TUM School of Engineering and Design
Technical University of Munich

Trajectory control of a flexible robot arm based on port-Hamiltonian numerical models

Mei Wang

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung des akademischen Grades einer

Doktorin der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. ir. Daniel J. Rixen
Prüfer der Dissertation:	1. Priv.-Doz. Dr.-Ing. habil. Paul Kotyczka
	2. Prof. Dr. Bernhard Maschke

Die Dissertation wurde am 05.10.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 14.02.2022 angenommen.

TO MY FAMILY.

Abstract

This dissertation presents a systematic design procedure, from modeling, analysis, numerical approximation to control design, for distributed parameter systems in one spatial dimension using the energy-based port-Hamiltonian (PH) system representation. The investigated control task is to realize the precise observer-based trajectory tracking of the tip of a flexible robot arm, which was designed and built at the Chair of Automatic Control as a practical outcome of this research.

The first design steps are structure-preserving spatial discretization of the flexible structure by means of a pseudo-spectral method and structure-preserving order reduction. The model structure is exploited for the inversion-based feedforward control and the observer-based control design in a two-degrees-of-freedom control scheme with model-based disturbance attenuation. Taking into account the time delays of the used industrial PLC, the presented methods are extended with the systematic consideration of time delays in a discrete-time control design. Experimental results for the tracking and disturbance behaviors of the flexible robot arm illustrate the quality of the design.

Acknowledgments

This present manuscript summarizes my research from 2015 to 2018 at the Chair of Automatic Control, Technical University of Munich. The result of this work is intended to bring the theory and the application of modern control methods closer together.

For the generous support of the work and trust placed in me, I sincerely appreciate that Professor Boris Lohmann allowed me to do my research at his chair. Especially, I am indebted to my supervisor, Priv.-Doz. Dr. Paul Kotyczka, for his always targeted supervision and constructive support of this research as well as the awakening of my interest in the wonderful port-Hamiltonian world. He gave me generous freedom to develop and implement ideas and strongly shaped my perspective on scientific work in general. I am grateful to Professor Bernhard Maschke for his continuous interest in this work and the willingness to be the second examiner. I have an unforgettable experience of my three-week research visit at the Laboratory of Automatic Control, Chemical and Pharmaceutical Engineering (LAGEPP) in Lyon in 2016. I would like to thank Professor Laurent Lefèvre for our discussions in Valence, in Bertinoro and during the IFAC World Congress 2017 in Toulouse. His professional competence also greatly inspired this work.

I want to thank my colleagues Dr. Maria Cruz, Dr. Nicole Gehring, Philipp Niermeyer and Linchong Gao for the kind discussions and helpful suggestions to my work. I express my grand gratitude to my colleagues Thomas Huber, Ralf Hübner and Klaus Albert, who helped me to build the laboratory manipulator.

I gratefully acknowledge the Deutsche Forschungsgemeinschaft (DFG) for supporting my whole PhD work.

My deepest gratitude goes to my parents and my sister. Their unconditional love and help over all the years brought me to the best. Many thanks and great love to my dear husband, Changyu, for his support, patience and encouragement in my life! During the time, the birth of Dongyan, my daughter, makes me understand the power of life: **Never give up**!

Garching, September 2021
Mei Wang

Contents

Chapter 1

Introduction

1.1 Motivation and Contributions

Modern control theories are committed to providing modeling, analysis, and control for even more complex systems. At the mention of complexity, on the one hand, the products that are smarter, lighter, and more powerful are becoming more popular and in demand. This evolution requires not only the development of new sensors and actuators but also the development of lightweight materials for such products. However, most of these components are difficult or even impossible to describe on the finite-dimensional space. In the field of production, logistics, etc., on the other hand, more distributed systems with decentralized degrees of freedom and more interactions between different controlled subsystems are required. As the systems grow in network size, a traditional central controller will no longer work. Moreover, the modeling for the connection of a physical system with the additional components has to be completely rebuilt according to the traditional modeling approaches, see the details in [14].

The *port-Hamiltonian* (PH) framework has been designed for the network modeling of physical systems with multiple domains such as mechanical, electrical, pneumatic, and hydraulic systems, etc., see [57]. The PH approach is able to describe not only lumped parameter systems on finite-dimensional spaces but also distributed parameter systems in the infinite-dimensional case. The former are mathematically described by ordinary differential equations (ODEs), while the latter are represented by partial differential equations (PDEs). The modeling paradigm is based on an underlying geometric structure (a *Dirac structure* for finite-dimensional systems) that represents the exchange of energy between energy forms and subsystems, and with the environment or other coupled systems, via pairs of *power/port variables*. For this reason, complex systems can be broken down into several smaller, interconnected components. In addition, the modularity of the modeling process enables the iterative execution of modeling and design tasks [9]. Due to their inherent passivity, PH models are a natural foundation for energy-based control schemes, e.g., *Control by Interconnection* (CBI) [65] or *Interconnection and Damping Assignment Passivity-based Control* (IDA-PBC) [62]. An overview of the various aspects of the PH framework can be found in the books [14], [86].

The purpose of this dissertation is to show how the PH framework can be exploited in the various steps of motion control design for an elastic rotating beam, which serves as a prototypical example of a flexible robot arm. The rotating beam is modeled as a

Timoshenko beam, see [52], and spatially discretized by the geometric pseudo-spectral approach described in [60] to preserve the structural properties in the lumped parameter model. In order to use the finite-dimensional model in real time for the control design, the discretized beam model is reduced by structure-preserving order reduction, see [30]. Based on the resulting (reduced) lumped parameter model, an inversion-based feedforward control (ibFFC) as described in [107] is designed and complemented by the observer-based state difference feedback.

This research was funded by Deutsche Forschungsgemeinschaft (DFG), project No. 260049780. The main novel contributions of this thesis are:

- *Structure-preserving discretization* and *model order reduction* of the Timoshenko beam using a pseudo-spectral method and a Krylov subspace method, respectively.
- The solution of a highly dynamic motion control task in a *holistic port-Hamiltonian frame*, from modeling, approximation, interconnection up to control design.
- The *design* and *construction* of a modular three-axis lightweight manipulator controlled by the industrial PLC (Programmable Logic Control).
- *Comparative experimental studies* of the control performance with a single-link flexible robot arm configuration, without and with the inclusion of communication time delays.

The dissertation mainly summarizes previous results on subsystem modeling, analysis, numerical approximation (spatial discretization and model order reduction for real-time implementation) and feedforward and feedback control design in a two-degrees-of-freedom structure with model-based disturbance attenuation [93], [97], [94], [95], [96], and completes it by the building of the laboratory manipulator based on the student works [105], [15], [32], [58], as well as the experimental results of different control strategies illustrating the tracking and disturbance behavior in practice.

1.2 Literature Review

As far as modeling, analysis, numerical approximation, and control of distributed parameter systems, the last 15 years have produced a number of important results. In the context of this dissertation, the following references are of particular importance in the PH framework: A functional analysis framework for linear distributed parameter systems in a one-dimensional (1D) spatial domain is presented in [36], where the characterization of the boundary conditions [46] defining a *Stokes-Dirac structure* [87] leads to a well-posed system description. Furthermore, in this section, we give an overview of the terms and methods and summarize some relevant references.

Modeling

The PH approach provides a systematic procedure for modeling (complex) physical systems (e.g., multi-body mechanical systems) based on the power-conserving interconnection of basic components (e.g., rigid bodies, flexible links, or kinematic pairs), see [54], [57]. The PH structure represents the power flows between subsystems and the environment, as well as the energy storage, conversion, and dissipation, see [14]. For the

mathematical description of the finite-dimensional systems, the so-called Dirac structures are crucial, which are described in [6], [7]. The Dirac structure is the linear subspace of power-conjugated port variables, in which the power conservation holds. It characterizes an underlying geometric structure of finite-dimensional PH systems. Its constitutive equations are derived from the total energy of the system, which is considered as Hamiltonian. The examples of Dirac structures for different physical systems are given in [14].

For the infinite-dimensional case, the systems can be described as distributed parameter PH systems, whose structures are considered as the extension of Dirac structures on infinite-dimensional spaces, so-called Stokes-Dirac structures, see [86]. Similar to the finite-dimensional case, the pairs of port variables on the spatial domain and on the boundary are associated to hold the power conservation on the Stokes-Dirac structure, see [87]. For control purposes, the boundary port variables can be defined as power-conjugated inputs and outputs, whose products represent the power exchange at the ports, see [46]. Unlike the finite-dimensional case, the constitutive equations are derived from a Hamiltonian functional, e.g., for a hyperbolic system of conservation laws.

The central task in the modeling and control of the elastic robot is the treatment of the vibrations of the flexible structure, which are described by PDEs. [72] gives a comprehensive overview of different PH representations for flexible mechanical systems. Beams are the simplest examples of flexible mechanical structures, see [67], [76], and [84] as an incomplete selection of texts on modeling, dynamical analysis and numerical approximation. The modeling and control of the Timoshenko beam in PH form were described in [52], exploiting the representation in energy variables and the Stokes-Dirac structure, while [46] approaches these tasks with a different choice of state variables and a jet bundle formulation. Taking into account the gravitational field, [78] extended the Hamiltonian density in terms of a gravitational potential density, see also [71]. An alternative PH beam model can be represented by a second-order differential operator based on the Euler-Bernoulli theory, see [4]. Both beam theories mentioned are based on the small deformation assumption. To describe the large deformation, [53], [85] described nonlinear beam models in PH framework according to the concept of the Stokes-Dirac structures.

Structure-Preserving Discretization

The fundamental problem of simulation and (early lumping) control of complex physical systems, including distributed parameter systems, focuses on finite-dimensional approximation. The classical discretization schemes, e.g., finite element methods, have the disadvantage that the structure of the system cannot be preserved since they define the basis functions before the actual spatial discretization. This can lead to undesired numerical instabilities or dissipation, see [88]. Therefore, *geometric* or *structure-preserving discretization schemes* are required to preserve certain (geometric or structural) properties in the lumped parameter model. In the context of PH systems, the basic idea of structure-preserving discretization is to take into account the different geometric nature of the power variables in the numerical approximation of their spatial distributions. This means that the infinite-dimensional (Stokes-)Dirac structure can be approximated by a finite-dimensional counterpart.

Several discretization approaches have been proposed in recent years. An early approach to discretize 1D distributed parameter systems while preserving their structure is presented in [28] using different finite element spaces for the dual power variables. This approach has been applied to an ideal transmission line described by the telegrapher's equation, to the 2D wave equation, and to the 1D Timoshenko Beam model, see [28], [102]. In addition, [3] proposed the partitioned finite element method (PFEM) which was used for the mindlin plate. Compared to this local discretization scheme, [60] presents a pseudo-spectral method in which global higher-order polynomials, e.g., Lagrange interpolation polynomials, are used as basis functions to approximate the distributed power variables at the collocation points. This approach was illustrated in the 1D resistive diffusion equation [90] and the 1D beam models [4], [97]. While using this global approach for the beams, the so-called shear locking phenomenon and spurious modes, which occur in the finite element analysis of the thin structure problems (see e.g. [8]), are not involved. This is also one of the reason why this approximation approach is chosen for our application. [38] proposed finite volume methods for open PH systems of conservation laws on staggered grids based on conservative generalized leapfrog schemes. It is able to define the boundary power variables on different grids in terms of their different geometric natures. A more detailed summary of the state of the art in numerical methods for distributed parameter PH systems can be found in [39].

Spatial discretization of the distributed parameter system can result in a large-scale lumped parameter system that is difficult to simulate or use for control design. Therefore, a so-called *structure-preserving model order reduction* is required to generate a suitable finite-dimensional model, see [27], [66], [100].

Feedforward Control

In practice, e.g. in production, elastic robots are used to track a predefined movement or to reach the desired position. In this case, a two-degrees-of-freedom control structure as described in [43] is considered. Control performance and accuracy can be improved by adding a feedforward control using a high-quality model. Based on the Timoshenko beam model, [70] used a flatness-based feedforward control approach to achieve finite time transitions between two equilibria. The state and input trajectories can be described by a flat output. The solution of PDEs, obtained using Mikusiński operators, provides the open-loop control of the beam. Further applications of the flatness-based feedforward control are presented for a heat-conducting rod on 1D space in [59], and for a fixed-bed tubular reactor in [18].

To avoid solving PDEs and to allow a feedforward control design for multi-dimensional systems, spatially discretized models are used, see [25]. One commonly used method is the so-called *inversion-based feedforward control* (ibFFC), which exploits the inverse plant model for control design [13]. However, for non-minimum phase systems, the classical inversion leads to internal instability. A bounded solution of the ibFFC was proposed in [107], [5], [106] using bounded state transition matrices. Some applications of this approach to the flexible beams can be found in [49], [12], [74] and in the PH framework [40] and [97].

Feedback Control

In order to suppress the residual oscillations and to achieve the highly dynamic motion of a flexible robot, (some) state variables must be fed back into the controller. In industry, the motor cascade control is a common, simple algorithm for robotics [24]. This type of controller does not depend on the plant model and can be adjusted according to the desired behavior. However, knowledge of the motor states alone is not sufficient to suppress the elastic robots due to their flexible structures. For this reason, in many applications, the beam deformations are additionally recorded with strain gauges or the beam acceleration and/or the angular velocity with internal measurement units (IMUs) for the feedback loop. [24] combined a PD joint controller with a torque feedforward control and a curvature feedback control for a three-axis elastic robot arm. Similar applications can be found in [26], [37] for two-axis robot arms. In [104], the acceleration of the beam tip was used for active vibration damping.

The PH approach is not only suitable for network modeling but also for controlling multi-physics systems. Energy shaping is the core of passivity-based control (PBC) and achieves (asymptotic) stabilization of the closed-loop systems by choosing the desired energy function, which may be a Lyapunov function candidate for the desired equilibrium. Based on this concept, the commonly used CbI and IDA-PBC were developed in [63], [65], [64], [62]. CbI adds a PH dynamic controller with a simple feedback interconnection to the PH system, while IDA-PBC controls the desired closed-loop dynamics of the PH system by solving matching constraints that restrict the interconnection and damping matrices as well as the closed-loop Hamiltonian such that the open-loop PH system becomes feedback-equivalent to a passive system. [51], [77] used the CbI technique for the distributed PH Timoshenko beam model, and [79] proposed a flatness-based feedforward control and an IDA-PBC for a three-axis elastic robot arm, where the acceleration and the angular velocity were recorded by an IMU for the feedback loop.

1.3 Overview of Contents

This dissertation is structured as follows. After outlining the motivation and highlights of this research in **Chapter 1**, the test bench of a flexible robot arm built in the frame of the project is presented in **Chapter 2**. The components of the servo system and the programmable logic control system (PLC) are described in detail in each section. It also presents the software configuration that has been set up in our application. Based on the system description and characteristics, some problems and limitations of the used real-time control system are pointed out, as they have a direct influence on the investigation of the chosen control algorithms and the corresponding control quality.

In **Chapter 3**, we first briefly present the necessary preliminaries of PH systems. We then summarize the PH modeling of a flexible robot arm, including structure-preserving discretization and order reduction of the beam model. The approximation quality of the discretized beam model is analyzed in terms of the eigenfrequencies and eigenfunctions as well as the transfer function. Moreover, different reduced PH beam models are compared with respect to the Bode diagram of the selected input and output.

In **Chapter 4**, we mainly show how to exploit the structure of the (reduced) beam model to design the model-based feedforward control. In order to generate a feasible trajectory for the tracking task, a point-to-point motion profile with sinusoidal interpo-

lation is introduced. Compared to the rigid model for control design, the problem of zero dynamics arises when the structural flexibility of the beam model is considered. To solve this problem, a stable inversion-based feedforward approach is applied to provide a bounded solution.

To compensate for the inaccuracy of the model and the unknown disturbances, a feedback loop is added to the control structure, resulting in the two-degrees-of-freedom control scheme. Since friction in servo motors plays an essential role in robot dynamics, it can be compensated on the one hand by a static friction model; on the other hand, it can be estimated and compensated online by a disturbance observer. **Chapter 5** presents a classical PD joint controller with static friction compensation and experimental friction identification with different models. In **Chapter 6**, the resulting PH model is used for the observer-based control design with model-based disturbance attenuation. Since some time delays exist in the chosen industrial control system that can potentially lead to degradation (up tp instability) of the control performance, the observer-based approach can be extended with a time-delay model that includes both the input and output time delays to make the control effect robust, see **Chapter 7**.

Chapter 8 demonstrates the implementation of the various control strategies. The experimental results illustrate the tracking and disturbance behaviors in practice.

Chapter 9 provides concluding remarks and directions for future work.

Chapter 2

Test Bench of the Flexible Manipulator

Figure 2.1: Modular, three-axis lightweight manipulator with structural elasticity[1]

The aim of this research is to find a uniform design procedure of the distributed parameter systems in the PH framework and also to validate the control methods in the experiments. For this purpose, a modular, three-axis lightweight robot with an industrial PLC was constructed in this work, see Figure 2.1. This manipulator consists of three gear motors and two flexible links. The upper arm consists of a steel square tube (length: 0.7m, edge length: 0.025m, wall thickness: 0.003m), whereas the lower arm is made of an aluminum square tube (length: 0.616m, edge length: 0.02m, wall thickness: 0.002m).

In the context of this dissertation, we focus on its uniaxial flexible link configuration, on which the experimental validation is performed.

[1] Adopted from an internal document at the Chair of Automatic Control, TU Munich.

2.1 Single-Link Flexible Robot Arm

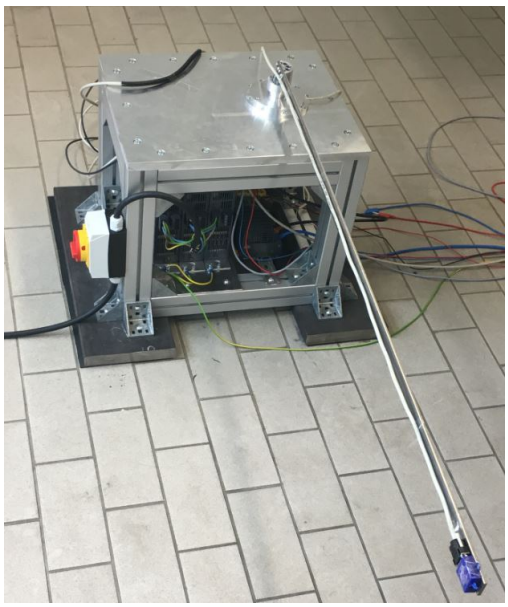


Figure 2.2: Photo of the Laboratory manipulator in its uniaxial flexible link configuration

Figure 2.2 is a photo of a single-link flexible robot arm. The test bench mainly consists of an industrial PLC, a servo drive, a servo motor, sensors, a steel beam as well as some safety components. The beam is mounted horizontally on the motor hub, which is driven by a servo motor. The mechanical part is a thin beam with the length of 1.2m, the height of 0.03m and the thickness of 0.005m. As a result, it oscillates with a fundamental natural frequency of about 3Hz.

Figure 2.3 shows the components of the test bench. The servo motor (RoboDrive Motor Gear Unit) RD85x13-HD i=160, manufactured by TQ-Systems GmbH, serves to drive the whole system. It contains a brushless DC electric motor (BLDC motor) and a Harmonic Drive (HD) gear. The transmission ratio is 160. The encoder attached to the motor communicates with the servo drive via a BiSS interface (bidirectional/serial/synchronous) and generally provides the servo drive with motor position and speed. The movement of the servo motor is controlled by an ACOPOS 8V1060.50 servo drive from B&R Industrial Automation GmbH. The motor control concept is based on the cascade structure integrated inside the servo drive. Various control approaches can be programmed in MATLAB/Simulink and then translated into machine-readable code for the PLC by automatic code generation. The data between the PLC and ACOPOS can be cyclical transferred by the Ethernet bus system named POWERLINK with a minimum cycle time of 400μs. An IMU is attached to the beam tip to capture the tip angular velocity and linear acceleration. The data between PLC and IMU is transmitted cyclically via the Controller Area Network (CAN) connection.

The products of the selected components in the test bench are listed in Table G.1. In the next sections, the basic concepts of the essential components are introduced.

[2]Adopted from Paul Kotyczka: internal document.

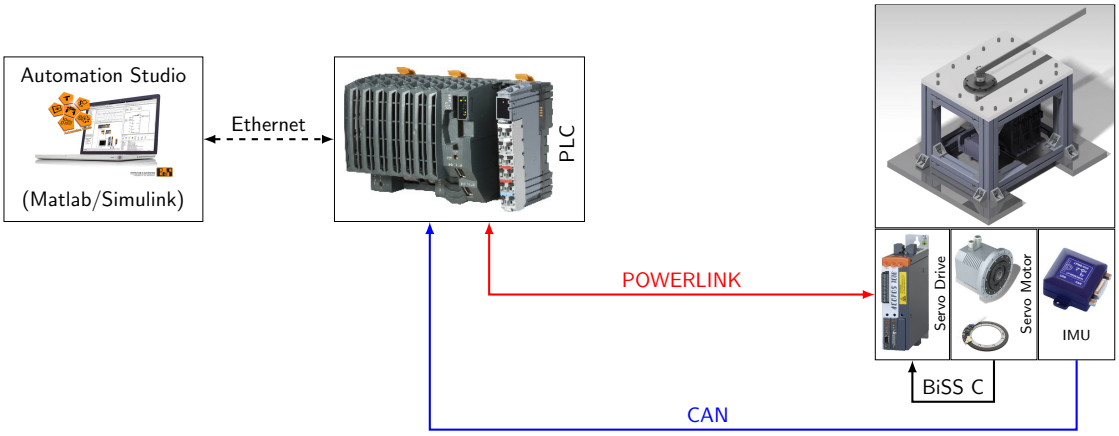


Figure 2.3: Components of the test rig[2]

2.2 Servo System

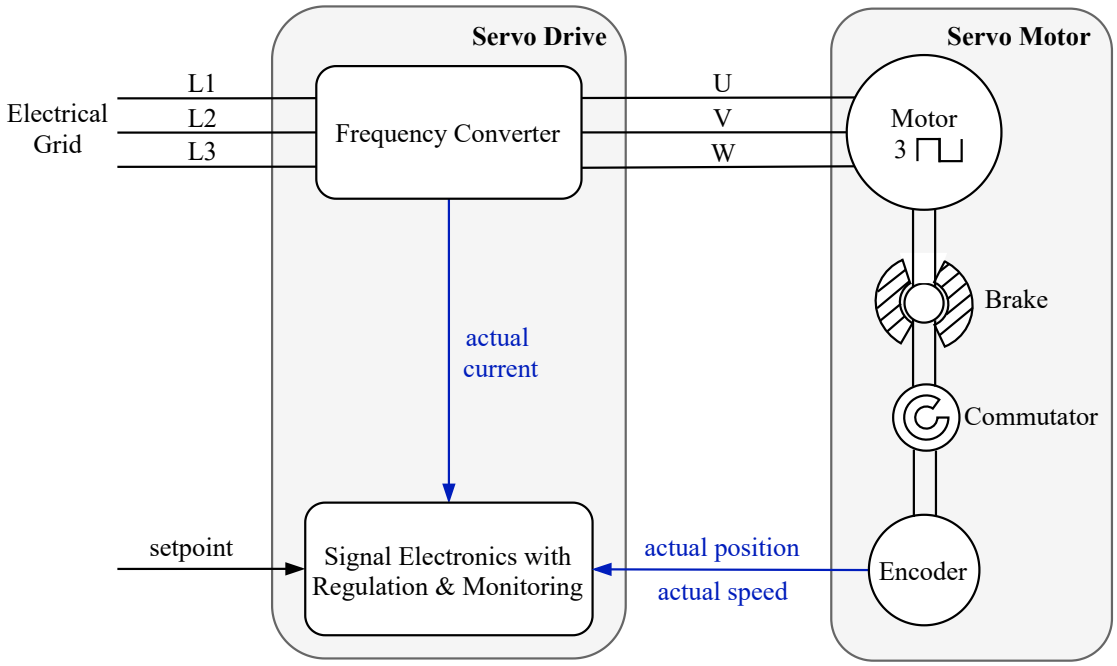


Figure 2.4: Construction of a servo system based on [98]

Figure 2.4 shows the structure of a servo system that corresponds to the one used in our application. In general, it consists of three primary components, namely a servo motor, a servo drive (mainly refers to a servo amplifier), and a feedback mechanism (in blue), see [98].

In many applications, a control system calculates the desired trajectory and triggers the servo drive with a generated command signal in relation to the desired position, speed, or torque. The servo drive amplifies this signal into electric voltage or current for the servo motor to achieve the required movement. At the same time, encoders detect the actual states of th motor and send them back to the servo drive, which compensates the

difference between the setpoint and actual states of the motor by adjusting the motor current.

2.2.1 Servo Motor

A servo motor is a high-performance actuator used for a precise control of angular or linear position, velocity, and acceleration. It contains a motor equipped with a resolver or a rotary encoder for position and speed feedback, see the servo motor in Figure 2.4. In general, servo motors are characterized by a low moment of inertia and high peak torque. The servo drive imposes the currents required for torque build-up very dynamically. They are able to operate over a wide speed range without overheating during acceleration and braking. The motor brakes must be also integrated into the robot arms in order to maintain the position of the arms in the event of a power failure.

Depending on the power source, the servo motors can be either DC motors or AC motors. A DC motor is an electric motor driven by direct current, such as batteries, DC power supplies, or an AC/DC power converter, while AC motors are fed by an alternating current or power converter. The motors typically used are brushed permanent magnet DC motors, the synchronous and induction motors (also called asynchronous motors). A DC motor with a brushed permanent magnet may change the direction of movement by means of an H-bridge. In contrast, the last two types use variable frequency drives to control their speed or motion. Their difference is that the synchronous motor could rotate at a rate that is locked to the grid frequency, whereas the rotation of an asynchronous motor depends on the current induction in the rotor winding so that it has to rotate slightly slower than the alternating current, see details in [98]. Today, servo motors are used very widely in robotics, in machine tools for wood- and metal-working, in printing machines, etc.

A BLDC motor is used in our servo system to execute the movement. It is also known as an electronically commutated motor (EC motor). Since its design corresponds to a permanent-magnet synchronous motor, it is also called a synchronous DC motor. Compared to a normal DC motor, the BLDC motor is commutated via an electrical circuit so that a mechanical commutator is no longer required. The stator consists of coils and the rotor is permanently excited. The commutation, that is, excitation and reversal of the current direction is carried out by the servo drive. The BLDC motor is normally coupled to a rotary encoder. There are two main types of rotary encoder: an incremental one retains the information about the motion of a shaft such as the changing position or speed, while an absolute one indicates the current shaft position, which is immediately available when power is applied. In our servo motor, an absolute encoder is fixed both on the motor side and on the load side, respectively, to determine the elasticity of the gearbox .

According to the motor datasheet, see Table G.2 for RD85x13-HD motor, the motor must first be configured before commissioning, see details in [15].

2.2.2 Servo Drive

The servo drive is an important component for determining the performance of a servo system. The essential role of the servo drive is to receive low-power command signals from a motion controller and amplify them to a specific amount of high-power voltage

and current for a servo motor to perform the required motion. For this reason, servo drives are sometimes called servo amplifiers.

It is noteworthy that the servo drive has a simple control logic for the motion control of the motor. Depending on the application, the servo drive can control the position, velocity, and torque of a motor. The servo drive attempts to adjust the voltage, frequency, or current of the motor to correct the difference between the actual and command signals. An additional motion control from a high-level control system can be superimposed to generate the setpoints for the motor control loop.

In our test rig, we use the ACOPOS servo drive family provided by B&R, which is capable of working with a fairly short sampling and communication time.

2.2.3 Cascade Control Structure

The core component of the B&R servo amplifier is the cascade control, which consists of the position, speed, and current control loop from the outside to the inside, see Figure 2.5. Table 2.1 describes the relevant parameters related to Figure 2.5. The subscript M designates the motor variables on the load side, while m designates that on the motor side. The blue-marked system is derived from the mechanical and electrical equations of a BLDC motor.

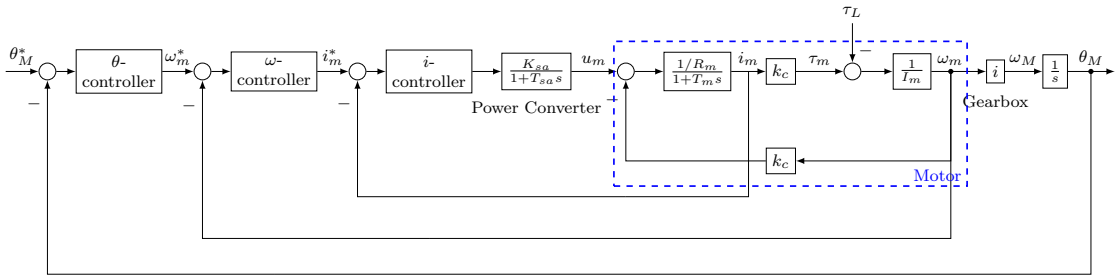


Figure 2.5: Cascade control structure of a servo motor according to [19]

Parameter	Description
θ_M	Load angle
ω_M	Load angular velocity
ω_m	Angular velocity of the motor
i_m	Rotor current
u_m	Voltage
M_m	Motor torque
M_L	Load torque
k_c	Motor constant

Table 2.1: Parameters of motor cascade control

The servo control is responsible for controlling the motor speed and position based on feedback signals. In most cases, the basic servo loop is the speed loop. To minimize the difference between the speed setpoint ω_m^* and actual value ω_m, the speed controller generates a corresponding torque. Since the torque generated by the motor is proportional

to the armature current, most industrial servo controllers rely on the current loop. The current loop takes the setpoint i_m^* from the speed loop and evaluates it with the current feedback i_m in order to set the required voltage u_m to the servo amplifier. When the system requires more torque, the current controller increases the voltage applied to the motor until the required current is reached. Compared to the speed loop, the current loop operates at much higher sampling rates. To prevent damage to BLDC motors due to unwanted and unsuitable control voltage, the current controller of the B&R servo amplifier is not accessible and must not be adjusted by the user. The current loops for motor control are normally tuned by the manufacturers.

For motor positioning, the motor angle[3] θ_M is considered the control target. The control structure can be extended with the position control loop outside the speed control. Apart from the angular velocity ω_m and the armature current i_m, the motor angle θ_M is required to be fed back to the $\theta-$controller. The position loop generates a reference speed for the speed loop. For further details, refer to [103].

Cascade control requires auxiliary control variables in order to set up the subordinate control loops. This control strategy is generally used for motor control due to its outstanding advantages. On the one hand, the inner control loop acts faster than the outer one and is completely unaffected by the slow dynamics of the outer loop. On the other hand, the slower outer loop could hardly be affected by the disturbances in the fast inner loop.

In the B&R servo amplifier, the control loops can be activated separately using the corresponding B&R function blocks, e.g., torque control loop. Instead of cascade control, the model-based motion control approaches for the manipulator are implemented and evaluated in the experiments. For this purpose, the outer control loops (speed and position) are deactivated, while the torque control loop is activated. In the next section, the torque control loop is briefly described.

Torque Control Loop

The tuning of the current controller gain is based on e.g. the motor stator inductance and resistance. The torque generated by the motor is proportional to the current in the windings and is based on the following relationship

$$\tau_m = \frac{k_c}{\sqrt{2}} \cdot i_m, \tag{2.1}$$

where k_c represents the motor constant, see Table G.2. For this reason, it is also considered a torque control loop. Since the electrical part of the BLDC motor is neglected in our model, the motor torque is the unique control input of the whole system. Since it is not permitted to access the torque (current) control loop, the torque trajectory calculated by the model-based control approach is specified as the torque setpoint for the torque control loop of the servo amplifier. Due to the maximum current, the motor torque must be limited. As already mentioned, when the torque control mode is activated, the other outer control loops for speed and position are automatically deactivated by the servo system to avoid overdetermination, see Figure 2.6. This control mode can be easily activated with the aid of the function block *MC_BR_TorqueControl*, see details in Section 2.3.2.

[3] In this case, the gear elasticity is ignored. Thus, $\theta_M = \frac{\theta_m}{i}$.

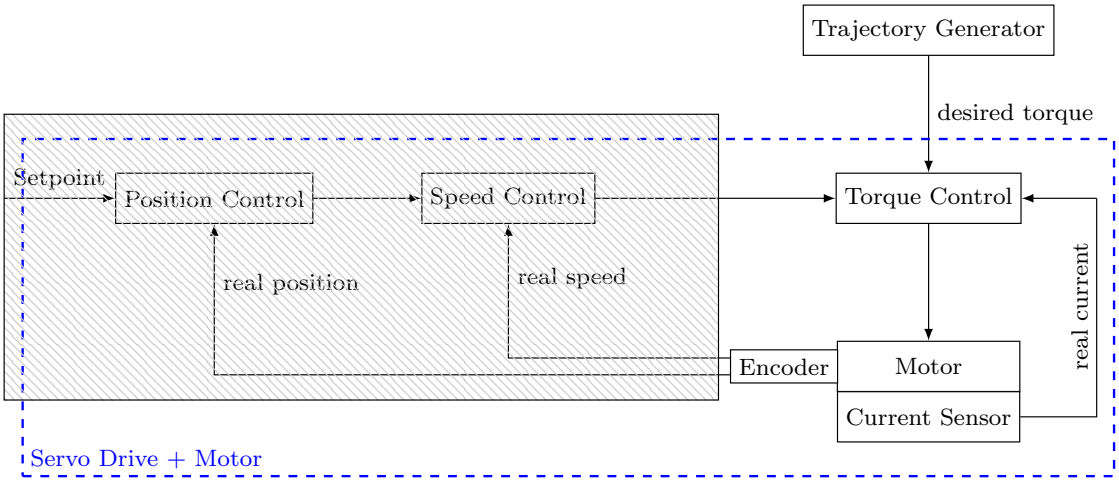


Figure 2.6: Schematic of the torque control loop

2.3 Programmable Logic Control

A PLC is a digital electronic system adapted to control manufacturing processes to ensure highly reliable control in any harsh environment. PLCs are used in a variety of industries, including hardness testing machines, injection molding machines, and fully automated production systems. It has a programmable memory for internal storage of user-oriented control commands for the implementation of exceptional features, such as logic control, sequence control, timer, counter, and arithmetic functions for controlling various types of machines and processes using digital or analog input and output signals, see [2].

PLCs have a robust design, are easily programmable, reliable, and have process fault diagnostics. Figure 2.7 shows an architecture of a typical PLC that includes the basic functional components of a central processing unit (CPU), a power supply unit, an input/output interface section, and a programmable device. Actuators are connected to the output interface of the PLC, while the sensors are located at the input interface. The PLC is capable of continuously monitoring the inputs of sensors and make output decisions to operate the actuators based on the program written by the user so that the machine operates in an intended way. The application program developed in the programming device is transferred to the memory unit of the PLC.

With regard to a PLC, it is implemented in different ways. This means that it can be realized as a single device, a software emulation, a PC plug-in card, etc. There are also modular solutions consisting of several components that are either plugged into a common rack or into a bus with expandable I/O capabilities. Each PLC system contains at least one CPU module, one power supply module, one or more I/O modules that are plugged together in the same rack.

- **CPU**
  The CPU contains a central processor and its memory. All necessary calculations and data processing are performed by the processor, which is capable of receiving the inputs and generating the corresponding outputs. The PLC memory can be divided into Read-Only Memory (ROM) and Random-Access Memory (RAM). The ROM memory consists of the operating system, the driver and the applica-

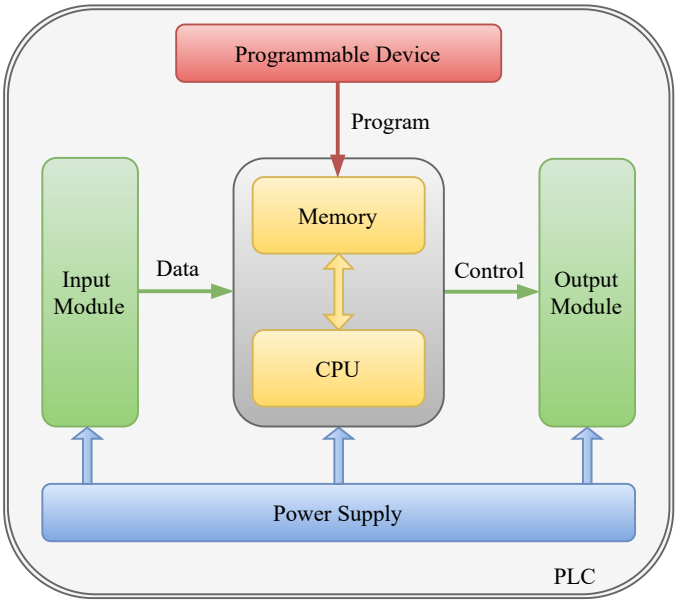


Figure 2.7: PLC architecture based on [2]

tion program, while the programs and working data written by the user are stored in RAM. If the power supply is interrupted or fails, the PLC uses the retentive memory to save user programs and data as well as to resume the execution of the user program after the power is restored. The retentive memory is configured by the user so that the values can be obtained by a power cycle or a program to execute a transition. Long-life batteries, EEPROM (Electrically Erasable Programmable Read-Only Memory) modules, and flash memory methods are required to implement retentive memory.

- **Power supply module**
  These modules are responsible for providing the required power to CPU and I/O modules by converting AC power to DC power.

- **I/O module**
  I/O modules of a PLC are used to connect the actuators and sensors to execute or measure real-time variables such as position, temperature, voltage, and pressure, etc. In general, the actuators could be electronic valves, drive control modules, power contactors for switching electric motors, etc. Limit switches, encoders, light barriers, and level sensors are usually selected as sensors. The I/O modules are divided into digital and analog types. The digital ones are connected to the digital sensors or actuators, which can be switched on or off for this purpose only, while the analog I/O modules are used for the sensors or actuators that generate analog electric signals such as voltage. To convert the analog signals into digital signals that can be read by the PLC, these modules usually have an integrated A/D converter.

Nowadays, PLC is becoming more and more popular in the industry. One of the main advantages is that the control algorithms can be easily implemented in the development phase. This means that PLC does not require any change in wiring compared to previous relay logic, and only the programs need to be modified. This saves time and allows a

late change of programs. In addition, the implementation and evaluation of a counter or timer, as well as the analysis of various states and connections, are carried out internally in the PLC. Furthermore, PLC requires little assembly work, materials, and wiring. For this reason, some projects can be realized in a much shorter time frame. The components can be also expanded later. Various programming languages are available for programming the control logic, such as ANSI C/C++, structured text (ST), continuous function chart (CFC), ladder diagram (LD), sequential function chart (SFC), see details in Bolton [2].

In our application, the PLC is used to implement the model-based feedforward and feedback control algorithms, which calculate the torque as the setpoint of the torque loop control for executing a given movement, and to include external sensors such as the IMU.

The following sections introduce two essential elements – ACOPOS parameter ID and function blocks.

2.3.1 ACOPOS Parameter ID

So-called parameter IDs (ParID) are available for communication with ACOPOS, which can be read and/or written by the user. The readable ParIDs represent different states of ACOPOS, such as the current motor position and velocity from the encoder, as well as the motor current, while the other ParIDs to be written are used for the adjustment of e.g. motor parameters or keep the user-defined parameters. With the function blocks *MC_BR_CyclicRead* and *MC_BR_CyclicWrite*, the parameters can be read or written cyclically via ParIDs. The two function blocks are presented in the next section.

ParID	Name	Access
91	Encoder1: Actual position	RD
92	CTRL Position Controller: Actual speed	RD
213	Set stator current quadrature component	RD
214	CTRL Current Controller: Actual stator current quadrature component	RD
277	Motor: Torque	RD
423	Encoder2: Actual position	RD
586	Function block: User R4 variable1	RD, WR

Table 2.2: Relevant ACOPOS parameter IDs from the *Automation Studio* 4.2 – Help (RD: read, WR: write, R4: data type float32)

Table 2.2 lists the main ACOPOS ParIDs used in our application. Since the rotary encoder is connected to ACOPOS via an RS232 connection, the data can be read directly via ParID 91, while the motor speed is accessible via ParID 92. The stator current and motor torque can be read by ParID 214 and ParID 277, respectively. The angle on

the load side can be monitored by an additional encoder that is also plugged into the ACOPOS. Similar to the motor encoder, the measurement is read by ParID 423. All the above-mentioned ParIDs are read-only. Some function blocks require freely available parameters whose values can be set by the user, e.g., ParID 586 is used for the function block *MC_BR_TorqueControl*, and the reference torque is written cyclically into this ParID.

Furthermore, there are ParIDs for motor configuration, see e.g. Table G.2. The datasheet is provided by the motor manufacturer, and the required values can be assigned to the corresponding ParIDs.

2.3.2 B&R Function Block

The motor control logic can be activated using available function blocks in Automation Studio, the software interface provided by B&R for drive control. It contains not only the PLCopen function blocks but also the self-developed function blocks for extended functionality, which in turn are based on the PLCopen blocks. The following sections introduce the major function blocks that are inserted into our application.

Cyclic Read Function Block

The cyclic read function block is called *MC_BR_CyclicRead*, see an example in Figure 2.8. It is able to enter the selected ParID into the "Cyclic data of the drive" and copy the read value into the user variable whose address is specified via "DataAddress". The data is copied cyclically as soon as "Enable" is set to 1. Otherwise, only the copy process to a user variable is stopped when "Enable" is changed to 0. However, it is not possible to disable the reading of ParID data from the drive. Instead of inserting the name of a variable in "Axis", the address of the variable must be determined here using the function block "ADR". The data type of the user variable is defined by "DataType". In addition, "Mode" is set to 1 to update the value at a constant update rate.

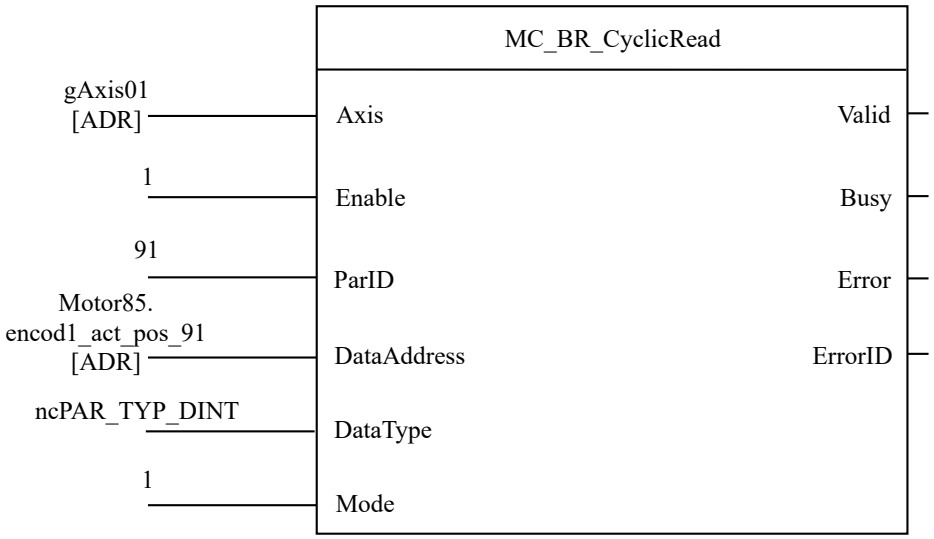


Figure 2.8: Diagram of the function block *MC_BR_CyclicRead* for an example of reading the motor position via ParID 91

Cyclic Write Function Block

In contrast, the function block *MC_BR_CyclicWrite* enters a ParID into the "Cyclic data to drive" and cyclically transfers the data of a user variable to the drive. The transmission process is the same as the one for *MC_BR_CyclicRead*. Figure 2.9 is an example of a setting to write the user-defined torque to ParID 586.

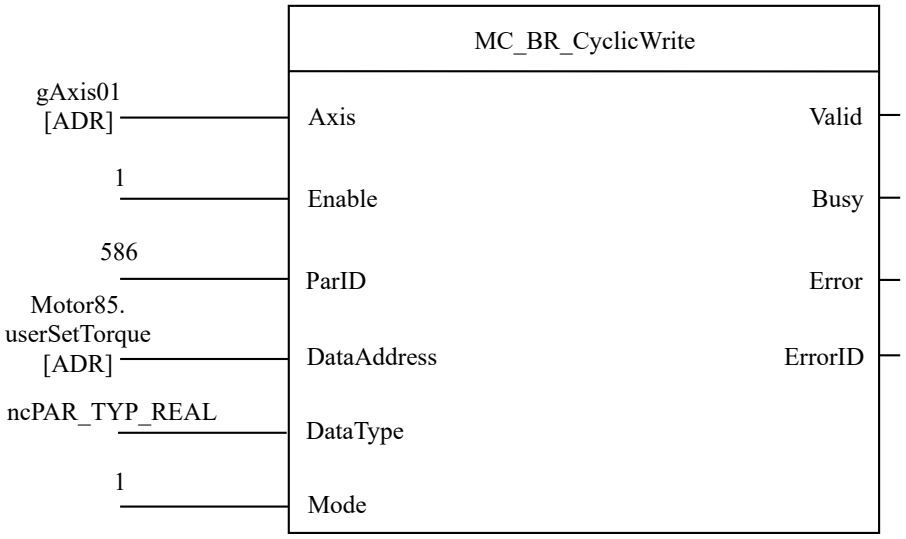


Figure 2.9: Diagram of the function block *MC_BR_CyclicWrite* for an example of writing the user-defined torque into ParID 586

Torque Control Function Block

Figure 2.10 shows an example of the function block *MC_BR_TorqueControl*. This block allows the user to specify the torque setpoint. As shown in Figure 2.6, the position and the speed control loop are automatically deactivated in this mode. Analogous to *MC_BR_CyclicRead*, the axis is defined by "ADR". To track the desired torque, both the "Enable" and "StartSignal" must be 1. The input "Torque" defines a constant value, and the torque can be ramped up to a configured setpoint with "TorqueRamp". The "InTorque" output indicates when the "Torque" is reached. Alternatively, the torque setpoint can be specified cyclically via "TorqueParID". In this case, the value of "Torque" and "TorqueRamp" must be set to 0. For "TorqueParID", a ParID is needed instead of a given value. There are several free ParIDs like ParID 586, which can be used for this purpose. The desired torque is cyclically written into the selected free ParID with *MC_BR_CyclicWrite*.

It is worth noting that the velocity and the acceleration must be limited. The velocity limitation is carried out both in the positive and negative direction by "PosMaxVelocity" and "NegMaxVelocity", respectively. When the maximum velocity is reached, this function block tries to reduce the torque until the velocity is below the maximum. This also applies to the acceleration limitation.

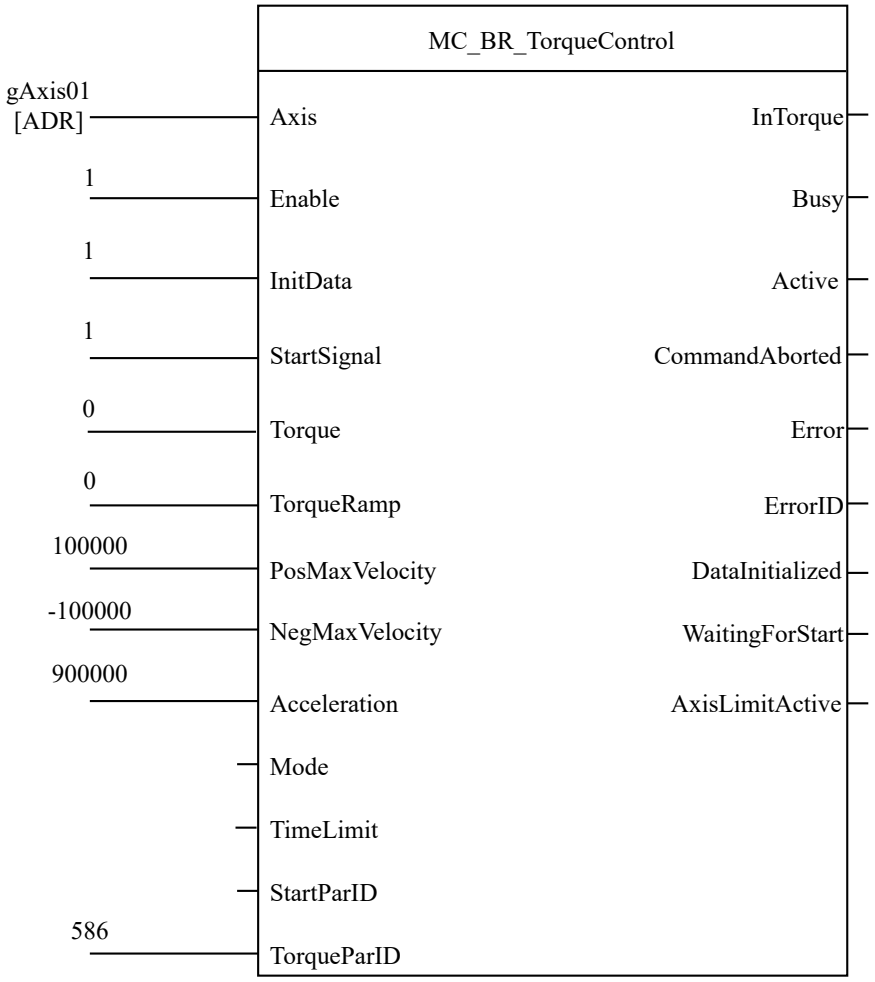


Figure 2.10: Diagram of the function block *MC_BR_TorqueControl* for an example of executing the user-defined torque trajectory

Figure 2.11 shows a schematic diagram for execution of a centralized control loop[4], in which the desired torque is calculated by the model-based feedforward and feedback control algorithms in the experiment. The desired torque trajectory τ_M^* is calculated in the PLC, and the sampled reference values are assigned to the global user-defined ParID 586 via *MC_BR_CyclicWrite*. This ParID is considered as "TorqueParID" for the function block *MC_BR_TorqueControl*. The PLC continues to transmit the desired torque $\tau_{M,PLC}^*$ to the servo amplifier and sets the torque setpoint $\tau_{M,ACOPOS}^*$ for the torque control loop. At the same time, the actual torque $\tau_{M,ACOPOS}$ is fed back to the torque controller in the servo amplifier. The measurement of the motor angle θ_M and angular velocity $\dot{\theta}_M$ can be monitored in the PLC with *MC_BR_CyclicRead*. Due to the possible time delays in the hardware system, the same variable with different notations *PLC* and *ACOPOS* displays the motor states in different components of the transmission process.

[4]The sensor signals from the IMU, which are transmitted by CAN bus system, are not covered in this thesis, see details in [58].

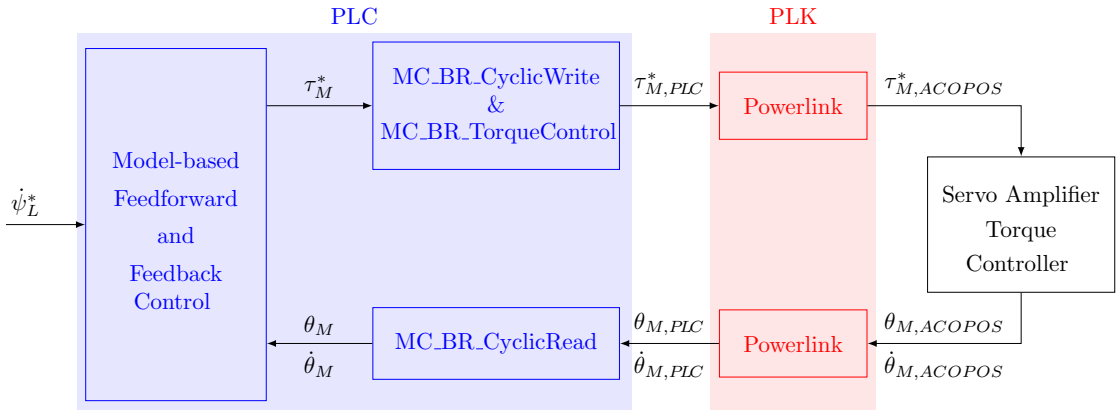


Figure 2.11: Diagram of the experimental torque control. The blue block represents the processing of the control system in PLC ($\dot{\psi}_L$: angular velocity of the beam tip, θ_M: motor angle, $\dot{\theta}_M$: motor angular velocity, τ_M: motor torque). The red block shows the signal transmission over the real-time Ethernet.

