, M.W.: Modeling and Control of Elastic Joint Robots. Journal of Dynamic Systems, Measurement, and Control, Vol. 109, No. 4, pp. 310–318, 1987.

[Spong94] Spong, M.: Partial feedback linearization of underactuated mechanical systems. Intelligent Robots and Systems '94. 'Advanced Robotic Systems and the Real World', IROS '94. Proceedings of the IEEE/RSJ/GI International Conference on, Vol. 1, pp. 314–321, 1994.

[Spong98] Spong, M.W.: Underactuated mechanical systems. pp. 135–150. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

[Spong06] Spong, M.W.: Robot modeling and control. Hoboken, N.J.: Wiley, 2006.

[Strang10] Strang, G.: Wissenschaftliches Rechnen. Berlin u.a.: Springer, 2010.

[Strang12] Strang, G.: Computational science and engineering. Wellesley, Mass.: Wellesley-Cambridge Press, 2012.

[SugiyamaSuda09] Sugiyama, H.; Suda, Y.: Non-linear elastic ring tyre model using the absolute nodal coordinate formulation. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics, Vol. 223, No. 3, pp. 211–219, 2009.

[Svaricek06] Svaricek, F.: Nulldynamik linearer und nichtlinearer Systeme: Definition, Eigenschaften und Anwendungen. Automatisierungstechnik, Vol. 54, No. 7, pp. 310–322, 2006.

[TalebiPatelKhorasani04] Talebi, H.A.; Patel, R.V.; Khorasani, K.: A Neural Network Controller for a Class of Nonlinear Non-Minimum Phase Systems with Application to a Flexible-Link Manipulator. Journal of Dynamic Systems, Measurement, and Control, Vol. 127, No. 2, pp. 289–294, 2004.

[LabVIEW15] LabVIEW. http://www.ni.com/labview/release-archive/2015/d/, 2015. Accessed: 2019-11-11.

[Theis10] Theis, J.: Beobachterentwurf für einen räumlich pendelnden Körper mit steuerbarer Aufhängung. Bachelorthesis, Teschnische Universität Hamburg-Harburg, 2010.

[ThrunBurgardFox06] Thrun, S.; Burgard, W.; Fox, D.: Probabilistic robotics. Cambridge, Mass. [u.a.]: MIT Press, 2006.

[TianEtAl07] Tian, Q.; Zhang, Y.Q.; Chen, L.P.; Yang, J.C.Y.: Two-link flexible manipulator modelling and tip trajectory tracking based on the absolute nodal coordinate method. In 13th National Conference on Mechanisms and Machines (NaCoMM07), pp. 251–258, 2007.

[VoharKeglRen08] Vohar, B.; Kegl, M.; Ren, Z.: Implementation of an ANCF beam finite element for dynamic response optimization of elastic manipulators. Engineering Optimization, Vol. 40, No. 12, pp. 1137–1150, 2008.

[WangVidyasagar91] Wang, D.; Vidyasagar, M.: Transfer functions for a single flexible link. The International Journal of Robotics Research, Vol. 10, No. 5, pp. 540–549, 1991.

[WasfyNoor03] Wasfy, T.M.; Noor, A.K.: Computational strategies for flexible multibody systems . Applied Mechanics Reviews, Vol. 56, No. 6, pp. 553–613, 2003.

[Williams92] Williams, T.: Transmission zeros of non-collocated flexible structures - Finite-dimensional effects. pp. 356–364. Dynamics Specialists Conference, 1992.

[Woernle13] Woernle, C.: Trajectory Tracking for a Three-Cable Suspension Manipulator by Nonlinear Feedforward and Linear Feedback Control. pp. 371–386. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[YangBetschZhang20] Yang, Y.; Betsch, P.; Zhang, W.: Numerical integration for the inverse dynamics of a large class of cranes. Multibody System Dynamics, Vol. 48, No. 1, pp. 1–40, 2020.

[YangKrishnanAng99] Yang, H.; Krishnan, H.; Ang, M.H.: Tip-trajectory tracking control of single-link flexible robots via output redefinition. In Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Vol. 2, pp. 1102–1107 vol.2, 1999.

Acronyms

ANCF	absolute nodal coordinate formulation
BDF	backwards differentiation formulas
BVP	boundary value problem
DAE	differential-algebraic equation
IMU	inertial measurement unit
LQR	linear quadratic regulator
LTI	linear time-invariant
MBS	multibody system
MIMO	multi-input multi-output
ODE	ordinary differential equation
Profibus	Process Field Bus
SISO	single-input single-output
SMC	sliding mode controller
UKF	unscented Kalman filter