2.3.3 Configuration

All function blocks can be assigned run-time properties that are designated by the "task", whether they read or write parameters or main programs, e.g., for the execution of a control logic. A typical PLC is capable of executing several programs simultaneously (*multitasking*) and deliberately prioritizing them, see [83]. The most important programs must be run as quickly as possible and should be placed in a higher priority task. However, it is not allowed to assign too many programs to the same task because the data volume and the computation time can lead to an overload of the CPU. The programs should therefore be distributed among different tasks according to their priorities.

X20CP1586 Configuration Parameter			Value
Timing	Cycle time of interface in μs		400
	Multiply cycle time by		0.25
	System tick in μs		0.25
	Idle time	Idle task class	Cyclic ♯2
		Task class idle time in μs	600
Resource	Number of cyclic resources		8
	Cyclic task classes	Cyclic ♯1: Duration in μs	1200
		Cyclic ♯2: Duration in μs	600
		...	...

Table 2.3: CPU configuration

How the task class is set for each program depends mainly on the cycle time of the POWERLINK interface and a parameter "Multiply cycle time by", which is multiplied by the cycle time to determine the system tick. The cycle time of the interface must be a multiple of 400μs, while the value of parameter "Multiply cycle time by" must be a positive integer or a positive number whose reciprocal is a positive integer. Depending on the "Number of cyclic resources", the cyclic resources can be divided into different

"Cyclic task classes" whose "Duration" must be a multiple of the system tick. A smaller number of cyclic task classes, e.g. Cyclic ♯1, indicates a higher priority. One of the task classes is selected as the "Idle task class", and the "Task class idle time" corresponds to the duration of this cyclic task class. Table 2.3 shows the essential parameters of the CPU configuration with a given example for the experiments.

The project source code such as function blocks, programs or libraries is managed in the Automation Studio *logical view*. To establish a relationship between a logical program and the hardware, the logical program should be assigned in the so-called *software configuration*. After the CPU configuration, many different PLC cyclic task classes are available. Depending on the cycle time requirements and the priority requirements of programs, these are assigned to different task classes.

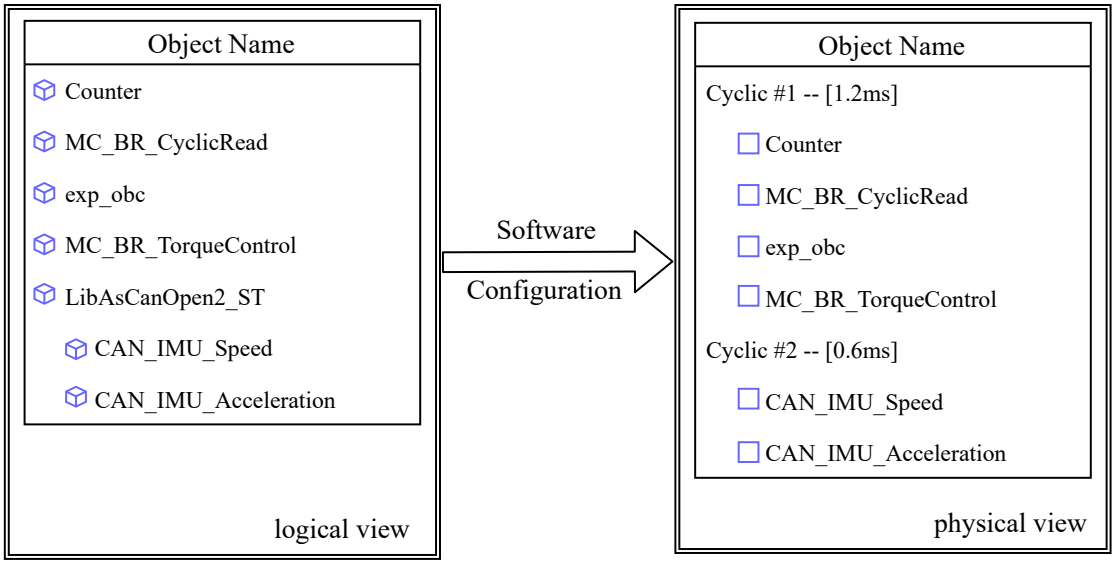


Figure 2.12: Software configuration: Assignment of programs and function blocks to cyclic tasks

Figure 2.12 shows how logical programs become PLC tasks. We take several source objects in *logic view* as an example. *MC_BR_CyclicRead*, *MC_BR_TorqueControl*, and *Counter* are based on the function blocks, while *exp_obc* is the generated C-code of a Simulink model. *LibAsCanOpen2_ST* is one of the available Automation Studio libraries that is used to read data from devices with a CAN interface. Since the IMU is connected to the PLC via the CAN interface, the measured angular velocity and translational acceleration of the beam tip can be recorded using this library. Based on the CPU configuration in Table 2.3, the main program *exp_obc* runs in a cycle time of 1.2ms, which is exactly a multiple of both cycle times of interface (400μs) and system tick (100μs). Note that the sampling time of the Simulink model must match the "Duration" of the selected cyclic task class. In Cyclic ♯1, the source objects are executed from top to bottom. The counter counts up with the running time as soon as the desired movement is activated. The required measurement is recorded every 1.2ms by *MC_BR_CyclicRead* and then returned to the model-based controller. *exp_obc* computes the command motor torque to achieve a specified movement and updates the setpoint of *MC_BR_TorqueControl*. At the same time, the measurement of the IMU is acquired in a cycle time of 0.6ms, since this library needs about four times 0.6ms to

read, write, and save the data. Depending on the applications, different combinations of the interface cycle time and the duration of cyclic task classes are possible.

Chapter 3

Port-Hamiltonian Modeling and Discretization

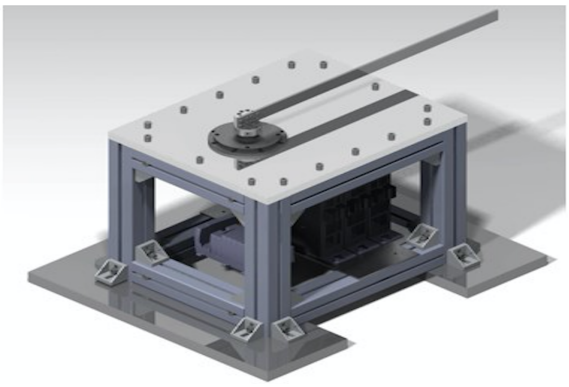


Figure 3.1: CAD sketch of the single-link flexible manipulator

Figure 3.1 shows the CAD sketch of the single-link flexible robot arm. The beam, which is connected to the gear motor via a hub, exerts a planar motion. The core task for modeling and control of the lightweight robot is to deal with the vibration of the flexible structure, which is described by PDEs, while the drive system can be easily described by ODEs. This chapter is, after a short introduction of PH systems, devoted to the finite- and infinite-dimensional PH modeling of the single-link flexible robot arm, and the structure-preserving spatial discretization and model order reduction to obtain a relatively low-dimensional, high-quality control model. The contents of this chapter are mainly excerpted from our published contributions [93], [96], [97].

3.1 Port-Hamiltonian Systems

PH systems, and their underlying geometric structures – Dirac or Stokes-Dirac structure – admit several kinds of formulations, see [14], [87]. In this section, we recall two of them. The explicit state representations[1] on a finite-dimensional and an infinite-dimensional state space will be used throughout the dissertation. We particularly stress the differ-

[1] We skip the time dependence of the variables in this chapter.

ential energy balance, which is to be preserved by appropriate numerical methods for discretization and model order reduction.

3.1.1 Dirac Structure

Dirac structures, originally introduced in [7], are characterized as the geometric structures formalizing general power-conserving interconnections, which permit the interconnected and constrained physical systems being represented in the Hamiltonian formulation, see [87].

Definition 3.1 (*Dirac structure*)**.** Given a linear space of flows $\boldsymbol{f} \in \mathcal{F}$ and its dual space $\mathcal{E} := \mathcal{F}^*$ defined for efforts $\boldsymbol{e} \in \mathcal{E}$, equipped with the duality pairing $\langle \cdot | \cdot \rangle : \mathcal{F} \times \mathcal{E} \to \mathbb{R}$. Define a symmetric bilinear form $\langle\langle \cdot | \cdot \rangle\rangle$ on $\mathcal{F} \times \mathcal{E}$ as

$$\langle\langle (\boldsymbol{f}_1, \boldsymbol{e}_1), (\boldsymbol{f}_2, \boldsymbol{e}_2) \rangle\rangle := \langle \boldsymbol{e}_1 | \boldsymbol{f}_2 \rangle + \langle \boldsymbol{e}_2 | \boldsymbol{f}_1 \rangle, \qquad (\boldsymbol{f}_i, \boldsymbol{e}_i) \in \mathcal{F} \times \mathcal{E}, \quad i = 1, 2. \tag{3.1}$$

A Dirac structure is a linear subspace $\mathcal{D} \subset \mathcal{F} \times \mathcal{E}$ with the property that $\mathcal{D} = \mathcal{D}^\perp$, where $\mathcal{D}^\perp$ denotes the orthogonal complement with respect to the bilinear form $\langle\langle \cdot | \cdot \rangle\rangle$.

According to the property $\mathcal{D} = \mathcal{D}^\perp$ of the Dirac structure, the bilinear form holds for every $(\boldsymbol{f}, \boldsymbol{e}) \in \mathcal{D}$ that

$$0 = \langle\langle (\boldsymbol{f}, \boldsymbol{e}), (\boldsymbol{f}, \boldsymbol{e}) \rangle\rangle = 2 \langle \boldsymbol{e} | \boldsymbol{f} \rangle. \tag{3.2}$$

Related to a pair of power variables $(\boldsymbol{f}, \boldsymbol{e})$, this condition corresponds to power conservation and implies that the total power exchange of a Dirac structure is zero.

Dirac structures admit different representations for finite-dimensional systems, see [9]. For the space of conjugated power variables $\mathcal{F} \times \mathcal{E} = \mathbb{R}^n \times \mathbb{R}^n$, three matrix representations will be presented here.

Proposition 3.1 (*Kernel and image representation*)**.** A Dirac structure $\mathcal{D} \subset \mathcal{F} \times \mathcal{E} = \mathbb{R}^n \times \mathbb{R}^n$ can be rewritten as

1) the kernel representation

$$\mathcal{D} = \{ (\boldsymbol{f}, \boldsymbol{e}) \in \mathcal{F} \times \mathcal{E} \mid \boldsymbol{F}\boldsymbol{f} + \boldsymbol{E}\boldsymbol{e} = \boldsymbol{0} \}; \tag{3.3}$$

2) the image representation

$$\mathcal{D} = \left\{ (\boldsymbol{f}, \boldsymbol{e}) \in \mathcal{F} \times \mathcal{E} \mid \boldsymbol{f} = \boldsymbol{E}^T \boldsymbol{\lambda},\, \boldsymbol{e} = \boldsymbol{F}^T \boldsymbol{\lambda},\, \boldsymbol{\lambda} \in \mathbb{R}^n \right\}, \tag{3.4}$$

if and only if there exist two matrices $\boldsymbol{F}, \boldsymbol{E} \in \mathbb{R}^{n \times n}$ satisfying the following conditions:

(i) skew-symmetry: $\boldsymbol{E}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{E}^T = \boldsymbol{0}$,

(ii) $\operatorname{rank} [\boldsymbol{F} : \boldsymbol{E}] = n$.

Proposition 3.2 (*Input-output representation*)**.** A Dirac structure $\mathcal{D} \subset \mathcal{F} \times \mathcal{E} = \mathbb{R}^n \times \mathbb{R}^n$ admits the input-output representation

$$\mathcal{D} = \left\{ (\boldsymbol{f}, \boldsymbol{e}) \in \mathcal{F} \times \mathcal{E} \;\middle|\; \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{f}_2 \end{bmatrix} \right\}, \tag{3.5}$$

if there exists a decomposition of the space of flows $(\boldsymbol{f}_1, \boldsymbol{f}_2) = \boldsymbol{f} \in \mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$ and space of efforts $(\boldsymbol{e}_1, \boldsymbol{e}_2) = \boldsymbol{e} \in \mathcal{E} = \mathcal{E}_1 \times \mathcal{E}_2$ as well as a skew-symmetric matrix $\boldsymbol{J} \in \mathbb{R}^{n \times n}$.

3.1.2 Finite-Dimensional PH Systems

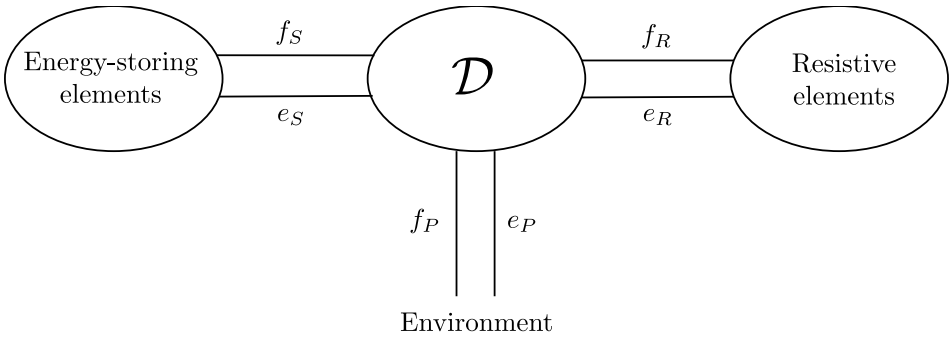


Figure 3.2: Port-Hamiltonian system [14]

The geometric definition of a PH system consists of three main ingredients as depicted in Figure 3.2. First, there are the energy-storing elements $(\boldsymbol{f}_S, \boldsymbol{e}_S)$ corresponding to the space of flows and efforts $\mathcal{F}_S \times \mathcal{E}_S$. Secondly, it contains the resistive elements $(\boldsymbol{f}_R, \boldsymbol{e}_R) \in \mathcal{F}_R \times \mathcal{E}_R$, Thirdly, there is a Dirac structure $\mathcal{D}$, which connects all the power variables with respect to the energy storage and dissipation as well as the external ports $(\boldsymbol{f}_P, \boldsymbol{e}_P)$ defined on the space $\mathcal{F}_P \times \mathcal{E}_P$ in a power-preserving manner, that is, the total power associated with the port variables imposed by the Dirac structure is zero. Thus, on $\mathcal{D}$ the structural power balance

$$\boldsymbol{e}_S^T \boldsymbol{f}_S + \boldsymbol{e}_R^T \boldsymbol{f}_R + \boldsymbol{e}_P^T \boldsymbol{f}_P = 0 \tag{3.6}$$

holds, where $\boldsymbol{f}_S$, $\boldsymbol{e}_S \in \mathbb{R}^n$, $\boldsymbol{f}_R$, $\boldsymbol{e}_R \in \mathbb{R}^q$, $\boldsymbol{f}_P$, $\boldsymbol{e}_P \in \mathbb{R}^m$.

Energy storage. The internal storage ports $(\boldsymbol{f}_S, \boldsymbol{e}_S)$ are interconnected to the energy storage of the system represented by a finite-dimensional state space for the *energy variables* $\boldsymbol{x} \in \mathbb{R}^n$, together with an *energy* or *Hamiltonian* function $H(\boldsymbol{x})$. The vectors $\boldsymbol{f}_S$ and $\boldsymbol{e}_S$ are called *flows* and *efforts* of the PH system, respectively. The energy storage can be represented as follows:

$$\begin{aligned} \dot{\boldsymbol{x}} &= -\boldsymbol{f}_S && \text{Dynamics} \\ \boldsymbol{e}_S &= \nabla H(\boldsymbol{x}) && \text{Constitutive equation} \end{aligned} \tag{3.7}$$

With the defined flows and efforts, the power balance equation can be written as

$$\dot{H} = \langle \nabla H(\boldsymbol{x}) | \dot{\boldsymbol{x}} \rangle = -\boldsymbol{e}_S^T \boldsymbol{f}_S. \tag{3.8}$$

Resistive elements. The resistive elements correspond to the internal energy dissipation, e.g. caused by the friction, resistance, etc., and are described by the port variables $(\boldsymbol{f}_R, \boldsymbol{e}_R)$. The input-output mapping of the resistive relation can be represented by

$$\boldsymbol{f}_R = -\boldsymbol{F}_R(\boldsymbol{e}_R), \tag{3.9}$$

with the resistive characteristic $\boldsymbol{F}_R : \mathbb{R}^q \to \mathbb{R}^q$ satisfying[2]

$$\boldsymbol{e}_R^T \boldsymbol{F}_R(\boldsymbol{e}_R) \geq 0, \tag{3.10}$$

for all $(\boldsymbol{f}_R, \boldsymbol{e}_R)$. In many cases, it restricts the linear resistive relations

$$\boldsymbol{f}_R = -\tilde{\boldsymbol{R}} \boldsymbol{e}_R, \qquad \tilde{\boldsymbol{R}} = \tilde{\boldsymbol{R}}^T \geq 0. \tag{3.11}$$

Regarding the external port variables $(\boldsymbol{f}_P, \boldsymbol{e}_P)$ as the inputs and the outputs $\boldsymbol{u}, \boldsymbol{y} \in \mathbb{R}^m$, $\boldsymbol{f}_P = \boldsymbol{u}$, $\boldsymbol{e}_P = \boldsymbol{y}$, the input-state-output representation of the Dirac structure $\mathcal{D}$ is written as

$$\begin{aligned} -\boldsymbol{f}_S &= (\boldsymbol{J} - \boldsymbol{R})\boldsymbol{e}_S + \boldsymbol{G}\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{G}^T \boldsymbol{e}_S + \boldsymbol{D}\boldsymbol{u}, \end{aligned} \tag{3.12}$$

where the internal interconnection structure is given by the skew-symmetric matrix $\boldsymbol{J} = -\boldsymbol{J}^T$, the resistive structure[3] $\boldsymbol{R} = \boldsymbol{R}^T \geq \boldsymbol{0}$, and the feedthrough matrix $\boldsymbol{D} = -\boldsymbol{D}^T$. According to (3.6), the structural power balance can be represented as

$$\boldsymbol{e}_S^T \boldsymbol{f}_S + \boldsymbol{e}_R^T \boldsymbol{f}_R + \boldsymbol{u}^T \boldsymbol{y} = 0. \tag{3.13}$$

With the dynamics and constitutive equation (3.7), the PH system (3.12) can be rewritten in the state space representation.

Definition 3.2 (*Finite-Dimensional port-Hamiltonian system*)**.** A dynamical system

$$\begin{aligned} \dot{\boldsymbol{x}} &= (\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{x}))\nabla H(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{G}^T(\boldsymbol{x})\nabla H(\boldsymbol{x}) + \boldsymbol{D}\boldsymbol{u}, \end{aligned} \tag{3.14}$$

where $H(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ is an energy (Hamiltonian) function, which is bounded from below, with interconnection matrix $\boldsymbol{J}(\boldsymbol{x}) = -\boldsymbol{J}^T(\boldsymbol{x})$, dissipation matrix $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{R}^T(\boldsymbol{x}) \geq \boldsymbol{0}$ and feedthrough matrix $\boldsymbol{D} = -\boldsymbol{D}$, is called a finite-dimensional PH system (with dissipation).

[2]Here, $\boldsymbol{e}_i^T \boldsymbol{f}_i$ describes respectively the power taken from the element (the energy storage, the resistive element, and the environment) – "generator sign convention".

[3]Notice that the flows and efforts of the resistive elements do not occur explicitly in the equations anymore. From a resistive port perspective, it can be specified by $\boldsymbol{R} = \boldsymbol{G}_R^T \tilde{\boldsymbol{R}}_R \boldsymbol{G}_R$ relating the linear resistive relation (3.11), where $\boldsymbol{G}_R$ corresponds to the input matrix of the resistive port, see details in [14, p. 70]

The structure of the PH state representation allows, by the skew-symmetry of $\boldsymbol{J}(\boldsymbol{x})$ and $\boldsymbol{D}(\boldsymbol{x})$ as well as the definition of the output (which is *collocated* and *power-conjugated*) to directly express the energy balance equation

$$\dot{H}(\boldsymbol{x}) = \boldsymbol{y}^T\boldsymbol{u} - \underbrace{(\nabla H(\boldsymbol{x}))^T\boldsymbol{R}(\boldsymbol{x})\nabla H(\boldsymbol{x})}_{\geq 0} \leq \boldsymbol{y}^T\boldsymbol{u}, \tag{3.15}$$

which shows passivity of (3.14) and *Lyapunov* stability for $\boldsymbol{u} = \boldsymbol{0}$.

3.1.3 Stokes-Dirac Structure

The distributed parameter systems can be formulated in terms of boundary PH systems based on the infinite-dimensional Dirac structure, that is, the so-called Stokes-Dirac structure. In this section, the generalized definition of Stokes-Dirac structures is recalled for systems of two conservation laws, with non-zero energy flow through the boundary. Instead of the vector calculus formulation, the differential forms on the spatial domain of the system and its boundary, permit giving a dimension-independent definition of Stokes-Dirac structures and boundary PH systems, see [14], [87].

Define n-dimensional smooth manifold $\mathcal{Z}$ with smooth $(n-1)$-dimensional boundary $\partial\mathcal{Z}$ to represent the space of spatial variables. The space of differential k-forms [17] on $\mathcal{Z}$ is indicated by $\Omega^k(\mathcal{Z})$, $k = 0, 1, \ldots, n$, whereas $\Omega^k(\partial\mathcal{Z})$, $k = 0, 1, \ldots, n-1$ for the space of k-forms on $\partial\mathcal{Z}$. Thus, the natural duality pairing between two differential forms $\alpha \in \Omega^k(\mathcal{Z})$ and $\beta \in \Omega^{n-k}(\mathcal{Z})$ is given by

$$\langle\beta|\alpha\rangle := \int_{\mathcal{Z}} \beta \wedge \alpha, \tag{3.16}$$

where $\wedge$ denotes the wedge product of differential forms to yield the n-form $\beta \wedge \alpha$. This pairing is non-degenerate if $\langle\beta|\alpha\rangle = 0$ for $\alpha \neq 0$ (for $\beta \neq 0$) only if $\beta = 0$ ($\alpha = 0$).

Similarly, the pairing relates differential forms on the boundary $\alpha \in \Omega^k(\partial\mathcal{Z})$, $\beta \in \Omega^{n-1-k}(\partial\mathcal{Z})$ as follows:

$$\langle\beta|\alpha\rangle := \int_{\partial\mathcal{Z}} \beta \wedge \alpha. \tag{3.17}$$

Consider now the linear space of distributed flows and efforts:

$$\begin{aligned} \mathcal{F} &:= \Omega^p(\mathcal{Z}) \times \Omega^q(\mathcal{Z}) \times \Omega^{n-p}(\partial\mathcal{Z}) \\ \mathcal{E} &:= \Omega^{n-p}(\mathcal{Z}) \times \Omega^{n-q}(\mathcal{Z}) \times \Omega^{n-q}(\partial\mathcal{Z}) \end{aligned} \tag{3.18}$$

for any pair p, q satisfying

$$p + q = n + 1. \tag{3.19}$$

The pairings (3.16) and (3.17) yield the bilinear form on $\mathcal{F} \times \mathcal{E}$

$$\begin{aligned} &\langle\langle (f_1^p, f_1^q, f_1^\partial, e_1^p, e_1^q, e_1^\partial), (f_2^p, f_2^q, f_2^\partial, e_2^p, e_2^q, e_2^\partial)\rangle\rangle := \\ &\langle e_1^p|f_2^p\rangle_{\mathcal{Z}} + \langle e_1^q|f_2^q\rangle_{\mathcal{Z}} + \langle e_1^\partial|f_2^\partial\rangle_{\partial\mathcal{Z}} + \langle e_2^p|f_1^p\rangle_{\mathcal{Z}} + \langle e_2^q|f_1^q\rangle_{\mathcal{Z}} + \langle e_2^\partial|f_1^\partial\rangle_{\partial\mathcal{Z}}, \end{aligned} \tag{3.20}$$

where for $i = 1, 2$

$$\begin{aligned} &f_i^p \in \Omega^p(\mathcal{Z}), \qquad f_i^q \in \Omega^q(\mathcal{Z}), \qquad f_i^\partial \in \Omega^{n-p}(\partial\mathcal{Z}), \\ &e_i^p \in \Omega^{n-p}(\mathcal{Z}), \qquad e_i^q \in \Omega^{n-q}(\mathcal{Z}), \qquad e_i^\partial \in \Omega^{n-q}(\partial\mathcal{Z}). \end{aligned} \tag{3.21}$$

It is worth noticing that the spaces of differential forms $\Omega^p(\mathcal{Z})$ and $\Omega^q(\mathcal{Z})$ denote the energy variables of two interacting physical energy domains, whereas $\Omega^p(\partial\mathcal{Z})$ and $\Omega^q(\partial\mathcal{Z})$ represent the boundary variables, whose product indicates the boundary energy flow, see [87].

Definition 3.3 (*Stokes-Dirac structure*)**.** Consider the linear space $\mathcal{F}$ and $\mathcal{E}$ defined in (3.18) respectively for flows and efforts with p and q satisfying Equation (3.19) and the duality pairing (3.20). The relation[4] of flows and efforts is given by

$$\begin{bmatrix} f^p \\ f^q \end{bmatrix} = \begin{bmatrix} 0 & (-1)^r d \\ d & 0 \end{bmatrix} \begin{bmatrix} e^p \\ e^q \end{bmatrix} \tag{3.22a}$$

$$\begin{bmatrix} f^\partial \\ e^\partial \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & (-1)^p \end{bmatrix} \begin{bmatrix} e^p|_{\partial\mathcal{Z}} \\ e^q|_{\partial\mathcal{Z}} \end{bmatrix}, \tag{3.22b}$$

where $r = pq+1$ and $|_{\partial\mathcal{Z}}$ denotes the restriction of the differential form to the boundary. Then, the linear subspace $\mathcal{D}$ of $\mathcal{F} \times \mathcal{E}$

$$\mathcal{D} = \left\{ \left(f^p, f^q f^\partial, e^p, e^q, e^\partial \right) \in \mathcal{F} \times \mathcal{E} | \, (3.22) \text{ holds} \right\} \tag{3.23}$$

is a Dirac structure, where $\mathcal{D} = \mathcal{D}^\perp$ holds.

By using the generalized Stokes' theorem, the properties, $\mathcal{D} \subset \mathcal{D}^\perp$ and $\mathcal{D}^\perp \subset \mathcal{D}$ can be proved [87]. Then, the corresponding power continuity relation is expressed by

$$\langle e^p | f^p \rangle_{\mathcal{Z}} + \langle e^q | f^q \rangle_{\mathcal{Z}} + \langle e^\partial | f^\partial \rangle_{\partial\mathcal{Z}} = 0. \tag{3.24}$$

3.1.4 Infinite-Dimensional PH Systems

For the purpose of beam modeling, we consider an infinite-dimensional extension of (3.14) on the space of smooth vector-valued functions over 1D domain $\mathcal{Z} = [0, L]$. Define the vector of state functions $\boldsymbol{x}(z) \in C^\infty(\mathcal{Z}; \mathbb{R}^n)$, which appears in the *energy* or *Hamiltonian functional* $H(\boldsymbol{x}(z)) = \int_{\mathcal{Z}} \mathcal{H}(\boldsymbol{x}(z))\, dz$ without spatial derivatives (energy variables), and the boundary in- and outputs $\boldsymbol{u}, \boldsymbol{y} \in \mathbb{R}^m$.

Definition 3.4 (*Boundary-controlled PH systems*)**.** The states of a boundary-controlled PH system (with dissipation) on a 1D domain $\mathcal{Z} \in [0, L]$ obey the abstract differential equation

$$\dot{\boldsymbol{x}}(z) = (\mathcal{J} - \mathcal{R})(\delta_{\boldsymbol{x}} H(\boldsymbol{x}(z)))^T \tag{3.25}$$

[4] Compared to the definition in [87], $-(-1)^{n-q}$ in [87] is replaced with $(-1)^p$ in Equation (3.22b) according to Equation (3.19), see [39].

with a formally[5] skew-adjoint matrix differential operator $\mathcal{J}$ and a self-adjoint, positive semi-definite dissipation operator $\mathcal{R}$. With the choice of boundary port variables[6], i.e., variational derivatives[7] of H restricted to the boundary of $\mathcal{Z}$,

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{W}_B \, (\delta_{\boldsymbol{x}} H)^T \big|_{\partial \mathcal{Z}} \\ \boldsymbol{y} &= \boldsymbol{W}_C \, (\delta_{\boldsymbol{x}} H)^T \big|_{\partial \mathcal{Z}} , \end{aligned} \tag{3.26}$$

where $\boldsymbol{W}_B, \boldsymbol{W}_C \in \mathbb{R}^{n \times 2n}$ are appropriate matrices, the energy balance reads

$$\dot{H}(\boldsymbol{x}(z)) = \boldsymbol{y}^T \boldsymbol{u} - \underbrace{\int_{\mathcal{Z}} \partial_{\boldsymbol{x}} \mathcal{H}(\boldsymbol{x}(z)) \mathcal{R} (\partial_{\boldsymbol{x}} \mathcal{H}(\boldsymbol{x}(z)))^T \, dz}_{\geq 0} . \tag{3.27}$$

Note that by the choice of energy variables, the row vector of variational derivatives coincides with the (transposed) gradient of the Hamiltonian density, $\delta_{\boldsymbol{x}} H = \partial_{\boldsymbol{x}} \mathcal{H}$.

Defining the efforts and flows for distributed energy storage as $\boldsymbol{f}(z) = -\dot{\boldsymbol{x}}(z)$ and $\boldsymbol{e}(z) = (\delta_{\boldsymbol{x}} H(\boldsymbol{x}(z)))^T$, together with appropriate distributed dissipation port variables $\boldsymbol{e}_R(z)$, $\boldsymbol{f}_R(z)$, these can be related, as in the finite-dimensional case, via

$$\int_{\mathcal{Z}} \boldsymbol{e}^T(z) \boldsymbol{f}(z) \, dz + \int_{\mathcal{Z}} \boldsymbol{e}_R^T(z) \boldsymbol{f}_R(z) \, dz + \boldsymbol{u}^T \boldsymbol{y} = 0. \tag{3.28}$$

The underlying linear infinite-dimensional space, which implies the structural balance equation (3.28), is a Stokes-Dirac structure [87], as defined above.

3.2 Drive System

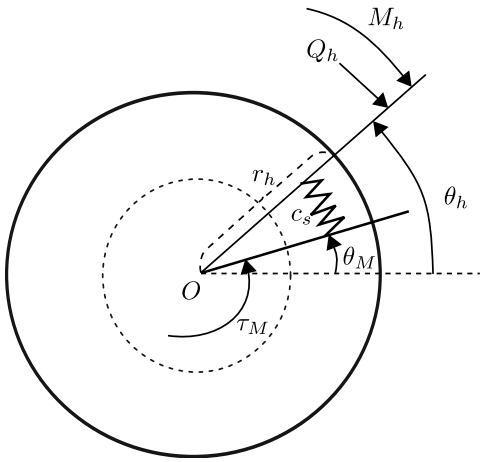


Figure 3.3: Schematic diagram of parameters and variables in the gear motor according to [24]

[5] I.e., under boundary conditions which make the boundary terms vanish after integration by parts.
[6] For a general discussion on this issue, see [46].
[7] Defined by $\delta H = \int_{\mathcal{Z}} \delta_{\boldsymbol{x}} H \cdot \delta \boldsymbol{x} \, dz$, where $\delta \boldsymbol{x}$ contains variations of the state functions.

The drive system of the manipulator consists of a BLDC electric motor with a *Harmonic Drive* gear, the elasticity of which is approximated by a linear torsion spring with stiffness c_s (see e.g. [24]). Figure 3.3 shows the angles at the motor and hub side θ_M, θ_h, the motor torque τ_M, as well as the radius of the rigid hub r_h, to which the flexible beam is attached. Q_h and M_h denote the reaction force and torque exerted to the hub (by the interconnected beam).

Equations of motion. According to Newton's second law, the equations of motion can be described by two degrees of freedom at the driven side[8]:

$$\begin{aligned} I_M\ddot{\theta}_M &= \tau_M - c_s(\theta_M - \theta_h) \\ I_h\ddot{\theta}_h &= c_s(\theta_M - \theta_h) - M_h - r_h Q_h \end{aligned} \tag{3.29}$$

with

$$I_M = n^2 I_m, \qquad \tau_M = n\tau_m, \qquad \theta_M = \frac{\theta_m}{n}. \tag{3.30}$$

I_M denotes the motor moment of inertia[9], while I_h describes the hub moment of inertia, which shouldn't be neglected in the modeling. In our case, the gear motor has a gear ratio $n > 1$.

Port-Hamiltonian representation. We define the state vector[10] $\boldsymbol{x}_a \in \mathbb{R}^4$ as

$$\boldsymbol{x}_a = \begin{bmatrix} \theta_M \\ \theta_h \\ p_M \\ p_h \end{bmatrix} \tag{3.31}$$

with the angular momenta $p_M = I_M\dot{\theta}_M$ and $p_h = I_h\dot{\theta}_h$. The Hamiltonian of the drive system is composed of the kinetic and elastic potential energy

$$H_a = \frac{1}{2}\left(\frac{p_M^2}{I_M} + \frac{p_h^2}{I_h} + c_s(\theta_M - \theta_h)^2\right). \tag{3.32}$$

The gradient of the quadratic Hamiltonian is defined as the co-energy vector $\boldsymbol{e}_a \in \mathbb{R}^4$

$$\boldsymbol{e}_a = \nabla H_a(\boldsymbol{x}_a) = \boldsymbol{Q}_a\boldsymbol{x}_a \tag{3.33}$$

with

$$\boldsymbol{Q}_a = \begin{bmatrix} c_s & -c_s & 0 & 0 \\ -c_s & c_s & 0 & 0 \\ 0 & 0 & \frac{1}{I_M} & 0 \\ 0 & 0 & 0 & \frac{1}{I_h} \end{bmatrix}. \tag{3.34}$$

[8] The indices m and M refer to quantities on the driving and driven side, respectively.

[9] I_m consists of the moment of inertia of the rotor and the fast-rotating gear part. If we consider the motor motion at the driven side, its angular velocity is reduced by the gearbox. Correspondingly, its moment of inertia I_m increases to I_M, so that the power is preserved (assumption without dissipation).

[10] The index "a" refers to the first subsystem.

Its components represent the transmitted torques, the motor and the hub angular velocity, respectively. The equations of motion (3.29) can be rewritten in linear PH form as

$$\begin{aligned} \dot{\boldsymbol{x}}_a &= \boldsymbol{J}_a \boldsymbol{Q}_a \boldsymbol{x}_a + \boldsymbol{B}_a \boldsymbol{u}_a \\ \boldsymbol{y}_a &= \boldsymbol{B}_a^T \boldsymbol{Q}_a \boldsymbol{x}_a \end{aligned} \tag{3.35}$$

with the interconnection and input matrix

$$\boldsymbol{J}_a = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$$

$$\boldsymbol{B}_a = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & r_h & 1 \end{bmatrix}.$$

The vectors $\boldsymbol{u}_a$, $\boldsymbol{y}_a \in \mathbb{R}^3$ contain the inputs and the collocated, power-conjugated outputs

$$\begin{aligned} \boldsymbol{u}_a &= \begin{bmatrix} u_{a_1} \\ \boldsymbol{u}_{a_2} \end{bmatrix} = \left[\begin{array}{c} \tau_M \\ \hdashline -Q_h \\ -M_h \end{array}\right] \\ \boldsymbol{y}_a &= \begin{bmatrix} y_{a_1} \\ \boldsymbol{y}_{a_2} \end{bmatrix} = \left[\begin{array}{c} \dot{\theta}_M \\ \hdashline v_h \\ \dot{\theta}_h \end{array}\right], \end{aligned} \tag{3.36}$$

where $v_h = r_h \dot{\theta}_h$ denotes the translational hub velocity.

3.3 Timoshenko Beam

To describe the flexible beam, we use the Timoshenko beam model[11], which leads to a PH state representation with a first-order differential operator. Different PH formulations are possible for the Timoshenko beam. Instead of the differential forms as presented in [52], we choose a formulation as in [46] with the system variables elements of appropriate Hilbert spaces[12]. Both representations rely on an underlying geometric structure of the PDEs that relates the power-conjugated variables.

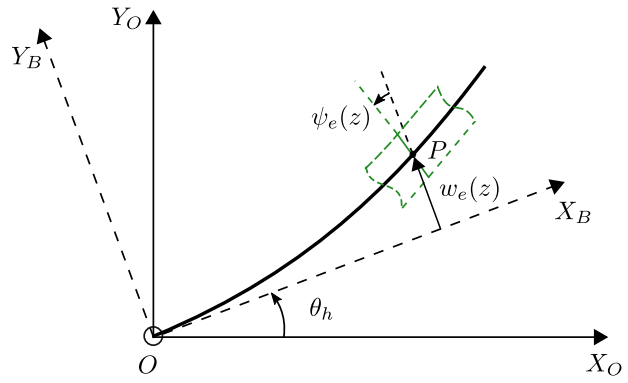


Figure 3.4: Timoshenko beam with coordinate systems and configuration variables based on [22] (X_0Y_0: inertial reference frame, X_BY_B: floating frame of reference)

3.3.1 Distributed Parameter Model

Figure 3.4 shows the coordinate systems and the configuration variables for a rotating Timoshenko beam. θ_h is the rigid body rotation angle, $w_e(z)$ the flexible displacement from the undeformed configuration, and $\psi_e(z)$ the rotation of the beam's cross section due to bending. Here, z denotes the spatial coordinate[13] along the undeformed beam to describe the 1D deformation. In many applications, as in our case, only small deformations occur compared to a reference configuration. Under the assumption of small deformation, the linearized model can be used, see details on the model derivation in Appendix A. In addition, we omit here the motion equation of the hub (A.27c), which is considered in Appendix A, and concentrate on the two linear second-order PDEs on $\mathbb{R}^+ \times [0, L]$

$$\begin{aligned} \rho\, \partial_t^2 w - K_s(\partial_z^2 w - \partial_z \psi) &= 0 \\ I_\rho\, \partial_t^2 \psi - K_b\, \partial_z^2 \psi + K_s(\psi - \partial_z w) &= 0 \end{aligned} \tag{3.37}$$

in terms of the global configuration variables described in the inertial reference frame

$$\begin{aligned} w(z) &= w_e(z) + z\theta_h \\ \psi(z) &= \psi_e(z) + \theta_h. \end{aligned} \tag{3.38}$$

In the small deformation case, $w(z)$ describes physically the arc length, which contains the dynamics of a rigid body rotation of the undeformed beam with a superposed flexible motion. Here, $\partial_t(\cdot)$ and $\partial_z(\cdot)$ denote the time and spatial derivatives, respectively.

The shear angle $\gamma(z) = \partial_z w_e(z) - \psi_e(z)$ is non-zero according to the Timoshenko theory. Additionally, it is assumed that the beam length doesn't change during the movement.

[11] For this work, we decided to stick to the Timoshenko beam model due to the straightforward adaptation of the pseudo-spectral method in [60] to the Timoshenko beam. While it is accepted that the Timoshenko theory is superior to the Euler-Bernoulli theory for "thick" beams, the Timoshenko model still does the job, just as the simpler Euler-Bernoulli model, for "slender" beams [1] as considered in this work.

[12] We assume C^∞, i.e., smooth functions in the spatial variable.

[13] Note that in Appendix A we use "x" as the spatial coordinate, with reference to the chosen coordinate systems.

The physical parameters ρ, I_ρ, K_s, and K_b are respectively the line density, the mass moment of inertia of the cross section, the shear and flexural stiffness. In most references, the latter are defined by $K_s = \kappa GA$ and $K_b = EI$, where the shear correction factor κ depends on the shape of the cross section, A is the cross-sectional area, G is the shear modulus, E is Young's modulus, and I is the area moment of inertia.

Without a tip mass, the boundary conditions are

$$\begin{aligned} z = 0: \qquad w(0) &= 0 \\ \psi(0) &= \theta_h \\ z = L: \qquad K_s(\partial_z w(L) - \psi(L)) &= 0 \\ K_b\, \partial_z \psi(L) &= 0 \quad . \end{aligned} \tag{3.39}$$

For $\theta_h = 0$, Equation (3.37) describe the quasistatic Timoshenko beam model, see [47].

3.3.2 Port-Hamiltonian Formulation

According to [46], the undamped Timoshenko beam model can be described as a system of first-order PDEs

$$\begin{bmatrix} \dot{p}_t \\ \dot{p}_r \\ \dot{\varepsilon}_t \\ \dot{\varepsilon}_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & \partial_z & 0 \\ 0 & 0 & 1 & \partial_z \\ \partial_z & -1 & 0 & 0 \\ 0 & \partial_z & 0 & 0 \end{bmatrix} \begin{bmatrix} p_t/\rho \\ p_r/I_\rho \\ K_s\varepsilon_t \\ K_b\varepsilon_r \end{bmatrix}. \tag{3.40}$$

Define the (distributed) state vector $\boldsymbol{\chi}(z) \in \mathbb{R}^4$ as

$$\boldsymbol{\chi} = \begin{bmatrix} p_t \\ p_r \\ \varepsilon_t \\ \varepsilon_r \end{bmatrix} = \begin{bmatrix} \rho\partial_t w \\ I_\rho \partial_t \psi \\ \partial_z w - \psi \\ \partial_z \psi \end{bmatrix} \tag{3.41}$$

with the translational and angular momentum densities $p_t(z)$ and $p_r(z)$ and the shear and rotational strains $\varepsilon_t(z)$ and $\varepsilon_r(z)$. In these variables, the total Hamiltonian is $H_b = \int_0^L \mathcal{H}_b dz$ with Hamiltonian density

$$\mathcal{H}_b = \frac{1}{2\rho} p_t^2 + \frac{1}{2I_\rho} p_r^2 + \frac{1}{2} K_s \varepsilon_t^2 + \frac{1}{2} K_b \varepsilon_r^2. \tag{3.42}$$

Due to the horizontal configuration of the manipulator, $\mathcal{H}_b$ only contains a kinetic and an elastic potential energy part. No gravitational energy term occurs.

The vector of the distributed effort variables $\boldsymbol{\eta}(z) \in \mathbb{R}^4$ can be expressed as

$$\boldsymbol{\eta} = \begin{bmatrix} \eta^{p_t} \\ \eta^{p_r} \\ \eta^{\varepsilon_t} \\ \eta^{\varepsilon_r} \end{bmatrix} = \begin{bmatrix} \delta_{p_t} H_b \\ \delta_{p_r} H_b \\ \delta_{\varepsilon_t} H_b \\ \delta_{\varepsilon_r} H_b \end{bmatrix} = \begin{bmatrix} \partial_{p_t} \mathcal{H}_b \\ \partial_{p_r} \mathcal{H}_b \\ \partial_{\varepsilon_t} \mathcal{H}_b \\ \partial_{\varepsilon_r} \mathcal{H}_b \end{bmatrix} = \begin{bmatrix} p_t/\rho \\ p_r/I_\rho \\ K_s \varepsilon_t \\ K_b \varepsilon_r \end{bmatrix}, \tag{3.43}$$

where $\delta_{\boldsymbol{\chi}} H_b$ denotes the (row) vector of variational derivatives of the energy functional. Physically, the effort variables represent the translational and angular velocities as well as the shear force and the bending moment.

Then, Equation (3.40) can be written as a PH system

$$\dot{\boldsymbol{\chi}} = \mathcal{J}_b (\delta_{\boldsymbol{\chi}} H_b)^T, \tag{3.44}$$

where $\mathcal{J}_b$ is a formally skew-adjoint differential operator, see [46]

$$\mathcal{J}_b = \begin{bmatrix} 0 & 0 & \partial_z & 0 \\ 0 & 0 & 1 & \partial_z \\ \partial_z & -1 & 0 & 0 \\ 0 & \partial_z & 0 & 0 \end{bmatrix}.$$

With the definition of the distributed flows $\boldsymbol{\varphi}(z) = -\dot{\boldsymbol{\chi}}(z) \in \mathbb{R}^4$

$$\boldsymbol{\varphi} = \begin{bmatrix} \varphi^{p_t} \\ \varphi^{p_r} \\ \varphi^{\varepsilon_r} \\ \varphi^{\varepsilon_r} \end{bmatrix} = \begin{bmatrix} -\dot{p}_t \\ -\dot{p}_r \\ -\dot{\varepsilon}_t \\ -\dot{\varepsilon}_r \end{bmatrix}, \tag{3.45}$$

the time derivative of the Hamiltonian can be written as

$$\dot{H}_b = \int_0^L \delta_{\boldsymbol{\chi}} H_b \, \dot{\boldsymbol{\chi}} \, dz = -\int_0^L \boldsymbol{\eta}^T \boldsymbol{\varphi} \, dz. \tag{3.46}$$

Replacing the right-hand side of Equation (3.40) for $\boldsymbol{\varphi}(z) = -\dot{\boldsymbol{\chi}}(z)$ in (3.46), and applying integration by parts, the energy balance reads

$$\dot{H}_b = (\eta^{p_t} \eta^{\varepsilon_t} + \eta^{p_r} \eta^{\varepsilon_r})|_0^L. \tag{3.47}$$

Defining the boundary flow and effort variables as

$$\begin{bmatrix} \boldsymbol{\varphi}_\partial^t \\ \boldsymbol{\varphi}_\partial^r \\ \boldsymbol{\eta}_\partial^t \\ \boldsymbol{\eta}_\partial^r \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}^{p_t}|_{\partial\mathcal{Z}} \\ \boldsymbol{\eta}^{p_r}|_{\partial\mathcal{Z}} \\ \boldsymbol{\zeta}\boldsymbol{\eta}^{\varepsilon_t}|_{\partial\mathcal{Z}} \\ \boldsymbol{\zeta}\boldsymbol{\eta}^{\varepsilon_r}|_{\partial\mathcal{Z}} \end{bmatrix}, \qquad \boldsymbol{\zeta} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.48}$$

where $\boldsymbol{\eta}^\nu|_{\partial\mathcal{Z}} = [\eta^\nu(0),\ \eta^\nu(L)]^T$, $\nu \in \{p_t, p_r, \varepsilon_t, \varepsilon_r\}$ denotes the restriction of the effort variables to the boundary of $\mathcal{Z} = [0, L]$, yields the power continuity equation[14] for the undamped case

$$\int_Z \boldsymbol{\eta}^T\, \boldsymbol{\varphi}\, dz + (\boldsymbol{\eta}^t_\partial \boldsymbol{\varphi}^t_\partial + \boldsymbol{\eta}^r_\partial \boldsymbol{\varphi}^r_\partial)\Big|_{\partial\mathcal{Z}} = 0. \tag{3.49}$$

The validity of (3.49) is only due to the linear relation $-\boldsymbol{\varphi} = \mathcal{J}_b\boldsymbol{\eta}$, the application of integration by parts (or Stokes' theorem, in general), and the definition of boundary port variables (3.48). The corresponding subset $\mathcal{D}$ of the bond space (i.e., the space of conjugated power variables), where the power continuity equation (3.49) holds, defines a so-called (Stokes-)Dirac structure (for details see [52], [87]):

$$\mathcal{D} = \left\{(\boldsymbol{\varphi}, \boldsymbol{\varphi}^t_\partial, \boldsymbol{\varphi}^r_\partial, \boldsymbol{\eta}, \boldsymbol{\eta}^t_\partial, \boldsymbol{\eta}^r_\partial) \in \mathcal{F}\times\mathcal{E} \mid \quad -\boldsymbol{\varphi} = \mathcal{J}_b\boldsymbol{\eta} \quad \text{and} \quad (3.48) \quad \text{holds}\right\}. \tag{3.50}$$

The spaces of distributed and boundary flows and efforts are

$$\mathcal{F} = L^2(\mathcal{Z}, \mathbb{R}^4) \times \mathbb{R}^4, \qquad \mathcal{E} = H^1(\mathcal{Z}, \mathbb{R}^4) \times \mathbb{R}^4. \tag{3.51}$$

$L^2(\mathcal{Z}, \mathbb{R}^4)$ refers to the distributed flow vector with four (Lebesgue) square-integrable functions on $\mathcal{Z}$. The distributed efforts make up a four-element vector of square-integrable and (weakly) differentiable functions on $\mathcal{Z}$: $H^1(\mathcal{Z}, \mathbb{R}^4)$. $\mathbb{R}^4$ refers to the finite-dimensional spaces of boundary flows and efforts, respectively.

3.3.3 PH Formulation with Dissipation

Dissipation in the beam can be modeled e.g. by constant viscous damping coefficients (per length) $r_t(z)$, $r_r(z) > 0$, which generate damping forces and momenta proportional to the translational and rotational velocities. These are considered distributed dissipation flows

$$\begin{aligned} \varphi^t_d &= -r_t \eta^{p_t} \\ \varphi^r_d &= -r_r \eta^{p_r}. \end{aligned} \tag{3.52}$$

The damping forces/momenta φ^t_d/φ^r_d, together with the velocities form a distributed dissipative port, which can be used to define a PH model with dissipation of the Timoshenko beam, see [52]. The beam equation (3.44) can be extended in the form

$$\dot{\boldsymbol{\chi}} = (\mathcal{J}_b - \mathcal{R}_b)(\delta_{\boldsymbol{\chi}} H)^T. \tag{3.53}$$

The formally skew-adjoint interconnection operator $\mathcal{J}_b$ is augmented by a positive semi-definite part $\mathcal{R}_b$, which is simply a diagonal matrix $\boldsymbol{R}_0$,

$$\mathcal{J}_b - \mathcal{R}_b = \boldsymbol{P}_1\partial_z + \boldsymbol{P}_0 - \boldsymbol{R}_0, \tag{3.54}$$

with

[14] $$\begin{aligned} (\boldsymbol{\eta}^t_\partial \boldsymbol{\varphi}^t_\partial + \boldsymbol{\eta}^r_\partial \boldsymbol{\varphi}^r_\partial)\big|_{\partial\mathcal{Z}} &= \eta^t_{\partial L}\varphi^t_{\partial L} + \eta^r_{\partial L}\varphi^r_{\partial L} + \eta^t_{\partial 0}\varphi^t_{\partial 0} + \eta^r_{\partial 0}\varphi^r_{\partial 0} \\ &= \eta^{\varepsilon_t}(L)\varphi^{p_t}(L) + \eta^{\varepsilon_r}(L)\varphi^{p_r}(L) - \eta^{\varepsilon_t}(0)\varphi^{p_t}(0) - \eta^{\varepsilon_r}(0)\varphi^{p_r}(0). \end{aligned}$$

$$\boldsymbol{P}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad \boldsymbol{P}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \boldsymbol{R}_0 = \begin{bmatrix} r_t & 0 & 0 & 0 \\ 0 & r_r & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

$\boldsymbol{P}_1 \in \mathbb{R}^{4\times4}$ represents the exchange of the kinetic and elastic energy between the translational and rotational state variables, respectively. $\boldsymbol{P}_0 \in \mathbb{R}^{4\times4}$ describes the coupling of the translational and rotational energy, and $\boldsymbol{R}_0 \in \mathbb{R}^{4\times4}$ models dissipation. The distributed part of the structure equations (linear relations between flows and efforts) hence can be rewritten as

$$-\boldsymbol{\varphi} = \boldsymbol{P}_1 \partial_z \boldsymbol{\eta} + (\boldsymbol{P}_0 - \boldsymbol{R}_0)\boldsymbol{\eta}_*, \qquad \boldsymbol{\eta}_* = \boldsymbol{\eta}, \tag{3.55}$$

where we use two notations $\boldsymbol{\eta}$ and $\boldsymbol{\eta}_*$ for the identical effort vector, depending on whether it is differentiated or not. Taking into account their different geometric nature, they can be approximated in the finite-dimensional space with different polynomial approximation bases by the so-called structure-preserving spatial discretization, which is introduced in the next section.

3.4 Structure-Preserving Spatial Discretization

A key step in structure-preserving discretization (and model order reduction) of PH systems is to map a Stokes-Dirac structure to an approximate Dirac structure. In a further step, the distributed constitutive equation $\boldsymbol{e} = (\delta_{\boldsymbol{x}} H)^T$ must be consistently approximated by a finite-dimensional correspondence $\boldsymbol{e}_d = \nabla H(\boldsymbol{x}_d)$.

The geometric discretization to approximate the beam model by a finite-dimensional PH system, see [97], is based on the *pseudo-spectral* (PS) *method* for canonical systems of two conservation laws presented in [60]. Taking into account the different geometric nature of flow and effort variables, different polynomial approximation bases are used. A suitable choice for the beam model is the Lagrange interpolation polynomials as approximation bases, which match the distributed variables at the collocation points. Particular attention is paid to the constitutive relations and the representation of the discretized efforts. Moreover, the boundary port variables are retained and appear as inputs and outputs in the resulting lumped parameter PH model, which, therefore, is a physically consistent approximation. This method is outstanding because fewer grid points are needed compared to local methods to achieve the same accuracy. It has been proven to provide a good approximation of system properties such as the spectrum of differential operators (for details see [60], [89]). Furthermore, in several space dimensions, the relative coarse grids that are sufficient for the most accurate requirements allow very time and memory-efficient computing, see [20].

In this chapter, we summarize the contents of [96] and [97], which presented the discretization procedure for the PH Timoshenko beam model and discussed the choice of collocation points and the numerical accuracy of the modal approximations.

3.4.1 Pseudo-Spectral Discretization of the PH Beam Model

Approximation bases. The idea, see [60], is to define different approximation bases for the flows φ^ν and the efforts η^ν, $\nu \in \{p_t, p_r, \varepsilon_t, \varepsilon_r\}$. The same holds for η^ν and η^ν_*, depending on whether the effort is differentiated or not[15]:

$$\begin{aligned}
\varphi^\nu(z) &\approx \sum_{k=0}^{N-1} f_k^\nu w_k^f(z) \\
\eta_*^\nu(z) &\approx \sum_{k=0}^{N-1} e_{*,k}^\nu w_k^f(z) \\
\eta^\nu(z) &\approx \sum_{i=0}^{N} e_i^\nu w_i^e(z).
\end{aligned} \tag{3.56}$$

The time-dependent coefficients are collected in the vectors $\boldsymbol{f}^\nu, \boldsymbol{e}_*^\nu \in \mathbb{R}^N$ and $\boldsymbol{e}^\nu \in \mathbb{R}^{N+1}$. $w_k^f(z)$, $k = 0, \dots, N-1$, and $w_i^e(z)$, $i = 0, \dots, N$, are the basis functions for flows and efforts that satisfy the exact differentiation or compatibility condition, see e.g. [89],

$$\begin{cases}
\bar{\mathcal{E}} &= \operatorname{span}\{w_0^e, \dots, w_N^e\} \\
\bar{\mathcal{F}} &= \operatorname{span}\{w_0^f, \dots, w_{N-1}^f\} \\
\partial_z(\bar{\mathcal{E}}) &= \bar{\mathcal{F}}
\end{cases}. \tag{3.57}$$

In our problem, the Lagrange interpolating polynomials of degree N and $N-1$ are a suitable choice:

$$\begin{aligned}
w_i^e(z) &= \prod_{j=0, j\neq i}^{N} \frac{z - \xi_j}{\xi_i - \xi_j} \\
w_k^f(z) &= \prod_{j=0, j\neq k}^{N-1} \frac{z - z_j}{z_k - z_j}.
\end{aligned} \tag{3.58}$$

$\xi_i \in [0,\, L]$, $i = 0, \dots, N$ and $z_k \in [0,\, L]$, $k = 0, \dots, N-1$, are the collocation points for w_i^e and w_k^f, respectively, and

$$\begin{aligned}
w_i^e(\xi_j) &= \delta_{ij} \\
w_k^f(z_j) &= \delta_{kj}
\end{aligned} \tag{3.59}$$

hold with δ_{ij} the Kronecker delta. The choice of collocation points is discussed in Section 3.4.2.

Approximation of the Stokes-Dirac structure. We first consider the undamped PH beam model (3.44) and approximate it in the finite-dimensional space.

[15] Instead of two different effort approximation spaces, one could also redefine the flows as in [75].

Proposition 3.3. Denote $\boldsymbol{w}^e = [w_0^e, \ldots, w_N^e]^T$ the vector of effort basis functions and

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{w}^e(0)^T \\ \boldsymbol{w}^e(L)^T \end{bmatrix}. \tag{3.60}$$

Let $\boldsymbol{f}_\partial^{t/r} = \boldsymbol{\varphi}_\partial^{t/r} = [\varphi_0^{t/r},\ \varphi_L^{t/r}]^T$ and $\boldsymbol{e}_\partial^{t/r} = \boldsymbol{\eta}_\partial^{t/r} = [\eta_0^{t/r},\ \eta_L^{t/r}]^T$ be the vectors of boundary flows and efforts corresponding to translational/rotational motion, see Equation (3.48). Inserting (3.56) into (3.55) and (3.48), one obtains the linear system of equations

$$\begin{aligned}
\begin{bmatrix} -\boldsymbol{f}^{p_t} \\ -\boldsymbol{f}^{p_r} \\ -\boldsymbol{f}^{\varepsilon_t} \\ -\boldsymbol{f}^{\varepsilon_r} \end{bmatrix} &= \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D} \\ \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_*^{p_t} \\ \boldsymbol{e}_*^{p_r} \\ \boldsymbol{e}_*^{\varepsilon_t} \\ \boldsymbol{e}_*^{\varepsilon_r} \end{bmatrix} \\
\begin{bmatrix} \boldsymbol{f}_\partial^t \\ \boldsymbol{f}_\partial^r \\ \boldsymbol{e}_\partial^t \\ \boldsymbol{e}_\partial^r \end{bmatrix} &= \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\zeta\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\zeta\Phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix}.
\end{aligned} \tag{3.61}$$

The elements of the derivative matrix $\boldsymbol{D} \in \mathbb{R}^{N\times(N+1)}$ are given by the spatial derivative of the effort basis functions at the flow collocation points:

$$[\boldsymbol{D}]_{k+1,i+1} = \partial_z w_i^e(z_k), \tag{3.62}$$

where $i = 0, \ldots, N$, $k = 0, \ldots, N-1$.

In accordance with the distributed parameter model, the elements of the matrix $\boldsymbol{P}_0$ map to the coupling terms with identity matrices $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ in the second term on the right-hand side of Equation (3.61).

Proposition 3.4. If the effort interpolation polynomials $w_i^e(z)$ are chosen with collocation points ξ_i, and $\boldsymbol{D}$ is determined according to Equation (3.62) for a given set of collocation points z_k, then the vectors of interpolation polynomials $w_k^f(z)$ are related via

$$\boldsymbol{w}^f(z) = (\boldsymbol{D}^+)^T \partial_z \boldsymbol{w}^e \tag{3.63}$$

with pseudo inverse $\boldsymbol{D}^+ = \boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}$.

Proof. Inserting (3.56) in (3.57), it becomes

$$\sum_{k=0}^{N-1} w_k^f(z) f_k = \sum_{i=0}^{N} \left[\partial_z w_i^e(z)\right] e_i, \tag{3.64}$$

or

$$\left[\boldsymbol{w}^f(z)\right]^T \boldsymbol{f} = \left[\partial_z \boldsymbol{w}^e(z)\right]^T \boldsymbol{e} \tag{3.65}$$

with

$$\boldsymbol{w}^f(z) = \begin{bmatrix} w_0^f(z) \\ \vdots \\ w_{N-1}^f(z) \end{bmatrix}$$

$$\partial_z \boldsymbol{w}^e(z) = \begin{bmatrix} \partial_z w_0^e(z) \\ \vdots \\ \partial_z w_N^e(z) \end{bmatrix}.$$

Since the derivation relation[16] between flows and efforts is $\boldsymbol{f} = \boldsymbol{D}\boldsymbol{e}$ (see also [60]), Equation (3.65) can be rewritten as

$$\left[\boldsymbol{w}^f(z)\right]^T \boldsymbol{D}\boldsymbol{e} = \left[\partial_z \boldsymbol{w}^e(z)\right]^T \boldsymbol{e}, \tag{3.66}$$

where

$$\boldsymbol{D}^T \boldsymbol{w}^f(z) = \partial_z \boldsymbol{w}^e(z). \tag{3.67}$$

Then, the basis functions of the flows can be directly determined by

$$\boldsymbol{w}^f(z) = \underbrace{\left[\boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}\right]^T}_{(\boldsymbol{D}^+)^T} \partial_z \boldsymbol{w}^e(z). \tag{3.68}$$

$\boldsymbol{D}^+ = \boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}$ denotes the pseudo inverse of $\boldsymbol{D}$. ■

According to Equation (3.20), we evaluate the bilinear product using the polynomial approximation bases of flows and efforts (3.56). This leads to a symmetric bilinear form on the finite-dimensional bond space of discrete flows and efforts $\mathcal{F}_r \times \mathcal{E}_r$ with $(\boldsymbol{f}^{p_t}, \boldsymbol{f}^{p_r}, \boldsymbol{f}^{\varepsilon_t}, \boldsymbol{f}^{\varepsilon_r}, \boldsymbol{f}^t_\partial, \boldsymbol{f}^r_\partial) \in \mathcal{F}_r = \mathbb{R}^{4N+4}$ and $(\boldsymbol{e}^{p_t}, \boldsymbol{e}^{p_r}, \boldsymbol{e}^{\varepsilon_t}, \boldsymbol{e}^{\varepsilon_r}, \boldsymbol{e}^t_\partial, \boldsymbol{e}^r_\partial) \in \mathcal{E}_r = \mathbb{R}^{4N+8}$:

$$\begin{bmatrix} \boldsymbol{e}_1^{p_t} \\ \boldsymbol{e}_1^{p_r} \\ \boldsymbol{e}_1^{\varepsilon_t} \\ \boldsymbol{e}_1^{\varepsilon_r} \\ \boldsymbol{e}_{1\partial}^t \\ \boldsymbol{e}_{1\partial}^r \\ \boldsymbol{f}_1^{p_t} \\ \boldsymbol{f}_1^{p_r} \\ \boldsymbol{f}_1^{\varepsilon_t} \\ \boldsymbol{f}_1^{\varepsilon_r} \\ \boldsymbol{f}_{1\partial}^t \\ \boldsymbol{f}_{1\partial}^r \end{bmatrix}^T \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\ \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_2^{p_t} \\ \boldsymbol{e}_2^{p_r} \\ \boldsymbol{e}_2^{\varepsilon_t} \\ \boldsymbol{e}_2^{\varepsilon_r} \\ \boldsymbol{e}_{2\partial}^t \\ \boldsymbol{e}_{2\partial}^r \\ \boldsymbol{f}_2^{p_t} \\ \boldsymbol{f}_2^{p_r} \\ \boldsymbol{f}_2^{\varepsilon_t} \\ \boldsymbol{f}_2^{\varepsilon_r} \\ \boldsymbol{f}_{2\partial}^t \\ \boldsymbol{f}_{2\partial}^r \end{bmatrix}. \tag{3.69}$$

[16] Note that the approximation bases for η_* are the same as ones for flows φ.

The elements of the non-square matrix $\boldsymbol{M} \in \mathbb{R}^{(N+1)\times N}$ are

$$[\boldsymbol{M}]_{i+1,k+1} = \int_0^L w_i^e(z) w_k^f(z)\, dz. \tag{3.70}$$

Then, the energy balance (3.46) can be approximated by this bilinear form between the vectors of discrete flows and efforts as follows

$$\dot{H}_b \approx \sum_{\nu\in\{p_t,p_r,\varepsilon_t,\varepsilon_r\}} (\boldsymbol{e}^\nu)^T \boldsymbol{M} \boldsymbol{f}^\nu. \tag{3.71}$$

As a consequence, the power continuity equation (3.49) is approximated by

$$\sum_{\nu\in\{p_t,p_r,\varepsilon_t,\varepsilon_r\}} (\boldsymbol{e}^\nu)^T \boldsymbol{M} \boldsymbol{f}^\nu + \sum_{\mu\in\{t,r\}} (\boldsymbol{e}_\partial^\mu)^T \boldsymbol{f}_\partial^\mu = 0. \tag{3.72}$$

Because of the dimension of the discrete flows and efforts, the kernel of matrix $\boldsymbol{M}$ is non-empty. Therefore, the symmetric pairing (3.69) is degenerated and is not suitable to define a Dirac structure on the discrete spaces $\mathcal{F}_r \times \mathcal{E}_r$.

Dirac structure. To obtain a non-degenerate power pairing, vectors of reduced discrete effort variables $\tilde{\boldsymbol{e}}^\nu \in \mathbb{R}^N$ are defined as

$$\tilde{\boldsymbol{e}}^\nu = \boldsymbol{M}^T \boldsymbol{e}^\nu. \tag{3.73}$$

These shall be – we discretize the constitutive equation $\boldsymbol{\eta} = (\delta_{\boldsymbol{\chi}} H_b)^T$ – derived from a discrete energy. Note that $\dot{\boldsymbol{\chi}} = -\boldsymbol{\varphi}$ holds, i.e., states and flows are discretized with respect to the same basis. We substitute $\chi^\nu(z) = \sum_{k=0}^{N-1} \tilde{x}_k^\nu w_k^f(z)$ in the Hamiltonian functional with χ^ν the components of $\boldsymbol{\chi}$ according to Equation (3.41). We obtain the energy approximation

$$\tilde{H}_b \approx \frac{1}{2} \sum_\nu c^\nu (\tilde{\boldsymbol{x}}^\nu)^T \boldsymbol{S} \tilde{\boldsymbol{x}}^\nu, \qquad c^\nu \in \{\frac{1}{\rho}, \frac{1}{I_\rho}, K_s, K_b\}, \tag{3.74}$$

where the matrix $\boldsymbol{S} = \boldsymbol{S}^T \in \mathbb{R}^{N\times N}$ consists of the elements

$$[\boldsymbol{S}]_{i+1,j+1} = \int_0^L w_i^f(z) w_j^f(z) dz. \tag{3.75}$$

On the one hand, we can derive the discretized constitutive equations

$$\tilde{\boldsymbol{e}}^\nu = \left(\frac{\partial \tilde{H}_b}{\partial \tilde{\boldsymbol{x}}^\nu}\right)^T = c^\nu \boldsymbol{S} \tilde{\boldsymbol{x}}^\nu. \tag{3.76}$$

On the other hand, the discretized effort vectors $\boldsymbol{e}_*^\nu$ are defined with respect to the same basis as the states according to Equation (3.43), so that the following relationship holds immediately:

$$\boldsymbol{e}_*^\nu = c^\nu \tilde{\boldsymbol{x}}^\nu = \boldsymbol{S}^{-1} \tilde{\boldsymbol{e}}^\nu = \boldsymbol{S}^{-1} \boldsymbol{M}^T \boldsymbol{e}^\nu. \tag{3.77}$$

Proposition 3.5. Define the flow variables $\tilde{\boldsymbol{f}}^T := (\boldsymbol{f}^{p_t}, \boldsymbol{f}^{p_r}, \boldsymbol{f}^{\varepsilon_t}, \boldsymbol{f}^{\varepsilon_r}, \boldsymbol{f}^t_\partial, \boldsymbol{f}^r_\partial)$ and the effort variables $\tilde{\boldsymbol{e}}^T := (\tilde{\boldsymbol{e}}^{p_t}, \tilde{\boldsymbol{e}}^{p_r}, \tilde{\boldsymbol{e}}^{\varepsilon_t}, \tilde{\boldsymbol{e}}^{\varepsilon_r}, \boldsymbol{e}^t_\partial, \boldsymbol{e}^r_\partial)$ in the bond space $\tilde{\mathcal{F}} \times \tilde{\mathcal{E}} = \mathbb{R}^{4N+4} \times \mathbb{R}^{4N+4}$. With the definition (3.73), the discretized relation (3.61) can be written in terms of the image representation of a Dirac structure

$$\begin{aligned}
\begin{bmatrix} \boldsymbol{f}^{p_t} \\ \boldsymbol{f}^{p_r} \\ \boldsymbol{f}^{\varepsilon_t} \\ \boldsymbol{f}^{\varepsilon_r} \\ \boldsymbol{f}^t_\partial \\ \boldsymbol{f}^r_\partial \end{bmatrix} &= \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{S}^{-1}\boldsymbol{M}^T & -\boldsymbol{D} \\ -\boldsymbol{D} & \boldsymbol{S}^{-1}\boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Phi} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{E}^T} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix} \\
\begin{bmatrix} \tilde{\boldsymbol{e}}^{p_t} \\ \tilde{\boldsymbol{e}}^{p_r} \\ \tilde{\boldsymbol{e}}^{\varepsilon_t} \\ \tilde{\boldsymbol{e}}^{\varepsilon_r} \\ \boldsymbol{e}^t_\partial \\ \boldsymbol{e}^r_\partial \end{bmatrix} &= \underbrace{\begin{bmatrix} \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\zeta\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\zeta\Phi} \end{bmatrix}}_{\boldsymbol{F}^T} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix},
\end{aligned} \tag{3.78}$$

where the vector subspace of $\tilde{\mathcal{F}} \times \tilde{\mathcal{E}}$ is defined as

$$\mathcal{D} = \left\{ \tilde{\boldsymbol{f}} \in \tilde{\mathcal{F}},\ \tilde{\boldsymbol{e}} \in \tilde{\mathcal{E}} \mid \tilde{\boldsymbol{f}} = \boldsymbol{E}^T\boldsymbol{\lambda},\ \tilde{\boldsymbol{e}} = \boldsymbol{F}^T\boldsymbol{\lambda},\ \boldsymbol{\lambda} \in \mathbb{R}^{4N+4} \right\}. \tag{3.79}$$

Finally the following approximate power continuity equation holds:

$$\sum_{\nu \in \{p_t, p_r, \varepsilon_t, \varepsilon_r\}} (\tilde{\boldsymbol{e}}^\nu)^T \boldsymbol{f}^\nu + \sum_{\mu \in \{t,r\}} (\boldsymbol{e}^\mu_\partial)^T \boldsymbol{f}^\mu_\partial = 0. \tag{3.80}$$

Proof. According to Proposition 3.1, two conditions on the structure matrices shall be checked.

- Skew-symmetry: $\boldsymbol{E}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{E}^T = \boldsymbol{0}$

  After the calculation, it results

$$\boldsymbol{E}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{E}^T = \begin{bmatrix} \boldsymbol{0} & \Psi \\ \Psi^T & \boldsymbol{0} \end{bmatrix} \tag{3.81}$$

  with

$$\Psi = \begin{bmatrix} -\boldsymbol{D}^T\boldsymbol{M}^T + \boldsymbol{\Phi}^T\boldsymbol{\zeta}\boldsymbol{\Phi} - \boldsymbol{M}\boldsymbol{D} & \boldsymbol{0} \\ \underbrace{\boldsymbol{M}\boldsymbol{S}^{-T}\boldsymbol{M}^T - \boldsymbol{M}\boldsymbol{S}^{-1}\boldsymbol{M}^T}_{=\boldsymbol{0}\text{ due to }\boldsymbol{S}=\boldsymbol{S}^T} & -\boldsymbol{D}^T\boldsymbol{M}^T + \boldsymbol{\Phi}^T\boldsymbol{\zeta}\boldsymbol{\Phi} - \boldsymbol{M}\boldsymbol{D} \end{bmatrix} \tag{3.82}$$

According to Condition (B.6), see the derivation in Appendix B, it can be verified that $\Psi = \mathbf{0}$ and therefore

$$\boldsymbol{E}\boldsymbol{F}^T + \boldsymbol{F}\boldsymbol{E}^T = \mathbf{0}. \tag{3.83}$$

- Rank condition: $[\boldsymbol{E} : \boldsymbol{F}]$ is full rank $4N + 4$

The structure matrices $\boldsymbol{F},\ \boldsymbol{E} \in \mathbb{R}^{(4N+4)\times(4N+4)}$ are

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{M} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{M} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{M} & \mathbf{0} & \boldsymbol{\Phi}^T\boldsymbol{\zeta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{M} & \mathbf{0} & \boldsymbol{\Phi}^T\boldsymbol{\zeta} \end{bmatrix}$$

$$\boldsymbol{E} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & -\boldsymbol{D}^T & \mathbf{0} & \boldsymbol{\Phi}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{M}\boldsymbol{S}^{-T} & -\boldsymbol{D}^T & \mathbf{0} & \boldsymbol{\Phi}^T \\ -\boldsymbol{D}^T & -\boldsymbol{M}\boldsymbol{S}^{-T} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{D}^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

To verify this condition, we shall only show the matrix $\boldsymbol{E}$ has rank $4N + 4$. Thus, according to the structure of the matrix $\boldsymbol{E}$, it is sufficient to show that

$$\{\boldsymbol{D}_1^T,\ \boldsymbol{D}_2^T,\ \ldots,\ \boldsymbol{D}_N^T,\ \boldsymbol{w}^e(0),\ \boldsymbol{w}^e(L)\} \tag{3.84}$$

is an independent set. $\boldsymbol{D}_k^T = \partial_z \boldsymbol{w}^e(z_k)$ are the N column vectors of the transpose derivative matrix $\boldsymbol{D}$ defined in Equation (3.62), where z_k, $k = 0, \ldots, N-1$ are the N collocation points for the interpolation polynomial $w_k^f(z)$.

According to [60], we assume that the mentioned set is dependent. In this case $\boldsymbol{w}^e(0)$ (the same for $\boldsymbol{w}^e(L)$) is a linear combination of $\partial_z \boldsymbol{w}^e(z_k)$. $\partial_z \boldsymbol{w}^e(z)$ are polynomials of order $N-1$ and uniquely determined by the values $\partial_z \boldsymbol{w}^e(z_k)$ at the collocation points z_k, $k = 0, 1, \ldots, N-1$. Thus, these N interpolating conditions would be sufficient to characterize uniquely the N-th order polynomials

$$\boldsymbol{w}^e(z) = \boldsymbol{w}^e(0) + \int_0^z \partial_z \boldsymbol{w}^e(\alpha) d\alpha. \tag{3.85}$$

According to the uniqueness of interpolating polynomials (see e.g. [55]), we need at least $N+1$ collocation points for a polynomial of degree at most N or else Equation (3.85) cannot be fully solved for. However, the above assumption contradicts this requirement. Hence, the set (3.84) is independent, and $[\boldsymbol{E} : \boldsymbol{F}]$ is full rank $4N + 4$.

■

Approximation of the energy dissipation. Similar to the lossless coupling terms, elements of the matrix $\boldsymbol{R}_0$ map to the dissipation terms with identity matrices $\boldsymbol{I} \in \mathbb{R}^{N\times N}$ in the second term on the right-hand side of Equation (3.61):

$$-\begin{bmatrix} \boldsymbol{f}^{p_t} \\ \boldsymbol{f}^{p_r} \\ \boldsymbol{f}^{\varepsilon_t} \\ \boldsymbol{f}^{\varepsilon_r} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{D} \\ \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix} + \begin{bmatrix} -r_t\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -r_r\boldsymbol{I} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_*^{p_t} \\ \boldsymbol{e}_*^{p_r} \\ \boldsymbol{e}_*^{\varepsilon_t} \\ \boldsymbol{e}_*^{\varepsilon_r} \end{bmatrix}.$$

Based on the relation (3.77), the image representation (3.78) can be extended by the dissipation terms, and the discrete structure equations become as

$$\begin{aligned} \begin{bmatrix} \boldsymbol{f}^{p_t} \\ \boldsymbol{f}^{p_r} \\ \boldsymbol{f}^{\varepsilon_t} \\ \boldsymbol{f}^{\varepsilon_r} \\ \boldsymbol{f}_\partial^t \\ \boldsymbol{f}_\partial^r \end{bmatrix} &= \begin{bmatrix} r_t\boldsymbol{S}^{-1}\boldsymbol{M}^T & \boldsymbol{0} & -\boldsymbol{D} & \boldsymbol{0} \\ \boldsymbol{0} & r_r\boldsymbol{S}^{-1}\boldsymbol{M}^T & -\boldsymbol{S}^{-1}\boldsymbol{M}^T & -\boldsymbol{D} \\ -\boldsymbol{D} & \boldsymbol{S}^{-1}\boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{D} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Phi} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Phi} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix} \\ \begin{bmatrix} \tilde{\boldsymbol{e}}^{p_t} \\ \tilde{\boldsymbol{e}}^{p_r} \\ \tilde{\boldsymbol{e}}^{\varepsilon_t} \\ \tilde{\boldsymbol{e}}^{\varepsilon_r} \\ \boldsymbol{e}_\partial^t \\ \boldsymbol{e}_\partial^r \end{bmatrix} &= \begin{bmatrix} \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{M}^T \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\zeta\Phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\zeta\Phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}^{p_t} \\ \boldsymbol{e}^{p_r} \\ \boldsymbol{e}^{\varepsilon_t} \\ \boldsymbol{e}^{\varepsilon_r} \end{bmatrix}. \end{aligned} \tag{3.86}$$

Input-/output representation. The I/O representation, which relates the approximate port variables with boundary port variables[17], is derived from elementary matrix operations.

[17] Notation: $\boldsymbol{f}_\partial^\mu = [f_{\partial 0}^\mu,\ f_{\partial L}^\mu]^T$, $\boldsymbol{e}_\partial^\mu = [e_{\partial 0}^\mu,\ e_{\partial L}^\mu]^T$, $\mu \in \{t, r\}$.

Proposition 3.6. We collect the boundary port variables which play the role of inputs on the right hand side to obtain the I/O representation of the Dirac structure $\mathcal{D}$ defined in the Proposition 3.5 as follows

$$\begin{bmatrix} \begin{pmatrix} \boldsymbol{f}^{p_t} \\ e^t_{\partial 0} \end{pmatrix} \\ \begin{pmatrix} \boldsymbol{f}^{p_r} \\ e^r_{\partial 0} \end{pmatrix} \\ \begin{pmatrix} \boldsymbol{f}^{\varepsilon_t} \\ f^t_{\partial L} \end{pmatrix} \\ \begin{pmatrix} \boldsymbol{f}^{\varepsilon_r} \\ f^r_{\partial L} \end{pmatrix} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{J}_1 \\ \boldsymbol{J}_2 & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \begin{pmatrix} \tilde{\boldsymbol{e}}^{p_t} \\ f^t_{\partial 0} \end{pmatrix} \\ \begin{pmatrix} \tilde{\boldsymbol{e}}^{p_r} \\ f^r_{\partial 0} \end{pmatrix} \\ \begin{pmatrix} \tilde{\boldsymbol{e}}^{\varepsilon_t} \\ e^t_{\partial L} \end{pmatrix} \\ \begin{pmatrix} \tilde{\boldsymbol{e}}^{\varepsilon_r} \\ e^r_{\partial L} \end{pmatrix} \end{bmatrix} \tag{3.87}$$

with[18]

$$\boldsymbol{J}_1 = \begin{bmatrix} \begin{pmatrix} -\boldsymbol{D} \\ -\boldsymbol{w}^e_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_L \end{pmatrix}^{-1} & \boldsymbol{0} \\ \begin{pmatrix} -\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_L \end{pmatrix}^{-1} & \begin{pmatrix} -\boldsymbol{D} \\ -\boldsymbol{w}^e_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_L \end{pmatrix}^{-1} \end{bmatrix}$$

$$\boldsymbol{J}_2 = \begin{bmatrix} \begin{pmatrix} -\boldsymbol{D} \\ \boldsymbol{w}^e_L \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_0 \end{pmatrix}^{-1} & \begin{pmatrix} \boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_0 \end{pmatrix}^{-1} \\ \boldsymbol{0} & \begin{pmatrix} -\boldsymbol{D} \\ \boldsymbol{w}^e_L \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_0 \end{pmatrix}^{-1} \end{bmatrix}$$

and

$$\boldsymbol{R}_1 = \begin{bmatrix} \begin{pmatrix} r_t\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_0 \end{pmatrix}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \begin{pmatrix} r_r\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}^e_0 \end{pmatrix}^{-1} \end{bmatrix}.$$

In fact, $\boldsymbol{J}_1 = -\boldsymbol{J}_2^T$, i.e., the conservative part of the Timoshenko beam model with boundary energy flow is correctly represented in the relations between the port variables (3.87).

[18] Short notation: $\boldsymbol{w}^e_0 = (\boldsymbol{w}^e(0))^T$, $\boldsymbol{w}^e_L = (\boldsymbol{w}^e(L))^T$

Proof. We compute the anti-diagonal elements' sum as follows

$$\begin{aligned}
&\boldsymbol{J}_1 + \boldsymbol{J}_2^T \\
&= \begin{bmatrix}
\begin{pmatrix} -\boldsymbol{D} \\ -\boldsymbol{w}_0^e \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} + \left(\begin{pmatrix} -\boldsymbol{D} \\ \boldsymbol{w}_L^e \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_0^e \end{pmatrix}^{-1} \right)^T & \boldsymbol{0} \\
\begin{pmatrix} -\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} + \left(\begin{pmatrix} \boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_0^e \end{pmatrix}^{-1} \right)^T & * \end{bmatrix},
\end{aligned} \tag{3.88}$$

where the element $*$ is the same as the first diagonal element. In order to show $\boldsymbol{J}_1 + \boldsymbol{J}_2^T = \boldsymbol{0}$, both of the following two elements should equal zero:

$$\begin{aligned}
&\begin{pmatrix} -\boldsymbol{D} \\ -\boldsymbol{w}_0^e \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} + \left(\begin{pmatrix} -\boldsymbol{D} \\ \boldsymbol{w}_L^e \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_0^e \end{pmatrix}^{-1} \right)^T \\
&= \begin{pmatrix} -\boldsymbol{D} \\ -\boldsymbol{w}_0^e \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} + \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix}^{-1} \begin{pmatrix} -\boldsymbol{D}^T & (\boldsymbol{w}_L^e)^T \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix}^{-1} \left(\begin{pmatrix} -\boldsymbol{D}^T & (\boldsymbol{w}_L^e)^T \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix} + \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix} \begin{pmatrix} -\boldsymbol{D} \\ -\boldsymbol{w}_0^e \end{pmatrix} \right) \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} \\
&= \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix}^{-1} \underbrace{\left(-\boldsymbol{D}^T\boldsymbol{M}^T + (\boldsymbol{w}_L^e)^T\boldsymbol{w}_L^e - \boldsymbol{M}\boldsymbol{D} - (\boldsymbol{w}_0^e)^T\boldsymbol{w}_0^e \right)}_{=①} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1},
\end{aligned} \tag{3.89}$$

$$\begin{aligned}
&\begin{pmatrix} -\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} + \left(\begin{pmatrix} \boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_0^e \end{pmatrix}^{-1} \right)^T \\
&= \begin{pmatrix} -\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} + \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{M}\boldsymbol{S}^{-T} & \boldsymbol{0} \end{pmatrix} \\
&= \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix}^{-1} \left(\begin{pmatrix} \boldsymbol{M}\boldsymbol{S}^{-T} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix} + \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix} \begin{pmatrix} -\boldsymbol{S}^{-1}\boldsymbol{M}^T \\ \boldsymbol{0} \end{pmatrix} \right) \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1} \\
&= \begin{pmatrix} \boldsymbol{M} & (\boldsymbol{w}_0^e)^T \end{pmatrix}^{-1} \underbrace{\left(\boldsymbol{M}\boldsymbol{S}^{-T}\boldsymbol{M}^T - \boldsymbol{M}\boldsymbol{S}^{-1}\boldsymbol{M}^T \right)}_{=\boldsymbol{0} \text{ due to } \boldsymbol{S}=\boldsymbol{S}^T} \begin{pmatrix} \boldsymbol{M}^T \\ \boldsymbol{w}_L^e \end{pmatrix}^{-1}.
\end{aligned} \tag{3.90}$$

According to Condition (B.6), $① = \boldsymbol{0}$. Thus, it is verified that $\boldsymbol{J}_1 = -\boldsymbol{J}_2^T$.

■

PH state space model. We denote $\boldsymbol{u}_b$, $\boldsymbol{y}_b \in \mathbb{R}^4$ the vectors of boundary inputs, and collocated, power-conjugated outputs, which are composed of the elements of the boundary flow and effort vectors $\boldsymbol{f}_\partial^t$, $\boldsymbol{f}_\partial^r$, $\boldsymbol{e}_\partial^t$, $\boldsymbol{e}_\partial^r$, see [46]. In terms of the physical boundary variables, we have

$$\boldsymbol{u}_b = \begin{bmatrix} \boldsymbol{u}_b^0 \\ \boldsymbol{u}_b^L \end{bmatrix} = \left[\begin{array}{c} f_{\partial 0}^t \\ f_{\partial 0}^r \\ \hdashline e_{\partial L}^t \\ e_{\partial L}^r \end{array}\right] = \left[\begin{array}{c} \eta^{p_t}(0) \\ \eta^{p_r}(0) \\ \hdashline \eta^{\varepsilon_t}(L) \\ \eta^{\varepsilon_r}(L) \end{array}\right] = \left[\begin{array}{c} \dot{w}(0) \\ \dot{\psi}(0) \\ \hdashline Q_s(L) \\ M_b(L) \end{array}\right] \tag{3.91}$$

$$\boldsymbol{y}_b = \begin{bmatrix} \boldsymbol{y}_b^0 \\ \boldsymbol{y}_b^L \end{bmatrix} = \left[\begin{array}{c} e_{\partial 0}^t \\ e_{\partial 0}^r \\ \hdashline f_{\partial L}^t \\ f_{\partial L}^r \end{array}\right] = \left[\begin{array}{c} -\eta^{\varepsilon_t}(0) \\ -\eta^{\varepsilon_r}(0) \\ \hdashline \eta^{p_t}(L) \\ \eta^{p_r}(L) \end{array}\right] = \left[\begin{array}{c} -Q_s(0) \\ -M_b(0) \\ \hdashline \dot{w}(L) \\ \dot{\psi}(L) \end{array}\right], \tag{3.92}$$

where $\dot{w}$ and $\dot{\psi}$ denote the translational and angular velocities, and Q_s and M_b are the shear force and the bending moment, respectively.

Substitution of $-\boldsymbol{f}^\nu = \dot{\tilde{\boldsymbol{x}}}^\nu$, as well as $\tilde{\boldsymbol{e}}^\nu = c^\nu \boldsymbol{S}\tilde{\boldsymbol{x}}^\nu$ in Equation (3.87), rearrangement of rows and columns, and the concatenation of the states in the overall state vector $\tilde{\boldsymbol{x}}_b \in \mathbb{R}^{4N}$ gives a linear PH state space model of the beam in the form[19]

$$\dot{\tilde{\boldsymbol{x}}}_b = (\tilde{\boldsymbol{J}}_b - \tilde{\boldsymbol{R}}_b)\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{x}}_b + \tilde{\boldsymbol{B}}_b\boldsymbol{u}_b \tag{3.93a}$$

$$\boldsymbol{y}_b = \tilde{\boldsymbol{B}}_b^T\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{x}}_b + \tilde{\boldsymbol{D}}_b\boldsymbol{u}_b \tag{3.93b}$$

with $\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{x}}_b = \nabla\tilde{H}_b(\tilde{\boldsymbol{x}}_b)$ the linear gradient of the quadratic approximate Hamiltonian $\tilde{H}_b = \frac{1}{2}(\tilde{\boldsymbol{x}}_b)^T\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{x}}_b$, where $\tilde{\boldsymbol{Q}}_b = \text{blockdiag}\left[\frac{1}{\rho}\boldsymbol{S},\ \frac{1}{I_\rho}\boldsymbol{S},\ K_b\boldsymbol{S},\ K_s\boldsymbol{S}\right] \in \mathbb{R}^{4N\times 4N}$. The input and feedthrough matrices can be split into the corresponding parts for the boundary at $z = 0$ and $z = L$, respectively:

$$\tilde{\boldsymbol{B}}_b = \begin{bmatrix} \tilde{\boldsymbol{B}}_{b_0} & \tilde{\boldsymbol{B}}_{b_L} \end{bmatrix} \tag{3.94}$$

$$\tilde{\boldsymbol{D}}_b = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix}. \tag{3.95}$$

The occurrence of the feedthrough matrix stems from the applied discretization scheme, which allows an interpretation in terms of the rigid body motion of the beam.

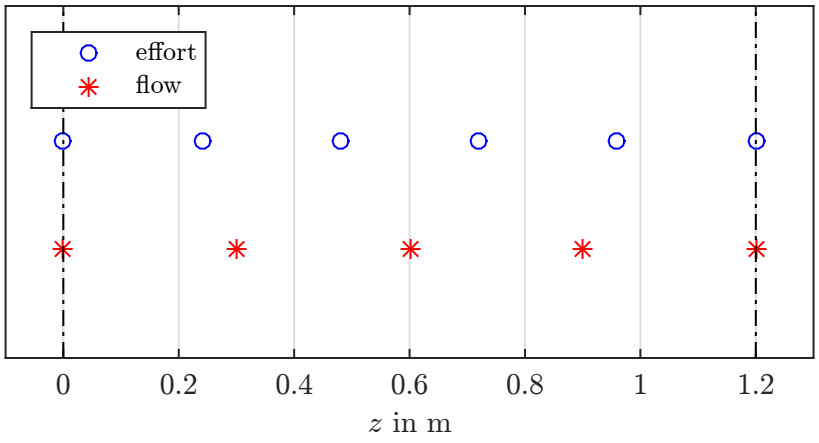
3.4.2 Approximation Quality Analysis of the Discretized Beam Model

In this section, we are discussing some issues about the application of the proposed pseudo-spectral method as well as the approximation quality of the beam model. In order to evaluate the approximation quality of the finite-dimensional model based on the geometric discretization, we consider the system (3.93a), (3.93b) without dissipation

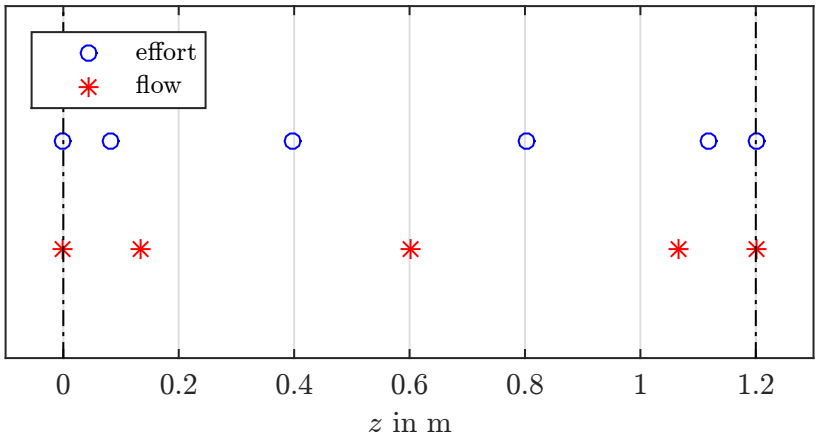
[19] The tilde will be removed in the reduced model, see Section 3.5.

($\tilde{\boldsymbol{R}}_b = \mathbf{0}$). Since we are handling the boundary control problem, two situations are considered: homogeneous and time-varying boundary conditions. In the first case, the numerical approximations of the spectrum and the eigenfunctions of the differential operator are analyzed. The approximation of a transfer function is considered in the second case.

Choice of collocation points. A core ingredient of the geometric PS method is the choice of the approximation space. The choice of approximation bases depends on the problem, see e.g. [89], where the (Bessel) eigenfunctions of the differential operator are a more appropriate choice. As mentioned in Section 3.4, Lagrange interpolating polynomials (3.58), which satisfy the compatibility condition (3.57), are chosen as basis functions in our case. The collocation points, where the approximate solution should match the exact one, can be typically either uniformly distributed or chosen as the zeros of Legendre or Chebyshev polynomials. Since the boundary flows and efforts depend on the boundary conditions, the two endpoints of the interval $[0, L]$ are also taken into account, see [23], [47].



(a) Equidistant collocation points



(b) Gauss-Legendre collocation points

Figure 3.5: Collocation points for efforts and flows with $N = 5$

We take different distributions of collocation points as an example: uniformly distributed and in the zeros of Gauss-Legendre polynomials plus the two endpoints. Figure 3.5 depicts the distribution of the collocation points for efforts and flows with $N = 5$.

Approximation of the eigenfrequencies and eigenfunctions. We compare the eigenfrequencies and eigenfunctions of the discretization with the numerical solutions of the exact eigenvalue problem for the Timoshenko beam as described in [56]. The solution of PDEs (3.37) is parametrized using separation of variables, which yields a fourth-order differential equation in the displacement with frequency-dependent coefficients. Under given boundary conditions, the eigenfrequencies are determined as nontrivial solutions of the characteristic equation. The derivation of the numerical solution is detailed in Appendix C.2.

Let us consider the clamped-free boundary conditions ($\theta_h = 0$, free tip) in terms of the boundary port variables:

$$\begin{aligned} \text{Clamped at } z = 0: \qquad \dot{w}(0) &= 0 \\ \dot{\psi}(0) &= 0 \\ \text{Free at } z = L: \qquad Q_s(L) &= 0 \\ M_b(L) &= 0 \quad . \end{aligned} \tag{3.96}$$

The numerical solutions of the eigenvalue problem for the uniform Timoshenko beam are computed using computer algebra software like *Maple*. Table 3.1 compares the natural frequencies of the discretized beam model, which is based on the material parameters in Table 8.3, for different collocation points with the exact values. The method is conservative, i.e., no numerical dissipation occurs in the approximate models. It is remarkable that the approximate eigenfrequencies do not depend on the choice of collocation points, as proven in [60]. Note that the first five eigenfrequencies – which are dominant for control design – are very well approximated even with a relatively small number of collocation points.

Exact frequency	Equidistant Points			Gauss-Legendre Points		
	$N = 5$	$N = 9$	$N = 12$	$N = 5$	$N = 9$	$N = 12$
2.9343	2.9343	2.9343	2.9343	2.9343	2.9343	2.9343
18.3873	18.3992	18.3873	18.3873	18.3992	18.3873	18.3873
51.4781	53.5262	51.4782	51.4781	53.5262	51.4782	51.4781
100.8570	148.5115	100.8914	100.8570	148.5115	100.8914	100.8570
166.6820	858.2190	168.2584	166.6859	858.2190	168.2584	166.6859

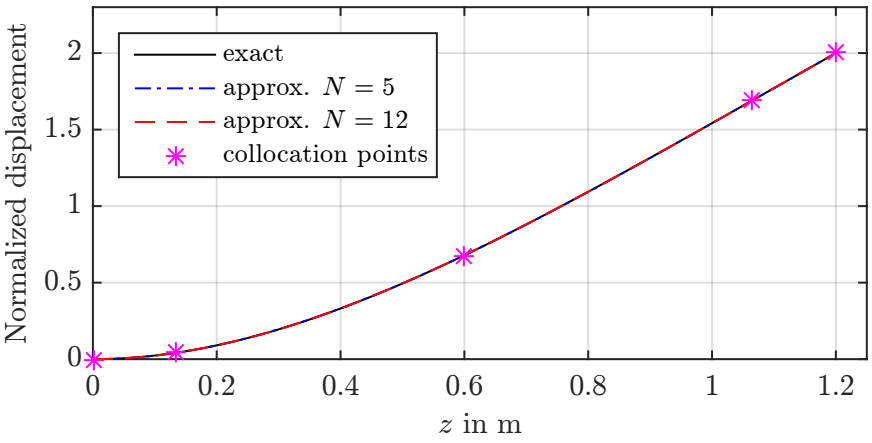
Table 3.1: First five natural frequencies of the discretized model for the uniform beam in Hz

Contrary to the eigenfrequencies, the distribution of collocation points has a strong impact on the approximation of eigenfunctions. In order to avoid the occurrence of numerical oscillations at the boundaries of the interval $[0, L]$ with increasing number N of collocation points (*Runge's phenomenon*, see the examples in Appendix D), we choose the interior Gauss-Legendre collocation points together with the two endpoints in this work.

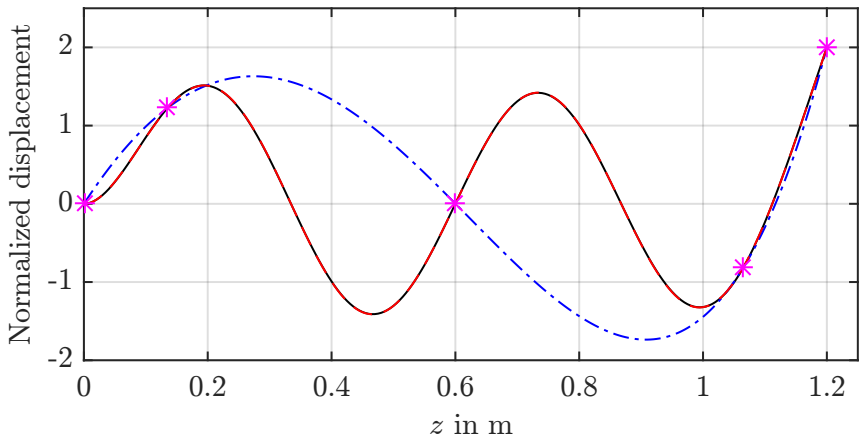
Now, we compare the eigenfunctions based on the single fourth-order displacement differential equation as derived in Appendix C.2 with the numerical approximations by the geometric PS method. Under homogeneous boundary conditions, it is easy to obtain the exact eigenfunctions (or eigenmodes) for the corresponding eigenfrequencies. The approximate eigenfunctions can be represented as the weighted sum of corresponding flow/state basis functions

$$\hat{X}_i(z) = \sum_{k=0}^{N-1} X_i(z_k) w_k^f(z) \tag{3.97}$$

with $X_i(z_k)$ the exact values of the i-th eigenfunction in the collocation points z_k (see also [89]). Figure 3.6 represents the approximations of the first (a) and fifth (b) eigenfunctions, compared to the exact eigenfunctions. The first eigenfunction is perfectly approximated with $N = 5$, while for the fifth one, $N = 12$ is large enough to achieve "perfect" matching.



(a) First eigenfunction



(b) Fifth eigenfunction

Figure 3.6: Exact and approximate eigenfunctions for the first and fifth mode

Approximation of the transfer function. Neglecting the drive system, we consider the forced rotation of the beam around the joint at $z = 0$, which means that one of the boundary conditions (3.96) becomes time-varying: $\dot{\psi}(0) = \dot{\theta}_h$. The exact transfer function between $\dot{\psi}(0)$ and $\dot{\psi}(L)$ is evaluated numerically, see Appendix C.3. In Figure 3.7, the exact amplitude-frequency response (black solid line) is compared with the transfer functions of the discretized models with $N = 5$ (blue dash-dotted line) and $N = 12$ (red dashed line). The latter approximation perfectly fits the exact frequency response in the considered range.

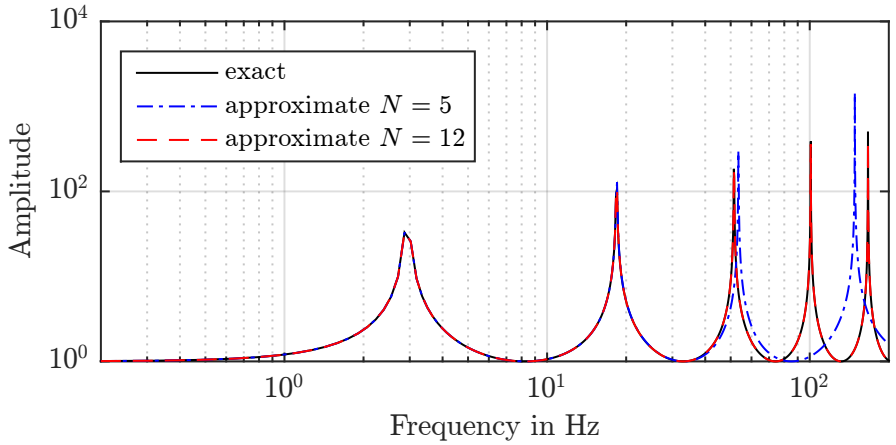


Figure 3.7: Exact and approximate amplitude-frequency response from $\dot{\psi}(0)$ to $\dot{\psi}(L)$

To sum up this section, the geometric discretization approach using Lagrange interpolating polynomials gives an accurate approximation of eigenfrequencies, eigenfunctions, and the transfer behavior, even for small N.

3.5 Structure-Preserving Model Order Reduction

In order to obtain an accurate finite-dimensional model, spatial discretization can lead to large-scale systems with a high number of differential equations. Since the performance and the CPU memory of a real-time control system are limited to a certain extent, in reality, it can be difficult to use the resulting large-scale system for control design and implementation. For this reason, the spatially discretized PH model must be reduced in such a way that the reduced model not only has a low order and displays the dominant dynamics but also maintains the PH structure and related system properties, such as passivity and energy conservation in the lossless case. In this sense, a so-called structure-preserving model order reduction can be used.

In recent years, several well-established algorithms for the linear PH model order reduction have been proposed, see e.g. [27], [30], [66], [100]. We first give a brief review of the basic concept and present the reduced model structure in this section. In the end, we compare the reduced beam models based on the *Krylov subspace method* and the *modal truncation* (see e.g. [19]) in the frequency domain.

3.5.1 Krylov Subspace Methods

Considering the conservation of the inherent PH structure, a structure-preserving model reduction using Krylov subspace method is used to reduce the spatially discretized beam model derived in Section 3.4, see [100]. This approach is usually utilized for large and very large-scale linear PH systems. The corresponding transfer function of a linear system is well approximated with moment matching.

According to [30], we perform model reduction for the linear PH beam model (3.93a), (3.93b) using the projection matrix $\boldsymbol{V} \in \mathbb{R}^{n\times r}$, $n = 4N$. The columns of $\boldsymbol{V}$ span the subspace, in which the state vector $\tilde{\boldsymbol{x}}_b \in \mathbb{R}^n$ is approximated by reduced state vector $\boldsymbol{x}_b \in \mathbb{R}^r$, $\tilde{\boldsymbol{x}}_b \approx \boldsymbol{V}\boldsymbol{x}_b$, and are constructed around a set of appropriately chosen shift points. Taking into account different bases of the state and co-state variables, we multiply Equations (3.93a), (3.93b) with the transpose of a matrix $\boldsymbol{W} \in \mathbb{R}^{n\times r}$, $\boldsymbol{W} = \tilde{\boldsymbol{Q}}_b\boldsymbol{V}$. It becomes

$$\begin{aligned}\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V}\dot{\boldsymbol{x}}_b &= \boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b(\tilde{\boldsymbol{J}}_b - \tilde{\boldsymbol{R}}_b)\tilde{\boldsymbol{Q}}_b\boldsymbol{V}\boldsymbol{x}_b + \boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{B}}_b\boldsymbol{u}_b\\ \boldsymbol{y}_b &= \tilde{\boldsymbol{B}}_b^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V}\boldsymbol{x}_b + \tilde{\boldsymbol{D}}_b\boldsymbol{u}_b.\end{aligned} \tag{3.98}$$

By definition of $\boldsymbol{Q}_b = \boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V}$ and pre-multiplication with $(\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V})^{-1}$, the reduced PH beam model can be formulated as follows:

$$\begin{aligned}\dot{\boldsymbol{x}}_b &= (\boldsymbol{J}_b - \boldsymbol{R}_b)\boldsymbol{Q}_b\boldsymbol{x}_b + \boldsymbol{B}_b\boldsymbol{u}_b\\ \boldsymbol{y}_b &= \boldsymbol{B}_b^T\boldsymbol{Q}_b\boldsymbol{x}_b + \boldsymbol{D}_b\boldsymbol{u}_b,\end{aligned} \tag{3.99}$$

with the reduced system matrices $\boldsymbol{J}_b$, $\boldsymbol{R}_b$, $\boldsymbol{Q}_b \in \mathbb{R}^{r\times r}$, and $\boldsymbol{B}_b \in \mathbb{R}^{r\times m}$

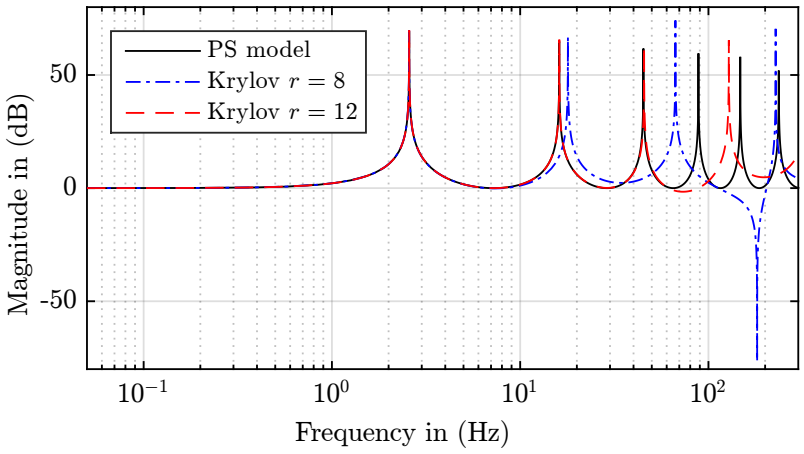
$$\begin{aligned}\boldsymbol{J}_b &= (\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V})^{-1}\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{J}}_b\tilde{\boldsymbol{Q}}_b\boldsymbol{V}(\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V})^{-1}\\ \boldsymbol{R}_b &= (\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V})^{-1}\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{R}}_b\tilde{\boldsymbol{Q}}_b\boldsymbol{V}(\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V})^{-1}\\ \boldsymbol{Q}_b &= \boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V}\\ \boldsymbol{B}_b &= (\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\boldsymbol{V})^{-1}\boldsymbol{V}^T\tilde{\boldsymbol{Q}}_b\tilde{\boldsymbol{B}}_b\\ \boldsymbol{D}_b &= \tilde{\boldsymbol{D}}_b.\end{aligned}$$

3.5.2 Approximation Quality Analysis of the Reduced Beam Model

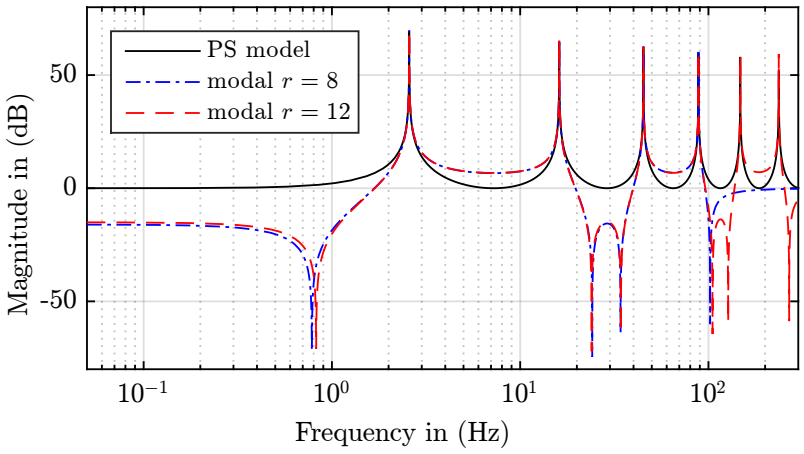
In this section, we discuss the approximation quality of reduced beam models. As an original model to be reduced, we choose a spatially discretized PH beam model based on the PS method with $N = 9$ collocation points, which leads to the system order of $4N = 36$. Based on the Krylov subspace method and modal truncation, we generate two reduced beam models with system orders of $r = 8$ and $r = 12$, respectively, see the corresponding parameters in Table 8.4. These two models are directly generated using the MATLAB Toolbox *sssMOR*[20], which was developed by the Chair of Automatic Control at TU Munich. The transfer function between the angular velocities of the hub (as input) and the beam tip (as output) is taken into account.

[20]See details under
https://www.epc.ed.tum.de/rt/forschung/modellordnungsreduktion/software/sssmor/.

Figure 3.8 shows the frequency responses of reduced models with the two methods. It is obvious to recognize that the Krylov-based model reduction provides a better approximation than the modal truncation. Comparing the blue dash-dotted line for the system order of $r = 8$ and the red dashed line for $r = 12$ in Figure 3.8(a), the approximation is getting more accurate in higher frequencies if the number of chosen shift points increases. Moreover, the shift points can be chosen around the first eigenfrequencies so that the approximate frequency response matches the original one much better. On the contrary, the frequency response of the reduced model based on modal truncation differs from the original one even in the lower frequencies, although this approach provides exact eigenfrequencies. In order to approximate the frequency response and the transient behavior of the beam model as accurately as possible, the Krylov-based reduced model with $r = 12$ is a suitable choice for our case.



(a)



(b)

Figure 3.8: Bode diagram of the reduced model based on (a) the Krylov subspace method and (b) the modal truncation with different system dimensions

Figure 3.9 compares a Krylov-based reduced model (red dashed line) with a PS discretized model (blue dash-dotted line), both of which have a system order of 12. The reference model (black solid line) is a PS model with $N = 9$ collocation points. The reduced model is based on this reference model, while a small PS discretized model is

generated by choosing $N = 3$ collocation points. Note that (not only in our case) the approximation quality of higher modes in a numerical model can be improved if fine discretization is followed by appropriate model reduction compared to a direct coarser discretization that leads to the same system order, see e.g., [101].

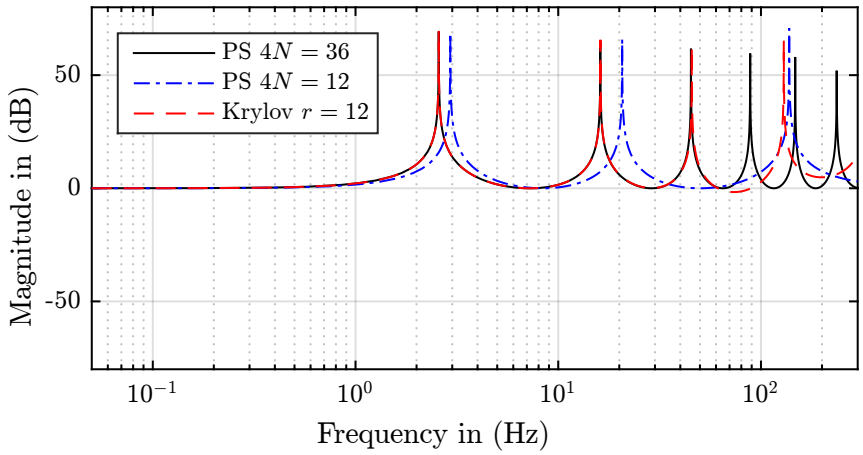


Figure 3.9: Comparison of the PS beam model with $4N = 12$ and the reduced Krylov-based beam model with $r = 12$

3.6 Coupling of the Subsystems

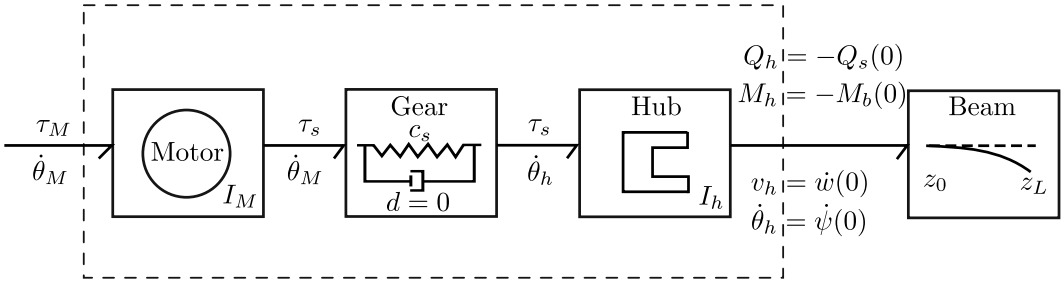


Figure 3.10: Symbolic representation of the interconnected system: The half arrows indicate the direction of the power flow.

The flexible beam is attached to the rigid hub. The point $z = 0$ along the beam corresponds to the radius of the hub r_h. The (reaction) shear force and the bending torque are localized here. The power-preserving interconnection conditions, which are visualized in Figure 3.10, are

$$\begin{aligned} \boldsymbol{u}_{a_2} &= \begin{bmatrix} -Q_h \\ -M_h \end{bmatrix} = \begin{bmatrix} Q_s(0) \\ M_b(0) \end{bmatrix} = -\boldsymbol{y}_b^0 \\ \boldsymbol{u}_b^0 &= \begin{bmatrix} \dot{w}(0) \\ \dot{\psi}(0) \end{bmatrix} = \begin{bmatrix} v_h \\ \dot{\theta}_h \end{bmatrix} \quad = \quad \boldsymbol{y}_{a_2}. \end{aligned} \tag{3.100}$$

Inserting the corresponding parts of (3.35) and (3.99) into (3.100), we obtain

$$\begin{aligned} \boldsymbol{u}_{a_2} &= -\left(\boldsymbol{B}_{b_0}^T \nabla H_b + \boldsymbol{u}_b^L\right) \\ \boldsymbol{u}_b^0 &= \quad \boldsymbol{B}_{a_2}^T \nabla H_a. \end{aligned} \tag{3.101}$$

Consequently, the dynamics[21] of the drive system (3.35) and the beam model (3.99) become

$$\begin{aligned} \dot{\boldsymbol{x}}_a &= \boldsymbol{J}_a \nabla H_a - \boldsymbol{B}_{a_2} \boldsymbol{B}_{b_0}^T \nabla H_a + \boldsymbol{B}_{a_1} u_{a_1} - \boldsymbol{B}_{a_2} \boldsymbol{u}_b^L \\ \dot{\boldsymbol{x}}_b &= \boldsymbol{B}_{b_0} \boldsymbol{B}_{a_2}^T \nabla H_a + \boldsymbol{J}_b \nabla H_a + \boldsymbol{B}_{b_L} \boldsymbol{u}_b^L. \end{aligned} \tag{3.102}$$

Defining the global state vector $\boldsymbol{x} = [\boldsymbol{x}_a^T,\ \boldsymbol{x}_b^T]^T \in \mathbb{R}^{r+4}$ and the global energy $H(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}_a^T \boldsymbol{Q}_a \boldsymbol{x}_a + \frac{1}{2}\boldsymbol{x}_b^T \boldsymbol{Q}_b \boldsymbol{x}_b$, results in the PH representation of the overall system[22]

$$\begin{aligned} \dot{\boldsymbol{x}} &= \begin{bmatrix} \boldsymbol{J}_a & -\boldsymbol{B}_{a_2} \boldsymbol{B}_{b_0}^T \\ \boldsymbol{B}_{b_0} \boldsymbol{B}_{a_2}^T & \boldsymbol{J}_b - \boldsymbol{R}_b \end{bmatrix} \nabla H(\boldsymbol{x}) + \begin{bmatrix} \boldsymbol{B}_{a_1} & -\boldsymbol{B}_{a_2} \\ \boldsymbol{0} & \boldsymbol{B}_{b_L} \end{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{B}_{a_1}^T & \boldsymbol{0} \\ -\boldsymbol{B}_{a_2}^T & \boldsymbol{B}_{b_L}^T \end{bmatrix} \nabla H(\boldsymbol{x}) \end{aligned} \tag{3.103}$$

with the global inputs and their collocated, power-conjugated outputs

$$\begin{aligned} \boldsymbol{u} &= \begin{bmatrix} u_{a_1} \\ \boldsymbol{u}_b^L \end{bmatrix} = \begin{bmatrix} \tau_M \\ \hdashline Q_s(L) \\ M_b(L) \end{bmatrix} \\ \boldsymbol{y} &= \begin{bmatrix} y_{a_1} \\ \boldsymbol{y}_b^L \end{bmatrix} = \begin{bmatrix} \dot{\theta}_M \\ \hdashline \dot{w}(L) \\ \dot{\psi}(L) \end{bmatrix}. \end{aligned} \tag{3.104}$$

It is noticed that the entire system (3.103) is again a PH system. Without a tip mass, $\boldsymbol{u}_b^L = [Q_s(L),\ M_b(L)]^T = \boldsymbol{0}$, and the interconnected model of the flexible setup is driven only by the motor torque. In what follows, we understand $u(t) = u_{a_1}(t) = \tau_M(t) \in \mathbb{R}$ as the scalar input and $\boldsymbol{y}(t) \in \mathbb{R}^3$ as the vector of output variables[23]. The configuration variable $\psi(t, L)$, or its time derivative $y_c(t) = y_{b,2}(t) = \dot{\psi}(t, L)$ is considered the controlled variable, whose smooth and calm transitions is to be realized by control.

[21] Compared to the linearized hub and beam model (A.27) derived in Appendix A, the drive system takes also into account the motor dynamics in addition to the hub. Moreover, the shear force between the hub and the beam is considered in the interconnection conditions, see Equation (3.100), and appears as the external force in the hub dynamics (A.27c).

[22] The same can be done with the unreduced beam model in "tilde" coordinates.

[23] In the following chapters, $\boldsymbol{y}(t)$ denotes the measured output vector, which includes the motor angle and angular velocity as well as the angular velocity of the beam tip.

Chapter 4

Model-Based Feedforward Control

Robots are generally used in order to do some of the human work that could be dangerous and difficult for human beings to do. Their applications are e.g. in space, underwater, in production, and in cooperation with humans. The advantages of robots lie in the saving of money and time. Most robot tasks involve positioning or tracking predefined movement.

Lightweight robots are becoming increasingly popular due to their large payload/mass ratio. In addition, these robot types can be used to realize highly dynamic movements. However, their structural flexibility leads to undesired elastic deformations during movement. In order to achieve precise motion and improve the control quality, the control strategy always takes into account the so-called model-based feedforward control. Here, it refers to the pre-calculated motor torque, which is based on the plant model to relieve the feedback control. The feedback controller is used to compensate for the model error, changing parameters, and unknown disturbances. Depending on the accuracy of the model used, the tracking error can be reduced to a certain degree. In particular, the oscillation of the flexible arm during a fast movement can be suppressed if the structural elasticity is taken into account during feedforward control.

In this chapter, we apply an inversion-based feedforward control to compute the desired input and state trajectories for the desired output trajectory, in particular for the flexible beam. To generate a suitable and feasible trajectory, a sinusoidal interpolation is first introduced. For comparison, we also consider a feedforward control based on the rigid body model, which neglects structural elasticity.

4.1 Trajectory Planning

The goal of trajectory planning is to prescribe a predefined movement to the manipulator, which is then executed by the industrial control system, see [73]. The desired movement, which describes the sequential configuration of robots between a start position and an end position of the joint or end effector, is called trajectory, see e.g. [99]. It is usually defined by the user with a set of parameters. The transition should be characterized by some performance criteria, constraints, and boundary conditions for the actuators, e.g.

to keep the saturation limits or to avoid excitation of the unmodeled resonant modes of the structure.

PTP and CP movement. Trajectory planning can be carried out either in joint space or operational (Cartesian) space, which can be defined as a point-to-point (PTP) movement and the continuous path (CP), respectively. In the PTP motion, the robot moves from an initial to a final joint coordinate in a given time. The robot leaves a curved trajectory in the Cartesian coordinate system and ends at the predefined position. Taking into account the speed and acceleration limitations, it could reach the end position by the fastest route. Since the steering action takes place directly in the joint space, the inverse kinematics is not necessary for this trajectory planning. For this reason, the movement of the end effector with respect to the nonlinear direct kinematics could not be easily predicted, see [73]. In contrast, CP planning generates a chronological sequence of variables to describe the position and the orientation of the point in the operational space, taking into account the constraints imposed. To obtain the corresponding joint coordinates for the robot control, the CP must be transformed into the joint space using inverse kinematics, see [24]. The unsuitable specification of the velocity in the Cartesian coordinate system could lead to singularity during the transformation. In addition, the limitation of angular velocity and acceleration can be exceeded. Therefore, the original trajectory should be iteratively adjusted until it meets the requirements in the joint space, see [99].

Since we consider only a single-link robot arm, PTP trajectory planning is suitable for our application. Furthermore, inverse kinematics can be also avoided. This movement is normally divided into three phases, i.e., acceleration, uniform movement and braking. First of all, the time profile of the acceleration for each axis should be determined, taking into account the maximum angular velocity and acceleration. The velocity profile is designed in such a way that the speed is initially zero, then increases to a maximum value and maintains this value for a time, and finally drops back to zero. Furthermore, the joint can be gently accelerated and braked by limiting the jerk. In this way, it reduces the stress on mechanics and machine wear.

Parameter	Description
s_e	Length of the total distance
$\dot{q}_m$	Maximum velocity
$\ddot{q}_m$	Maximum acceleration
t_a	Time at the end of the acceleration
t_e	Time required for the total distance
t_b	Time at the beginning of braking

Table 4.1: Definition of the trajectory parameters for the sinusoidal profile

Interpolation with a sinusoidal profile. In order to obtain a smooth trajectory, a PTP motion profile with sinusoidal interpolation is constructed, which forces a sinusoidal acceleration profile in the starting and braking phase, according to [73]. The movement begins at the starting point $q = q_s$ at $t = t_s$ ($t_s = 0$) and ends at the endpoint $q = q_e$ at $t = t_e$. $s_e = |q_e - q_s|$ indicates the total distance. To determine the trajectory, two

intermediate points between the starting point and the endpoint are added, namely lift-off at $t = t_a$ and set-down at $t = t_b$, where $t_b = t_e - t_a$. Table 4.1 lists the relevant trajectory parameters. $\dot{q}_m$ and $\ddot{q}_m$ indicate the maximum velocity and acceleration, respectively.

Figure 4.1 shows the PTP trajectory interpolated by a sinusoidal profile in terms of position, velocity, and acceleration. Due to the equal duration of acceleration and deacceleration, the symmetrical property for the velocity profile in relation to the time results in $\bar{t} = t_e/2$. Then, the average point satisfies $\bar{q} = (q_s + q_e)/2$, see [73]. The initial and final velocities are zero. Note that at the end of the acceleration phase $t = t_a$, the velocity must reach its maximum value $\dot{q}_m$ and remain constant for a time $t = t_b - t_a$. According to [21], the acceleration time t_a can be determined by the permissible maximum velocity $\dot{q}_m$ and acceleration $\ddot{q}_m$

$$t_a = 2\frac{\dot{q}_m}{\ddot{q}_m}, \tag{4.1}$$

which satisfy the following relationship

$$\dot{q}_m \stackrel{!}{=} \frac{1}{2}\ddot{q}_m t_a. \tag{4.2}$$

The minimum time t_e for the total distance s_e can be calculated by

$$t_e^{min} = \frac{s_e}{\dot{q}_m} + t_a. \tag{4.3}$$

If the specified total time t_e is greater than its minimum value t_e^{min}, we maintain the maximum acceleration $\ddot{q}_m$ and reduce only the maximum velocity $\dot{q}_m$ as

$$\dot{q}_m = \frac{1}{2}\left(\frac{\ddot{q}_m t_e}{2} - \sqrt{\frac{\ddot{q}_m^2 t_e^2}{4} - 2\ddot{q}_m s_e}\right). \tag{4.4}$$

As a result, the acceleration time t_a is reduced due to the new $\dot{q}_m$ and must be recalculated according to Equation (4.1). The following equations generate the trajectories with the sinusoidal profile in terms of position, velocity, and acceleration, which are shown in Figure 4.1.

- Acceleration: $0 \leq t < t_a$:

$$\begin{aligned} \ddot{q}(t) &= \ddot{q}_m \sin^2\left(\frac{\pi}{t_a}t\right) \\ \dot{q}(t) &= \ddot{q}_m\left(\frac{1}{2}t - \frac{t_a}{4\pi}\sin\left(\frac{2\pi}{t_a}t\right)\right) \\ q(t) &= \ddot{q}_m\left(\frac{1}{4}t^2 + \frac{t_b^2}{8\pi^2}\left(\cos\left(\frac{2\pi}{t_a}t\right) - 1\right)\right) \end{aligned} \tag{4.5}$$

- Uniform movement: $t_a \leq t < t_b$:

$$\begin{aligned} \ddot{q}(t) &= 0 \\ \dot{q}(t) &= \dot{q}_m \\ q(t) &= \dot{q}_m(t - \frac{1}{2}t_a) \end{aligned} \tag{4.6}$$

- Braking: $t_b \leq t < t_e$:

$$\begin{aligned}
\ddot{q}(t) &= -\ddot{q}_m \sin^2\left(\frac{\pi}{t_e - t_b} t\right) \\
\dot{q}(t) &= \dot{q}_m - \ddot{q}_m \left(\frac{1}{2}(t - t_b) - \frac{t_a}{4\pi} \sin\left(\frac{2\pi}{t_a}(t - t_b)\right)\right) \\
q(t) &= \frac{1}{2}\ddot{q}_m \left(t_e(t + t_a) - \frac{1}{2}(t^2 + t_e^2 + 2t_a^2) + \frac{t_a^2}{4\pi^2}\left(1 - \cos\left(\frac{2\pi}{t_a}(t - t_b)\right)\right)\right)
\end{aligned} \tag{4.7}$$

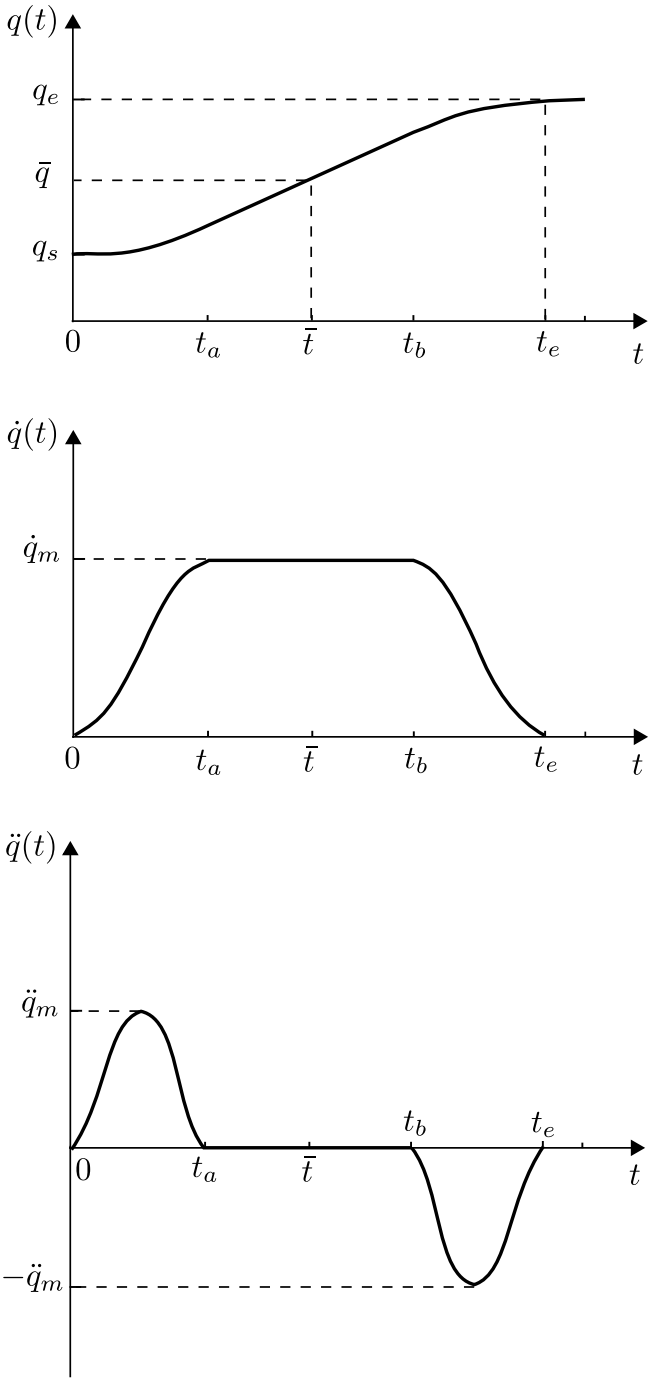


Figure 4.1: Position, velocity and acceleration profile of a PTP motion with the sinusoidal interpolation [73]

4.2 Rigid-Body-Based Feedforward Control

When controlling industrial robots, feedforward control based on a rigid body model is often used to calculate the reference motor torque for the desired movement. For comparison, we assume that the robot arm is rigid and only the gear elasticity is present in the drive system.

The total energy is composed of the kinematic energies of the gear motor and the rigid link[1] as well as the potential energy of the spring

$$H_{rb} = \frac{1}{2}\left(\frac{p_M^2(t)}{I_M} + \frac{p_l^2(t)}{I_l} + c_s(\theta_M(t) - \theta_l(t))\right), \tag{4.8}$$

where $\theta_l(t)$ is the angle of the rigid link.

We define a state vector $\boldsymbol{x}_{rb}(t) \in \mathbb{R}^4$

$$\boldsymbol{x}_{rb}(t) = \begin{bmatrix} \theta_M(t) \\ \theta_l(t) \\ p_M(t) \\ p_l(t) \end{bmatrix}, \tag{4.9}$$

which contains the motor angle, the link angle, and their corresponding angular momentum, respectively. Then, the co-energy vector $\boldsymbol{e}_{rb}(t) \in \mathbb{R}^4$ are defined by the gradient of the Hamiltonian

$$\boldsymbol{e}_{rb}(t) = \nabla H_{rb}(\boldsymbol{x}_{rb}(t)) = \boldsymbol{Q}_{rb}\boldsymbol{x}_{rb}(t) \tag{4.10}$$

with

$$\boldsymbol{Q}_{rb} = \begin{bmatrix} c_s & -c_s & 0 & 0 \\ -c_s & c_s & 0 & 0 \\ 0 & 0 & \frac{1}{I_M} & 0 \\ 0 & 0 & 0 & \frac{1}{I_l} \end{bmatrix}. \tag{4.11}$$

They represent the spring (reaction) forces and the rotational speeds of the motor and the link. With the above definitions, we obtain the state space representation in the standard PH form

$$\begin{aligned} \dot{\boldsymbol{x}}_{rb}(t) &= \boldsymbol{J}_{rb}\boldsymbol{Q}_{rb}\boldsymbol{x}_{rb}(t) + \boldsymbol{B}_{rb}u(t) \\ y(t) &= \boldsymbol{B}_{rb}^T\boldsymbol{Q}_{rb}\boldsymbol{x}_{rb}(t), \end{aligned} \tag{4.12}$$

with the skew-symmetric interconnection matrix $\boldsymbol{J}_{rb}$ and the input vector $\boldsymbol{B}_{rb}$

$$\boldsymbol{J}_{rb} = \begin{bmatrix} \boldsymbol{0}_{2\times 2} & \boldsymbol{I}_2 \\ -\boldsymbol{I}_2 & \boldsymbol{0}_{2\times 2} \end{bmatrix}, \qquad \boldsymbol{B}_{rb} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}.$$

[1] The hub and beam are combined to one component with the substituted moment of inertia for the link $I_l = I_h + I_b$.

The control input is again the motor torque $u(t) = \tau_M(t)$, while the motor angular velocity is its collocated, power-conjugated output $y(t) = \dot{\theta}_M(t)$.

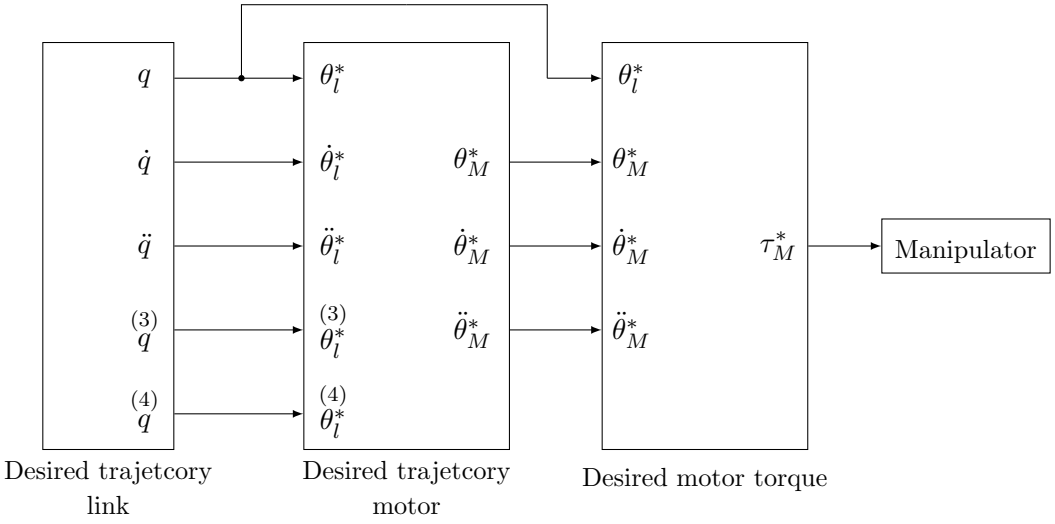


Figure 4.2: Sketch of the feedforward control based on the rigid body model

Figure 4.2 shows the design sketch of the feedforward control based on the rigid robot model. The desired trajectory for the link is generated by a sinusoidal profile, which is shown in Figure 4.1 with

$$\theta_l^*(t) = q(t), \quad \dot{\theta}_l^*(t) = \dot{q}(t), \quad \ddot{\theta}_l^*(t) = \ddot{q}(t), \quad \overset{(3)}{\theta}_l^*(t) = \overset{(3)}{q}(t), \quad \overset{(4)}{\theta}_l^*(t) = \overset{(4)}{q}(t). \tag{4.13}$$

According to Equation (4.12), we obtain the desired motor angle $\theta_M^*(t)$ in relation to the desired link angle $\theta_l^*(t)$ and angular acceleration $\ddot{\theta}_l^*(t)$

$$\theta_M^*(t) = \frac{1}{c_s} I_l \ddot{\theta}_l^*(t) + \theta_l^*(t). \tag{4.14}$$

To achieve the desired trajectory, the reference motor torque $\tau_M^*(t)$ is calculated by the motor dynamics[2]

$$\tau_M^*(t) = I_M \ddot{\theta}_M^*(t) + c_s(\theta_M^*(t) - \theta_l^*(t)) \tag{4.15}$$

with the time derivatives of the desired motor angle

$$\dot{\theta}_M^*(t) = \frac{1}{c_s} I_M \overset{(3)}{\theta}_l^*(t) + \dot{\theta}_l^*(t) \tag{4.16}$$

$$\ddot{\theta}_M^*(t) = \frac{1}{c_s} I_l \overset{(4)}{\theta}_l^*(t) + \ddot{\theta}_l^*(t). \tag{4.17}$$

Note that the time derivatives are required up to the fourth order of the link angle. It follows that the predefined trajectory must be differentiable at least up to the fourth order. Therefore, we choose the sinusoidal profile presented in Section 4.1 to implement the trajectory.

[2]Here, we use undamped motor dynamics for the design of feedforward control. The damping or friction term is explicitly described with a friction model which is used for friction compensation in the control law, see Section 5.3.

4.3 Inversion-Based Feedforward Control

To calculate the required control input to achieve exact output tracking, an inversion-based feedforward control (ibFFC) can be investigated, see e.g. [11]. The required input of the forward model is calculated by solving the system inverse for a predefined output trajectory. It is distinguished whether the considered system is a minimum-phase or non-minimum-phase system (for details, see Section 4.3.1), since the internal dynamics is numerically integrated to determine the required input trajectory, see [29]. If the internal dynamics is unstable, i.e., the system is non-minimum-phase, so its solution is not bounded. From a control-theoretic point of view, the stable inversion-based approaches are required to compute the bounded state and input trajectories for these non-minimum-phase systems for a given desired output trajectory.

In the following sections, we give a short overview of the problem of zero dynamics, see [80]. According to [5], a stable inversion-based feedforward control approach for the finite-dimensional beam model is presented.

4.3.1 Zero Dynamics

The zero dynamics describes a system internal dynamics for the case that the output $y(t)$ for a certain input $u(t)$ is zero, and initial conditions $\boldsymbol{x}(0)$ for all times $t \geq t_0$ (see the definition in [80]):

$$y(t) = \dot{y}(t) = \ddot{y}(t) = \cdots = \overset{(n)}{y}(t) = 0, \tag{4.18}$$

where n is the system order.

Now, we consider a linear, time-invariant SISO system in state space representation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{b}u(t) \tag{4.19a}$$

$$y(t) = \boldsymbol{c}^T\boldsymbol{x}(t). \tag{4.19b}$$

The output (4.19b) is derived in terms of time

$$\begin{aligned} \dot{y}(t) &= \boldsymbol{c}^T\dot{\boldsymbol{x}}(t) \\ &= \boldsymbol{c}^T\boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{c}^T\boldsymbol{b}u(t). \end{aligned} \tag{4.20}$$

If $\boldsymbol{c}^T\boldsymbol{b} = 0$, the second derivative of the output results from Equation (4.20)

$$\begin{aligned} \ddot{y}(t) &= \boldsymbol{c}^T\boldsymbol{A}\dot{\boldsymbol{x}}(t) \\ &= \boldsymbol{c}^T\boldsymbol{A}^2\boldsymbol{x}(t) + \boldsymbol{c}^T\boldsymbol{A}\boldsymbol{b}u(t). \end{aligned} \tag{4.21}$$

If $\boldsymbol{c}^T\boldsymbol{A}\boldsymbol{b} = 0$, the output should be further derived until the input $u(t)$ comes out in the derivation of the output, where $\boldsymbol{c}^T\boldsymbol{A}^i\boldsymbol{b} \neq 0$. The lowest order of its derivation is called relative degree δ.

In summary, the derivatives of the output can be formulated as follows:

$$\overset{(i)}{y}(t) = \boldsymbol{c}^T\boldsymbol{A}^i\boldsymbol{x}(t) \quad \text{for} \quad i = 1, \ldots, \delta - 1 \tag{4.22a}$$

$$\overset{(\delta)}{y}(t) = \boldsymbol{c}^T\boldsymbol{A}^\delta\boldsymbol{x}(t) + \boldsymbol{c}^T\boldsymbol{A}^{\delta-1}\boldsymbol{b}u(t). \tag{4.22b}$$

According to Equations (4.22b) and (4.18), we obtain the input, which makes the output zero

$$u(t) = -\frac{\boldsymbol{c}^T \boldsymbol{A}^{\delta}}{\boldsymbol{c}^T \boldsymbol{A}^{\delta-1} \boldsymbol{b}} \boldsymbol{x}(t). \tag{4.23}$$

In the linear SISO case, the zero dynamics also characterizes the zeros (roots of the uncancelled numerator polynomials $Z(s)$) of the corresponding transfer function $G(s)$ of the system (4.19a), (4.19b)

$$\begin{aligned} G(s) &= \boldsymbol{c}^T (s\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{b} \\ &= \frac{\boldsymbol{c}^T \mathrm{adj}(s\boldsymbol{I} - \boldsymbol{A}) \boldsymbol{b}}{\det(s\boldsymbol{I} - \boldsymbol{A})} \\ &= \frac{Z(s)}{N(s)} \\ &= K \frac{\beta_0 + \beta_1 s + \cdots + \beta_{m-1} s^{m-1} + s^m}{\alpha_0 + \alpha_1 s + \cdots + \alpha_{n-1} s^{n-1} + s^n} \\ &= K \frac{\prod_{i=1}^{m} (s - N_i)}{\prod_{j=1}^{n} (s - P_j)}, \end{aligned} \tag{4.24}$$

where $\mathrm{adj}(\boldsymbol{X})$ and $\det(\boldsymbol{X})$ are the adjugate and determinant of matrix $\boldsymbol{X}$, respectively. The numerator $Z(s)$ has the degree m, while the denominator degree is $N(s)$ is n, where $m \leq n$. The difference in degree between denominator and numerator corresponds to the relative degree $\delta = n - m$.

As mentioned in [80], the zeros have a decisive influence both on the behavior of open-loop systems and on the achievable control performance of closed-loop systems in the linear case, see [16]. Depending on the location of the zeros of $G(s)$, the system is minimum-phase or non-minimum-phase. If at least one of the zeros lies on the right side of the imaginary axis, the system is called non-minimum-phase with unstable inverse dynamics. In contrast, the system is minimum-phase when the zero dynamics is asymptotically stable, see [35]. It corresponds to the case that all zeros of $G(s)$ are on the left side.

Example 4.1: To better understand the behavior of non-minimum-phase systems, we take the transfer function $G_1(s)$ and $G_2(s)$ as an example

$$G_1(s) = \frac{s+2}{(s+1)(s+3)} \tag{4.25}$$

$$G_2(s) = \frac{-s+2}{(s+1)(s+3)}, \tag{4.26}$$

which have two equal stable poles $s = -1$, $s = -3$. The difference is that $G_1(s)$ has a stable zero $s = -2$, which is on the left side of the imaginary axis, while that of $G_2(s)$ is on the right side with $s = 2$. Thus, the system $G_1(s)$ is minimum-phase and the system $G_2(s)$ is non-minimum-phase.

Figure 4.3 shows the step responses of both systems. It is obvious to see that the non-minimum-phase system $G_2(s)$ has more delay than $G_1(s)$. Initially, $G_2(s)$ moves in the opposite direction and then tends to the same final value as $G_1(s)$. The transfer function $G_2(s)$ can be divided into a minimum-phase part $G_1(s)$ and the correction part $G_3(s)$

$$G_2(s) = \frac{-s+2}{(s+1)(s+3)} = \frac{s+2}{(s+1)(s+3)} - \frac{2s}{(s+1)(s+3)} = G_1(s) + G_3(s). \quad (4.27)$$

$G_3(s)$ also has a strongly decreasing stage at the beginning of the response and slower convergence to the final value 0.

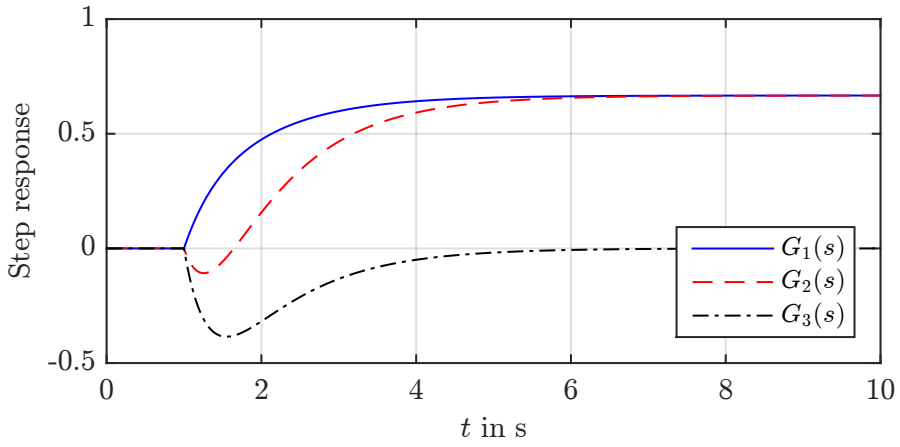


Figure 4.3: Step responses of systems $G_1(s)$ and $G_2(s)$ as well as $G_3(s)$

4.3.2 Bounded Solution of the Inverse Beam Dynamics

We consider a fast PTP movement of the beam tip. For this purpose, the required control input $u^*(t) = \tau_M^*(t)$ is computed from a desired trajectory $y_c^*(t) = \dot{\psi}^*(t, L)$ using the inversion of the system dynamics. However, the inverse dynamics of the elastic robot arm, which is a non-minimum-phase system[3], is unstable and non-causal. In order to find a bounded input trajectory despite the instability, we apply a stable inversion approach according to [5], [107] as shown in [97]. In what follows, we sketch the inversion of the beam model. The resulting interconnection port variables at the motor hub can then be used to determine the necessary motor torque by inversion of the drive model (3.35), see Section 4.3.3.

Inverse beam model. According to the definition of beam inputs and outputs (3.91), we specify a desired trajectory for the total angle $\psi^*(t, L)$ from which the desired angular velocity $\dot{\psi}^*(t, L) = y_{b,4}^*(t)$ follows. The corresponding control input for executing the motion task is the hub angular velocity $\dot{\psi}^*(t, 0) = u_{b,2}^*(t)$, see Figure 4.4.

To solve this control problem, an inverse model needs to be set up. It is obtained from the forward model by solving the output equation (3.93b) for the control input and replacing the result in the state differential equation (3.93a). The full rank of the

[3]The instability of the internal or zero dynamics is frequently referred to as non-minimum phase property, see Section 4.3.1.

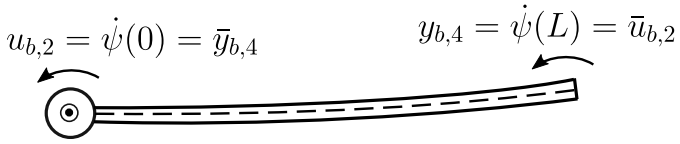


Figure 4.4: Inputs and output in the forward and the inverse beam model

feedthrough matrix $\boldsymbol{D}_b$, which stems from the rigid body motion, allows this partial model inversion.

The inverse beam model reads

$$\dot{\boldsymbol{x}}_b = \underbrace{\bar{\boldsymbol{F}}_b \boldsymbol{Q}_b}_{\bar{\boldsymbol{A}}_b} \boldsymbol{x}_b + \bar{\boldsymbol{B}}_b \bar{\boldsymbol{u}}_b \tag{4.28a}$$

$$\bar{\boldsymbol{y}}_b = \underbrace{\bar{\boldsymbol{B}}_b^T \boldsymbol{Q}_b}_{\bar{\boldsymbol{C}}_b} \boldsymbol{x}_b + \bar{\boldsymbol{D}}_b \bar{\boldsymbol{u}}_b \tag{4.28b}$$

with

$$\begin{aligned} \bar{\boldsymbol{u}}_b &= \begin{bmatrix} u_{b,1} \\ y_{b,4} \\ u_{b,3} \\ u_{b,4} \end{bmatrix} = \begin{bmatrix} \dot{w}(0) \\ \dot{\psi}(L) \\ Q_s(L) \\ M_b(L) \end{bmatrix} \\ \bar{\boldsymbol{y}}_b &= \begin{bmatrix} y_{b,1} \\ y_{b,2} \\ y_{b,3} \\ u_{b,2} \end{bmatrix} = \begin{bmatrix} -Q_s(0) \\ -M_b(0) \\ \dot{w}(L) \\ \dot{\psi}(0) \end{bmatrix}. \end{aligned} \tag{4.29}$$

The properties of the matrices $\bar{\boldsymbol{F}}_b$ and accordingly $\bar{\boldsymbol{A}}_b$ were analyzed numerically. The results are comparable to the inverse models of hyperbolic systems of two conservation laws discretized with the FE method according to [28] as reported in [40]: In the undamped case, the eigenvalues of $\bar{\boldsymbol{A}}_b$ lie on the imaginary axis. For $r_t, r_r > 0$, the inverse (or internal) dynamics is hyperbolic with half of the eigenvalues in the left and the right open half plane.

Bounded solution of the inverse model. In our application, the inverse model is supplied with a single non-zero input

$$\bar{\boldsymbol{u}}_b(t) = \begin{bmatrix} 0 & \dot{\psi}^*(t, L) & 0 & 0 \end{bmatrix}^T, \tag{4.30}$$

and the output that determines the hub angular velocity is

$$\bar{y}_{b,4}(t) = \bar{\boldsymbol{C}}_{b,4}^T \boldsymbol{x}_b(t) + \bar{D}_{b,42} \bar{u}_{b,2}(t) \tag{4.31}$$

with $\bar{\boldsymbol{C}}_{b,4}^T$ and $\bar{D}_{b,42}$ the corresponding row of $\bar{\boldsymbol{C}}_b$ and element of $\bar{\boldsymbol{D}}_b$.

If we could solve the ODE (4.28a) for $\boldsymbol{x}_b(t)$, the solution $\bar{y}_{b,4}(t)$ can be derived directly from the trajectory $\bar{\boldsymbol{u}}_b(t)$ described above. Now, the following questions must be explained:

- Does the solution of Equation (4.28a) at the end of the time correspond to the state on the desired trajectory?
- Can the ideal tracking behavior be guaranteed by the serially connecting of the feedforward part, which is based on an inverse model, to the plant model?
- Will it also prove itself by multiplying the transfer function?

To answer these questions, we take a SISO system as an example (see the mathematical derivation in Appendix E). The system is considered in the Laplace domain, and its inverse model can be easily derived. It is noticeable that the state variables of the inverse model (E.10) have the same physical meaning as those of the forward model (E.1). To achieve the desired output of the forward model, the required control input is derived, which is the output of the inverse model. The ideal tracking behavior can actually be realized by feedforward control based on the inverse model.

However, the inverse model – as in most cases, the internal dynamics for flexible beams when a TCP trajectory is imposed – is unstable. In order to obtain a bounded input trajectory for the desired output, a stable approach of the ibFFC is applied, see [107] for linear systems. This approach is based on determining the relative degree and decoupling the stable and unstable internal dynamics. The bounded solution of the internal dynamics is calculated using bounded state transition matrices.

For the considered case with viscous damping r_t, $r_t > 0$ and therefore hyperbolic inverse dynamics, the bounded solution results as follows. The inverse model (4.28) can be transformed into the Jordan normal form

$$\dot{\boldsymbol{\sigma}}(t) = \boldsymbol{\Lambda}\boldsymbol{\sigma}(t) + \boldsymbol{B}_\sigma \bar{\boldsymbol{u}}_b(t), \qquad \boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_r\}, \tag{4.32}$$

with $\boldsymbol{B}_\sigma = \boldsymbol{V}_b^{-1}\bar{\boldsymbol{B}}_b$, $\boldsymbol{V}_b \in \mathbb{R}^{r\times r}$ the matrix of right eigenvectors of $\bar{\boldsymbol{A}}_b$ and $\boldsymbol{\sigma} = \boldsymbol{V}_b^{-1}\boldsymbol{x}_b$ the vector of modal coordinates. Now, the dynamical system can be split into a stable and an unstable part

$$\dot{\boldsymbol{\sigma}}_s(t) = \boldsymbol{\Lambda}_s\boldsymbol{\sigma}_s(t) + \boldsymbol{B}_{\sigma,s}\bar{\boldsymbol{u}}_b(t) \tag{4.33a}$$

$$\dot{\boldsymbol{\sigma}}_u(t) = \boldsymbol{\Lambda}_u\boldsymbol{\sigma}_u(t) + \boldsymbol{B}_{\sigma,u}\bar{\boldsymbol{u}}_b(t), \tag{4.33b}$$

with $\boldsymbol{\Lambda}_s \in \mathbb{R}^{r_s\times r_s}$ and $\boldsymbol{\Lambda}_u \in \mathbb{R}^{(r-r_s)\times(r-r_s)}$, the Jordan matrices of the stable and unstable eigenvalues, respectively. The bounded solution of (4.32) for the given input is obtained by

$$\boldsymbol{\sigma}_s(t) = \int_{-\infty}^{t} e^{\boldsymbol{\Lambda}_s(t-\tau)}\boldsymbol{B}_{\sigma,s}\bar{\boldsymbol{u}}_b(\tau)\,d\tau \tag{4.34a}$$

$$\boldsymbol{\sigma}_u(t) = -\int_{t}^{\infty} e^{-\boldsymbol{\Lambda}_u(\tau-t)}\boldsymbol{B}_{\sigma,u}\bar{\boldsymbol{u}}_b(\tau)\,d\tau. \tag{4.34b}$$

The stable subsystem (4.34a) is integrated in forward time, whereas the unstable subsystem (4.34b) is integrated in reverse time. The bounded solution, which connects two

stationary points on the time interval $[t_{\text{start}},\, t_{\text{end}}]$, is achieved by integrating both subsystems on a sufficiently large time interval $I \supset [t_{\text{start}},\, t_{\text{end}}]$, which allows phases of pre- and post-actuation, see e.g. Figure 8.6. The bounded solution of the unstable part (4.34b) is derived extensively in Appendix F. The corresponding simulation/experimental results are presented in Section 8.4.

4.3.3 Calculation of the Bounded Motor Control Torque

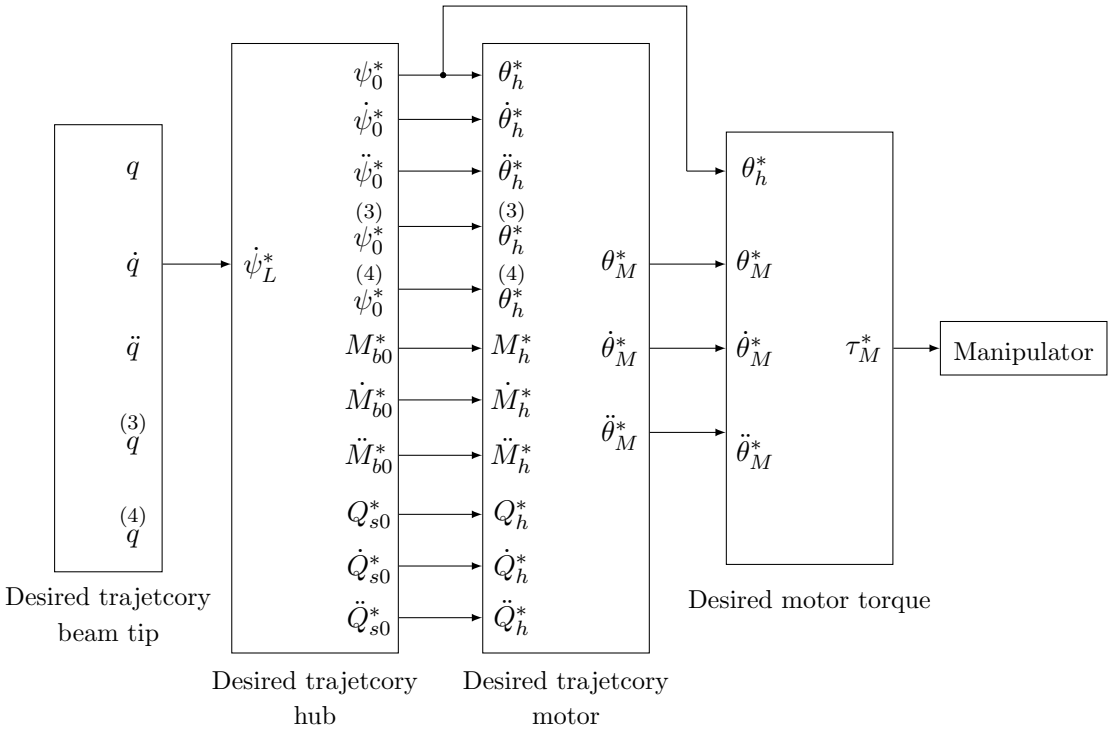


Figure 4.5: Sketch of the feedforward control based on the flexible body in relation to Figure 3.10 (Notation: Rotation of the beam's cross section $\psi_0 = \psi(t,0)$, $\psi_L = \psi(t,L)$, bending moment $M_{b0} = M_b(t,0)$, and shear force $Q_{s0} = Q_s(t,0)$)

In the section above, the design of the feedforward control is based only on the (reduced) beam model. To calculate the required motor torque, the motor dynamics, and the boundary conditions between the hub and the beam are used. Figure 4.5 shows the corresponding design sketch.

The desired angular velocity of the beam tip is generated by a sinusoidal profile (see Figure 4.1)

$$\dot{\psi}^*(t,L) = \dot{q}(t). \tag{4.35}$$

The desired trajectories of the hub are determined by the following boundary conditions (and their derivatives)

$$\begin{aligned} \theta_h^*(t) &= \quad \psi^*(t,0) \\ M_h^*(t) &= -M_b^*(t,0) \\ Q_h^*(t) &= -Q_s^*(t,0), \end{aligned} \tag{4.36}$$

where $\psi^*(t,0)$, $M_b^*(t,0)$ and $Q_s^*(t,0)$ are computed by the ibFFC. Based on Equation (3.35), we obtain the desired motor angle $\theta_M^*(t)$ as follows

$$\theta_M^*(t) = \frac{1}{c_s}(I_h\,\ddot{\theta}_h^*(t) + M_h^*(t) + r_h\,Q_h^*(t)) + \theta_h^*(t). \tag{4.37}$$

The reference motor torque $\tau_M^*(t)$ is calculated by the motor dynamics

$$\tau_M^*(t) = I_M\,\ddot{\theta}_M^*(t) + c_s(\theta_M^*(t) - \theta_h^*(t)) \tag{4.38}$$

with

$$\dot{\theta}_M^*(t) = \frac{1}{c_s}(I_h\,\overset{(3)}{\theta}_h^*(t) + \dot{M}_h^*(t) + r_h\,\dot{Q}_h^*(t)) + \dot{\theta}_h^*(t) \tag{4.39}$$

$$\ddot{\theta}_M^*(t) = \frac{1}{c_s}(I_h\,\overset{(4)}{\theta}_h^*(t) + \ddot{M}_h^*(t) + r_h\,\ddot{Q}_h^*(t)) + \ddot{\theta}_h^*(t). \tag{4.40}$$

It is recognized that the hub angular velocity, as well as its derivatives up to fourth order, are needed for the motor dynamics. That is also the reason why the given output trajectory should be smooth enough (differentiable at least up to the fourth order).

Chapter 5

PD Feedback Control with Static Friction Compensation

5.1 Two-Degrees-of-Freedom Regulation

The model-based feedforward control requires a high accuracy of the model. In reality, there are not only model errors but also some disturbances like friction, which cannot be modeled or identified exactly. Feedback control is therefore necessary to compensate for these errors. Together with the feedforward control, a two-degrees-of-freedom control structure is established. It does not matter what type of feedforward or feedback control is used, this control structure remains essentially the same.

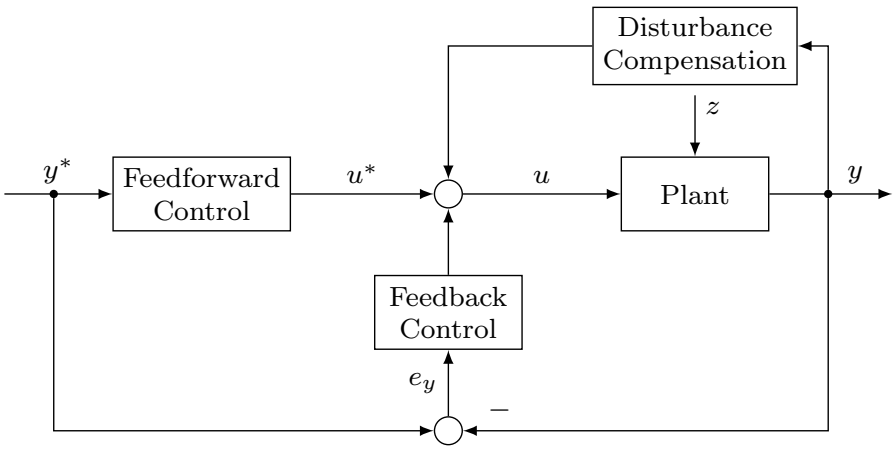


Figure 5.1: Two-degrees-of-freedom control structure, consisting of feedforward control, output feedback, and disturbance compensation

Figure 5.1 shows a two-degrees-of-freedom control structure including the feedforward control, the output feedback, and the disturbance compensation. It is characterized by the fact that it determines the reference input response independently of the disturbance response. In this way, the design of both the feedforward and feedback control can be implemented systematically, taking into account the disturbances. For this reason, this control structure corresponds to our control purpose.

To set the reference input response, the system is initially assumed to be without initial value and external disturbances. The feedforward control is responsible for generating the nominal control input $u^*(t)$ according to the desired trajectory $y^*(t)$. For a known $y^*(t)$, $u^*(t)$ relies only on the system description. Since there are ideally no disturbances, the reference input response (assuming the plant is stable) can only be achieved under the feedforward control. In this case, the control input $u^*(t)$ causes the controlled variable $y(t)$ to follow exactly the desired trajectory $y^*(t)$.

The trajectory $y^*(t)$ must be specified so that the plant is able to follow it as closely as possible while maintaining actuator saturation. Only in this way, the tracking error $e_y(t) \neq 0$ can be avoided. The control input $u(t)$ for achieving a tracking behavior can be calculated offline using an inverse plant model for a given desired trajectory $y^*(t)$. Certainly, the stability of this inverse plant model plays an essential role in the numerical solution of the corresponding differential equations, see the previous chapter. The desired trajectory $y^*(t)$ and the control input $u^*(t)$ are stored in advance and interpolated in real-time by the industrial PLC.

Due to different initial values between the plant and its model, not-measurable disturbances, as well as the model uncertainty, a tracking error $e_y(t) \neq 0$ occurs. To compensate for this, an output feedback is taken into account. This controller only operates if there is a disturbance response. Due to some unknown disturbances, it is not possible to avoid exceeding the saturation of the control input, which can lead to instability of the closed-loop system. If we are able to describe the disturbance, the control scheme can be completed by the disturbance compensation to improve the control effect. In our case, there is unknown friction in the servo motor. On the one hand, it can be described with a static or a dynamic friction model, the parameters of which are identified experimentally, see e.g. [42], [31]. On the other hand, it can be estimated online with a disturbance observer, see e.g. [69].

In this chapter, we focus on the PD feedback with disturbance compensation based on the static friction models .

5.2 PD Feedback Control

The control target of the flexible robot arm is not only to follow a given trajectory but also to suppress the residual oscillation. Ultimately, it is based mainly on the motor motion control. The simplest way is PD (**P**roportional–**D**erivative) feedback control, where only the motor position and speed are used. The biggest advantage of this feedback control is that it is not dependent on the system model. It merely attempts to adjust the control input so that the difference between desired and actual motor position or speed is minimized. This approach is generally applied to industrial robots. Therefore, we first consider PD feedback control and later want to make a comparison with other developed control algorithms.

Figure 5.2 extends the two-degrees-freedom control structure by PD feedback control and friction compensation. The motor torque τ_M consists of the nominal value τ_M^*, the PD feedback part, and the compensated friction torque $\hat{\tau}_f$ as follows

$$\tau_M = \tau_M^* + K_P \cdot (\theta_M^* - \theta_M) + K_D \cdot (\dot{\theta}_M^* - \dot{\theta}_M) + \hat{\tau}_f. \tag{5.1}$$

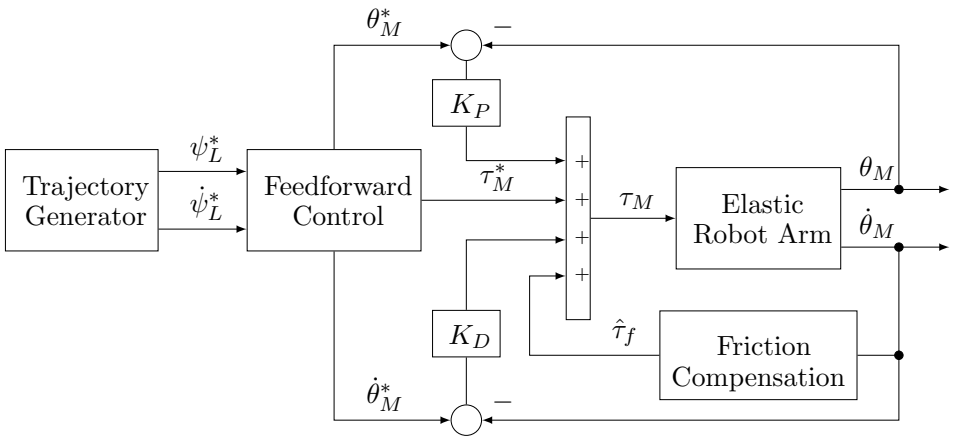


Figure 5.2: Two-degrees-of-freedom regulation structure, consisting of PD feedback control and friction compensation

A trajectory generator produces the desired position ψ_L^* and speed $\dot{\psi}_L^*$ of the beam tip to fulfill a given movement of the manipulator, see Section 4.1. Based on the desired values, the desired trajectory of motor position θ_M^* and speed $\dot{\theta}_M^*$, as well as the nominal motor torque τ_M^*, can be determined by a feedforward control approach, see e.g. Sections 4.2 and 4.3. A PD controller is used to calculate the motor torque required to compensate for the existing tracking error. For this purpose, the actual motor position θ_M and speed $\dot{\theta}_M$ must be fed back to the controller. In general, the proportional gain K_P and the derivative gain K_D are adjusted according to the Ziegler–Nichols method.

During the commissioning of the motor, it is found that friction has a significant influence on the robot dynamics. To compensate for the lack of motor torque, a friction compensation $\hat{\tau}_f$ based on a static friction model is added to the control input. As a result, the tracking error is smaller than without friction compensation, and the corresponding control energy is also reduced. This prevents an abrupt change of the motor torque and the voltage at the beginning of a movement.

5.3 Friction Modeling and Identification

Friction occurs when the bodies in contact move and counteracts their acceleration. In mechatronics, friction is one of the main causes for a deterioration in the achievable control quality of the closed-loop system. It can lead to limit cycle, permanent control error, and stick-slip effect, see [31]. In the light of robotics, the friction in the servo motor attacks the high-precision positioning in case of a near standstill.

In servo motors, friction depends on environmental conditions such as temperature, types of lubrication, and operating conditions, e.g. speed and direction of motion. It can be modeled with the help of a friction characteristic, which describes a discontinuous mapping of the motor speed to the friction torque. Especially from standstill to movement, the friction jumps abruptly due to the stiction effect. In order to reduce the influence of friction on the system dynamics, the control loop system can be supplemented by a model-based static friction compensation.

In this section, some static friction models and the results of friction identification based on the experimental data are presented.

5.3.1 Static Friction Models

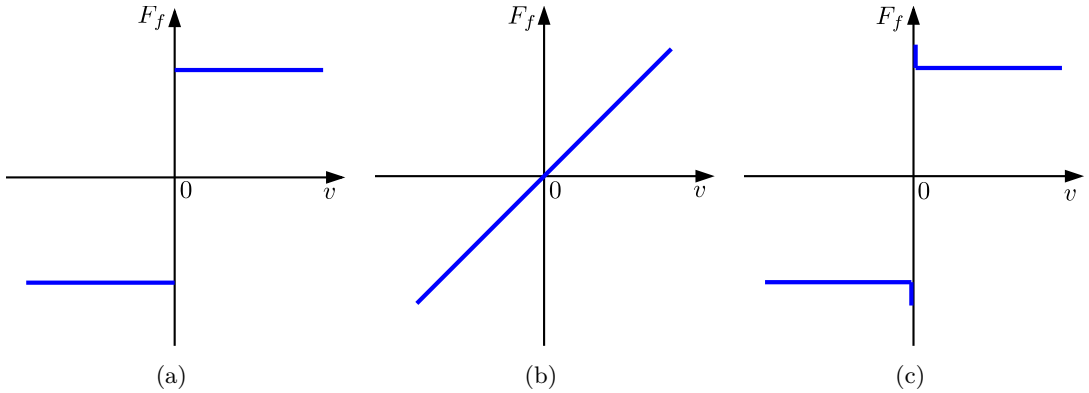


Figure 5.3: Static characteristic of (a) Coulomb friction, (b) viscous friction, and (c) Coulomb friction with stiction based on [44]

Depending on the phenomena that occur, there are various friction models to describe their behavior, which are determined experimentally. Generally, they are a function of speed, external force, or temperature. The classical friction components are divided into Coulomb friction, viscous friction, and stiction friction. Figure 5.3 shows their corresponding properties in terms of speed (for details, see [42], [44]). In most cases, the friction can be described by the combined classical friction models.

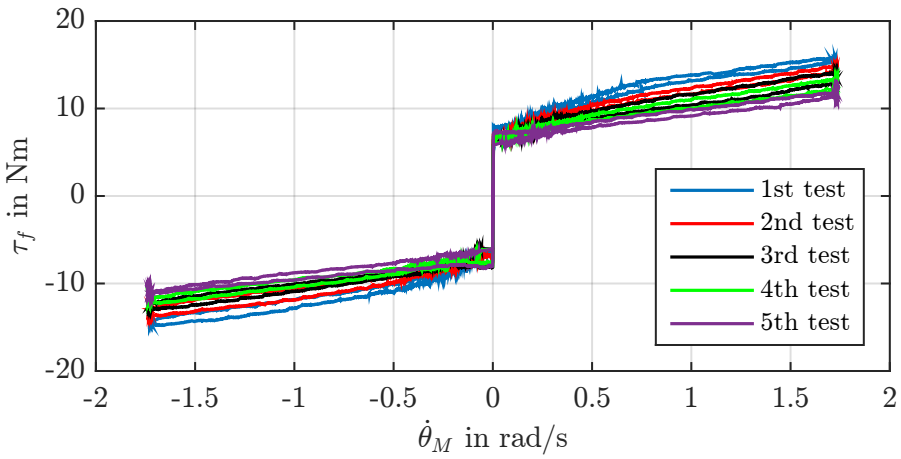


Figure 5.4: Friction torque τ_f as a function of motor speed $\dot{\theta}_M$. The friction torque decreases with increasing temperature.

Figure 5.4 shows the measured friction characteristic of the RD85x13 servo motor for five experimental sets. At the beginning of a movement, it is obvious to recognize the Coulomb friction with the stiction effect. Furthermore, the friction torque[1] τ_f has a component proportional to velocity in the case of $\dot{\theta}_M \neq 0$. During acceleration and deceleration, the hysteresis effect occurs due to the structural damping of the flexspline in the Harmonic Drive gearbox. In other words, the friction torque can have more than

[1] τ_f denotes the measured friction torque, whereas $\hat{\tau}_f$ describes the modeled one.

one value at the same motor speed, depending on the change in direction of the speed. In practice, the hysteresis effect is quite difficult to model and is often neglected in control design.

As researched in [69], the time-varying operating condition, especially the operating temperature, causes additional uncertainty for the modeling of the friction. For example, the viscosity of the lubricating oil decreases almost exponentially with increasing temperature. Although explicit modeling of the friction effect as a function of the operating temperature appears to be extremely advantageous, this strategy is not pursued in the compensation-oriented approaches. On the one hand, the complex thermodynamic laws speak against the easily parameterizable and practical representation of friction. On the other hand, the determination of the operating temperature directly at the contact surfaces of the friction bodies is either associated with a metrological effort, in the case of a state estimation, or with the excessive model uncertainties.

This means that classic friction components can be combined with each other in different ways:

- **Linear Model**

  The most basic friction models consist of Coulomb and viscous friction. The simplest formulation is a linear combination of both friction models, in which the friction torque $\hat{\tau}_f$ is linearly dependent on the motor speed $\dot{\theta}_M$, according to [48]:

  $$\hat{\tau}_f = b_1^l \cdot \dot{\theta}_M + b_0^l \cdot \mathrm{sgn}(\dot{\theta}_M). \tag{5.2}$$

  b_1^l is the increase in viscous friction, while b_0^l represents the Coulomb friction term.

- **Asymmetric Model**

  Taking into account different friction characteristics in the positive and negative direction, the linear friction model can be extended to an asymmetric model, see e.g. [81]. As a result, the friction torque is described by the Coulomb and viscous friction, with velocity-direction dependent coefficient as follows

  $$\begin{aligned}\hat{\tau}_f = & b_1^n \cdot \min(\mathrm{sgn}(-\dot{\theta}_M) + 1, 1) \cdot \dot{\theta}_M + b_1^p \cdot \min(\mathrm{sgn}(\dot{\theta}_M) + 1, 1) \cdot \dot{\theta}_M \\ & + b_0^n \cdot \min(\mathrm{sgn}(-\dot{\theta}_M) + 1, 1) \cdot \mathrm{sgn}(\dot{\theta}_M) \\ & + b_0^p \cdot \min(\mathrm{sgn}(\dot{\theta}_M) + 1, 1) \cdot \mathrm{sgn}(\dot{\theta}_M).\end{aligned} \tag{5.3}$$

  The difference between linear and asymmetric friction models is shown in Figure 5.5(a).

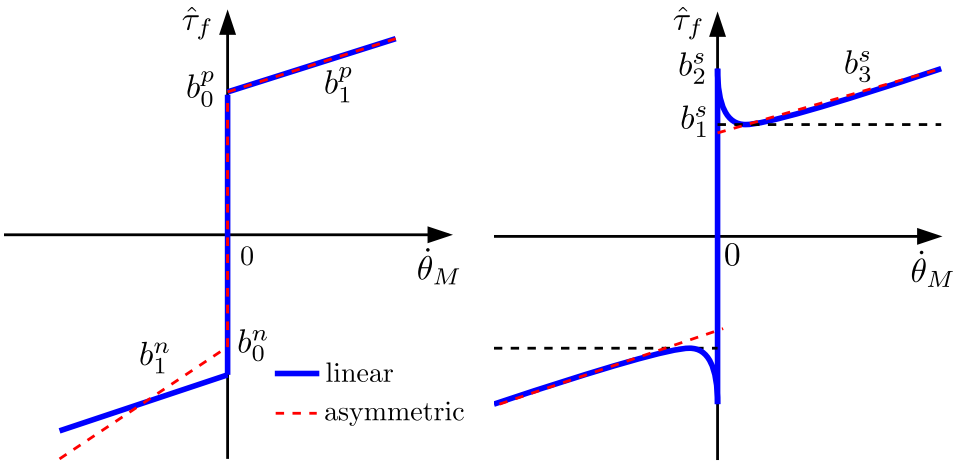
- **Stribeck Model**

  For most situations, it is observed that friction continuously decreases with increasing speed in a low-speed range. This effect is the so-called Stribeck friction. The common mathematical form for the Stribeck effect related to the viscous term is described by

  $$\hat{\tau}_f = (b_1^s + (b_2^s - b_1^s) \cdot \exp(-\left|\frac{\dot{\theta}_M}{\dot{\theta}_s}\right|^{\delta})) \cdot \mathrm{sgn}(\dot{\theta}_M) + b_3^s \cdot \dot{\theta}_M, \tag{5.4}$$

whereby the static characteristic is limited by the stiction friction upwards and by the constant Coulomb friction downwards. The parameter $\dot{\theta}_s$ corresponds to the Stribeck velocity, which determines how fast the Stribeck curve converges to Coulomb friction, while the Stribeck form factor δ specifies the speed range in which the transition between the stiction and sliding friction is made. This form factor δ determines an exponential transition and can be either a positive or negative amount for different surfaces, for details see [69]. According to [42], Stribeck parameters are usually selected as follows: $\dot{\theta}_s = 0.001\text{rad/s}$ and $\delta = 1$ or 2. With increasing speed, viscous friction dominates. Figure 5.5(b) shows an example of a Stribeck curve with $\delta = 1$.

From the linear combination of the Coulomb and viscous friction to an asymmetric model, whose friction behavior is different in positive and negative directions, and the Stribeck model, which includes stiction effect in addition to the Coulomb and viscous friction, the main difficulty of their simulation is the discontinuity at $\dot{\theta}_M = 0$. For this reason, the $\text{sgn}(\dot{\theta}_M)$ function can practically be approximated by a $\tanh(k\dot{\theta}_M)$ function, in which a high slope factor k forces the function to behave like the sgn function. In our case, $k = 60$.



(a) Linear and asymmetric curves (b) Stribeck curve with $\delta = 1$

Figure 5.5: Friction models for identification

5.3.2 Experimental Parameter Identification

As mentioned above, we use a static friction model to compensate for the lack of control torque. Based on a large amount of experimental data, the parameters of the friction models used must first be identified. The aim of this section is to choose a simple and precise friction model for control purposes rather than to present different identification methods. Thus, the design procedure and the experimental results of parameter identification from our previous work described in [58] is summarized in the following paragraphs.

Design procedure. In order to record the friction information accurately, a suitable motor torque must meet the following requirements:

1) First, make sure that the measured torque includes the pure friction torque.
2) Second, the test signal is designed to cover the speed and torque range as widely as possible.

To the first point, we measure the actual motor torque to indirectly calculate the friction, since there is no sensor to measure it directly. Now, we concentrate on the motor dynamics on the driven side

$$I_M\ddot{\theta}_M = \tau_M - \tau_f - \underbrace{\tau_L}_{=0}, \tag{5.5}$$

where $\ddot{\theta}_M$ represents the motor angular acceleration, and the motor torque τ_M is transformed by τ_m on the motor side according to Equation (3.30). The motor characteristics should only be indicated by the motor speed and torque. So, in the experiments, the motor is driven without any load $\tau_L = 0$. In this case, we also neglect the gear elasticity, as there is only a small difference between the speed on the motor and the driven side. Furthermore, the term $\ddot{\theta}_M$ will disappear when the motor speed $\dot{\theta}_M = \text{const}$. Therefore, the friction torque τ_f at a given speed $\dot{\theta}_M$ can be derived from the actual motor torque with the following relationship:

$$\tau_f = \tau_M = n\tau_m, \qquad \dot{\theta}_M = \text{const}. \tag{5.6}$$

The second point is to consider the motor movement in both positive and negative directions. Furthermore, the specified maximum torque must be limited under the rated motor torque τ_m^{rated}. Based on the simplified motor dynamics, the maximum acceleration $\ddot{\theta}_m^{max}$ on the motor side is roughly determined by

$$\begin{aligned} \ddot{\theta}_m^{max} &= \frac{\tau_m^{rated}}{I_m} \\ &= \frac{0.598\text{Nm}}{2.29 \cdot 10^{-4}\text{kgm}^2} = 2611.4\text{rad/s}^2, \end{aligned} \tag{5.7}$$

where the relevant motor parameters are listed in Appendix G.2.

In the Automation Studio application project, the motor states such as position, speed, and acceleration must be converted into the so-called (internal) "Units", where 1000 Units represent 1 degree on the driven side. This allows the maximum speed and acceleration of the motor RD85x13 to be:

$$\begin{aligned} PosMaxVelocity &= \dot{\theta}_m^{max}\,\frac{\text{rad}}{\text{s}} \cdot \frac{180}{\pi} \cdot 1000 \cdot \frac{1}{n} \\ &= 284.84 \cdot \frac{180}{\pi} \cdot 1000 \cdot \frac{1}{160} \\ &= 1.02 \cdot 10^5\,\frac{\text{Units}}{\text{s}} \end{aligned} \tag{5.8}$$

$$\begin{aligned} Acceleration &= \ddot{\theta}_m^{max}\,\frac{\text{rad}}{\text{s}^2} \cdot \frac{180}{\pi} \cdot 1000 \cdot \frac{1}{n} \\ &= 2611.4 \cdot \frac{180}{\pi} \cdot 1000 \cdot \frac{1}{160}\,. \\ &= 1.4962 \cdot 10^6\,\frac{\text{Units}}{\text{s}} \end{aligned} \tag{5.9}$$

In summary, the speed and acceleration limits of the function block *MC_BR_TorqueControl*, as shown in Figure 2.10, are set according to the following table:

Motor	PosMaxVelocity in $\frac{\text{Units}}{\text{s}}$	NegMaxVelocity in $\frac{\text{Units}}{\text{s}}$	Acceleration in $\frac{\text{Units}}{\text{s}^2}$
RD85x13	100000	-100000	900000

Table 5.1: Torque configuration of the RD85x13 motor for friction identification

In order to obtain the friction information for different motor speeds, the given motor torque should be designed so that the speed remains constant for some time after the speed increases. For this purpose, a triangular signal is selected as the motor drive torque τ_M, which contains several piecewise constant torques for a certain period of time, see Figure 5.6. The motor drive torque is limited to 0.1Nm and increases by 0.004Nm with each step. Each commanded torque value remains constant for 4s.

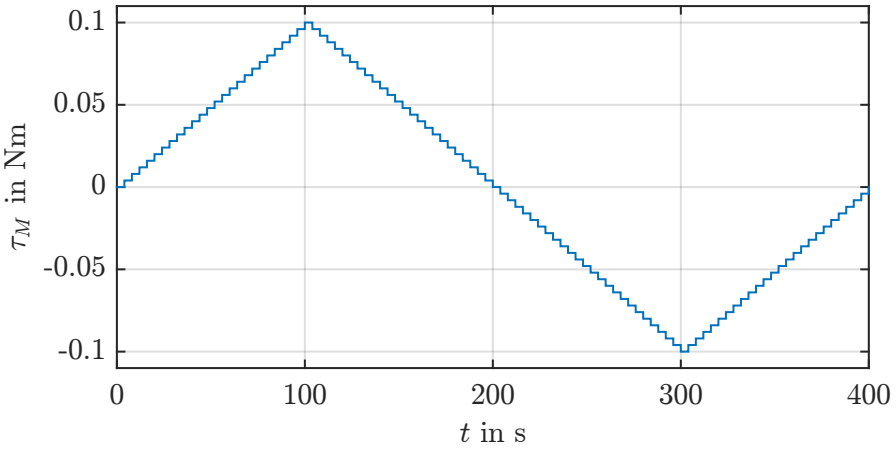


Figure 5.6: Motor drive torque for friction identification based on [58]

Experimental results of friction identification. Figure 5.4 shows the different friction behavior at high and low temperatures. If we compare the motor behavior before and after warming up, the friction is obviously reduced with the increasing operating temperature at the same speed. Since the static friction model is used for the compensation, whose parameters are constant and identified offline, we perform the same friction identification five times after warming up. Based on Equation (5.6), only the measured data for the constant speed will be considered. How to truncate the data is explained in [58].

Using the MATLAB *Curve Fitting Toolbox*, the parameters of the three friction models presented in Section 5.3.1 can be easily identified based on the same measurement data. Figure 5.7 shows the corresponding simulated results. The blue cross represents the measurement data of five experiments. It is observed that three models provide the same behavior when the speed is small, i.e., between -0.5rad/s and 0.5rad/s. The linear model agrees with the Stribeck model, while the asymmetric model differs slightly from the other two. The identified parameters are summarized in Table 5.2. In addition, the fitting quality of the different models can be directly evaluated by some quality indices in the *Curve Fitting Toolbox*. According to [58], R-square is chosen as the evaluating indicator, which describes the ratio between the sum of the squares of the regression

and the total sum of the squares. It allows a value between 0 and 1. Closer to 1 means a larger proportion of the variance due to the model. For the RD85x13 motor, the R-square values of the three models are all around 89% due to the obvious hysteresis phenomenon.

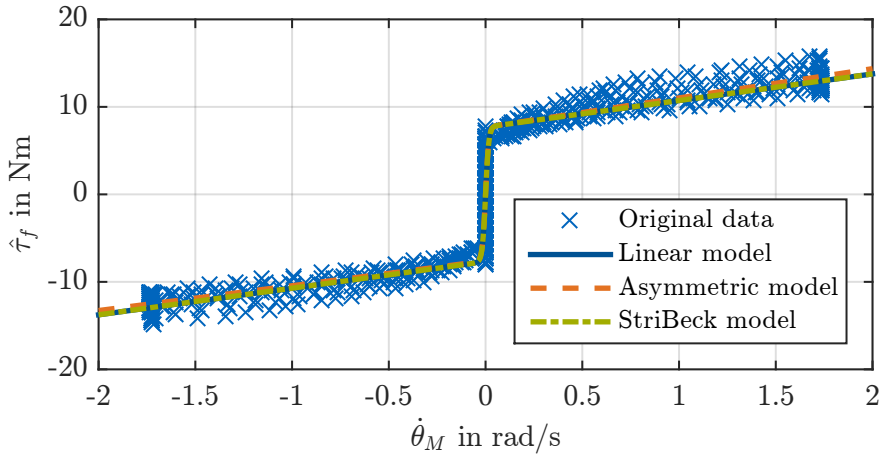


Figure 5.7: Parameter identification of the different motor friction models based on the measurement data shown in Figure 5.4

Model	Parameters	RD85x13	R-Square evaluation
Linear	b_0^l	7.684	0.8905
	b_1^l	3.040	
Asymmetric	b_0^n	7.607	0.8911
	b_0^p	7.684	
	b_1^n	2.835	
	b_1^p	3.341	
Stribeck	b_1^s	7.684	0.8907
	b_2^s	10.66	
	b_3^s	3.041	

Table 5.2: Parameter identification results of the different friction models from [58]

In summary, a simple linear friction model (5.2) can be used for friction compensation, see e.g. [48]. On the one hand, it delivers a feasible result. On the other hand, it is easily implemented and calculated in real time.

Chapter 6

Observer-Based Trajectory Control

Until now we used structure-preserving methods to obtain physically consistent control methods, and we exploited them for motion planning. For closing the control loop by observer-based feedback, in order to achieve smooth transitions and good disturbance behavior, also in the presence of modeling inaccuracy, we use standard approaches from linear state space control. In our case, the LQG regulator, involving both the Kalman filter and the Riccati controller, is applied to the state difference vector $\boldsymbol{e}_x(t) = \boldsymbol{x}^*(t) - \boldsymbol{x}(t)$. This control strategy requires only supplying the desired output trajectories $\boldsymbol{y}^*(t)$ to the feedback controller, instead of the full state reference $\boldsymbol{x}^*(t)$, which saves storage and execution time on the PLC system. Furthermore, the observer is extended with a disturbance model to estimate and compensate the unknown friction.

First of all, we recall the design of the optimal control and the state observer according to [19], [50]. We then introduce the LQG regulator based on the difference model to achieve a PTP movement of the beam tip and to suppress the occurring oscillations.

6.1 Controllability and Observability

The terms controllability and observability were first time introduced by *R. Kalman* in 1960. These two terms play an essential role in most applications, as they represent the necessary (and sufficient) prerequisites for the feasibility of various control algorithms.

We consider a linear, time-invariant (LTI) system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \tag{6.1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t), \tag{6.1b}$$

where the system matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are constant. Disturbances are not taken into account. The output $\boldsymbol{y}(t)$ is normally measured by sensors and can be either a set of state variables or linear combinations thereof. The initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ is generally unknown.

The goal of a feedforward and feedback control is that the control input $\boldsymbol{u}(t)$ ensures the movement of state variables $\boldsymbol{x}(t)$ from the initial state $\boldsymbol{x}_0$ to the desired final state in a finite time. According to [19], the system (6.1) is controllable, if $\boldsymbol{x}(t)$ can be moved

from $\boldsymbol{x}_0$ to $\mathbf{0}$ in a finite time by suitable selection of $\boldsymbol{u}(t)$. This situation is sometimes referred to as completely controllable.

Depending on the measurement of the output $\boldsymbol{y}(t)$, the initial state $\boldsymbol{x}_0$ can be determined from its course over a finite time period. Therefore, a system is observable if $\boldsymbol{x}_0$ can be determined from $\boldsymbol{y}(t)$ over a finite time period with a known control signal $\boldsymbol{u}(t)$, no matter where it is located, see the definition in [19]. It follows from this definition that the initial state $\boldsymbol{x}_0$ is based on the future measurement data for $t > t_0$.

Criterion	Controllability	Observability	System class
Kalman's criterion	$\operatorname{rank}(\boldsymbol{Q}_c) = n$	$\operatorname{rank}(\boldsymbol{Q}_o) = n$	SISO for overall system simple/multiple eigenvalues
Hautus' criterion	$\operatorname{rank}(\begin{bmatrix}\lambda_i \boldsymbol{I} - \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}) = n$	$\operatorname{rank}(\begin{bmatrix}\lambda_i \boldsymbol{I} - \boldsymbol{A} \\ \boldsymbol{C}\end{bmatrix}) = n$	SISO/MIMO for single eigenvalue simple/multiple eigenvalues
Gilbert's criterion	$\hat{\boldsymbol{B}} = \boldsymbol{V}^{-1}\boldsymbol{B}$ $\hat{\boldsymbol{B}}_i^T \neq \mathbf{0}$	$\hat{\boldsymbol{C}} = \boldsymbol{C}\boldsymbol{V}$ $\hat{\boldsymbol{C}}_i \neq \mathbf{0}$	SISO/MIMO for single eigenvalue simple eigenvalues (in Jordan normal form)

Table 6.1: Criteria for the controllability and observability

For linear systems, controllability and observability are investigated first. Depending on the system class, there are different criteria for assessing them, see Table 6.1. The Kalman's criterion is suitable for the overall system, while the other criteria can be applied to assign controllability or observability to single eigenvalues. Compared to the Hautus' criterion, the Gilbert's criterion requires the Jordan normal form of a system and checks whether rows or columns of the input and output matrix are non-zero, see details in [19]. The controllability matrix $\boldsymbol{Q}_c$ and the observability matrix $\boldsymbol{Q}_o$ are defined as follows

$$\boldsymbol{Q}_c = \begin{bmatrix}\boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B}\end{bmatrix}$$

$$\boldsymbol{Q}_o = \begin{bmatrix}\boldsymbol{C} \\ \boldsymbol{C}\boldsymbol{A} \\ \vdots \\ \boldsymbol{C}\boldsymbol{A}^{n-1}\end{bmatrix}.$$

6.2 Optimal Control

6.2.1 Basic Idea

The core task of a state feedback control is to design the dynamic properties of a closed-loop system. On the one hand, it is intended to stabilize an unstable system. On

the other hand, it can assign eigenvalues of a controllable system to arbitrary values. Moreover, the state feedback controller works with a proportional type so that the implementation of the control law only requires simple multiplications and additions. Now, we assume that all state variables of the LTI system (6.1) are known or measured and define the control law for the state feedback as follows

$$\boldsymbol{u}(t) = -\boldsymbol{K}\boldsymbol{x}(t). \tag{6.2}$$

The controller gain $\boldsymbol{K} \in \mathbb{R}^{m \times n}$ must make the closed-loop system asymptotically stable. This means that each state variable tends asymptotically to a constant for $t \to \infty$ against the external disturbance $\boldsymbol{z}(t)$ or the initial disturbance $\boldsymbol{x}_0$. Figure 6.1 shows the block diagram for the state feedback control.

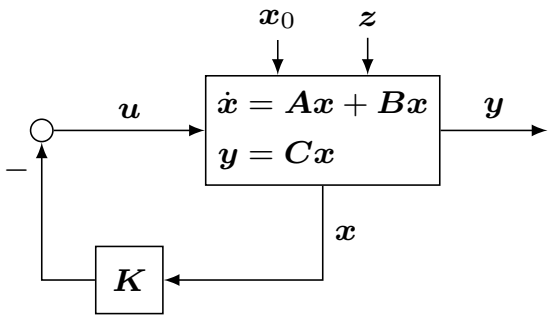


Figure 6.1: State feedback control according to [19]

In the simplest case, the desired closed-loop system can be realized by pole placement, see e.g. [19], [50]. If the system requirements can be formulated as the minimization of a cost functional related to the behaviors of system states and inputs, the controller is considered as the solution of the optimization issue, see [50]. Compared to pole placement, optimal control is suitable for systems with many state variables since it focuses on minimizing the cost functional and not on the placement of all eigenvalues, whose locations must be chosen sensibly. That is also the reason why we chose this controller for our application.

The controller gain $\boldsymbol{K}$ is determined to meet the following two requirements, see [19]:

- The transition from the initial state $\boldsymbol{x}_0$ to the final state $\mathbf{0}$ should not be too slow and should not oscillate too much;
- The control energy required for the transition should be as low as possible.

Consequently, the cost functional can be described in a quadratic form

$$J = \frac{1}{2}\int_0^\infty [\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{S}\boldsymbol{u}(t)]dt, \tag{6.3}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{S} \in \mathbb{R}^{m \times m}$ are the weighting matrices of the state and input vector, respectively. The matrix $\boldsymbol{Q}$ is symmetric and positive semidefinite, while $\boldsymbol{S}$ is a positive definite symmetric matrix.

6.2.2 Riccati Controller

We insert the control law (6.2) into the plant model (6.1) to obtain the closed-loop system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_K \boldsymbol{x}(t) \tag{6.4}$$

with the system matrix

$$\boldsymbol{A}_K = \boldsymbol{A} - \boldsymbol{B}\boldsymbol{K}.$$

The cost functional (6.3) is then changed to

$$J = \frac{1}{2} \int_0^\infty \boldsymbol{x}^T(t)\, \boldsymbol{Q}_K\, \boldsymbol{x}(t) dt \tag{6.5}$$

with a symmetric and positive definite matrix $\boldsymbol{Q}_K \in \mathbb{R}^{n \times n}$

$$\boldsymbol{Q}_K = \boldsymbol{Q} + \boldsymbol{K}^T \boldsymbol{S} \boldsymbol{K}.$$

With the help of the solution of closed-loop system (6.4)

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}_K t} \boldsymbol{x}_0, \qquad \boldsymbol{x}_0 = \boldsymbol{x}(0), \tag{6.6}$$

the cost functional becomes

$$J = \frac{1}{2}\, \boldsymbol{x}_0^T\, \boldsymbol{P}\, \boldsymbol{x}_0\, dt \tag{6.7}$$

with a symmetric and positive definite matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$

$$\boldsymbol{P} = \int_0^\infty e^{\boldsymbol{A}_K^T t} \boldsymbol{Q}_K e^{\boldsymbol{A}_K t} dt. \tag{6.8}$$

$\boldsymbol{K}$ should be parameterized so that the system matrix $\boldsymbol{A}_K$ is stable, whose eigenvalues lie on the left side of the imaginary axis. When using integration by parts for (6.8), it becomes

$$\boldsymbol{P} = \left[e^{\boldsymbol{A}_K^T t} \boldsymbol{Q}_K \boldsymbol{A}_K^{-1} e^{\boldsymbol{A}_K t} \right]_{t=0}^{t=\infty} - \int_0^\infty \boldsymbol{A}_K^T\, e^{\boldsymbol{A}_K^T t} \cdot \boldsymbol{Q}_K \boldsymbol{A}_K^{-1} e^{\boldsymbol{A}_K t} dt. \tag{6.9}$$

As $\boldsymbol{A}_K$ is stable, the e-function tends toward zero for $t \to +\infty$. Therefore, the above equation is simplified as follows

$$\boldsymbol{P} = -\boldsymbol{Q}_K \boldsymbol{A}_K^{-1} - \boldsymbol{A}_K^T \int_0^\infty e^{\boldsymbol{A}_K^T t} \boldsymbol{Q}_K e^{\boldsymbol{A}_K t} dt \cdot \boldsymbol{A}_K^{-1}. \tag{6.10}$$

By right multiplication with $\boldsymbol{A}_K$, we obtain the Lyapunov equation

$$\boldsymbol{A}_K^T \boldsymbol{P} + \boldsymbol{P} \boldsymbol{A}_K = -\boldsymbol{Q}_K, \tag{6.11}$$

which plays an important role in the stability analysis of the closed-loop system. The Lyapunov equation (6.11) has exactly one symmetric, positive definite solution $\boldsymbol{P}$ for

any arbitrary given symmetric, positive definite matrix $\boldsymbol{Q}_K$ if the system matrix $\boldsymbol{A}_K$ is asymptotically stable, see e.g. Proposition 7.1 in [50]. Since the cost functional is associated with the controller gain $\boldsymbol{K}$, its minimum (6.7) can be expressed by the derivative of the cost functional through all elements of $\boldsymbol{K}$. To solve this optimization problem[1], [50] derived the necessary condition for the control law (6.44) as follows

$$\boldsymbol{K} = \boldsymbol{S}^{-1}\boldsymbol{B}^T\boldsymbol{P}, \tag{6.12}$$

where $\boldsymbol{S}^{-1}$ exists and the matrix $\boldsymbol{P} = \boldsymbol{P} > \boldsymbol{0}$ is calculated from the Riccati equation

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0}. \tag{6.13}$$

This type of controller is also called the *Riccati controller*.

In general, we choose the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{S}$ in diagonal form

$$\boldsymbol{Q} = \operatorname{diag}\begin{bmatrix} q_1, & \cdots, & q_n \end{bmatrix}$$
$$\boldsymbol{S} = \operatorname{diag}\begin{bmatrix} s_1, & \cdots, & s_m \end{bmatrix}.$$

q_i, $i = 1, \ldots, n$ is the associated weighting factor of x_i, while s_j, $j = 1, \ldots, m$ is the weighting factor of corresponding control input u_j. There is a MATLAB function *lqr(·)* to determine the Riccati controller, which directly calculates $\boldsymbol{K}$, $\boldsymbol{P}$, as well as the eigenvalues $\boldsymbol{E}_c$ of the closed-loop system:

$$\begin{bmatrix} \boldsymbol{K}, & \boldsymbol{P}, & \boldsymbol{E}_c \end{bmatrix} = \operatorname{lqr}(\ \boldsymbol{A}, \quad \boldsymbol{B}, \quad \boldsymbol{Q}, \quad \boldsymbol{S}\ ). \tag{6.14}$$

6.3 State Observer

To realize the control law (6.2), it was assumed that all state variables $\boldsymbol{x}(t)$ are known or measured. However, this prerequisite is not met in practice. On the one hand, it is expensive and inconvenient to measure all state variables accurately. On the other hand, it is difficult to capture some particular state variables in reality. For these reasons, an observer is used to reconstruct the unknown state variables. Consequently, the control law (6.2) is adapted with the estimated state vector $\hat{\boldsymbol{x}}(t)$ as follows

$$\boldsymbol{u}(t) = -\boldsymbol{K}\hat{\boldsymbol{x}}(t), \tag{6.15}$$

where $\hat{\boldsymbol{x}}(t)$ is determined by the plant model with consideration of both inputs and outputs[2]. [50] states that the behavior of the closed-loop system with the estimated state feedback is possibly the same as the one based on the real state variables. In addition, the closed-loop system must function against the disturbance.

At first, we recall the structure and the principle of a state observer, in particular the Kalman filter. Then, we give an overview of the LQG regulator, which is based on the principle of separation of estimation and control. Finally, the state observer is extended with a disturbance model to further estimate unknown disturbances.

[1] In [19], the solution of this optimization problem is derived using a symmetric matrix $\boldsymbol{W}$, which depends on the initial condition. Although the derivation is slightly different from [50], the results are the same.

[2] Here, the mentioned outputs correspond to the measured output variables.

6.3.1 Observer Structure

An observer is actually a parallel model of the system (6.1). If there is no difference between the system and the observer dynamics, $\hat{\boldsymbol{x}}(t)$ coincides with $\boldsymbol{x}(t)$. However, the system dynamics also depends on the initial condition $\boldsymbol{x}_0$, which is practically difficult to recognize, so it makes $\hat{\boldsymbol{x}}(t)$ different from $\boldsymbol{x}(t)$. For this reason, the observation error is adjusted by the correction term $\boldsymbol{r}(t)$, see Figure 6.2.

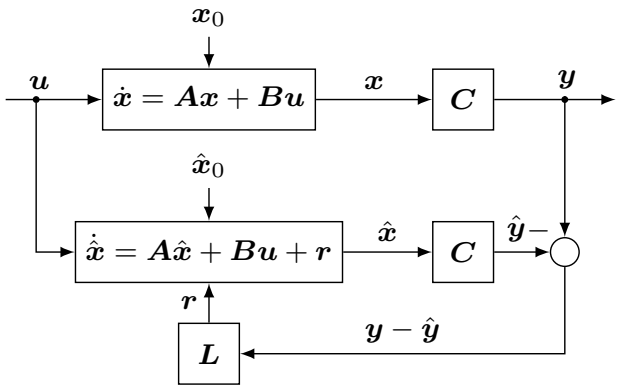


Figure 6.2: Structure of the state observer according to [19]

Based on the above structure, the observer dynamics is described as follows

$$\begin{aligned} \dot{\hat{\boldsymbol{x}}}(t) &= \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{L}(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t)), \qquad \hat{\boldsymbol{x}}(0) = \hat{\boldsymbol{x}}_0 \\ \hat{\boldsymbol{y}}(t) &= \boldsymbol{C}\hat{\boldsymbol{x}}(t) \end{aligned} \tag{6.16}$$

with the correction term $\boldsymbol{r}(t) = \boldsymbol{L}(\boldsymbol{y}(t) - \hat{\boldsymbol{y}}(t))$. It is noteworthy that the observer requires information on both the control input $\boldsymbol{u}(t)$ and the measured output $\boldsymbol{y}(t)$.

The correction matrix $\boldsymbol{L}$ is determined in such a way that the observation error

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \hat{\boldsymbol{x}}(t) \tag{6.17}$$

tends toward zero for $t \to \infty$ with arbitrary initial conditions $\boldsymbol{x}_0$ and $\hat{\boldsymbol{x}}_0$. Using the model (6.1) and (6.16), we obtain the error dynamics

$$\dot{\boldsymbol{e}}(t) = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{e}(t), \qquad \boldsymbol{e}(0) = \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0. \tag{6.18}$$

The core ingredient of the observer design is how to determine the correction matrix $\boldsymbol{L}$. One possibility is to set the eigenvalues of $(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ in the open left complex half-plane, thus ensuring the asymptotic stability of the equilibrium $\boldsymbol{e}^*(t) = \boldsymbol{0}$ of the error dynamics (6.18). This means that the observed state variables approach the real ones. Design of the correction matrix $\boldsymbol{L}$ can be formulated as the control design problem for a dual system:

$$\begin{aligned} \dot{\boldsymbol{x}}_d(t) &= \boldsymbol{A}^T\boldsymbol{x}_d(t) + \boldsymbol{C}^T\boldsymbol{u}_d(t) \\ \boldsymbol{u}_d(t) &= -\boldsymbol{L}^T\boldsymbol{x}_d(t). \end{aligned} \tag{6.19}$$

Table 6.2 shows the correspondence between the matrices for the observer and the control design problem, see [19].

Observer		Controller
$\boldsymbol{A}^T$	$\leftrightarrow$	$\boldsymbol{A}$
$\boldsymbol{C}^T$	$\leftrightarrow$	$\boldsymbol{B}$
$\boldsymbol{L}^T$	$\leftrightarrow$	$\boldsymbol{R}$

Table 6.2: Correspondence between the observer and the control design matrices

6.3.2 Kalman Filter

Depending on the methods of determining the matrix $\boldsymbol{L}$, there is the Luenberger observer and the Kalman filter. Unlike the pole placement by the Luenberger observer, the Kalman filter is based on the optimization of a quadratic cost functional. Both types of observers have the same structure as shown in Figure 6.2 but they are used for different situations. The Luenberger observer focuses on reconstructing state variables only with the initial disturbance $\boldsymbol{x}_0$, while the Kalman filter works well for systems with stochastic disturbances, which falsify the measured outputs during observation. For the latter systems, the observation error can not disappear asymptotically if the Luenberger observer is applied.

For this purpose, an LTI system (6.1) is extended with the process noise $\boldsymbol{n}_x(t)$ and the measured noise $\boldsymbol{n}_y(t)$:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{n}_x(t) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{n}_y(t). \end{aligned} \tag{6.20}$$

$\boldsymbol{n}_x(t)$ and $\boldsymbol{n}_y(t)$ must be white Gaussian noise processes. The white noise process with zero-mean can be described as

$$\begin{aligned} \sigma\{\boldsymbol{n}_x(t), \boldsymbol{n}_x(\tau)\} &= E\{\boldsymbol{n}_x(t)\boldsymbol{n}_x(\tau)^T\} = \boldsymbol{Q}\delta(t-\tau) \\ \sigma\{\boldsymbol{n}_y(t), \boldsymbol{n}_y(\tau)\} &= E\{\boldsymbol{n}_y(t)\boldsymbol{n}_y(\tau)^T\} = \boldsymbol{S}\delta(t-\tau), \end{aligned} \tag{6.21}$$

where the constant and symmetric matrices $\boldsymbol{Q}$ and $\boldsymbol{S}$ are the covariance matrices of $\boldsymbol{n}_x(t)$ and $\boldsymbol{n}_y(t)$, respectively.

The estimation error dynamics (6.16) is then adapted for the stochastically disturbed system (6.20) as follows (see details in(6.18))

$$\dot{\boldsymbol{e}}(t) = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\boldsymbol{e}(t) + \boldsymbol{n}_x(t) - \boldsymbol{L}\boldsymbol{n}_y(t), \qquad \boldsymbol{e}(0) = \boldsymbol{x}_0 - \hat{\boldsymbol{x}}_0. \tag{6.22}$$

Now, we choose a cost functional related to the expected value of the quadratic estimation error

$$J = \sum_{i=1}^{n} E\{e_i^2\} = \sum_{i=1}^{n} \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} e_i^2(t)dt. \tag{6.23}$$

Matrix $\boldsymbol{L}$ should be determined to match the smallest possible cost functional J. Based on the duality of the observer and controller matrices shown in Table 6.2, the observation gain $\boldsymbol{L}$ can be determined by the optimal control approach presented in Section 6.2.2. Therefore, the optimal solution for the Kalman filter is

$$\boldsymbol{L} = \boldsymbol{P}\boldsymbol{C}^T\boldsymbol{S}^{-1} \tag{6.24}$$

with the positive definite matrix $\boldsymbol{P}$ calculated by the Riccati equation

$$\boldsymbol{AP} + \boldsymbol{PA}^T - \boldsymbol{PC}^T\boldsymbol{S}^{-1}\boldsymbol{CP} + \boldsymbol{Q} = \boldsymbol{0}. \tag{6.25}$$

The solution (6.24) is the same as the one solved directly by (6.23). MATLAB function *lqr(·)* can be used to solve this optimal observation problem of the dual system that fits well with our application

$$\left[\ \boldsymbol{L}^T,\ \ \boldsymbol{P},\ \ \boldsymbol{E}_o\ \right] = \mathrm{lqr}(\ \boldsymbol{A}^T,\ \ \boldsymbol{C}^T,\ \ \boldsymbol{Q},\ \ \boldsymbol{S}\ ). \tag{6.26}$$

It is remarkable that the dual approach allows direct weighting of the state variables or the outputs concerned.

6.3.3 LQG Controller

Since an optimal controller, which is combined with a Kalman filter, is applied for the linear system in consideration of the white Gaussian noise, it belongs to the linear-quadratic-Gaussian (LQG) control problem. So far, the design of the controller and the observer are carried out separately. The controller is built on the basis of all known state variables. Actually, the estimated state vector $\hat{\boldsymbol{x}}$ is used for state feedback, see Equation (6.15).

When inserting both the control law (6.15) and the observer dynamics (6.16) into the system (6.1), it becomes

$$\begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\hat{\boldsymbol{x}}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A} & -\boldsymbol{BR}\\ \boldsymbol{LC} & \boldsymbol{A}-\boldsymbol{LC}-\boldsymbol{BR}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \hat{\boldsymbol{x}}\end{bmatrix}. \tag{6.27}$$

Instead of the observer dynamics, the error dynamics (6.18) is considered in the entire state space representation

$$\begin{bmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{e}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}-\boldsymbol{BR} & \boldsymbol{BR}\\ \boldsymbol{0} & \boldsymbol{A}-\boldsymbol{LC}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{e}\end{bmatrix}. \tag{6.28}$$

The set of all eigenvalues of (6.28) can be determined by the following characteristic equation

$$\det\left(s\boldsymbol{I}_n - (\boldsymbol{A}-\boldsymbol{BR})\right)\cdot\det\left(s\boldsymbol{I}_n - (\boldsymbol{A}-\boldsymbol{LC})\right) = 0. \tag{6.29}$$

This equation reveals that the eigenvalues of a closed-loop system without a state observer are not modified by an additive observer. Based on the separation principle, see [50], it is worth mentioning that the controller and the observer can be designed separately for linear systems.

6.3.4 Disturbance Observer

As already mentioned, the disturbances can be described with some suitable models, and the model parameters are identified offline in the experiments, see Section 5.3. In most cases, they are unknown or difficult to identify accurately. For this reason, we are trying to reconstruct them online using the information from the inputs and the measured outputs as well as a disturbance model.

The plant model (6.1) is now extended with an external disturbance input $\boldsymbol{z} \in \mathbb{R}^p$

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{z}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{y} &= \boldsymbol{C}\boldsymbol{x}. \end{aligned} \tag{6.30}$$

To approximate the actual disturbances, a disturbance model will be applied. Although there are different types of disturbance models, they all can be represented by homogeneous differential equations and transformed into a state space representation, see e.g. [19],

$$\begin{aligned} \dot{\boldsymbol{x}}_s &= \boldsymbol{A}_s\boldsymbol{x}_s, \qquad \boldsymbol{x}_s(0) = \boldsymbol{x}_{s_0} \\ \boldsymbol{z} &= \boldsymbol{C}_s\boldsymbol{x}_s, \end{aligned} \tag{6.31}$$

where $\boldsymbol{x}_s \in \mathbb{R}^s$ is the disturbance state vector, and $\boldsymbol{z}$ is considered as the output. This output does not match the actual disturbances, and this model is only used to capture the behavior of the disturbances.

By inserting (6.31) into (6.30), the plant model is expanded as follows

$$\begin{aligned} \begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{x}}_s \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{E}\boldsymbol{C}_s \\ \boldsymbol{0} & \boldsymbol{A}_s \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} \end{aligned} \tag{6.32}$$

with new state vector $\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{x}^T, & \boldsymbol{x}_s^T \end{bmatrix}^T \in \mathbb{R}^{n+s}$.

As for the classical observer, the control input $\boldsymbol{u}$ and the measured output $\boldsymbol{y}$ are provided to the disturbance observer. The observer dynamics can be described for state and disturbance estimation $\hat{\boldsymbol{x}}_e = \begin{bmatrix} \hat{\boldsymbol{x}}^T, & \hat{\boldsymbol{x}}_s^T \end{bmatrix}^T$:

$$\begin{aligned} \begin{bmatrix} \dot{\hat{\boldsymbol{x}}} \\ \dot{\hat{\boldsymbol{x}}}_s \end{bmatrix} &= \begin{bmatrix} \boldsymbol{A} - \boldsymbol{L}_x\boldsymbol{C} & \boldsymbol{E}\boldsymbol{C}_s \\ -\boldsymbol{L}_{x_s}\boldsymbol{C} & \boldsymbol{A}_s \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{x}}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{B} & \boldsymbol{L}_x \\ \boldsymbol{0} & \boldsymbol{L}_{x_s} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{y} \end{bmatrix} \\ \hat{\boldsymbol{y}} &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{x}} \\ \hat{\boldsymbol{x}}_s \end{bmatrix}, \end{aligned} \tag{6.33}$$

where $\boldsymbol{L}_e = \begin{bmatrix} \boldsymbol{L}_x^T & \boldsymbol{L}_{x_s}^T \end{bmatrix}^T \in \mathbb{R}^{(n+s)\times m}$ is the correction matrix. $\boldsymbol{L}_e$ must be determined in such a way that the estimation error

$$\boldsymbol{e}(t) = \boldsymbol{x}_e(t) - \hat{\boldsymbol{x}}_e(t)$$

tends to zero for $t \to \infty$ with arbitrary initial conditions $\boldsymbol{x}_e(0) = \left[\boldsymbol{x}_0^T, \quad \boldsymbol{x}_{s0}^T\right]^T$.

The LQG regulator (6.15) is now extended with the feedback of the estimated disturbance state $\hat{\boldsymbol{x}}_s$:

$$\boldsymbol{u} = -\boldsymbol{K}\hat{\boldsymbol{x}} - \boldsymbol{K}_s\hat{\boldsymbol{x}}_s. \tag{6.34}$$

6.4 Observer-Based State Difference Feedback

The desired trajectories for both the control input and the state variables, as well as for the outputs, can be generated offline by the feedforward control. In practice, these trajectories are stored in the lookup table and interpolated at each sampling time so that the CPU can be heavily loaded. Therefore, the control structure is applied for the state difference $\boldsymbol{e}_x = \boldsymbol{x}^* - \boldsymbol{x}$, which requires only supplying the desired output trajectories $\boldsymbol{y}^*$ to the feedback controller, instead of the full state reference $\boldsymbol{x}^*$, which saves storage and execution time on the PLC system, see Figure 6.3. Furthermore, the observer is extended with a disturbance model to estimate and compensate the unknown friction.

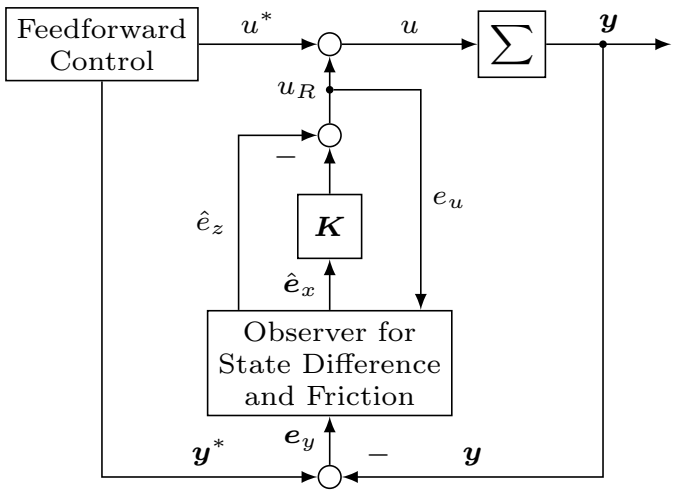


Figure 6.3: Control scheme with feedforward control and state feedback based on the estimated state difference and disturbance. $\sum$ denotes the plant model (6.30).

Difference model. We consider a nominal model for the feedforward control

$$\begin{aligned} \dot{\boldsymbol{x}}^* &= \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{B}u^* + \boldsymbol{E}z^* \\ \boldsymbol{y}^* &= \boldsymbol{C}\boldsymbol{x}^*, \end{aligned} \tag{6.35}$$

on the basis of which the desired reference trajectories are calculated.

An LTI state space model

$$\begin{aligned} \dot{\boldsymbol{e}}_x &= \boldsymbol{A}\boldsymbol{e}_x + \boldsymbol{B}e_u + \boldsymbol{E}e_z \\ \boldsymbol{e}_y &= \boldsymbol{C}\boldsymbol{e}_x \end{aligned} \tag{6.36}$$

is described for the differences of states, input, disturbance, and measured outputs from their reference values

$$\boldsymbol{e}_x = \boldsymbol{x}^* - \boldsymbol{x}, \qquad e_u = u^* - u, \qquad e_z = z^* - z = -z, \qquad \boldsymbol{e}_y = \boldsymbol{y}^* - \boldsymbol{y},$$

whereby we assume zero disturbance $z^* = 0$ in the nominal case, for details , see [19]. A disturbance observer can be used for the difference system (6.36) to estimate the difference variables $\hat{\boldsymbol{e}}_x$ and $\hat{e}_z$. In our case, e_z is equal to the negative friction torque.

The control input u is composed of the feedforward part u^* and the feedback part u_R:

$$u = u^* + u_R, \tag{6.37}$$

see Figure 6.3, such that the control input of the difference model (6.36) becomes

$$e_u = u^* - u = -u_R. \tag{6.38}$$

Extended difference model with a disturbance model. In most cases, the friction torque can be modeled as a piecewise constant function, i.e., as the integrator model

$$\begin{aligned} \dot{e}_s &= a_s e_s \\ e_z &= c_s e_s, \end{aligned} \tag{6.39}$$

with $a_s = 0$ and $c_s = 1$. By inserting (6.39) into the difference model (6.36), we obtain an extended difference model with the state vector $\boldsymbol{e}_e = \begin{bmatrix} \boldsymbol{e}_x^T, & e_s \end{bmatrix}^T$

$$\begin{aligned} \dot{\boldsymbol{e}}_e &= \boldsymbol{A}_e \boldsymbol{e}_e + \boldsymbol{B}_e e_u \\ \boldsymbol{e}_y &= \boldsymbol{C}_e \boldsymbol{e}_e. \end{aligned} \tag{6.40}$$

The extended system matrices are

$$\begin{aligned} \boldsymbol{A}_e &= \begin{bmatrix} \boldsymbol{A} & \boldsymbol{E} \\ \boldsymbol{0}^T & 0 \end{bmatrix} \\ \boldsymbol{B}_e &= \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0} \end{bmatrix} \\ \boldsymbol{C}_e &= \begin{bmatrix} \boldsymbol{C} & \boldsymbol{0} \end{bmatrix}. \end{aligned} \tag{6.41}$$

.

Observer dynamics. According to [19], the observer dynamics for the estimated state vector $\hat{\boldsymbol{e}}_e = \begin{bmatrix} \hat{\boldsymbol{e}}_x^T, & \hat{e}_s \end{bmatrix}^T$ is described as

$$\dot{\hat{\boldsymbol{e}}}_e = \boldsymbol{A}_e \hat{\boldsymbol{e}}_e + \boldsymbol{B}_e \boldsymbol{e}_u + \boldsymbol{L}_e (\boldsymbol{e}_y - \hat{\boldsymbol{e}}_y) \tag{6.42}$$

with the correction term $\boldsymbol{L}_e(\boldsymbol{e}_y - \hat{\boldsymbol{e}}_y)$. The observer needs the information of both the control input difference e_u and the measured output difference $\boldsymbol{e}_y$.

Similar to the classical case, the dynamics

$$\Delta\dot{\boldsymbol{e}}_e = (\boldsymbol{A}_e - \boldsymbol{L}_e\boldsymbol{C}_e)\Delta\boldsymbol{e}_e \tag{6.43}$$

for the observation error $\Delta\boldsymbol{e}_e(t) = \boldsymbol{e}_e(t) - \hat{\boldsymbol{e}}_e(t)$ depends on the observer gain $\boldsymbol{L}_e$, which should be determined so that $\Delta\boldsymbol{e}_e(t)$ tends toward zero for $t \to \infty$ under arbitrary initial conditions $\boldsymbol{e}_e(0)$ and $\hat{\boldsymbol{e}}_e(0)$. In our case, $\boldsymbol{L}_e$ is determined by the Kalman filter, see Section 6.3.2.

Control law. The observer-based state feedback

$$e_u = -u_R = -\boldsymbol{K}\hat{\boldsymbol{e}}_x + \hat{e}_z, \tag{6.44}$$

which is extended by the disturbance compensation based on the estimation $\hat{z} = -\hat{e}_z$ of the unknown friction torque, is designed for the difference model (6.36). According to the control input (6.37), the control law, as shown in Figure 6.3, becomes

$$u = u^* + \boldsymbol{K}\hat{\boldsymbol{e}}_x - \hat{e}_z. \tag{6.45}$$

Note that the controller gain $\boldsymbol{K}$ is determined on the basis of the difference model (6.36) for the undisturbed case.

Figure 6.4 shows the implementation procedure of the LQG regulator in MATLAB. The dynamics of the entire system is mathematically described by a continuous-time model. Since the model-based control design runs in a real-time system, it must be transferred to a discrete-time domain. To design the state observer, the extended difference model (6.40) is first discretized in time with the MATLAB function $c2d(T)$, which discretizes the continuous-time dynamical system model using zero-order hold (ZOH) on the inputs at a sampling time of T. Based on this discrete-time model, a discrete-time observer can be determined using the MATLAB function $dlqr(\bullet)$ with the corresponding weighting matrices. In contrast, the discrete LQG regulator is designed for the continuous-time undisturbed plant model (6.36) using the MATLAB function $lqrd(T)$, which discretizes the plant with $c2d(T)$ and calculates the gain matrix from the discretized data with $dlqr(\bullet)$.

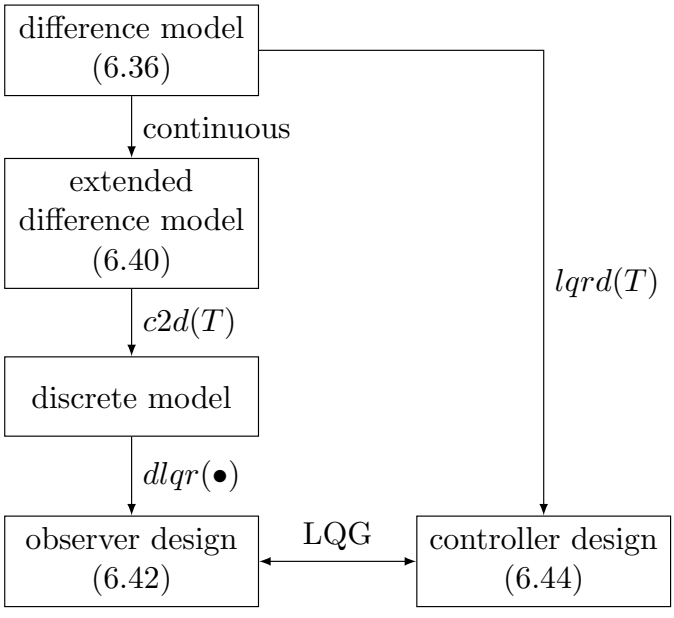


Figure 6.4: Design procedure of the LQG regulator

Chapter 7

Observer-Based Trajectory Control for Time-Delay Systems

7.1 Problem Formulation

Time delays occur in most technical processes and are typically caused by e.g. mass and energy transport, signal and information transmission, cyclic operation, and time load required for calculation or decision-making, see [68]. Time delays, in particular, are neglected in the modeling and control design process, as they may be unknown, unpredictable, stochastic, or difficult to identify. This can lead to a deterioration of the control quality or even to the instability of the controlled system. It is advantageous if we can at least identify the extent of the time delays and take them into account in the modeling process. On the basis of the model extended by time delays, the proposed observer-based control strategy can be applied.

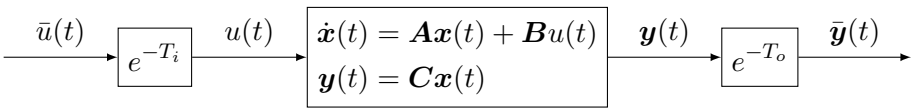


Figure 7.1: Time-delay system

Figure 7.1 shows the sketch of a system with time delays in our application[1]. There is not only the input delay T_i, which is caused by the communication process from the PLC to ACOPOS but also the output delay T_o, which occurs when the data is transferred from sensors to the PLC. Now, the system description is extended by the time delays in the continuous-time domain

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\bar{u}(t - T_i), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{7.1a}$$

$$\bar{\boldsymbol{y}}(t + T_o) = \boldsymbol{C}\boldsymbol{x}(t), \tag{7.1b}$$

where $u(t) = \bar{u}(t - T_i) \in \mathbb{R}$ and $\boldsymbol{y}(t - T_o) = \bar{\boldsymbol{y}}(t) \in \mathbb{R}^3$. It is assumed that the two time delays in our case are multiples of the sampling time T, i.e.,

$$\begin{aligned} T_i &= d_i\, T \\ T_o &= d_o\, T. \end{aligned} \tag{7.2}$$

[1] In our case, there is only one single input $u(t) \in \mathbb{R}$, namely the motor torque. Nevertheless, $\boldsymbol{B} \in \mathbb{R}^{n\times 1}$ here denotes the input column vector, while $\boldsymbol{C} \in \mathbb{R}^{3\times n}$ is the output matrix.

In the following sections, we introduce some time-delay models that are frequently used. In general, the continuous-time system (6.1) can be discretized in time and then transformed into a discrete standard form without explicit time delay by augmenting the number of states. The mentioned state (disturbance) observer is designed for these systems with multiple time delays.

7.2 Discrete-Time State Space Model

Up to now, the system (6.1) is considered in continuous time, and the solution is derived with the state transition matrices

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}t}\boldsymbol{x}_0 + \int_0^t e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}u(\tau)d\tau. \tag{7.3}$$

In reality, the controller operates on the digital control system, the structure of which is shown in Figure 7.2. The measured output $\boldsymbol{y}(t)$ is sampled with an A/D converter and fed back to the controller at each sampling step, while the control input $u[k]$ is held at the same time using the ZOH component. According to [50], we first introduce how to obtain the discrete-time state space model from the continuous-time representation.

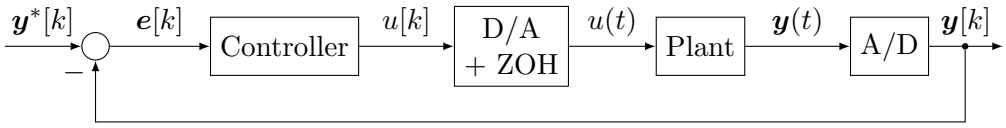


Figure 7.2: Digital control structure

Due to the ZOH component, the control input is constant within the sampling time

$$u(t) = u(kT) = const., \qquad kT \leq t < (k+1)T. \tag{7.4}$$

The output equation at the sampling time $t = kT$ can be derived by inserting $t = kT$ in (6.1b)

$$\boldsymbol{y}(kT) = \boldsymbol{C}\boldsymbol{x}(kT)$$

with the same output matrix $\boldsymbol{C}$ of the continuous-time system. Similarly, the Solution (7.3) is calculated separately for $t = (k+1)T$

$$\begin{aligned} \boldsymbol{x}((k+1)T) &= e^{\boldsymbol{A}T}\boldsymbol{x}(kT) + \int_{kT}^{(k+1)T} e^{\boldsymbol{A}((k+1)T-\tau)}\boldsymbol{B}u(\tau)d\tau \\ &= e^{\boldsymbol{A}T}\boldsymbol{x}(kT) + \int_0^T e^{\boldsymbol{A}(T-\tau')}\boldsymbol{B}d\tau' u(kT) \end{aligned} \tag{7.5}$$

with the substitution $\tau' = \tau - kT$. The result is a discrete-time state space system represented in a short form

$$\boldsymbol{x}[k+1] = \boldsymbol{A}'\boldsymbol{x}[k] + \boldsymbol{B}'u[k], \qquad \boldsymbol{x}[0] = \boldsymbol{x}_0 \tag{7.6a}$$

$$\boldsymbol{y}[k] = \boldsymbol{C}'\boldsymbol{x}[k] \tag{7.6b}$$

with the following definition of new matrices

$$\begin{aligned} \boldsymbol{A'} &= e^{\boldsymbol{A}T} \\ \boldsymbol{B'} &= \int_0^T e^{\boldsymbol{A}(T-\tau')}\boldsymbol{B}d\tau' \\ \boldsymbol{C'} &= \boldsymbol{C}. \end{aligned} \tag{7.7}$$

Note that the system order of a continuous-time system without time delays is not changed by time discretization. This description is derived exactly from the continuous-time form (6.1) at $t = kT$. Since the state variables are the same for both cases, the output equation remains unchanged. In addition, the discrete matrices depend on the sampling time T, which is explicitly defined in Equation (7.7).

7.3 Discrete Time-Delay System

As mentioned above, the real-time control system must be able to cope with input and output delays. The input delays mainly occur during the signal transmission in the Ethernet between the PLC and the servo amplifier. This delay process is caused by writing input signals to the PLC, storing data from the PLC in a cache, and transferring the data from the cache to the servo amplifier. At least, we can experimentally determine an average delay of the signal output in the PLC, see e.g. Figure 2.11.

The output delays occur mainly during the digital transmission of measurements. In our case, the encoder is directly connected to a plug-in card in the servo amplifier, see Figure 2.3. Its delay is much smaller than the one of the IMU, whose data is transmitted through the CAN bus. For this reason, we consider only the delayed signals from the IMU as output delay.

The time-delay model can be transformed into a discrete-time state space representation, see e.g. [50], [61]. If the time delay T_t is a multiple of the sampling time T, the discrete-time model (7.6) is simply extended by $\frac{T_t}{T}$ additional state variables with respect to time delays, as in our case (7.2). In the following sections, we introduce two time-delay models, i.e., the input and output delay model.

7.3.1 Input Delay Model

Equation (7.1a) shows that the calculated input $\bar{u}(t)$ is delayed by the time $T_i = d_i T$ and acts on the continuous-time system. In order to convert the control design into standard form without explicit time delay, we concentrate on the discrete-time domain

$$\bar{u}[k - d_i] = u[k]. \tag{7.8}$$

For this purpose, we define a set of discrete-time state variables ξ_i, $i = 1, \ldots, d_i$ according to [50]

$$\begin{aligned} u[k] &= \xi_1[k] \\ u[k+1] &= \xi_2[k] \\ &\vdots \\ u[k+(d_i-1)] &= \xi_{d_i}[k] \end{aligned} \tag{7.9a}$$

$$u[k+d_i] = \xi_{d_i}[k+1] = \bar{u}[k]. \tag{7.9b}$$

Furthermore, the external disturbance $z[k]$ is also considered in the discrete-time model. Since the estimated disturbance is later added to the control input $u[k]$, which is delayed, we now replace the undelayed disturbance $z[k]$ by a fictitious disturbance $\bar{z}(t)$, which can be inserted directly into $\bar{u}[k]$. Therefore, the control input (7.9b) is modified as follows

$$u[k+d_i] = \xi_{d_i}[k+1] = \bar{u}[k] - \bar{z}[k]. \tag{7.10}$$

Then, Equations (7.9a) and (7.10) are rewritten in a state space representation

$$\begin{aligned} \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_{d_i-1} \\ \xi_{d_i} \end{bmatrix}_{k+1} &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\boldsymbol{A}_\xi} \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_{d_i-1} \\ \xi_{d_i} \end{bmatrix}_k + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{\boldsymbol{B}_\xi} \bar{u}[k] + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix}}_{\boldsymbol{E}_\xi} \bar{z}[k] \\ y_\xi[k] = u[k] &= \underbrace{\begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}}_{\boldsymbol{C}_\xi} \boldsymbol{\xi}. \end{aligned} \tag{7.11}$$

By inserting the input delay model (7.11) into the discrete-time plant model (7.6), the state space model is now extended with the input delay

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix}_{k+1} &= \left[\begin{array}{c|c} \boldsymbol{A}' & \boldsymbol{B}' \quad \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{A}_\xi \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix}_k + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{B}_\xi \end{bmatrix} \bar{u}[k] + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{E}_\xi \end{bmatrix} \bar{z}[k] \\ \boldsymbol{y}[k] &= \begin{bmatrix} \boldsymbol{C}' & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \end{bmatrix}_k. \end{aligned} \tag{7.12}$$

7.3.2 Output Delay Model

As mentioned above, a large time delay exists in the transmission of data from the IMU to the PLC via CAN bus compared with the measurement from the encoder. Therefore, only an output delay model for the angular velocity of the beam tip is considered. Since the IMU is read with its own predefined cycle time, and its data is only returned to the digital controller at each sampling time T, this corresponds to a multiple of the sampling time.

Based on Equation (7.2), the output delay (7.1b) is converted in discrete-time form

$$\bar{y}_3[k] = y_3[k - d_o]. \tag{7.13}$$

Then, a new set of discrete-time state variables β_j, $j = 1, \ldots, d_o$ is used to model the output delays

$$\begin{aligned} \bar{y}_3[k] &= \beta_1[k] \\ \bar{y}_3[k+1] &= \beta_2[k] \\ &\vdots \\ \bar{y}_3[k + (d_o - 1)] &= \beta_{d_o}[k] \end{aligned} \tag{7.14a}$$

$$\bar{y}_3[k + d_o] = y_3[k] = \beta_{d_o}[k+1], \tag{7.14b}$$

or

$$\begin{aligned} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{d_i-1} \\ \beta_{d_i} \end{bmatrix}_{k+1} &= \underbrace{\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}}_{\boldsymbol{A}_\beta} \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{d_o-1} \\ \beta_{d_i} \end{bmatrix}_k + \underbrace{\begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_{\boldsymbol{B}_\beta} y_3[k] \\ \bar{y}_3[k] &= \underbrace{\begin{bmatrix} 1 & \mathbf{0} \end{bmatrix}}_{\boldsymbol{C}_\beta} \boldsymbol{\beta}[k]. \end{aligned} \tag{7.15}$$

$y_3[k]$ can be represented by the state vector $\boldsymbol{x}[k]$ as follows

$$y_3[k] = \boldsymbol{C}_3' \boldsymbol{x}[k] \tag{7.16}$$

with $\boldsymbol{C}_3' = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \boldsymbol{C}'$, see Equation (7.6b). Finally, the discrete-time model with the output delay can be described in state space form

$$\begin{aligned} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\beta} \end{bmatrix}_{k+1} &= \left[\begin{array}{c|c} \boldsymbol{A}' & \mathbf{0} \\ \hline \boldsymbol{B}_\beta \boldsymbol{C}_3' & \boldsymbol{A}_\beta \end{array}\right] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\beta} \end{bmatrix}_k + \begin{bmatrix} \boldsymbol{B}' \\ \mathbf{0} \end{bmatrix} \bar{u}[k] + \begin{bmatrix} \boldsymbol{E}' \\ \mathbf{0} \end{bmatrix} \bar{z}[k] \\ \begin{bmatrix} y_1 \\ y_2 \\ \hline \bar{y}_3 \end{bmatrix}_k &= \left[\begin{array}{c|c} \boldsymbol{C}_{12}' & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{C}_\beta \end{array}\right] \left[\begin{array}{c} \boldsymbol{x} \\ \hline \boldsymbol{\beta} \end{array}\right]_k \end{aligned} \tag{7.17}$$

with $\boldsymbol{C}_{12}' = \begin{bmatrix} \boldsymbol{I}_2 & \mathbf{0} \end{bmatrix} \boldsymbol{C}'$.

7.3.3 Input and Output Delay System

Taking into account the two delay models (7.12) and (7.17) in the system description, we obtain the state space representation of the input- and the output-delay system as follows

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \\ \hdashline \boldsymbol{\beta} \end{bmatrix}_{k+1} = \underbrace{\left[\begin{array}{ccc:c} \boldsymbol{A}' & \boldsymbol{B}' & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_\xi & & \\ \hdashline \boldsymbol{B}_\beta \boldsymbol{C}_3' & \boldsymbol{0} & & \boldsymbol{A}_\beta \end{array}\right]}_{\bar{\boldsymbol{A}}} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \\ \hdashline \boldsymbol{\beta} \end{bmatrix}_k + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{B}_\xi \\ \hdashline \boldsymbol{0} \end{bmatrix}}_{\bar{\boldsymbol{B}}} \bar{u}[k] + \underbrace{\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{E}_\xi \\ \hdashline \boldsymbol{0} \end{bmatrix}}_{\bar{\boldsymbol{E}}} \bar{z}[k]$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \hdashline \bar{y}_3 \end{bmatrix}_k = \underbrace{\left[\begin{array}{ccc:c} \boldsymbol{C}_{12}' & \boldsymbol{0} & \boldsymbol{0} \\ \hdashline \boldsymbol{0} & & \boldsymbol{C}_\beta \end{array}\right]}_{\bar{\boldsymbol{C}}} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \\ \hdashline \boldsymbol{\beta} \end{bmatrix}_k \tag{7.18}$$

with new state vector $\bar{\boldsymbol{x}} \in \mathbb{R}^{n+d_i+d_o}$ and output vector $\bar{\boldsymbol{y}} \in \mathbb{R}^3$

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{\xi} \\ \boldsymbol{\beta} \end{bmatrix}, \qquad \bar{\boldsymbol{y}} = \begin{bmatrix} y_1 \\ y_2 \\ \bar{y}_3 \end{bmatrix}.$$

7.4 Observer-Based State Difference Feedback for Time-Delay Systems

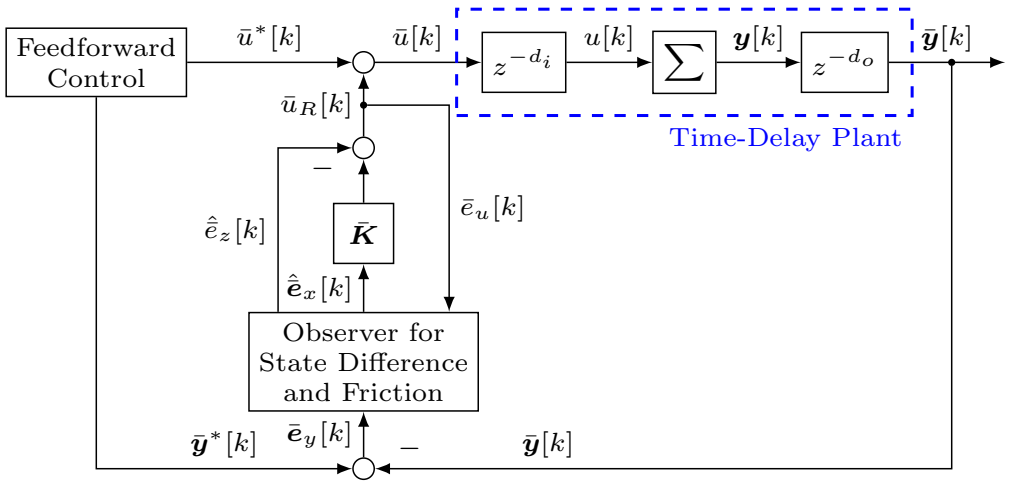


Figure 7.3: Control scheme with feedforward control and state feedback based on the estimated state difference and disturbance for a time-delay system. $\sum$ denotes the plant model (7.6).

Figure 7.3 shows the control scheme for a time-delay system, which includes both the input and output delay model, with the feedforward control and the observer-based state feedback. For the same reason mentioned in Section 6.4, the observer is designed

to estimate the state difference $\bar{\boldsymbol{e}}_x[k] = \bar{\boldsymbol{x}}^*[k] - \bar{\boldsymbol{x}}[k]$ so that only the output trajectory $\bar{\boldsymbol{y}}^*[k]$ is needed. Thus, we consider a difference model for the time-delay system (7.18) as follows

$$\begin{aligned} \bar{\boldsymbol{e}}_x[k+1] &= \bar{\boldsymbol{A}}\bar{\boldsymbol{e}}_x[k] + \bar{\boldsymbol{B}}\bar{e}_u[k] + \bar{\boldsymbol{E}}\bar{e}_z[k] \\ \bar{\boldsymbol{e}}_y[k] &= \bar{\boldsymbol{C}}\bar{\boldsymbol{e}}_x[k] \end{aligned} \tag{7.19}$$

with

$$\begin{aligned} \bar{\boldsymbol{e}}_x[k] &= \bar{\boldsymbol{x}}^*[k] - \bar{\boldsymbol{x}}[k] \\ \bar{\boldsymbol{e}}_y[k] &= \bar{\boldsymbol{y}}^*[k] - \bar{\boldsymbol{y}}[k] \\ \bar{e}_u[k] &= \bar{u}^*[k] - \bar{u}[k] \\ \bar{e}_z[k] &= \bar{z}^*[k] - \bar{z}[k] = -\bar{z}[k]. \end{aligned}$$

Similar to the continuous-time case (6.38), the observer-based state feedback

$$\bar{e}_u[k] = -\bar{u}_R[k] = -\bar{\boldsymbol{K}}\hat{\bar{\boldsymbol{e}}}_x[k] \tag{7.20}$$

is designed for the difference system (7.19) for the undisturbed case, where $\hat{\bar{\boldsymbol{e}}}_x[k]$ is the estimation of state difference vector, and $\bar{\boldsymbol{K}} \in \mathbb{R}^{1\times(n+d_i+d_o)}$ denotes the controller gain. In addition, the control law (7.20) can be extended to include the disturbance compensation based on the estimation $\hat{\bar{z}}[k] = -\hat{\bar{e}}_z[k]$. The control input $\bar{u}[k]$ for the system (7.18) is composed of the feedforward part $\bar{u}^*[k]$ and the feedback part, which is based on $\hat{\bar{\boldsymbol{e}}}_x[k]$ and $\hat{\bar{e}}_z[k]$

$$\bar{u}[k] = \bar{u}^*[k] + \bar{\boldsymbol{K}}\hat{\bar{\boldsymbol{e}}}_x[k] - \hat{\bar{e}}_z[k]. \tag{7.21}$$

In the following sections, a discrete-time LQG regulator is applied for the difference model (7.19). Note that we do not consider the delayed model for the design of the feedforward control in this section. It will be instantiated later for our application in Section 8.6.

7.4.1 Discrete-Time Optimal Control

Since there are many state variables, mainly from the spatial discretization and time-delay models, it is useful to directly apply a discrete-time optimal control for System (7.19) to determine the controller gain $\bar{\boldsymbol{K}}$ for the control law (7.20). The optimal control is obtained by minimizing the cost functional J

$$\min_{\bar{e}_u} J = \min_{\bar{e}_u} \sum_{k=0}^{\infty} (\bar{\boldsymbol{e}}_x^T[k]\bar{\boldsymbol{Q}}\bar{\boldsymbol{e}}_x[k] + \bar{e}_u[k]\bar{R}\bar{e}_u[k]). \tag{7.22}$$

$\bar{R} \in \mathbb{R}$ is a positive definite weighting factor for the control input $\bar{e}_u[k]$, while the state vector $\bar{\boldsymbol{e}}_x[k]$ is weighted by a positive semidefinite matrix $\bar{\boldsymbol{Q}} \in \mathbb{R}^{(n+d_i+d_o)\times(n+d_i+d_o)}$.

By solving the Riccati equation

$$\bar{\boldsymbol{P}} = \bar{\boldsymbol{Q}} + \bar{\boldsymbol{A}}^T\bar{\boldsymbol{P}}\bar{\boldsymbol{A}} - \bar{\boldsymbol{A}}^T\bar{\boldsymbol{P}}\bar{\boldsymbol{B}}(\bar{R} + \bar{\boldsymbol{B}}^T\bar{\boldsymbol{P}}\bar{\boldsymbol{B}})^{-1}\bar{\boldsymbol{B}}^T\bar{\boldsymbol{P}}\bar{\boldsymbol{A}}, \tag{7.23}$$

the optimal controller is calculated as follows

$$\bar{\boldsymbol{K}} = (\bar{R} + \bar{\boldsymbol{B}}^T\bar{\boldsymbol{P}}\bar{\boldsymbol{B}})^{-1}\bar{\boldsymbol{B}}^T\bar{\boldsymbol{P}}\bar{\boldsymbol{A}} \tag{7.24}$$

with a positive definite matrix $\bar{\boldsymbol{P}}$, for details see [50].

7.4.2 Discrete-Time Disturbance Observer

In order to describe the friction torque in discrete time, a piecewise constant function is chosen as the disturbance model

$$\begin{aligned} \bar{e}_s[k+1] &= a_s \bar{e}_s[k] \\ \bar{e}_z[k] &= c_s \bar{e}_s[k], \end{aligned} \tag{7.25}$$

where $a_s = 1$ and $c_s = 1$. The time-delay system (7.19) can be extended with disturbance model (7.25) for the substituted state vector $\bar{\boldsymbol{e}}_e = \begin{bmatrix} \bar{\boldsymbol{e}}_x^T, & \bar{e}_s \end{bmatrix}^T$

$$\begin{aligned} \bar{\boldsymbol{e}}_e[k+1] &= \bar{\boldsymbol{A}}_e \bar{\boldsymbol{e}}_e[k] + \bar{\boldsymbol{B}}_e \bar{e}_u[k] \\ \bar{\boldsymbol{e}}_y[k] &= \bar{\boldsymbol{C}}_e \bar{\boldsymbol{e}}_e[k] \end{aligned} \tag{7.26}$$

with system matrices

$$\begin{aligned} \bar{\boldsymbol{A}}_e &= \begin{bmatrix} \bar{\boldsymbol{A}} & \bar{\boldsymbol{E}} \\ \boldsymbol{0}^T & 0 \end{bmatrix} \\ \bar{\boldsymbol{B}}_e &= \begin{bmatrix} \bar{\boldsymbol{B}} \\ \boldsymbol{0} \end{bmatrix} \\ \bar{\boldsymbol{C}}_e &= \begin{bmatrix} \bar{\boldsymbol{C}} & \boldsymbol{0} \end{bmatrix}. \end{aligned}$$

According to [50], the observer dynamics for the estimated state vector $\hat{\bar{\boldsymbol{e}}}_e = \begin{bmatrix} \hat{\bar{\boldsymbol{e}}}_x^T, & \hat{\bar{e}}_s \end{bmatrix}^T$ is described as

$$\hat{\bar{\boldsymbol{e}}}_e[k+1] = \bar{\boldsymbol{A}}_e \hat{\bar{\boldsymbol{e}}}_e[k] + \bar{\boldsymbol{B}}_e \bar{e}_u[k] + \boldsymbol{L}_e(\bar{\boldsymbol{e}}_y[k] - \hat{\bar{\boldsymbol{e}}}_y[k]). \tag{7.27}$$

The observer gain $\boldsymbol{L}_e$ should be determined to eliminate the observation error for each sampling time

$$\lim_{k \to \infty} \left\| \bar{\boldsymbol{e}}_e[k] - \hat{\bar{\boldsymbol{e}}}_e[k] \right\| = 0. \tag{7.28}$$

It is noticeable that the input difference $\bar{e}_u[k]$ and the output difference $\bar{\boldsymbol{e}}_y[k]$ at time k are used to predict the state difference $\bar{\boldsymbol{e}}_e[k+1]$ at time $k+1$. If the calculated output difference $\hat{\bar{\boldsymbol{e}}}_y$ matches the measurement $\bar{\boldsymbol{e}}_y$, the observer simply uses the plant model to predict the state $\hat{\bar{\boldsymbol{e}}}_e[k+1]$. If an output error exists, the prediction is then corrected by the term $\boldsymbol{L}_e(\bar{\boldsymbol{e}}_y[k] - \hat{\bar{\boldsymbol{e}}}_y[k])$, see details in [50]. In our case, the observation gain $\boldsymbol{L}_e$ is determined by the Kalman filter.

7.4.3 Implementation Procedure in MATLAB

Figure 7.4 illustrates the implementation procedure of an LQG regulator for the time-delay system in MATLAB. Since the difference model (6.36) is linear time-invariant, its discrete-time model (7.6) is derived directly with the MATLAB function *c2d(T)*. Based on the discrete time-delay difference model (7.19), the Riccati controller is determined by the MATLAB function *dlqr*(•). In addition, the extended time-delay model (7.26)

is used to design the Kalman filter with $dlqr(\bullet)$, which calculates the observer gain $\boldsymbol{L}_e$ with the selected weighting matrices for the extended model in discrete time.

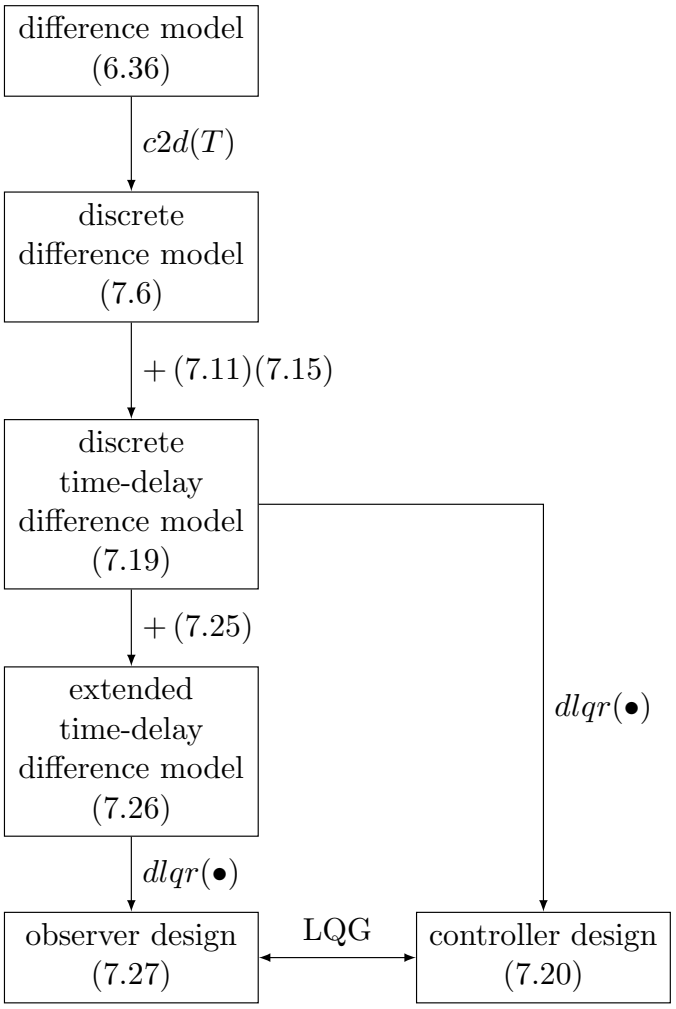


Figure 7.4: The LQG regulator's implementation procedure for the time-delay system

Chapter 8

Experimental Control Results for a Flexible Robot Arm

There are two control issues: On the one hand, the beam tip should follow the desired trajectory as quickly as possible; on the other hand, the oscillation of the flexible beam should be suppressed as much as possible. To this end, the motion task to demonstrate the quality of the selected control strategy is a fast PTP motion of a flexible robot by 90 degrees.

In this chapter, four control strategies are considered by combining different feedforward and feedback control approaches, see Table 8.1. Comparing Experiment A with B only makes a difference in the model chosen for the feedforward control design. One is based on a rigid body model (rbFFC), which is presented in Section 4.2, while the other uses an inverse model (ibFFC) that accounts for the beam elasticity, see Section 4.3. In both experiments, disturbance compensations based on a linear static friction model are added to the PD feedback control (pdFBC). The comparison of these two experiments shows the tracking quality of the different feedforward strategies. Both Experiment C and D combine an ibFFC and an LQG regulator with a disturbance observer. The difference is that in Experiment D, the time delay model is considered for the feedforward and feedback control design. In the following, two experimental scenarios are considered: (a) the undisturbed execution of the motion task and (b) the reaction of the control system to an exogenous disturbance – after the PTP motion, the beam is hit with a broomstick.

Experiment	FFC		FBC		Friction Compensation		Delay Model	
	rbFFC	ibFFC	pdFBC	LQG	static	dynamical	without	with
A	√		√		√		√	
B		√	√		√		√	
C		√		√		√	√	
D		√		√		√		√

Table 8.1: Experiment variants (FFC: feedforward control, FBC: feedback control)

8.1 Modeling and Discretization Parameters

This section contains the tables with the parameters for the drive unit, the steel beam, the structure-preserving discretization and model order reduction used to generate the model for the control design.

Parameter	Value	Unit
I_M	$2.29 \cdot 10^{-4}$	$\mathrm{kg\,m^2}$
I_h	$1.69 \cdot 10^{-4}$	$\mathrm{kg\,m^2}$
c_s	$5 \cdot 10^4$	Nm/rad
r_h	0.025	m

Table 8.2: Gear motor parameters

Table 8.2 shows the model parameters of the drive unit derived from the CAD design. As mentioned in Section 3.2, the moments of inertia of the motor and the hub are considered separately. According to the characteristic of the torsional stiffness of the Harmonic Drive gear, its elasticity is approximated by a linear torsion spring with stiffness c_s.

Parameter	Value	Unit
Length	1.20	m
Height	0.03	m
Thickness	0.005	m
Volumetric mass density	7856	$\mathrm{kg\,m^{-3}}$
Young's modulus	215	GPa
Poisson's ratio	0.28	–
Shear correction factor	$\frac{5}{6}$	–

Table 8.3: Parameters of the steel beam

Table 8.3 lists the beam model parameters. The thin steel beam has a length of 1.2m, a height of 0.03m, and a thickness of 0.005m. Since the beam is modeled as the Timoshenko beam model, we choose the commonly used Poisson's ratio and shear correction factor from [45].

Parameter	Value
Number of collocation points N	9
Order of the discretized beam model $4N$	36
Order of the reduced beam model r	12
Shift points s_0	[16, 18.43, 115.6]

Table 8.4: Parameters of the spatial discretization and the model reduction of the beam model

The beam model is spatially discretized using the PS method, see Section 3.4.1. As shown in Section 3.4.2, the more collocation points are selected, the more accurate the model is. Thus, $N = 9$ collocation points are chosen as the zeros of Legendre or Chebyshev polynomials to approximate well the first low eigenfrequencies and eigenfunctions of

the beam, see Table 3.1 and Figure 3.6. This beam model leads to a system order of $4 \cdot 9 = 36$, which brought the selected PLC to its limits. To obtain a small-scale, finite-dimensional beam model that retains the first eigenmode, the discretized beam model is reduced using the Krylov subspace method. Thus, 3 shift points are selected near the first natural frequency to produce a reduced beam model with a system order of $r = 12$, see Figure 3.8(a). This reduced beam model is used for the proposed control design, see e.g. Sections 6.4 and 7.4.

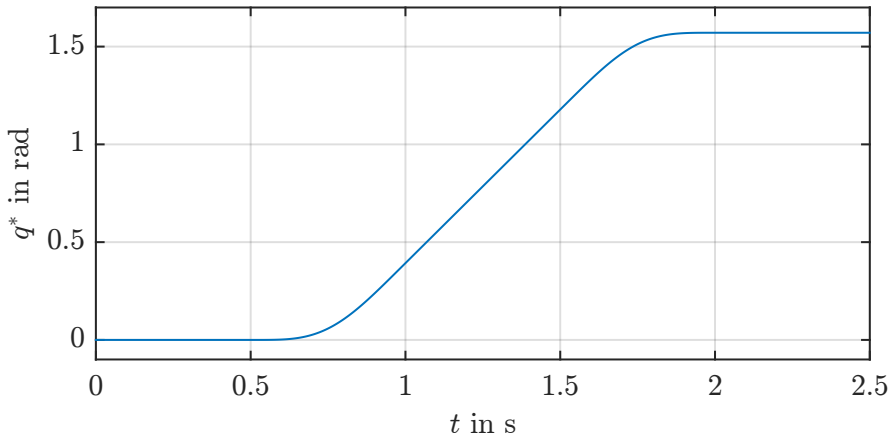
8.2 Parameterization of the Desired Trajectory

To generate the desired trajectory for PTP motion control, the trajectory planning is specified according to Section 4.1. The configuration parameters are listed in Table 8.5.

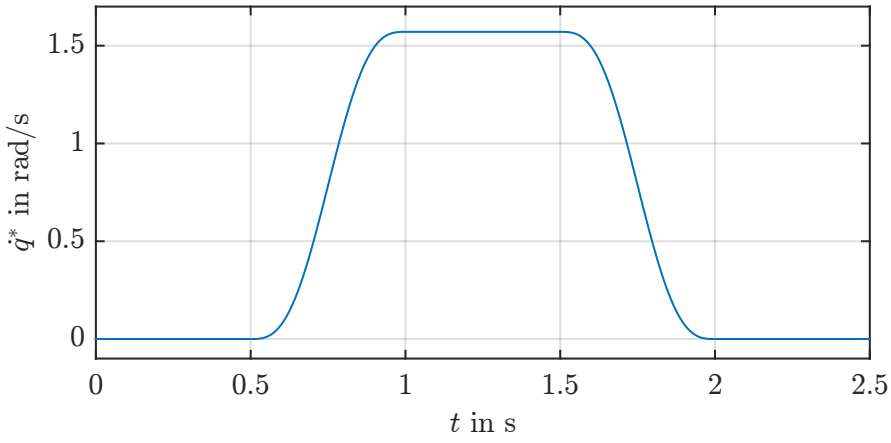
Parameter	Value	Unit
q_s	0	rad
q_e	$\frac{\pi}{2}$	rad
$\dot{q}_m$	$\frac{\pi}{2}$	rad/s
$\ddot{q}_m$	2π	rad/s^2
t_s	0.5	s
ΔT	1.5	s
t_e	2	s

Table 8.5: Parameter configuration of the path

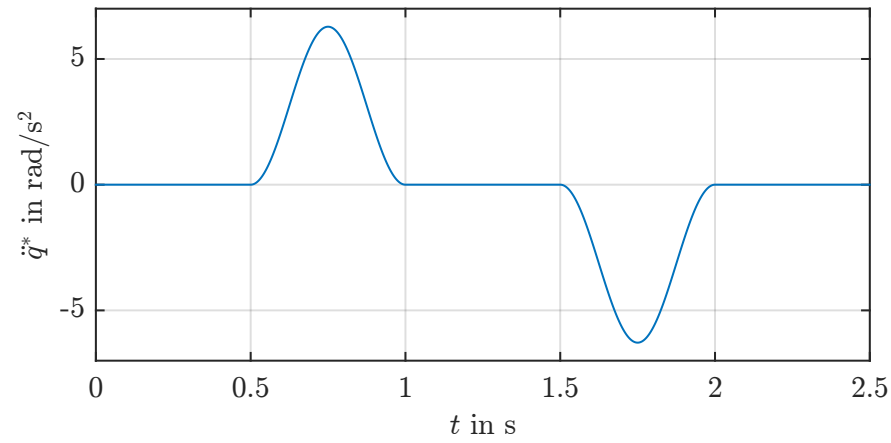
Figure 8.1 shows the simulated trajectories of angle, angular velocity, and acceleration. The movement starts at $t_s = 0.5$s from position $q_s = 0$ and ends at $t_e = 2$s at position $q_e = \frac{\pi}{2}$rad, see Figure 8.1(a). At the beginning of the motion, the speed increases to the maximum value $\dot{q}_m = \frac{\pi}{2}$rad/s within 0.5s. In this phase, the acceleration is given by a sinusoidal profile whose maximum value is $\ddot{q}_m = 2\pi$rad/s^2. After a uniform movement, braking is performed with the maximum deceleration until the speed is reduced to zero, see Figures 8.1(b) and 8.1(c). The entire route takes $\Delta T = 1.5$s. This path specifies the desired trajectory of the beam tip for our tracking task. Based on this path, the required trajectories of the control input and the state variables are calculated by the feedforward control.



(a) Desired angle $q^*(t)$



(b) Desired angular velocity $\dot{q}^*(t)$



(c) Desired angular acceleration $\ddot{q}^*(t)$

Figure 8.1: Simulation of the desired path

8.3 Experiment A: Rigid-Body-Based FFC and PD Joint Feedback Control

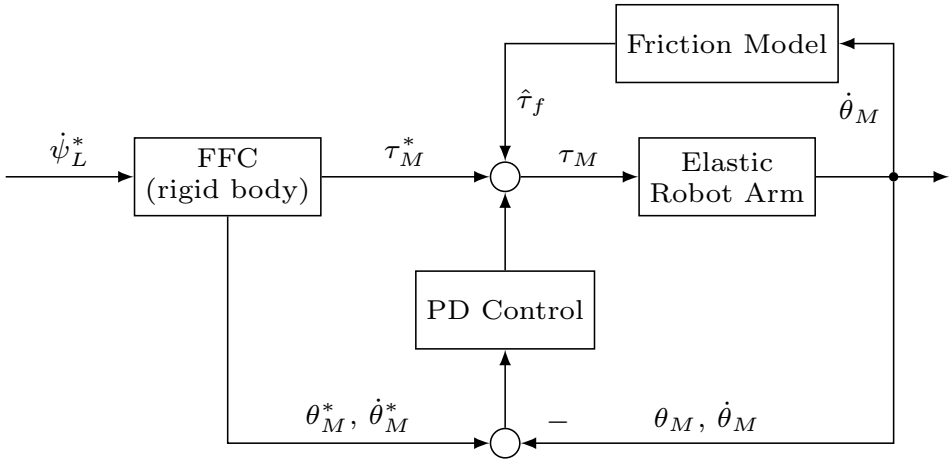


Figure 8.2: Two-degrees-of-freedom regulation: A rigid-body-based feedforward control (rbFFC) and a PD joint feedback with a static friction compensation

To illustrate the flexibility of the beam, Experiment A is performed with a feedforward control based on a rigid body model and with a simple PD joint feedback using the measurement of motor angle θ_M and angular velocity $\dot{\theta}_M$, see Figure 8.2. The nominal motor torque $\tau_M^*(t)$ and the desired trajectories of θ_M^* and $\dot{\theta}_M^*$ are calculated by Equations (4.15), (4.14), and (4.16), respectively. Moreover, the friction compensation in this setting is based on the linear static model (5.2) using a set of identified nominal parameters. The parameters of the friction model and the PD controller are listed in Table 8.6.

	Parameter	Value	Unit
Friction Model	b_1^l	3.04	Nm · s/rad
	b_0^l	7.684	Nm
	k	60	–
PD Controller	K_P	320	Nm/rad
	K_D	200	Nm · s/rad

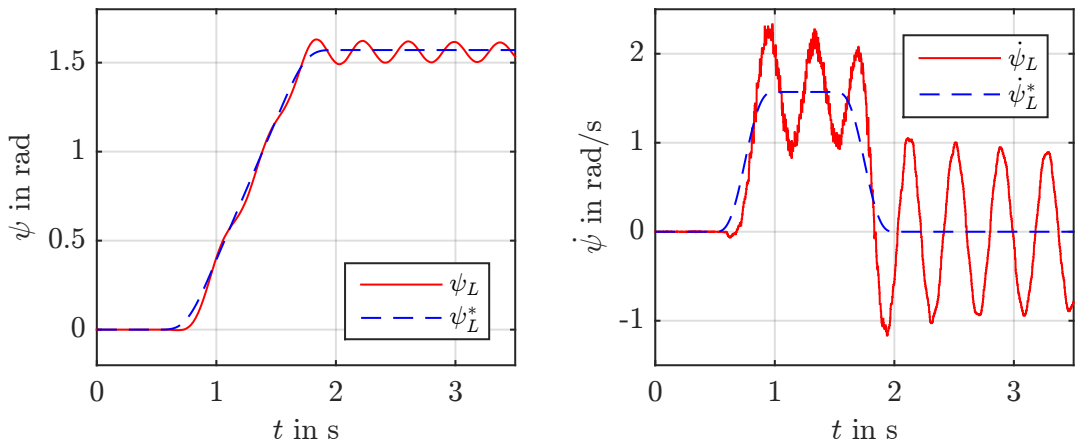
Table 8.6: Experiment A: Parameters of the friction model and the PD controller

In this experiment, we only consider the tracking behavior[1] of the beam tip with respect to the angle ψ_L and the angular velocity $\dot{\psi}_L$, see Figure 8.3. The results give evidence of the high flexibility and the low natural damping of the beam. The blue dashed line and the red solid line represent the desired trajectory and the corresponding measurement, respectively. At the beginning of the acceleration at $t = 0.5$s, the beam tip moves in the opposite direction due to the inertia of the structural flexibility. From a control point of view, the flexible robot arm is a non-minimum-phase system with respect to the selected

[1] See the video of the corresponding experiment under https://youtu.be/KzOa1Fq6B_g.

input and output, which has unstable internal dynamics. The delay of the input signal is more pronounced in these systems compared to the minimum-phase case, see details in Section 4.3.1. After the movement, there is still a very large oscillation.

To carry out the experiment, the calculated control torque is given as the setpoint τ_M^{set} in the torque control loop, see Figure 2.6. The setpoint τ_M^{set} (blue dashed line) and the measured torque τ_M^{act} (red solid line)) are illustrated in Figure 8.4. Since the current controller works quite fast with a cycle time of 50μs, the measurement almost matches the desired value. After $t = 2$s, there is some control effort due to the remaining stationary error.



(a) Beam tip angle

(b) Beam tip angular velocity

Figure 8.3: Experiment A: Tracking behavior of the beam tip angle and angular velocity using rbFFC and PD joint feedback

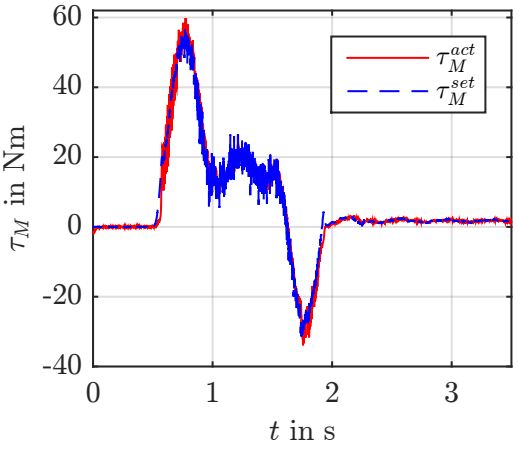


Figure 8.4: Experiment A: Motor torque for tracking the desired trajectory using rbFFC and PD joint feedback

8.4 Experiment B: Inversion-Based FFC and PD Joint Feedback Control

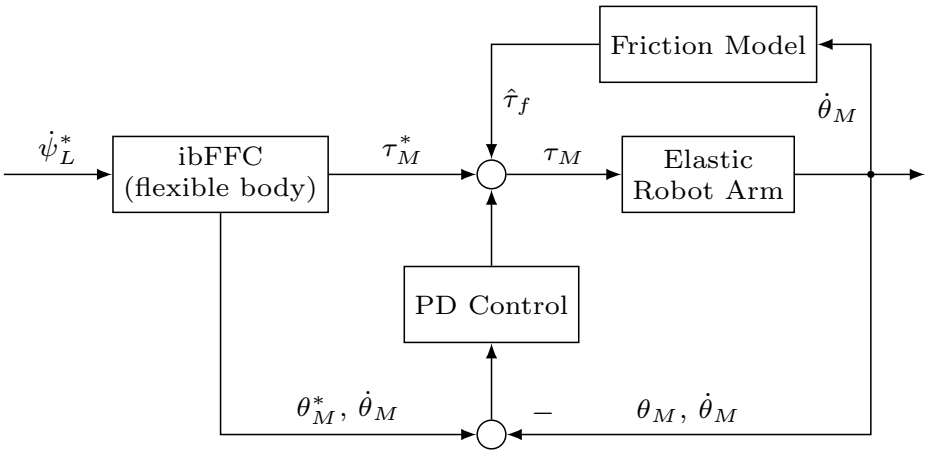


Figure 8.5: Two-degrees-of-freedom regulation: An inversion-based feedforward control (ibFFC) and a PD joint feedback with a static friction compensation

In Experiment B, a flexible beam model is exploited to design an ibFFC, see Figure 8.5. The desired trajectory for the tip angle $\psi^*(t, L)$ is generated based on the initial and final conditions, under the constraints of maximum joint velocity and acceleration, see Figure 8.1(a). The desired angular velocity at the beam tip $\dot{\psi}^*(t, L)$ serves as input to the inverse model (4.28) for generating $\dot{\psi}^*(t, 0)$ at the hub using ibFFC. Figure 8.6 shows the desired angular velocity at the beam tip (blue dashed line) and the calculated angular velocity at the hub (red solid line). If we compare the two lines, in particular before $t = 0.5$s and after $t = 2$s, we can see the pre- and post-actuation input outside the transition interval $T = [0.5\text{s}, 2\text{s}]$. The required motor torque can be calculated using the dynamic model of the drive unit (3.35). In this experiment, the parameters of the PD controller and the friction model are the same as in the rigid case, see Table 8.6. The experiment is carried out for both the tracking and disturbance behaviors[2].

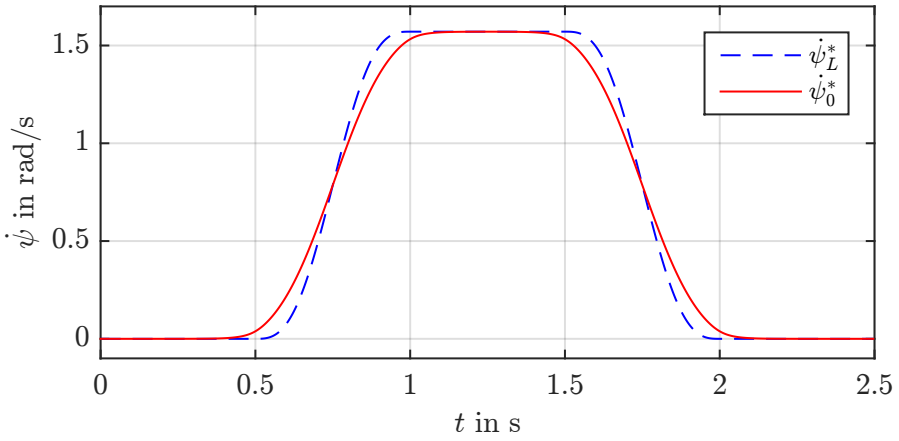
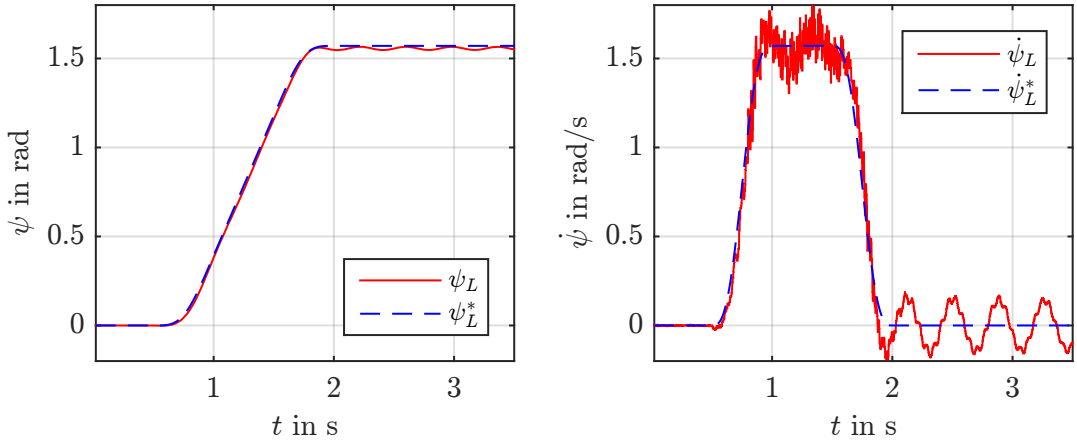


Figure 8.6: Experiment B: Desired and computed trajectories for output and input ($\dot{\psi}_L^*$: desired angular velocity at the beam tip, $\dot{\psi}_0^*$: computed angular velocity at the hub)

[2]See the video of the corresponding experiment under https://youtu.be/yme5FREaSog.

Figure 8.7 shows the tracking behavior of the beam tip with respect to the angle ψ_L and the angular velocity $\dot{\psi}_L$. At the beginning of the movement, the beam tip follows the desired trajectory faster, and the amplitude of the angular velocity in the opposite direction is much smaller compared to Figure 8.3(b). Although the beam tip is still oscillating in the dominant first mode after $t = 2$s, the oscillation amplitude is greatly reduced compared to Experiment A.

Figure 8.8 represents the corresponding torque setpoint τ_M^{set} (blue dashed line) and the actual value τ_M^{act} (red solid line). The range of motor torque is approx. between -20Nm and 50Nm, while Experiment A requires more motor torque – between -40Nm and 60Nm – for the same tracking task, see Figure 8.4. In robotics, feedforward control design generally requires an accurate model to avoid a large tracking error that can lead to saturation in feedback control. In this way, the drive system can be protected and spared.



(a) Beam tip angle

(b) Beam tip angular velocity

Figure 8.7: Experiment B: Tracking behavior of the beam tip angle and angular velocity using ibFFC and PD joint feedback

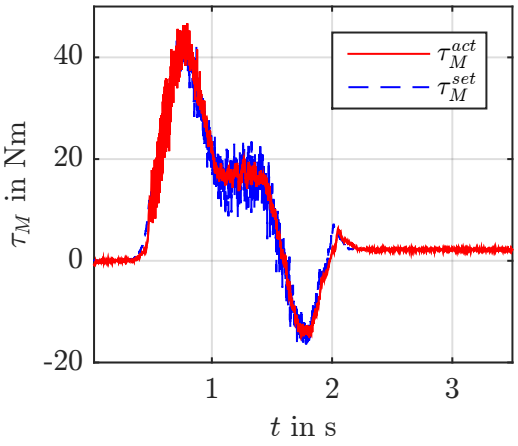
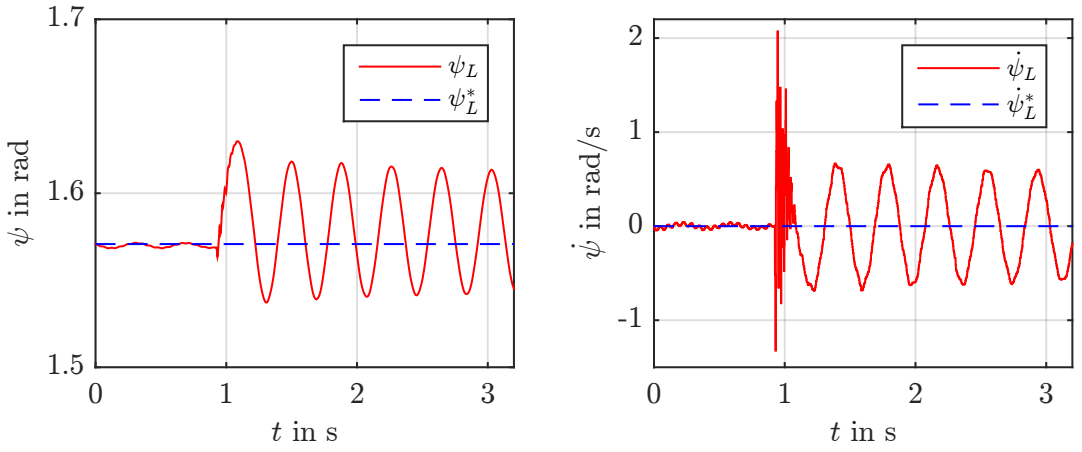


Figure 8.8: Experiment B: Motor torque for tracking the desired trajectory using ibFFC and PD joint feedback

Figure 8.9 shows the disturbance behavior of the beam tip, which is hit with a broomstick after the PTP movement. It is obvious that the PD controller is not able to suppress the oscillations of the beam in the presence of external disturbance.



(a) beam tip angle

(b) beam tip angular velocity

Figure 8.9: Experiment B: Disturbance behavior of the beam tip angle and angular velocity using ibFFC and PD joint feedback

8.5 Experiment C: Observer-Based Trajectory Control

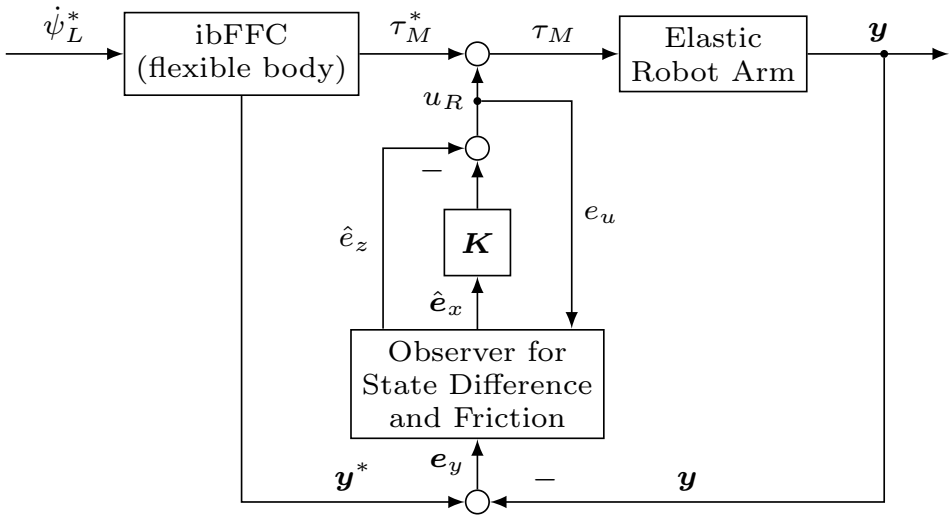


Figure 8.10: Two-degrees-of-freedom regulation: An inversion-based feedforward control (ibFFC) and a state feedback based on the observation of the state difference and the friction torque

Figure 8.10 shows the control scheme with ibFFC and state feedback based on the observation of state difference $\hat{\boldsymbol{e}}_x$ and friction torque $\hat{e}_z$. The state (disturbance) observer is designed using the method described in Section 6.4. In addition to the motor angle and angular velocity, the tip angular velocity, which is measured with an IMU, is fed

back to the controller. In this experiment, we evaluate the quality of this control strategy in terms of the tracking and disturbance behavior[3] of the beam tip.

8.5.1 Parametrization of the LQG Regulator

At first, we parametrize the LQG regulator, which involves the Kalman filter and the Riccati controller. In general, the weighting matrices are selected as identity matrices, whose individual elements are then adjusted, depending on the practical application and the control objective.

Parametrization of the Kalman filter. The goal of a disturbance observer is to reconstruct the unknown state variables and the missing friction torque. For this purpose, the Kalman filter is designed in the modeling state variables. According to the extended difference model (6.40), the weighting matrix $\boldsymbol{Q}_o \in \mathbb{R}^{17\times 17}$ for state vector $\boldsymbol{e}_e$ can be directly decomposed into the submatrices for drive system $\boldsymbol{Q}_o^a \in \mathbb{R}^{4\times 4}$, beam model $\boldsymbol{Q}_o^b \in \mathbb{R}^{12\times 12}$, and disturbance model $Q_o^s \in \mathbb{R}$ as follows

$$\boldsymbol{Q}_o = \begin{bmatrix} \boldsymbol{Q}_o^a & & \\ & \boldsymbol{Q}_o^b & \\ & & Q_o^s \end{bmatrix}. \tag{8.1}$$

In this way, we can roughly determine the weighting factors for each subsystem separately and fine tune them according to their coupling effects. Moreover, the weighting matrix for the measurement $\boldsymbol{S}_o \in \mathbb{R}^{3\times 3}$ is chosen as the identity matrix.

Parametrization of the Riccati controller. In order to suppress the oscillations of the beam, the weighting factors of the Riccati controller can be adjusted in terms of the different eigenmodes. In our configuration, the low-frequency eigenmodes dominate the movement. So, the Riccati controller is designed based on the Jordan normal form of the difference model (6.36)

$$\dot{\boldsymbol{\sigma}}_x = \tilde{\boldsymbol{\Lambda}}_x \boldsymbol{\sigma}_x + \tilde{\boldsymbol{B}} e_u, \qquad \tilde{\boldsymbol{\Lambda}} = \operatorname{diag}\left\{\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n\right\}, \tag{8.2}$$

where $\tilde{\boldsymbol{B}} = \boldsymbol{V}_x^{-1}\boldsymbol{B}$, $\boldsymbol{V}_x \in \mathbb{R}^{n\times n}$ is the matrix of right eigenvectors of $\boldsymbol{A}$, and $\boldsymbol{\sigma}_x = \boldsymbol{V}_x^{-1}\boldsymbol{e}_x$ is the vector of modal coordinates. The cost functional (6.3) is also rewritten in modal coordinates as follows

$$\begin{aligned} J &= \frac{1}{2}\int_0^\infty (\boldsymbol{e}_x^T \boldsymbol{Q}_c \boldsymbol{e}_x + S_c e_u^2) dt \\ &= \frac{1}{2}\int_0^\infty (\boldsymbol{\sigma}_x^T \tilde{\boldsymbol{Q}}_c \boldsymbol{\sigma}_x + S_c e_u^2) dt \end{aligned} \tag{8.3}$$

with new weighting matrix $\tilde{\boldsymbol{Q}}_c \in \mathbb{R}^{16\times 16}$

$$\tilde{\boldsymbol{Q}}_c = \boldsymbol{V}_x^T \boldsymbol{Q}_c \boldsymbol{V}_x = \begin{bmatrix} \tilde{\boldsymbol{Q}}_c^h & \\ & \tilde{\boldsymbol{Q}}_c^l \end{bmatrix}, \tag{8.4}$$

[3] See the video of the corresponding experiment under https://youtu.be/17VVtNAMGlQ.

where $\tilde{\boldsymbol{Q}}_c^l$ and $\tilde{\boldsymbol{Q}}_c^h$ are the weighting matrices for the low-frequency and remaining high-frequency eigenmodes, respectively. The low-frequency eigenmodes – in our case the first and second eigenmodes – are heavily weighted by choosing large values of the corresponding matrix $\tilde{\boldsymbol{Q}}_c^l$. The weighting matrix $\boldsymbol{Q}_c \in \mathbb{R}^{16\times16}$ for the modeling state variables is then calculated by

$$\boldsymbol{Q}_c = \boldsymbol{V}_x^{-T} \tilde{\boldsymbol{Q}}_c \boldsymbol{V}_x^{-1}. \tag{8.5}$$

Note that the weighting factor[4] $S_c \in \mathbb{R}$ for the control input is the same in both forms.

Table 8.7 lists the reference parameters of the LQG regulator for Experiment C. It is worth noting that the disturbance state is weighted more heavily in the Kalman filter than the motor and beam state variables. Moreover, the first and second eigenmodes are dominant in the control design, while the other high-frequency eigenmodes are only weakly weighted. The weighting matrices S_c and $\boldsymbol{S}_o$ are defined as the identity matrix. Based on the design procedure as shown in Figure 6.4, the LQG regulator can be determined directly in MATLAB.

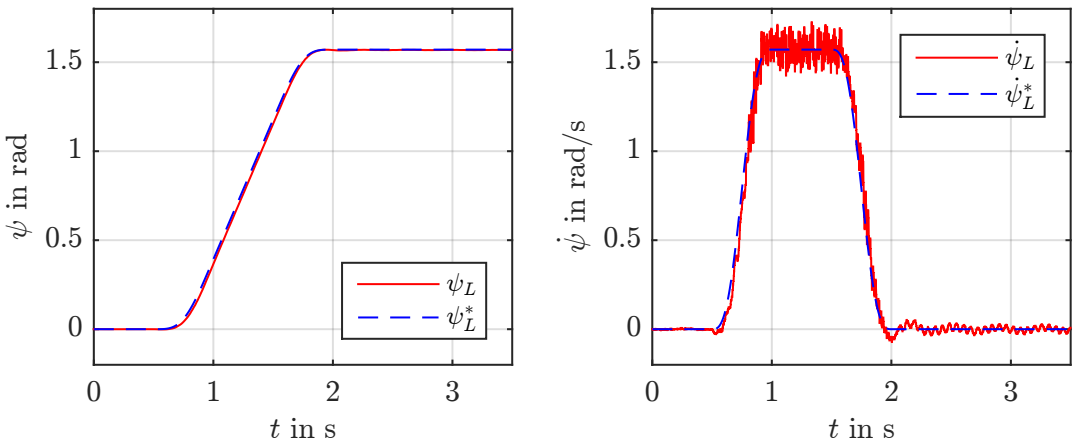
LQG	Parameter	Value
Kalman Filter	$\boldsymbol{Q}_o^a$	$\boldsymbol{0}_4$
	$\boldsymbol{Q}_o^b$	$10^{-4} \cdot \boldsymbol{I}_{12}$
	Q_o^s	$5 \cdot 10^2$
	$\boldsymbol{S}_o$	$\boldsymbol{I}_3$
Riccati Controller	$\tilde{\boldsymbol{Q}}_c^h$	$5 \cdot 10^{-3} \cdot \boldsymbol{I}_{12}$
	$\tilde{\boldsymbol{Q}}_c^l$	$3 \cdot 10^3 \cdot \boldsymbol{I}_4$
	S_c	1

Table 8.7: Experiment C: Parameters of the LQG regulator. $\boldsymbol{Q}_o^a$, $\boldsymbol{Q}_o^b$, and Q_o^s: Observer weighting matrices for the (modeling) state variables of the drive (a) and the beam model (b), as well as the disturbance (s), respectively. $\tilde{\boldsymbol{Q}}_c^l$ and $\tilde{\boldsymbol{Q}}_c^h$: Controller weighting matrices for the first and second eigenmodes (l) and the remaining high-frequency eigenmodes (h), respectively.

8.5.2 Experimental Results

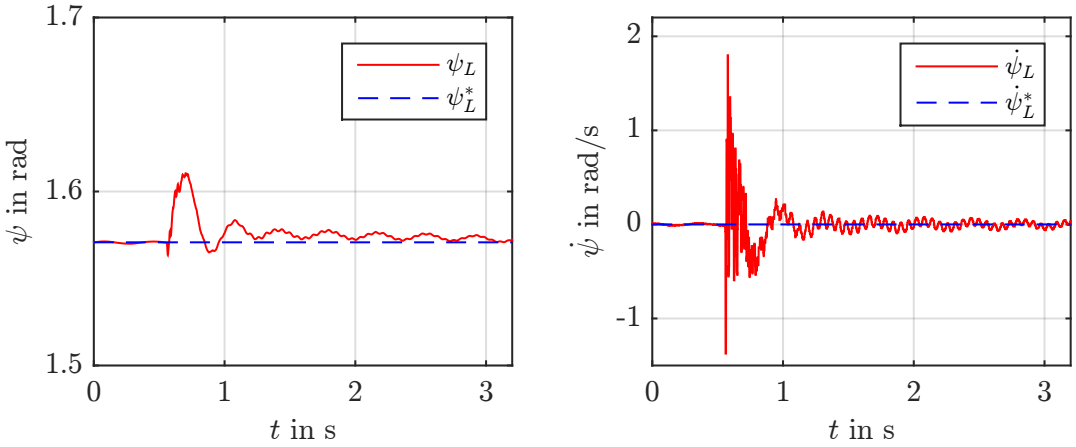
Figures 8.11 and 8.12 illustrate the quality of the control strategy outlined in Figure 8.10. The beam tip perfectly follows the desired trajectory, and the beam oscillation is quickly suppressed in the dominant first mode in response to the external disturbance. Residual oscillations with two distinguished frequencies are visible in both the angle and velocity signals. These can be a consequence of the motor's stiction torque, which must be overcome for movements close to the rest position, and which is not captured in the simple friction model.

[4] Because of $e_u \in \mathbb{R}$, S_c is a scalar.



(a) Beam tip angle (b) Beam tip angular velocity

Figure 8.11: Experiment C: Tracking behavior of the beam tip angle and angular velocity using ibFFC and observer-based state difference feedback



(a) Beam tip angle (b) Beam tip angular velocity

Figure 8.12: Experiment C: Disturbance behavior of the beam tip angle and angular velocity using ibFFC and observer-based state difference feedback

8.6 Experiment D: Observer-Based Trajectory Control for Delayed Systems

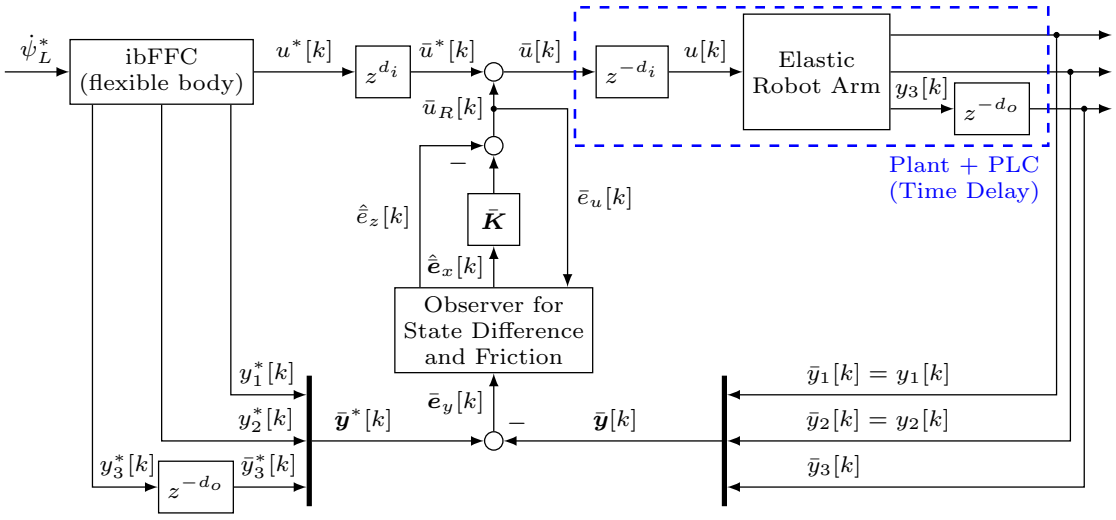


Figure 8.13: Two-degrees-of-freedom regulation: An inversion-based feedforward control (ibFFC) and an observer-based state difference feedback for the time-delay system

As in Chapter 7, the cyclic and hierarchical task execution of the PLC causes time delays of different magnitudes. Therefore, in our case, we consider the input delay for the control torque and the output delay for the measurement from the IMU when modeling. Figure 8.13 represents the two-degrees-of-freedom control scheme for a time-delay system, which consists of the ibFFC and the state feedback based on the estimated state difference and friction torque. The time delay models are used for both the feedforward and feedback control design.

Considering the input delay $u[k] = \bar{u}[k - d_i]$ in the feedforward control, the nominal control torque $u^*[k]$ is replaced by $\bar{u}^*[k] = u^*[k + d_i]$. Thus, the control input $u[k]$ can be actually set to the desired value $u^*[k]$ at the time k

$$u[k] = \bar{u}[k - d_i] \stackrel{!}{=} \bar{u}^*[k - d_i] = u^*[k].$$

Compared to the measured signals from the encoder, there is a relatively large delay in the measurement from the IMU. Similarly, this output delay $\bar{y}_3[k] = y_3[k - d_o]$ should also be considered in feedforward control, which calculates the desired output trajectory $\bar{y}_3^*[k]$

$$\bar{y}_3^* = y_3^*[k - d_o]$$

instead of the undelayed value $y_3^*[k]$. Thus, the desired and measured angular velocity of the beam tip at the same time k are evaluated.

Moreover, the LQG regulator is designed based on the time-delay plant model, which finally results in the control law

$$\bar{u}[k] = \bar{u}^*[k] + \bar{u}_R[k],$$

where $\bar{u}[k]$ is calculated in the PLC plane, while $u[k]$ is executed in the servo amplifier plane for the motor torque control loop, see Figure 2.11.

8.6.1 Parametrization of the LQG Regulator

The weighting matrices of both the Kalman filter and the Riccati controller for a time-delay system can be chosen according to the concept in Experiment C, see Section 8.5.1.

Parametrization of the Kalman filter. The observer design is based on the difference model (7.26) in the modeling coordinates to estimate the state difference vector $\bar{\boldsymbol{e}}_e = \left[\bar{\boldsymbol{e}}_x^T, \quad \bar{e}_s\right]^T$. Thus, the weighting matrix $\bar{\boldsymbol{Q}}_o \in \mathbb{R}^{24\times 24}$ can be divided into two parts as follows

$$\bar{\boldsymbol{Q}}_o = \begin{bmatrix} \bar{\boldsymbol{Q}}_o^x & \\ & \bar{Q}_o^s \end{bmatrix}, \tag{8.6}$$

where $\bar{\boldsymbol{Q}}_o^x \in \mathbb{R}^{23\times 23}$ stands for the state vector $\bar{\boldsymbol{e}}_x$, and $\bar{Q}_o^s \in \mathbb{R}$ stands for the disturbance state $\bar{e}_s$. In our case, the weighting matrix $\bar{\boldsymbol{S}}_o$ is the identity matrix.

Parametrization of the Riccati controller. Since the state difference vector $\bar{\boldsymbol{e}}_x = \left[\boldsymbol{e}_x^T, \quad \boldsymbol{e}_\xi^T, \quad \boldsymbol{e}_\beta^T\right]^T$ can extend the plant state vector $\boldsymbol{e}_x$ with the delay difference vectors $\boldsymbol{e}_\xi$ and $\boldsymbol{e}_\beta$, the cost functional for the Riccati controller is extended by the delay state vector with the corresponding weighting matrix $\boldsymbol{Q}^{\xi\beta} \in \mathbb{R}^{7\times 7}$ as follows

$$\bar{J} = \frac{1}{2}\int_0^\infty (\boldsymbol{e}_x^T \boldsymbol{Q}_c \boldsymbol{e}_x + \begin{bmatrix} \boldsymbol{e}_\xi \\ \boldsymbol{e}_\beta \end{bmatrix}^T \boldsymbol{Q}_c^{\xi\beta} \begin{bmatrix} \boldsymbol{e}_\xi \\ \boldsymbol{e}_\beta \end{bmatrix} + \bar{S}_c \bar{e}_u^2) dt. \tag{8.7}$$

To parameterize $\boldsymbol{Q}_c$, we use the Jordan normal form (8.2) for $\boldsymbol{e}_x$. The low-frequency eigenmodes are considered for the control design mainly by choosing large $\bar{\boldsymbol{Q}}_c^l$, see Equation (8.4). The weighting matrix $\boldsymbol{Q}_c$ corresponding to the modeling state variables is then determined by the back transformation (8.5). Moreover, $\bar{S}_c$ is chosen as 1 to weight the input difference. The reference parameters of the LQG regulator are listed in Table 8.8.

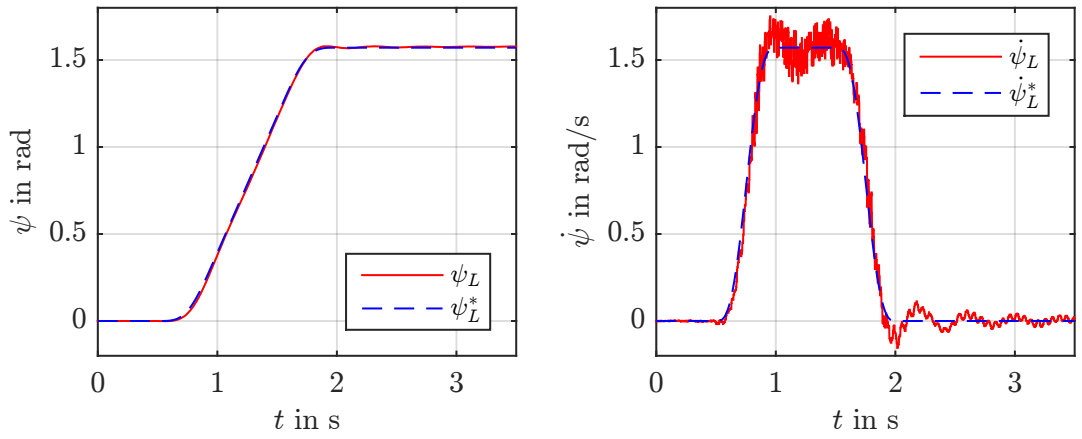
	Parameter	Value
Time Delay	T	1.2ms
	d_i	4
	d_o	3
Kalman Filter	$\bar{\boldsymbol{Q}}_o^x$	$10^{-4} \cdot I_{23}$
	$\bar{Q}_o^s$	10^4
	$\bar{\boldsymbol{S}}_o$	$\boldsymbol{I}_3$
Riccati Controller	$\tilde{\boldsymbol{Q}}_c^h$	$10^{-2} \cdot \boldsymbol{I}_{12}$
	$\tilde{\boldsymbol{Q}}_c^l$	$5 \cdot 10^3 \cdot \boldsymbol{I}_4$
	$\boldsymbol{Q}_c^{\xi\beta}$	$\boldsymbol{0}_7$
	$\bar{S}_c$	1

Table 8.8: Experiment D: Parameters of the LQG regulator for the time-delay system. $\bar{\boldsymbol{Q}}_o^x$ and $\bar{\boldsymbol{Q}}_o^s$: observer weighting matrices for the state (x) and disturbance (s) variables, respectively. $\tilde{\boldsymbol{Q}}_c^l$, $\tilde{\boldsymbol{Q}}_c^h$, and $\boldsymbol{Q}_c^{\varepsilon\beta}$: controller weighting matrices for the first and second eigenmodes (l), the remaining high-frequency eigenmodes (h) and the time delay model ($\varepsilon\beta$), respectively.

8.6.2 Experimental Results

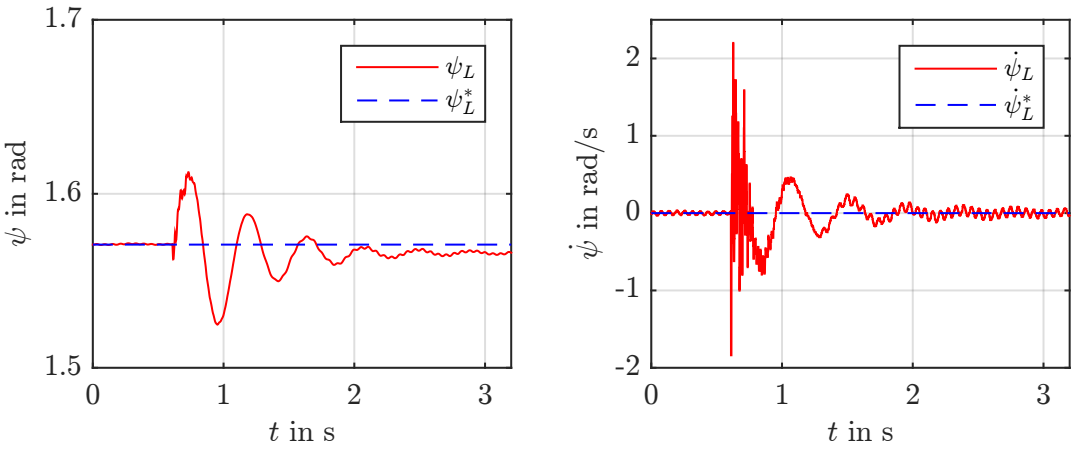
Figures 8.14 and 8.15 show the tracking and disturbance behavior[5] of the beam tip with respect to the angle and the angular velocity, respectively. In the tracking behavior, the beam tip responds very quickly at the beginning of the movement and moves almost directly in the positive direction. After deacceleration, the angular velocity is also quickly reduced to zero. However, the beam tip oscillates more strongly in the uniform movement and at the end of the movement compared to Experiment C. In this case, the state variables of the drive system and the beam model, as well as of the time delay model are weighted identically in the observer, while the weighting matrices for the LQG regulator are larger than the values of the Experiment C. Thus, the controller compensates the tracking error faster and is sensitive to the external disturbance.

[5]See the video of the corresponding experiment under https://youtu.be/fZMYw9lO2_M.



(a) Beam tip angle (b) Beam tip angular velocity

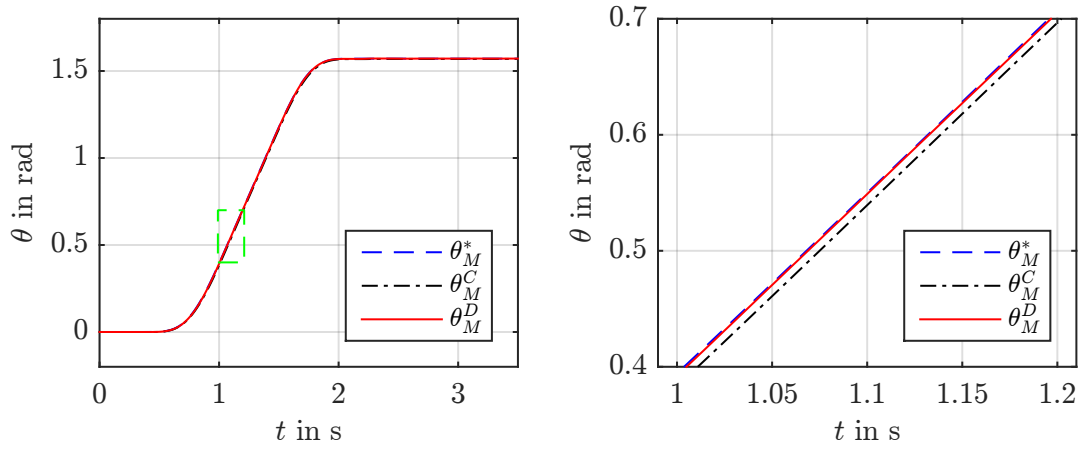
Figure 8.14: Experiment D: Tracking behavior of the beam tip angle and angular velocity using ibFFC and observer-based state difference feedback based on the time-delay model



(a) Beam tip angle (b) Beam tip angular velocity

Figure 8.15: Experiment D: Disturbance behavior of the beam tip angle and angular velocity using ibFFC and observer-based state difference feedback based on the time-delay model

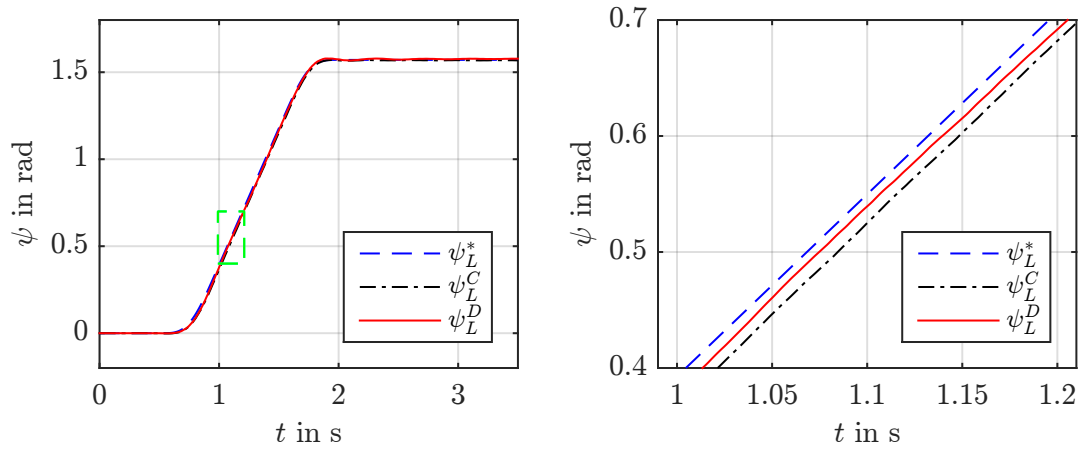
Figures 8.16 and 8.17 compare the tracking behaviors related to the motor angle and the beam tip angle and show the zoom of the chosen transition interval (green rectangle) $t = [1\text{s}, 1.2\text{s}]$. Compared with the experimental results of Experiment C, we obtain a shorter time delay between the measured and desired value in both Figures 8.16(b) and 8.17(b). It is observed that the measured motor angle θ_M^D matches almost the desired value θ_M^* (from about 0.006s to 0.0012s), while the transmission of the beam tip angle is reduced from 0.015s (see ψ_L^C) to 0.006s (see ψ_L^D). Some remaining time delays still exist because CAN bus is asynchronous communication where the time interval of the transmission is random and cannot be predicted.



(a) Motor angle

(b) Zoom of the motor angle

Figure 8.16: Comparison of the motor angle (θ_M^*: desired value, θ_M^C: measurement from Experiment C, θ_M^D: Measurement from Experiment D)



(a) Beam tip angle

(b) Zoom of the beam tip angle

Figure 8.17: Comparison of the beam tip angle (φ_L^*: desired value, φ_L^C: measurement from Experiment C, φ_L^D: Measurement from Experiment D)

Chapter 9

Conclusions

In this thesis, a holistic approach for the trajectory control design of an elastic robot arm based on the PH representation has been presented. The flexible arm is driven horizontally by a brushless DC motor with a Harmonic Drive gear. The drive system was directly described by a finite-dimensional PH model, while the flexible beam model was discretized and reduced using structure-preserving numerical methods that preserve certain (geometric or structural) properties in the lumped parameter model. The two subsystems were coupled by interconnection to form an overall PH system. The structure of the obtained finite-dimensional beam model has been shown to be suitable for the design of inversion-based feedforward control. An LQG regulator, which incorporates a simple estimate of the disturbance (friction), complemented the two-degrees-of-freedom controller structure to suppress the residual oscillations and achieve a high-fidelity tracking of the reference trajectory. The control quality of the proposed approach was demonstrated by experimental validation and compared with other control strategies.

This dissertation has shown that the energy-based modeling and the use of physically consistent numerical methods can contribute greatly to the successful design of control systems, especially with respect to a structured and modular approach and the task of motion planning. Furthermore, the results show (once again) the high utility of the linear state space tools in their application within the – also modular – two-degrees-of-freedom control structure.

The results of this work were generated on a test bench controlled by an industrial PLC, which was designed and constructed in the frame of this thesis. The cyclic and hierarchical task execution of the PLC generates time delays of different magnitude (multiples of the lowest sampling time). This can lead to a deterioration of the control performance or even to the instability of the controlled system. Therefore, the proposed approach was extended to account for the time delays in the discrete-time domain, using a simple Euler integration scheme for the discrete-time implementation of the continuously derived control.

A topic for future work on this flexible robot arm configuration, in addition to structure preservation in the spatial discretization, is the application of structure-preserving time integration schemes for discrete-time control design, such as in [33], [41], [82] to the PLC controlled flexible manipulator.

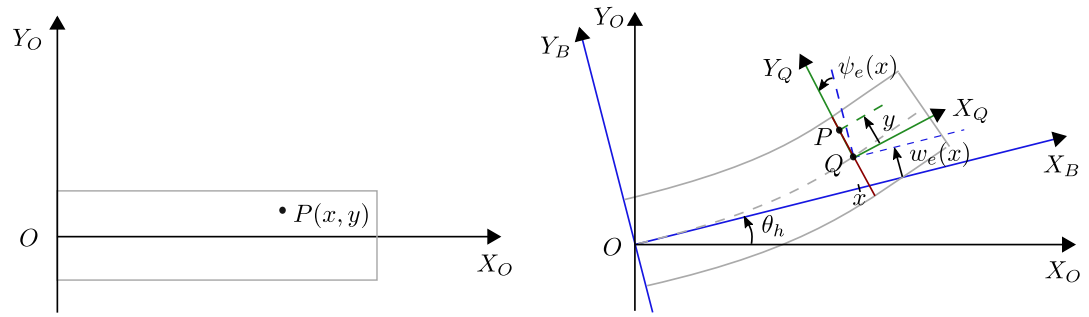
The present results serve as a reference for the systematic approach from modeling to control design and implementation in a structured energy-based framework. The flexible

three-link setup depicted in Figure 2.1 is an example for a real world lightweight robot arm. For this system class, the seamless integration of modern control design in modular automation systems poses numerous questions for ongoing and future works.

Appendix A

Dynamical Modeling of the Timoshenko Beam

In this appendix, we first derive the nonlinear motion equations of a flexible beam using Hamilton's principle. Then, some simplifications are taken into account according to the small deformation assumption. Furthermore, it is assumed that the motor torque τ_M acts directly on the hub[1], i.e., $r_h = 0$.



(a) Undeformed beam

(b) Deformed beam

Figure A.1: On the coordinates of a point $P(x, y)$ on the cross section of the beam under deformation ($w_e(x)$: deformation, $\psi_e(x)$: rotation)

Coordinate system. The beam moves in the horizontal plane and the gravity is neglected. Therefore, we define some 2D Cartesian coordinate systems[2] to express the coordinate transformations of the movement, see Figure A.1.

- Inertial frame of reference $X_O Y_O$: It describes the absolute movement in the inertial coordinate system.
- Floating frame of reference $X_B Y_B$: The deformation only occurs in relation to this coordinate system. Using the floating frame of reference approach, flexible bodies can be decomposed into rigid body motion and elastic deformation.

[1] Here, the dynamics of the gear motor is neglected.
[2] In our case, the manipulator configuration follows the Denavit-Hartenberg convention.

- Coordinate system of a point X_QY_Q: In order to describe the motion of an arbitrary point of the beam, we define a point-fixed coordinate system based on the deformed beam's cross section.

Rotation matrix. Based on Figure A.1, the relative motion between X_OY_O and X_BY_B, and between X_BY_B and X_QY_Q can be expressed by the rotation matrices

$$\boldsymbol{A}_{OB} = \begin{bmatrix} \cos(\theta_h) & -\sin(\theta_h) \\ \sin(\theta_h) & \cos(\theta_h) \end{bmatrix}$$
$$\boldsymbol{A}_{BQ} = \begin{bmatrix} \cos(\psi_e) & -\sin(\psi_e) \\ \sin(\psi_e) & \cos(\psi_e) \end{bmatrix},$$

and a translation ${}_B\boldsymbol{r}_Q$ of the origin of X_QY_Q in the floating frame X_BY_B

$${}_B\boldsymbol{r}_Q = \begin{bmatrix} x \\ w_e \end{bmatrix}.$$

θ_h describes the rigid body rotation, and ψ_e denotes the rotation of the deformed cross section.

Absolute position, velocity of a point P. We consider an arbitrary point $P(x, y)$ on the deformed beam's cross section with respect to X_QY_Q:

$${}_Q\mathbf{r}_P = \begin{bmatrix} 0 \\ y \end{bmatrix}. \tag{A.1}$$

It is assumed that the beam length is not changed after the deformation. Then, P can be transformed in the coordinate system X_BY_B with the rotation matrix $\boldsymbol{A}_{BQ}$ and the translation vector ${}_B\boldsymbol{r}_Q$:

$$\begin{aligned} {}_B\mathbf{r}_P &= {}_B\mathbf{r}_Q + \boldsymbol{A}_{BQ}\ {}_Q\mathbf{r}_P \\ &= \begin{bmatrix} x \\ w_e \end{bmatrix} + \begin{bmatrix} \cos(\psi_e) & -\sin(\psi_e) \\ \sin(\psi_e) & \cos(\psi_e) \end{bmatrix} \begin{bmatrix} 0 \\ y \end{bmatrix} \\ &= \begin{bmatrix} x - y\ \sin(\psi_e) \\ w_e + y\ \cos(\psi_e) \end{bmatrix}. \end{aligned} \tag{A.2}$$

Since the absolute velocity is originally defined in the inertial frame of reference, we describe the position vector $\boldsymbol{r}_P$ of P after the deformation in $X_O Y_O$ as follows

$$\begin{aligned} {}_O\mathbf{r}_P &= \boldsymbol{A}_{OB}\ {}_B\mathbf{r}_P \\ &= \begin{bmatrix} \cos(\theta_h) & -\sin(\theta_h) \\ \sin(\theta_h) & \cos(\theta_h) \end{bmatrix} \begin{bmatrix} x - y\,\sin(\psi_e) \\ w_e + y\,\cos(\psi_e) \end{bmatrix} \\ &= \begin{bmatrix} x\,\cos(\theta_h) - w_e\,\sin(\theta_h) - y\,\sin(\theta_h + \psi_e) \\ x\,\sin(\theta_h) + w_e\,\cos(\theta_h) + y\,\cos(\theta_h + \psi_e) \end{bmatrix}. \end{aligned} \tag{A.3}$$

The absolute velocity ${}_O\mathbf{v}_P = {}_O\dot{\mathbf{r}}_P$ results

$$\begin{aligned} {}_O\mathbf{v}_P &= \dot{\boldsymbol{A}}_{OB}\ {}_B\mathrm{r}_P + \boldsymbol{A}_{OB}\ {}_B\dot{\mathbf{r}}_P \\ &= \begin{bmatrix} -x\,\sin(\theta_h)\,\dot{\theta}_h - \dot{w}_e\,\sin(\theta_h) - w_e\,\cos(\theta_h)\,\dot{\theta}_h - y\,\cos(\theta_h + \psi_e)\,(\dot{\theta}_h + \dot{\psi}_e) \\ x\,\cos(\theta_h)\,\dot{\theta}_h + \dot{w}_e\,\cos(\theta_h) - w_e\,\sin(\theta_h)\,\dot{\theta}_h - y\,\sin(\theta_h + \psi_e)\,(\dot{\theta}_h + \dot{\psi}_e) \end{bmatrix}. \end{aligned} \tag{A.4}$$

Energy. The kinetic energy of the rotating beam is defined as

$$\begin{aligned} T_b = &\ \frac{1}{2}\int_0^L \iint_A \rho_v\, {}_O\mathbf{v}_P^T\, {}_O\mathbf{v}_P\, dA\, dx \\ = &\ \frac{1}{2}\int_0^L \rho_v \Big\{ \iint_A [(x\,\dot{\theta}_h + \dot{w}_e)^2 + (w_e\,\dot{\theta}_h)^2 + y^2\,(\dot{\theta}_h + \dot{\psi}_e)^2]\, dA \\ &- \iint_A 2\,y\,\sin(\psi_e)[x\,(\dot{\theta}_h + \dot{\psi}_e) + \dot{w}_e\,(\dot{\theta}_h + \dot{\psi}_e)] dA \Big\}. \end{aligned} \tag{A.5}$$

Using the following relations

$$\iint_A y dA = 0, \qquad \iint_A y^2 dA = I_Z, \qquad I_\rho = \rho_v I_Z, \quad \rho = \rho_v A,$$

Equation (A.5) is simplified to

$$T_b = \frac{1}{2}\int_0^L [\rho(x\,\dot{\theta}_h + \dot{w}_e)^2 + \rho(w_e\dot{\theta}_h)^2 + I_\rho(\dot{\theta}_h + \dot{\psi}_e)^2] dx, \tag{A.6}$$

where I_z and ρ denote the area moment of inertia and the linear density, respectively. The potential energy of the beam is defined as

$$P_b = \frac{1}{2}\int_0^L [EI(\psi_e')^2 + \kappa GA(\psi_e - w_e')^2] dx. \tag{A.7}$$

The total energy of the flexible manipulator is composed of the kinetic energy T considering the parts of the beam T_b and the hub T_h with

$$\begin{aligned} T = &\quad T_b + T_h \\ = &\quad \frac{1}{2}\int_0^L [\rho(x\,\dot{\theta}_h + \dot{w}_e)^2 + \rho(w_e\dot{\theta}_h)^2 + I_\rho(\dot{\theta}_h + \dot{\psi}_e)^2]dx + \frac{1}{2}I_h\dot{\theta}_h^2, \end{aligned} \tag{A.8}$$

and the potential energy $P = P_b$. Hamilton's principle, i.e., stationarity of the action functional reads

$$\int_{t_0}^{t_f} (\delta T + \delta W - \delta P)dt = 0, \tag{A.9}$$

where $\delta W = \tau_M \delta\theta_h$ denotes the variation of the external work.

In the following paragraphs, the variational calculation of Equation (A.9) is separately performed with respect to the variables θ_h, w_e, and ψ_e.

Variation with respect to variable $\boldsymbol{\theta_h}$. The variation[3] of kinetic energy (A.8) relating to θ_h is written as

$$\delta_{\theta_h} T = I_h\dot{\theta}_h\delta\dot{\theta}_h + \int_0^L [\rho x(x\dot{\theta}_h + \dot{w}_e) + \rho w_e^2\dot{\theta}_h + I_\rho(\dot{\theta}_h + \dot{\psi}_e)]\delta\dot{\theta}_h dx. \tag{A.10}$$

Because the potential energy (A.7) does not depend on θ_h, its variation is

$$\delta_{\theta_h} P = 0. \tag{A.11}$$

The virtual work is given by

$$\delta_{\theta_h} W = \tau_M \delta\theta_h. \tag{A.12}$$

Substituting Equations (A.10), (A.11), and (A.12) into Hamilton's principle (A.9) results

$$\begin{aligned} &\int_{t_0}^{t_f} (\delta_{\theta_h} T + \delta_{\theta_h} W - \delta_{\theta_h} P)dt \\ &= \int_{t_0}^{t_f} I_h\dot{\theta}_h\delta\dot{\theta}_h dt + \int_{t_0}^{t_f}\int_0^L [\rho x(x\dot{\theta}_h + \dot{w}_e) + \rho w_e^2\dot{\theta}_h + I_\rho(\dot{\theta}_h + \dot{\psi}_e)\delta\dot{\theta}_h dx dt] \\ &\quad + \int_{t_0}^{t_f} \tau_M\delta\theta_h dt. \end{aligned} \tag{A.13}$$

We perform the integration by parts for the first and second integral terms, and consider that $\delta\theta_h$ vanishes at $t = t_0$ and $t = t_f$. Then, it is simplified to

$$\begin{aligned} &\delta_{\theta_h}\int_{t_0}^{t_f} (T + W - P)dt \quad \stackrel{!}{=} \quad 0 \\ &= -\int_{t_0}^{t_f} \underbrace{\{I_h\ddot{\theta}_h + \int_0^L [\rho(x^2 + w_e^2)\ddot{\theta}_h + 2\rho w_e\dot{w}_e\dot{\theta}_h + \rho x\ddot{w}_e + I_\rho(\ddot{\theta}_h + \ddot{\psi}_e)]dx - \tau_M\}}_{=①} \delta\theta_h dt. \end{aligned} \tag{A.14}$$

[3] Notice that $\delta_{\theta_h} T = \delta_{\dot{\theta}_h} T$, and $\delta\dot{\theta}_h = \frac{d}{dt}(\delta\theta_h)$.

Variation with respect to variable w_e. Now, the variational derivation of the kinetic energy (A.8) with respect to w_e is represented as

$$\delta_{w_e} T = \int_0^L [\rho(x\dot{\theta}_h + \dot{w}_e)\delta\dot{w}_e + \rho w_e \dot{\theta}_h^2 \delta w_e] dx. \tag{A.15}$$

Using spatial integration by parts, the variation of the potential energy (A.7) related to w_e is derived by

$$\begin{aligned} \delta_{w_e} P &= \int_0^L -kGA(\psi_e - w_e')\delta w_e' dx \\ &= -[kGA(\psi_e - w_e')\delta w_e]_0^L + \int_0^L kGA(\psi_e' - w_e'')\delta w_e dx. \end{aligned} \tag{A.16}$$

Because the external work is not the function of w_e, the virtual work is

$$\delta_{w_e} W = 0. \tag{A.17}$$

Hamilton's principle corresponding to w_e is calculated by

$$\begin{aligned} &\int_{t_0}^{t_f} (\delta_{w_e} T + \delta_{w_e} W - \delta_{w_e} P) dt \\ &= \int_{t_0}^{t_f} \int_0^L [\rho(x\dot{\theta}_h + \dot{w}_e)\delta\dot{w}_e + \rho w_e \dot{\theta}_h^2 \delta w_e - kGA(\psi_e' - w_e'')\delta w_e] dx dt \\ &\quad + \int_{t_0}^{t_f} [kGA(\psi_e - w_e')\delta w_e]_0^L dt. \end{aligned} \tag{A.18}$$

Applying the integration by parts for the first and second integral terms, it reads

$$\begin{aligned} &\delta_{w_e} \int_{t_0}^{t_f} (T + W - P) dt \quad \stackrel{!}{=} \quad 0 \\ &= \int_{t_0}^{t_f} \int_0^L \{ \underbrace{-[\rho(x\ddot{\theta}_h + \ddot{w}_e - w_e\dot{\theta}_h^2) + \kappa GA(\psi_e' - w_e'')]}_{=②} \delta w_e \} dx dt \\ &\quad + \int_{t_0}^{t_f} \underbrace{[\kappa GA(\psi_e - w_e')\delta w_e]_0^L}_{=*} dt \end{aligned} \tag{A.19}$$

with $\delta w_e = 0$ at $t = t_0,\, t_f$.

Variation with respect to variable ψ_e. The variation of the kinetic energy (A.8) corresponding to ψ_e is given by

$$\delta_{\psi_e} T = \int_0^L I_\rho(\dot{\theta}_h + \dot{\psi}_e)\delta\dot{\psi}_e dx. \tag{A.20}$$

Using the spatial integration by parts, the variation of the potential energy (A.7) is derived by

$$\begin{aligned} \delta_{\psi_e} P &= \int_0^L EI\psi_e'\delta\psi_e' dx + \int_0^L kGA(\psi_e - w_e')\delta\psi_e dx \\ &= [EI\psi_e'\delta\psi_e]_0^L - \int_0^L [EI\psi_e'' - kGA(\psi_e - w_e')]\delta\psi_e dx. \end{aligned} \tag{A.21}$$

Similarly, there is no virtual work relating to ψ_e

$$\delta_{\psi_e} W = 0. \tag{A.22}$$

Substituting the above equations into Hamilton's principle (A.9), it becomes

$$\begin{aligned}
&\int_{t_0}^{t_f} (\delta_{\psi_e} T + \delta_{\psi_e} W - \delta_{\psi_e} P) dt \\
&= \int_{t_0}^{t_f} \int_0^L \{ I_\rho (\dot{\theta}_h + \dot{\psi}_e) \delta \dot{\psi}_e + [EI \psi_e'' - kGA(\psi_e - w_e')] \delta \psi_e \} dx dt \\
&\quad - \int_{t_0}^{t_f} [EI \psi_e' \delta \psi_e]_0^L dt.
\end{aligned} \tag{A.23}$$

Applying the integration by parts for the first integral term with $\delta\psi_e = 0$ at $t = t_0$ and $t = t_f$, we obtain

$$\begin{aligned}
&\delta_{\psi_e} \int_{t_0}^{t_f} (T + W - P) dt \quad \stackrel{!}{=} \quad 0 \\
&= - \int_{t_0}^{t_f} \underbrace{[EI \psi_e' \delta \psi_e]_0^L}_{=**} dt + \int_{t_0}^{t_f} \int_0^L \{ \underbrace{-I_\rho (\ddot{\theta}_h + \ddot{\psi}_e) + [EI \psi_e'' - \kappa GA(\psi_e - w_e')]}_{=③} \} \delta \psi_e dx dt.
\end{aligned} \tag{A.24}$$

Nonlinear Timoshenko beam equations. The boundary conditions of the clamped-free beam with respect to the deformation variables are:

$$\begin{aligned}
\text{Clamped at } x = 0: \qquad w_e(0) &= 0 \\
\psi_e(0) &= 0 \\
\text{Free at } x = L: \qquad \kappa GA(\psi_e(L) - w_e'(L)) &= 0 \\
EI \psi_e'(t, L) &= 0 \quad .
\end{aligned} \tag{A.25}$$

Substituting (A.25) into the terms $*$, $**$, they vanish

$$\begin{aligned}
* &= 0 \\
** &= 0.
\end{aligned}$$

In order to satisfy conditions (A.14), (A.19), and (A.24), terms ①, ②, and ③ must be null. Consequently, the dynamics of the flexible manipulator is rewritten as

$$\rho \left(x \ddot{\theta}_h + \ddot{w}_e - \underline{w_e \dot{\theta}_h^2} \right) + \kappa GA \left(\psi_e' - w_e'' \right) = 0 \tag{A.26a}$$

$$I_\rho \left(\ddot{\theta}_h + \ddot{\psi}_e \right) - EI \, \psi_e'' + \kappa GA \left(\psi_e - w_e' \right) = 0 \tag{A.26b}$$

$$I_h \ddot{\theta}_h + \int_0^L [\rho x (x \ddot{\theta}_h + \ddot{w}_e) + \rho (\underline{2 w_e \dot{w}_e \dot{\theta}_h} + \underline{w_e^2 \ddot{\theta}_h}) + I_\rho (\ddot{\theta}_h + \ddot{\psi}_e)] dx = \tau_M. \tag{A.26c}$$

It is remarkable that the first two equations characterize the dynamical motion of the beam, which consists of the rigid body rotation and the deformations, while the last one describes the dynamics of the hub with reaction effects from the deformed beam.

Linearized Timoshenko beam model. In many engineering applications, as in our case, only small deformations occur compared to a reference configuration, so that nonlinear beam model (A.26) can be linearized, see e.g. [22], [92]. Under the small deformation assumption, the deformation terms in (A.26), i.e., the higher-order terms and the higher-order product, can be ignored[4]. Here, $w_e\dot{\theta}_h^2 \approx 0$, $w_e^2\ddot{\theta}_h \approx 0$ and $2w_e\dot{w}_e\dot{\theta}_h \approx 0$.

Defining the global configuration variables

$$\begin{aligned} w &= w_e + x\theta_h \\ \psi &= \psi_e + \theta_h, \end{aligned}$$

and inserting them into Equation (A.26), the nonlinear Timoshenko beam can be linearized as follows

$$\rho\,\ddot{w} + \kappa GA\,(\psi' - w'') = 0 \tag{A.27a}$$

$$I_\rho\,\ddot{\psi} - EI\,\psi'' + \kappa GA\,(\psi - w') = 0, \tag{A.27b}$$

$$I_h\ddot{\theta}_h + \underbrace{\int_0^L (\rho x\ddot{w} + I_\rho\ddot{\psi})dx}_{=-EI\psi'(0)} = \tau_M, \tag{A.27c}$$

where Equation (A.27c) is derived using integration by parts with the following boundary conditions:

$$\begin{aligned} x = 0\text{:} \qquad w(0) &= 0 \\ \psi(0) &= \theta_h \\ x = L\text{:} \qquad \kappa GA(\psi(L) - w'(L)) &= 0 \\ EI\psi'(L) &= 0 \quad . \end{aligned} \tag{A.28}$$

Note that the linearized equation (A.27) describes a planar motion of the Timoshenko beam, which is dependent on both time and space. In this context, without loss of generality, we define a generalized coordinate $z \in \mathcal{Z} = \{0,\, L\}$ as the spatial coordinate along the beam X_B axis to describe the one-dimensional deformation, i.e., $w(z)$ and $\psi(z)$. Thus, $w' = \partial_z w(z)$ and $\psi' = \partial_z \psi(z)$. This notation is used throughout Chapter 3.

[4]The nonlinear terms play a major role in the dynamics of an elastic arm only if the motion speed is extremely high, see details in [22].

Appendix B

Restricted Bilinear Form and Resulting Conditions

Isotropy condition of the PH beam model. In order to describe the isotropy condition on the bond space (3.50) with respect to the vectors $\boldsymbol{\eta}_i = [\eta_i^{p_t},\, \eta_i^{p_r},\, \eta_i^{\varepsilon t},\, \eta_i^{\varepsilon r}]^T$ and $\boldsymbol{\varphi}_i = [\varphi_i^{p_t},\, \varphi_i^{p_r},\, \varphi_i^{\varepsilon t},\, \varphi_i^{\varepsilon r}]^T$, $i \in \{1,\, 2\}$, we consider the bilinear form[1] between $\mathcal{F}$ and $\mathcal{E}$:

$$\begin{aligned}&\langle\langle(\boldsymbol{\eta}_1,\, \boldsymbol{\eta}_{1\partial}^t,\, \boldsymbol{\eta}_{1\partial}^r,\, \boldsymbol{\varphi}_1,\, \boldsymbol{\varphi}_{1\partial}^t,\, \boldsymbol{\varphi}_{1\partial}^r),\, (\boldsymbol{\eta}_2,\, \boldsymbol{\eta}_{2\partial}^t,\, \boldsymbol{\eta}_{2\partial}^r,\, \boldsymbol{\varphi}_2,\, \boldsymbol{\varphi}_{2\partial}^t,\, \boldsymbol{\varphi}_{2\partial}^r)\rangle\rangle \\ &:= \int_0^L (\boldsymbol{\eta}_1^T\boldsymbol{\varphi}_2 + \boldsymbol{\eta}_2^T\boldsymbol{\varphi}_1)dz + (\boldsymbol{\eta}_{1\partial}^t)^T\boldsymbol{\varphi}_{2\partial}^t + (\boldsymbol{\eta}_{1\partial}^r)^T\boldsymbol{\varphi}_{2\partial}^r + (\boldsymbol{\varphi}_{1\partial}^t)^T\boldsymbol{\eta}_{2\partial}^t + (\boldsymbol{\varphi}_{1\partial}^r)^T\boldsymbol{\eta}_{2\partial}^r.\end{aligned} \tag{B.1}$$

Inserting the undamped beam model (3.44) in Equation (B.1), it becomes

$$\begin{aligned}\text{(B.1)} &= -\int_0^L (\boldsymbol{\eta}_1^T\boldsymbol{P}_1\partial_z\boldsymbol{\eta}_2 + \boldsymbol{\eta}_1^T\boldsymbol{P}_0\boldsymbol{\eta}_2 + \boldsymbol{\eta}_2^T\boldsymbol{P}_1\partial_z\boldsymbol{\eta}_1 + \boldsymbol{\eta}_2^T\boldsymbol{P}_0\boldsymbol{\eta}_1)dz \\ &\quad + (\boldsymbol{\eta}_{1\partial}^t)^T\boldsymbol{\varphi}_{2\partial}^t + (\boldsymbol{\eta}_{1\partial}^r)^T\boldsymbol{\varphi}_{2\partial}^r + (\boldsymbol{\varphi}_{1\partial}^t)^T\boldsymbol{\eta}_{2\partial}^t + (\boldsymbol{\varphi}_{1\partial}^r)^T\boldsymbol{\eta}_{2\partial}^r \\ &= -\int_0^L (\boldsymbol{\eta}_1^T\boldsymbol{P}_1\partial_z\boldsymbol{\eta}_2 + \boldsymbol{\eta}_2^T\boldsymbol{P}_1\partial_z\boldsymbol{\eta}_1)dz \\ &\quad + (\boldsymbol{\eta}_{1\partial}^t)^T\boldsymbol{\varphi}_{2\partial}^t + (\boldsymbol{\eta}_{1\partial}^r)^T\boldsymbol{\varphi}_{2\partial}^r + (\boldsymbol{\varphi}_{1\partial}^t)^T\boldsymbol{\eta}_{2\partial}^t + (\boldsymbol{\varphi}_{1\partial}^r)^T\boldsymbol{\eta}_{2\partial}^r\end{aligned} \tag{B.2}$$

with

$$\boldsymbol{P}_1 = \begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\end{bmatrix}, \qquad \boldsymbol{P}_0 = \begin{bmatrix}0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}.$$

Using integration by parts for Equation (B.2) gives

$$\begin{aligned}\text{(B.1)} &= -\int_0^L d(\boldsymbol{\eta}_1^T\boldsymbol{P}_1\boldsymbol{\eta}_2) + (\boldsymbol{\eta}_{1\partial}^t)^T\boldsymbol{\varphi}_{2\partial}^t + (\boldsymbol{\eta}_{1\partial}^r)^T\boldsymbol{\varphi}_{2\partial}^r + (\boldsymbol{\varphi}_{1\partial}^t)^T\boldsymbol{\eta}_{2\partial}^t + (\boldsymbol{\varphi}_{1\partial}^r)^T\boldsymbol{\eta}_{2\partial}^r \\ &= -\left[\boldsymbol{\eta}_1^T\boldsymbol{P}_1\boldsymbol{\eta}_2\right]\Big|_0^L + (\boldsymbol{\eta}_{1\partial}^t)^T\boldsymbol{\varphi}_{2\partial}^t + (\boldsymbol{\eta}_{1\partial}^r)^T\boldsymbol{\varphi}_{2\partial}^r + (\boldsymbol{\varphi}_{1\partial}^t)^T\boldsymbol{\eta}_{2\partial}^t + (\boldsymbol{\varphi}_{1\partial}^r)^T\boldsymbol{\eta}_{2\partial}^r \\ &= 0.\end{aligned} \tag{B.3}$$

[1] Notation: $\boldsymbol{\eta}_{i\partial}^t = [-\eta_i^{\varepsilon t}(0),\, \eta_i^{\varepsilon t}(L)]$, $\boldsymbol{\eta}_{i\partial}^r = [-\eta_i^{\varepsilon r}(0),\, \eta_i^{\varepsilon r}(L)]$, $\boldsymbol{\varphi}_{i\partial}^t = [\eta_i^{p t}(0),\, \eta_i^{p t}(L)]$, $\boldsymbol{\varphi}_{i\partial}^r = [\eta_i^{p r}(0),\, \eta_i^{p r}(L)]$, $i \in \{1,\, 2\}$.

Approximation of the bilinear form. Based on the discretization, the isotropy condition of the Stokes-Dirac structure can be also approximated on the finite-dimensional space, see e.g. [60].

Inserting (3.56) and (3.70) into (B.1), the following relation is obtained:

$$\begin{aligned}(\text{B.1}) = &(\boldsymbol{e}_1^{p_t})^T \boldsymbol{M} \boldsymbol{f}_2^{p_t} + (\boldsymbol{e}_1^{p_r})^T \boldsymbol{M} \boldsymbol{f}_2^{p_r} + (\boldsymbol{e}_1^{\varepsilon_t})^T \boldsymbol{M} \boldsymbol{f}_2^{\varepsilon_t} + (\boldsymbol{e}_1^{\varepsilon_r})^T \boldsymbol{M} \boldsymbol{f}_2^{\varepsilon_r} \\ &+ (\boldsymbol{f}_1^{p_t})^T \boldsymbol{M}^T \boldsymbol{e}_2^{p_t} + (\boldsymbol{f}_1^{p_r})^T \boldsymbol{M}^T \boldsymbol{e}_2^{p_r} + (\boldsymbol{f}_1^{\varepsilon_t})^T \boldsymbol{M}^T \boldsymbol{e}_2^{\varepsilon_t} + (\boldsymbol{f}_1^{\varepsilon_r})^T \boldsymbol{M}^T \boldsymbol{e}_2^{\varepsilon_r} \\ &+ (\boldsymbol{e}_{1\partial}^{t})^T \boldsymbol{f}_{2\partial}^{t} + (\boldsymbol{e}_{1\partial}^{r})^T \boldsymbol{f}_{2\partial}^{r} + (\boldsymbol{f}_{1\partial}^{t})^T \boldsymbol{e}_{2\partial}^{t} + (\boldsymbol{f}_{1\partial}^{r})^T \boldsymbol{e}_{2\partial}^{r}.\end{aligned} \tag{B.4}$$

It is noteworthy that the expression of the isotropy condition depends on the discrete port variables $\boldsymbol{f}^{\nu}$, $\boldsymbol{e}^{\nu}$, $\nu \in \{p_t, p_r, \varepsilon_t, \varepsilon_r\}$ and the discrete bilinear form (3.69), which appears on the right side of Equation (B.4). Moreover, inserting the relation (3.78) in (B.4), the bilinear form can be expressed only in terms of the discrete effort variables $(\boldsymbol{e}_1^{\mu_i}, \boldsymbol{e}_2^{\nu_i})$, $\mu \in \{p_t, p_r, \varepsilon_t, \varepsilon_r\}$, $\nu \in \{\varepsilon_t, \varepsilon_r, p_t, p_r\}$, $i = 1, 2, 3, 4$ as:

$$\begin{aligned}(\text{B.4}) = &(\boldsymbol{e}_1^{p_t})^T \boldsymbol{M}(-\boldsymbol{D}\boldsymbol{e}_2^{\varepsilon_t}) + (\boldsymbol{e}_1^{p_r})^T \boldsymbol{M}(-\boldsymbol{D}\boldsymbol{e}_2^{\varepsilon_r} - \boldsymbol{S}^{-1}\boldsymbol{M}^T \boldsymbol{e}_2^{\varepsilon_t}) \\ &+ (\boldsymbol{e}_1^{\varepsilon_t})^T \boldsymbol{M}(-\boldsymbol{D}\boldsymbol{e}_2^{p_t} + \boldsymbol{S}^{-1}\boldsymbol{M}^T \boldsymbol{e}_2^{p_r}) + (\boldsymbol{e}_1^{\varepsilon_r})^T \boldsymbol{M}(-\boldsymbol{D}\boldsymbol{e}_2^{p_r}) \\ &+ (\boldsymbol{e}_1^{\varepsilon_t})^T (-\boldsymbol{D}^T)\boldsymbol{M}^T \boldsymbol{e}_2^{p_t} + (\boldsymbol{e}_1^{\varepsilon_r})^T (-\boldsymbol{D}^T - \boldsymbol{M}\boldsymbol{S}^{-T})\boldsymbol{M}^T \boldsymbol{e}_2^{p_r} \\ &+ (\boldsymbol{e}_1^{p_t})^T (-\boldsymbol{D}^T + \boldsymbol{M}\boldsymbol{S}^{-T})\boldsymbol{M}^T \boldsymbol{e}_2^{\varepsilon_t} + (\boldsymbol{e}_1^{p_r})^T (-\boldsymbol{D}^T)\boldsymbol{M}^T \boldsymbol{e}_2^{\varepsilon_r} \\ &+ (\boldsymbol{e}_1^{\varepsilon_t})^T \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi} \boldsymbol{e}_2^{p_t} + (\boldsymbol{e}_1^{\varepsilon_r})^T \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi} \boldsymbol{e}_2^{p_r} + (\boldsymbol{e}_1^{p_t})^T \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi} \boldsymbol{e}_2^{\varepsilon_t} + (\boldsymbol{e}_1^{p_r})^T \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi} \boldsymbol{e}_2^{\varepsilon_r} \\ = &(\boldsymbol{e}_1^{p_t})^T (-\boldsymbol{M}\boldsymbol{D} - \boldsymbol{D}^T \boldsymbol{M}^T + \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi}) \boldsymbol{e}_2^{\varepsilon_t} \\ &+ (\boldsymbol{e}_1^{p_r})^T (-\boldsymbol{M}\boldsymbol{D} - \boldsymbol{D}^T \boldsymbol{M}^T + \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi}) \boldsymbol{e}_2^{\varepsilon_r} \\ &+ (\boldsymbol{e}_1^{\varepsilon_t})^T (-\boldsymbol{M}\boldsymbol{D} - \boldsymbol{D}^T \boldsymbol{M}^T + \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi}) \boldsymbol{e}_2^{p_t} \\ &+ (\boldsymbol{e}_1^{\varepsilon_r})^T (-\boldsymbol{M}\boldsymbol{D} - \boldsymbol{D}^T \boldsymbol{M}^T + \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi}) \boldsymbol{e}_2^{p_r} \\ &+ (\boldsymbol{e}_1^{p_t})^T \underbrace{(\boldsymbol{M}\boldsymbol{S}^{-T}\boldsymbol{M}^T - \boldsymbol{M}\boldsymbol{S}^{-1}\boldsymbol{M}^T)}_{=\boldsymbol{0} \text{ due to } \boldsymbol{S}=\boldsymbol{S}^T} \boldsymbol{e}_2^{\varepsilon_t} + (\boldsymbol{e}_1^{\varepsilon_t})^T \underbrace{(\boldsymbol{M}\boldsymbol{S}^{-1}\boldsymbol{M}^T - \boldsymbol{M}\boldsymbol{S}^{-T}\boldsymbol{M}^T)}_{=\boldsymbol{0}} \boldsymbol{e}_2^{p_r} \\ = &\, 0.\end{aligned} \tag{B.5}$$

Resulting condition. Since (B.5) holds for any $(\boldsymbol{e}_1^{\mu_i}, \boldsymbol{e}_2^{\nu_i})$, the following condition is deduced, see also [60]:

$$-\boldsymbol{M}\boldsymbol{D} + \boldsymbol{D}^T \boldsymbol{M}^T + \boldsymbol{\Phi}^T \boldsymbol{\zeta} \boldsymbol{\Phi} = \boldsymbol{0}. \tag{B.6}$$

Appendix C

Free and Forced Vibrations of the Timoshenko Beam

We sketch the derivation of exact expressions for the eigenforms and the computation of the natural frequencies of the Timoshenko beam as described in [56]. It is based on a single fourth-order differential equation for the free and forced vibrations of the Timoshenko beam. Moreover, we summarize the reasoning of [10] to obtain transfer functions for the case of inhomogeneous boundary conditions (BCs).

C.1 Eigenvalue Problem

To characterize the free vibrations of the Timoshenko beam, we employ a separation of variables ansatz

$$\begin{aligned} w(t,z) &= T(t)X(z) \\ \psi(t,z) &= T(t)Y(z), \end{aligned} \tag{C.1}$$

which leads the variables $w(t,z)$ and $\psi(t,z)$ to the form of a product of a spatial function $X(z)$, $Y(z)$ and a time function $T(t)$. Here, the same time function is considered for both variables.

Then, the beam equations (3.37) become

$$\begin{aligned} \rho X(z)\ddot{T}(t) - K_s[X^{''}(z)T(t) - Y^{'}(z)T(t)] &= 0 \\ I_\rho Y(z)\ddot{T}(t) + K_s[Y(z)T(t) - X^{'}(z)T(t)] - K_b Y^{''}T(t) &= 0. \end{aligned} \tag{C.2}$$

Separation of variables yields

$$\frac{\ddot{T}}{T} = \frac{K_s(X'' - Y')}{\rho X} = \frac{K_b Y'' - K_s(Y - X')}{I_\rho Y}. \tag{C.3}$$

For harmonic vibrations with frequency ω, the time function satisfies $\ddot{T}(t) + \omega^2 T(t) = 0$. Using this relation, we arrive quickly at the expressions

$$Y' = X'' + a(\omega)X \tag{C.4}$$

$$-b(\omega)Y = X''' + [a(\omega) + c(\omega)]X' \tag{C.5}$$

with the frequency-dependent coefficients

$$\begin{aligned} a(\omega) &= \frac{\rho\omega^2}{K_s} \\ b(\omega) &= \frac{I_\rho\omega^2}{K_b} - \frac{K_s}{K_b} \\ c(\omega) &= \frac{K_s}{K_b}. \end{aligned} \tag{C.6}$$

From Equations (C.4) and (C.5), we eliminate $Y(z)$ to obtain the single fourth-order differential equation for the displacement function

$$X^{(4)} + d(\omega)X'' + e(\omega)X = 0, \tag{C.7}$$

where[1]

$$\begin{aligned} d(\omega) &= a(\omega) + b(\omega) + c(\omega) \\ &= \omega^2(\frac{I_\rho}{K_b} + \frac{\rho}{K_s}) \\ e(\omega) &= a(\omega)b(\omega) \\ &= \frac{\rho\omega^2}{K_b}(\frac{I_\rho\omega^2}{K_s} - 1). \end{aligned} \tag{C.8}$$

The function $Y(z)$ depends on the derivatives of the spatial function $X(z)$. Thus, Equations (C.4) and (C.5) can be used to define the boundary conditions, which are only in terms of $X(z)$ as well as its derivatives.

By parameterizing the solutions of Equation (C.7) via $X(z) = Ce^{\lambda z}$, we obtain the characteristic equation

$$\lambda^{(4)} + d(\omega)\lambda^2 + e(\omega) = 0. \tag{C.9}$$

With $r := \lambda^2$, this becomes a quadratic equation whose roots are given by

$$r_{1/2} = -\frac{1}{2}d(\omega) \pm \frac{1}{2}\sqrt{\Delta(\omega)}, \qquad \Delta(\omega) = d^2(\omega) - 4e(\omega) > 0 \quad \forall\omega. \tag{C.10}$$

At the critical frequency $\omega_c = \sqrt{\frac{\kappa GA}{I\rho_v}}$, the root r_1 changes sign, while $r_2 < 0$ for all ω. We distinguish the cases

1. $\omega > \omega_c \;\Rightarrow\; r_1 < 0$. Only pairs of complex eigenvalues occur:

$$\pm i\lambda_1 = \pm i\sqrt{-r_1}, \quad \pm i\lambda_2 = \pm i\sqrt{-r_2}. \tag{C.11}$$

The corresponding solution of the displacement function $X(z)$ for the high frequencies can be described only in terms of trigonometric functions $\cos(\bullet)$ and $\sin(\bullet)$ using Euler's formula:

$$\begin{aligned} X(z) &= C_1e^{i\lambda_1 z} + C_2e^{-i\lambda_1 z} + C_3e^{i\lambda_2 z} + C_4e^{-i\lambda_2 z} \\ &= P_1\cos(\lambda_1 z) + P_2\sin(\lambda_1 z) + P_3\cos(\lambda_2 z) + P_4\sin(\lambda_2 z). \end{aligned} \tag{C.12}$$

[1] The coefficients in terms of the volume density $\rho_v = \frac{\rho}{A}$ and the moment of inertia $I = \frac{I_\rho}{\rho_v}$ read

$$a = \frac{\rho_v\omega^2}{\kappa G}, \quad b = \frac{\rho_v\omega^2}{E} - c(\omega), \quad c = \frac{\kappa GA}{EI}, \quad d = \omega^2\rho_v(\frac{1}{E} + \frac{1}{\kappa G}), \quad e = \frac{\omega^2}{EI}(\frac{I\rho_v^2\omega^2}{\kappa G} - A\rho_v).$$

2. $\omega < \omega_c \Rightarrow r_1 > 0$. The resulting pairs of real and complex eigenvalues are

$$\pm\lambda_1 = \pm\sqrt{r_1}, \quad \pm i\lambda_2 = \pm i\sqrt{-r_2}. \tag{C.13}$$

The general solution of the displacement function $X(z)$ can be written in the form of trigonometric and hyperbolic function

$$\begin{aligned} X(z) &= C_1 e^{\lambda_1 z} + C_2 e^{-\lambda_1 z} + C_3 e^{i\lambda_2 z} + C_4 e^{-i\lambda_2 z} \\ &= P_1 \cosh(\lambda_1 z) + P_2 \sinh(\lambda_1 z) + P_3 \cos(\lambda_2 z) + P_4 \sin(\lambda_2 z) \end{aligned} \tag{C.14}$$

with $P_{1/2} = C_1 \pm C_2$ and $P_{3/4} = C_3 \pm C_4$. For our application, we consider only the second case and describe the eigenvalues and eigenforms for the (dominant) lower frequencies.

C.2 Boundary Conditions, Eigenforms and Natural Frequencies

The clamped-free boundary conditions with $\theta_h = 0$ are translated into conditions on $X(z)$:

$$\begin{aligned} z = 0: \qquad X(0) &= 0 \\ X'''(0) + (a(\omega) + c(\omega))X'(0) &= 0 \\ z = L: \qquad X'''(L) + dX'(L) &= 0 \\ X''(L) + aX(L) &= 0 \ . . \end{aligned} \tag{C.15}$$

With the general solution (C.14) for $X(z)$ and its derivatives, we can formulate a system of equations for the free parameters that satisfy the boundary conditions (abbreviations: $A = \cosh(\lambda_1 L)$, $B = \sinh(\lambda_1 L)$, $C = \cos(\lambda_2 L)$, $D = \sin(\lambda_2 L)$):

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & \lambda_1((a+c)+\lambda_1^2) & 0 & \lambda_2((a+c)-\lambda_2^2) \\ \lambda_1(\lambda_1^2+d)B & \lambda_1(\lambda_1^2+d)A & \lambda_2(\lambda_2^2-d)D & -\lambda_2(\lambda_2^2-d)C \\ (\lambda_1^2+a)A & (\lambda_1^2+a)B & (-\lambda_2^2+a)C & (-\lambda_2^2+a)D \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{C.16}$$

This system of equations has the form

$$\boldsymbol{M}(\omega)\boldsymbol{P} = \boldsymbol{0} \tag{C.17}$$

with a matrix $\boldsymbol{M}(\omega)$, whose entries depend on the frequency ω, and $\boldsymbol{P} = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 \end{bmatrix}^T$. The frequencies, at which $\det \boldsymbol{M}(\omega) = 0$, are the natural frequencies or eigenfrequencies ω_i of the cantilever beam. The corresponding non-zero values of $\boldsymbol{P}$ parametrize the associated eigenfunctions/eigenforms $X_i(z)$ of the free vibration. Normally, there is a free tuning parameter in $\boldsymbol{P}$, e.g. P_1, while the other parameters linearly depend on it. To determine P_1, either choose $P_1 = 1$ to get the simple normalized eigenfunctions, or the eigenfunctions can be orthonormalized by requiring (see e.g. [91]):

$$\langle X_i(z), X_j(z)\rangle = \begin{cases}\delta_{ij}, & i = j\\ 0, & i \neq j\end{cases}, \qquad \forall i, j.$$

Table C.1 lists the five low eigenfrequencies for the BCs (C.15) and the corresponding eigenfunctions are depicted in Figure C.1.

Eigenfrequencies	f_1	f_2	f_3	f_4	f_5
Hz	2.9343	18.3873	51.4781	100.8570	166.6820

Table C.1: First five natural frequencies of the Timoshenko beam for the uniform beam in Hz ($\omega = 2\pi f$).

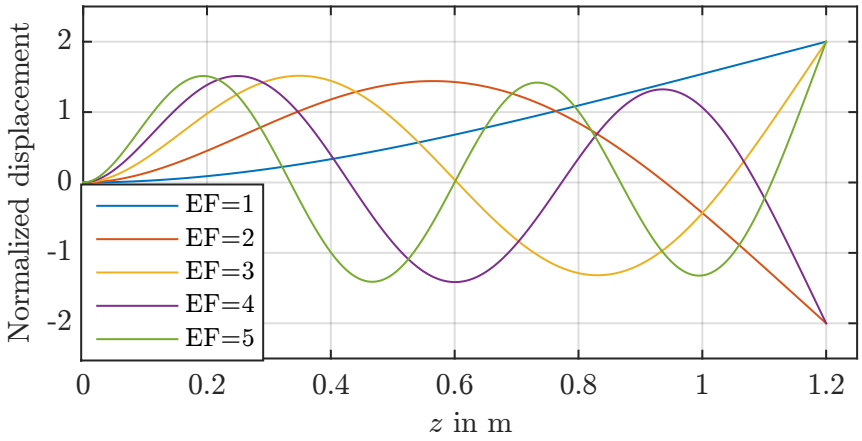


Figure C.1: First five eigenforms of the Timoshenko beam.

C.3 Inhomogeneous Boundary Conditions

To obtain a transfer function from the hub angular velocity to the angular velocity of the beam tip, we assume a harmonic excitation $\theta_h(\omega)$ of the hub angle. The second boundary condition in Equation (C.15) is in this case replaced by

$$-b(\omega)\theta_h(\omega) = X''' + (a(\omega) + c(\omega))X'. \tag{C.18}$$

The system of Equation (C.17) becomes inhomogeneous

$$\boldsymbol{M}(\omega)\boldsymbol{P} = \boldsymbol{K}(\omega), \tag{C.19}$$

with $\boldsymbol{K}(\omega) = \begin{bmatrix}0 & -b(\omega)\theta_h(\omega) & 0 & 0\end{bmatrix}^T$. For $\det \boldsymbol{M}(\omega) \neq 0$, we can express the coefficients of the general solution as $\boldsymbol{P} = \boldsymbol{M}^{-1}(\omega)\boldsymbol{K}(\omega)$, which allows us to establish the relation

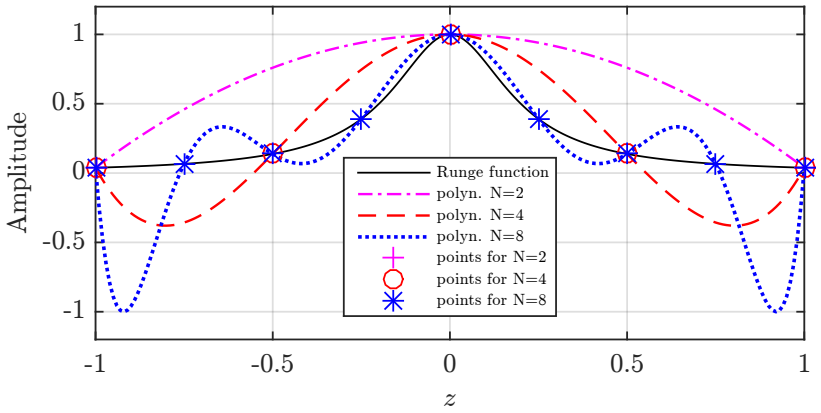
$$Y(\omega, L) = G(\omega)\theta_h(\omega) \tag{C.20}$$

between the hub angle and the rotation of the beam tip. The same transfer function $G(\omega)$ describes the ratio between the angular velocities, see Figure 3.10.

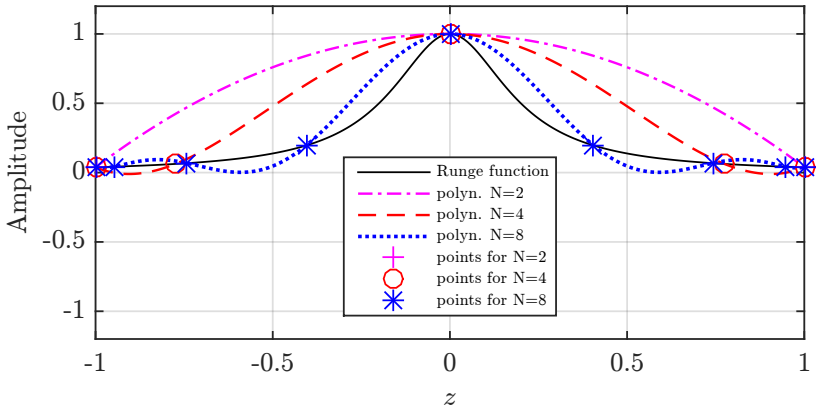
Appendix D

Runge's Phenomenon

In the mathematical field of numerical analysis, Runge's phenomenon illustrates the numerical oscillation at the edge of an interval, that appears when constructing the high-degree polynomial interpolation over a set of equidistant interpolation points, see e.g. [34].



(a) Interpolating polynomials with equidistant collocation points



(b) Interpolating polynomials with Gauss-Legendre collocation points

Figure D.1: Runge's phenomenon with different distribution of the collocation points

Consider a Runge function $f(z)$

$$f(z) = \frac{1}{25z^2},$$

which is interpolated at equidistant points between -1 and 1 with Lagrange interpolating polynomials of degree $N = 2, 4, 8$, respectively, see Figure D.1(a). It is observed that the interpolation oscillates toward the end of the interval, and the error is increased when the degree of the polynomials increases. In contrast, this phenomenon can be avoided by choosing the zeros of Gauss-Legendre polynomials, see Figure D.1(b). That is also the reason why we chose the latter as interpolation points.

Appendix E

Inverse State Space Model

We consider a linear, time-invariant SISO system

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}u \tag{E.1}$$

$$y = \boldsymbol{c}^T\boldsymbol{x} + du \tag{E.2}$$

with $d \neq 0$. The Laplace transform of Equations (E.1) and (E.2)

$$s\boldsymbol{X}(s) - \boldsymbol{x}_0 = \boldsymbol{A}\boldsymbol{X}(s) + \boldsymbol{b}U(s) \tag{E.3}$$

$$Y(s) = \boldsymbol{c}^T\boldsymbol{X}(s) + dU(s) \tag{E.4}$$

can be described using Rosenbrock-matrix in the following form

$$\begin{bmatrix} \boldsymbol{A} - s\boldsymbol{I} & \boldsymbol{b} \\ \boldsymbol{c}^T & d \end{bmatrix} \begin{bmatrix} \boldsymbol{X}(s) \\ U(s) \end{bmatrix} = \begin{bmatrix} -\boldsymbol{x}_0 \\ Y(s) \end{bmatrix}. \tag{E.5}$$

The values of s, which make the system singular, are the so-called invariant zeros of the state space model. It is calculated by the determinant of the Rosenbrock matrix as follows

$$\begin{aligned} \det \begin{bmatrix} \boldsymbol{A} - s\boldsymbol{I} & \boldsymbol{b} \\ \boldsymbol{c}^T & \boldsymbol{d} \end{bmatrix} &= \det((\boldsymbol{A} - s\boldsymbol{I})d - \boldsymbol{b}\boldsymbol{c}^T) \\ &= d\det(\boldsymbol{A} - s\boldsymbol{I} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T). \end{aligned} \tag{E.6}$$

Now, we solve the output equation (E.4) for $U(s)$

$$U(s) = \frac{1}{d}Y(s) - \frac{1}{d}\boldsymbol{c}^T\boldsymbol{X}(s), \tag{E.7}$$

and insert it into Equation (E.3) to obtain the relation between $\boldsymbol{X}(s)$ and $Y(s)$

$$(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})\boldsymbol{X}(s) + \frac{1}{d}\boldsymbol{b}Y(s) = -\boldsymbol{x}_0. \tag{E.8}$$

We represent the above two equations in the matrix form

$$\begin{bmatrix} \boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I} & \frac{1}{d}\boldsymbol{b} \\ -\frac{1}{d}\boldsymbol{c}^T & \frac{1}{d} \end{bmatrix} \begin{bmatrix} \boldsymbol{X}(s) \\ Y(s) \end{bmatrix} = \begin{bmatrix} -\boldsymbol{x}_0 \\ U(s) \end{bmatrix}. \tag{E.9}$$

As a consequence, the state space representation of Equation (E.9) is obtained by inverse Laplace transformation

$$\dot{\boldsymbol{x}} = (\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T)\boldsymbol{x} + \frac{1}{d}\boldsymbol{b}y \tag{E.10}$$

$$u = -\frac{1}{d}\boldsymbol{c}^T\boldsymbol{x} + \frac{1}{d}y. \tag{E.11}$$

Note that there are no new additional state variables in Equation (E.10) compared with Equation (E.1). The signals $\boldsymbol{X}(s)$, $U(s)$, $Y(s)$, and $\boldsymbol{x}_0$, which satisfy the original system (E.1), (E.2), satisfy the system (E.10), (E.11) as well. This means that both models have identical trajectories of state variables.

- If $u(t)$, $\boldsymbol{x}_0$ are given for determining $\boldsymbol{x}(t)$, $y(t)$, there exists a solution of Equations (E.1), (E.2) (e.g. numerical integration, discrete convolution);
- If $y(t)$, $\boldsymbol{x}_0$ are given for determining $\boldsymbol{x}(t)$, $u(t)$, there exists a solution of Equations (E.10), (E.11) (e.g. numerical integration, discrete convolution).

Because the Rosenbrock matrix in the inverse model (E.10) can be rewritten as

$$\frac{1}{d}\det(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I} + \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T) = \frac{1}{d}\det(\boldsymbol{A} - s\boldsymbol{I}), \tag{E.12}$$

the invariant zeros of the model (E.10), (E.11) are identical with the eigenvalues of the model (E.1), (E.2).

According to Equation (E.5), $U(s)$ can be formulated only related to $Y(s)$ and $\boldsymbol{x}_0$ using Woodbury matrix identity

$$U(s) = \frac{1}{d}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\boldsymbol{x}_0 + (\frac{1}{d^2}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\boldsymbol{b} + \frac{1}{d})Y(s). \tag{E.13}$$

To determine $Y(s)$, we insert Equation (E.13) and the same $\boldsymbol{x}_0$ into Equation (E.9) to obtain

$$\begin{aligned} Y(s) = &-\left[\frac{1}{d} + \frac{1}{d}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\frac{1}{d}\boldsymbol{b}\right]^{-1}\frac{1}{d}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\boldsymbol{x}_0 \\ &+\left[\frac{1}{d} + \frac{1}{d}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\right]U(s). \end{aligned} \tag{E.14}$$

Then, $U(s)$ is replaced by Equation (E.13), and it results

$$\begin{aligned} Y(s) &= \left[\frac{1}{d} + \frac{1}{d}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\frac{1}{d}\boldsymbol{b}\right]^{-1}\left[\frac{1}{d^2}\boldsymbol{c}^T(\boldsymbol{A} - \frac{1}{d}\boldsymbol{b}\boldsymbol{c}^T - s\boldsymbol{I})^{-1}\boldsymbol{b} + \frac{1}{d}\right]Y(s) \\ &= Y(s). \end{aligned} \tag{E.15}$$

It is recognized that the perfect trajectory tracking can be realized by interconnecting the feedforward part based on Equations (E.10), (E.11) with the plant (E.1), (E.2). Consequently, this tracking behavior can be also indicated by multiplication of the transfer function, see Equation (E.15).

Appendix F

Bounded Solution of the Inversion-based Feedforward Control

In this chapter, we only consider the unstable part of the inverse model in Jordan normal form

$$\dot{\boldsymbol{\sigma}}_u(t) = \boldsymbol{\Lambda}_u \boldsymbol{\sigma}_u(t) + \boldsymbol{B}_{\sigma,u} \bar{\boldsymbol{u}}_b(t). \tag{F.1}$$

If we integrate it in the forward time, it leads to an unbounded solution, which is not wanted. In order to obtain a stable solution, Equation (F.1) should be integrated in reverse time. To this end, we define a time variable v to satisfy the following relations:

$$v = T - t, \qquad dv = -dt, \tag{F.2}$$

with a time interval T.

Substituting Equation (F.2) into Equation (F.1), and eliminating t, it becomes

$$\begin{aligned} \frac{d}{dv}\,\tilde{\boldsymbol{\sigma}}_u(v) &= \frac{d\boldsymbol{\sigma}_u(t)}{dt} \cdot \frac{dt}{dv} \\ &= -\boldsymbol{\Lambda}_u\,\tilde{\boldsymbol{\sigma}}_u(v) - \boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(v) \end{aligned} \tag{F.3}$$

with new state vector $\tilde{\boldsymbol{\sigma}}_u(v)$ and new input $\tilde{\boldsymbol{u}}_b(v)$ in terms of v

$$\begin{aligned} \tilde{\boldsymbol{\sigma}}_u(v) &= \boldsymbol{\sigma}_u(T - v) \\ \tilde{\boldsymbol{u}}_b(v) &= \bar{\boldsymbol{u}}_b(T - v). \end{aligned} \tag{F.4}$$

Multiplying the state-transition matrix $e^{\boldsymbol{\Lambda}_u v}$ on the left of Equation gives (F.3)

$$e^{\boldsymbol{\Lambda}_u v}\,\dot{\tilde{\boldsymbol{\sigma}}}_u(v) + e^{\boldsymbol{\Lambda}_u v}\,\boldsymbol{\Lambda}_u\,\tilde{\boldsymbol{\sigma}}_u(v) = -e^{\boldsymbol{\Lambda}_u v}\,\boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(v). \tag{F.5}$$

Using the integration by parts on the left terms

$$\frac{d}{dv}(e^{\boldsymbol{\Lambda}_u v}\,\tilde{\boldsymbol{\sigma}}_u(v)) = -e^{\boldsymbol{\Lambda}_u v}\,\boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(v), \tag{F.6}$$

and integrating Equation (F.6) in the time interval $s \in [0, v]$

$$\begin{aligned} \int_0^v \frac{d}{ds}(e^{\boldsymbol{\Lambda}_u s}\,\tilde{\boldsymbol{\sigma}}_u(s)) &= \int_0^v -e^{\boldsymbol{\Lambda}_u s}\,\boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(s)\,ds \\ e^{\boldsymbol{\Lambda}_u v}\,\tilde{\boldsymbol{\sigma}}_u(v) - e^{\boldsymbol{\Lambda}_u 0}\,\tilde{\boldsymbol{\sigma}}_u(0) &= \int_0^v -e^{\boldsymbol{\Lambda}_u s}\,\boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(s)\,ds, \end{aligned} \tag{F.7}$$

the solution of Equation (F.3) is represented as follows

$$\tilde{\boldsymbol{\sigma}}_u(v) = -\int_0^v e^{-\boldsymbol{\Lambda}_u(v-s)}\,\boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(s)\,ds. \tag{F.8}$$

Based on the transformation (F.2), we replace v in Equation (F.4) by t

$$\begin{aligned} \tilde{\boldsymbol{\sigma}}_u(T-t) &= \boldsymbol{\sigma}_u(T-(T-t)) &&= \boldsymbol{\sigma}_u(t) \\ \tilde{\boldsymbol{u}}_b(T-t) &= \bar{\boldsymbol{u}}_b(T-(T-t)) &&= \bar{\boldsymbol{u}}_b(t). \end{aligned} \tag{F.9}$$

Then, the solution of $\boldsymbol{\sigma}_u(t)$ can be derived by back-transforming (F.8)

$$\begin{aligned} \boldsymbol{\sigma}_u(t) &= -\int_0^{T-t} e^{-\boldsymbol{\Lambda}_u(T-t-s)}\,\boldsymbol{B}_{\sigma,u}\,\tilde{\boldsymbol{u}}_b(s)\,ds \\ &= \int_T^t e^{-\boldsymbol{\Lambda}_u(\tau-t)}\,\boldsymbol{B}_{\sigma,u}\,\bar{\boldsymbol{u}}_b(\tau)\,dv \end{aligned} \tag{F.10}$$

with the definition $\tau = T - s$. Since $0 < s < T - t$, the integration time interval for v is determined by $T > \tau > t$. Note that the inertial condition starts with the time T. In case $T \to \infty$, the solution becomes

$$\begin{aligned} \boldsymbol{\sigma}_u(t) &= \int_\infty^t e^{-\boldsymbol{\Lambda}_u(\tau-t)}\,\boldsymbol{B}_{\sigma,u}\,\bar{\boldsymbol{u}}_b(\tau)\,d\tau \\ &= -\int_t^\infty e^{-\boldsymbol{\Lambda}_u(\tau-t)}\,\boldsymbol{B}_{\sigma,u}\,\bar{\boldsymbol{u}}_b(\tau)\,d\tau. \end{aligned} \tag{F.11}$$

Appendix G

Hardware and Software Configuration

G.1 Components of the Test Bench

Type	Components	Supplier/Amount
Servo Motor	**RD85x13-HD i=160**	TQ-Systems GmbH
– Motor	– ILM85x13	TQ-Systems GmbH
– Gear	– CPL-25A-160-2A	Harmonic Drive AG
– Encoder	– AksIM MBA4	Renishaw plc
Servo Drive	**ACOPOS 8V1016.50-2**	B&R Industrial Automation GmbH
–Interface	–POWERLINK V2 Interface 8AC114.60-2	1
–Interface	–Biss Interface 5V 8AC125.60-2	2
–Cable	–POWERLINK X20CA0E61.00100	1
Industrial PLC	**X20CP1586**	B&R Industrial Automation GmbH
–Bus module	–X20BM33	1
–CAN module	–X20IF2772	1
–Safety input	–X20SLX410	1
–Safety output	–X20SO4110	1
–Field terminal	–X20TB52	1
–Cable	–Ethernet Standard	1
Sensor IMU	**LPMS-CU2**	LP-Research Inc.
–Cable	–CAN DB9-5pin Cable	1

Table G.1: Components of the test rig

G.2 Technical Specifications of Motors

ID	Motor Parameters	Description	Value
30	MOTOR_TYPE	Motor: Type	16#0002
31	MOTOR_COMPATIBILITY	Software compatibility	16#0201
46	MOTOR_WIND_CONNECT	Winding connection	1
47	MOTOR_POLEPAIRS	Number of pole-pairs	10
42	MOTOR_BRAKE_CURR_RATED	Holding brake: Rated current	0.5A
43	MOTOR_BRAKE_TORQ_RATED	Holding brake: Rated torque	1.68Nm
44	MOTOR_BRAKE_ON_TIME	Holding brake: Engaging delay	0.020s
45	MOTOR_BRAKE_OFF_TIME	Holding brake: Release delay	0.04s
64	MOTOR_TEMPSENS_PAR1	Temperature sensor: Parameter 1	0
65	MOTOR_TEMPSENS_PAR2	Temperature sensor: Parameter 2	0
66	MOTOR_TEMPSENS_PAR3	Temperature sensor: Parameter 3	0
67	MOTOR_TEMPSENS_PAR4	Temperature sensor: Parameter 4	0
68	MOTOR_TEMPSENS_PAR5	Temperature sensor: Parameter 5	0
69	MOTOR_TEMPSENS_PAR6	Temperature sensor: Parameter 6	0
70	MOTOR_TEMPSENS_PAR7	Temperature sensor: Parameter 7	0
71	MOTOR_TEMPSENS_PAR8	Temperature sensor: Parameter 8	0
72	MOTOR_TEMPSENS_PAR9	Temperature sensor: Parameter 9	0
73	MOTOR_TEMPSENS_PAR10	Temperature sensor: Parameter 10	0
75	MOTOR_THERMAL_CONST	Thermal time constant	20s
48	MOTOR_VOLTAGE_RATED	Rated voltage	48V
49	MOTOR_VOLTAGE_CONST	Voltage constant	$17.65\text{mV} \cdot \text{min}$
50	MOTOR_SPEED_RATED	Rated speed	$2720\frac{1}{\text{min}}$
51	MOTOR_SPEED_MAX	Maximum speed	$2720\frac{1}{\text{min}}$
52	MOTOR_TORQ_STALL	Stall torque	0.598Nm
53	MOTOR_TORQ_RATED	Rated torque	0.598Nm
54	MOTOR_TORQ_MAX	Peak torque	1.571Nm
55	MOTOR_TORQ_CONST	Torque constant	$0.13\frac{\text{Nm}}{\text{A}}$
56	MOTOR_CURR_STALL	Stall current	4.6A
57	MOTOR_CURR_RATED	Rated current	4.6A
58	MOTOR_CURR_MAX	Peak current	12.085A
59	MOTOR_WIND_CROSS_SECT	Phase cross section	0.985mm^2
60	MOTOR_STATOR_RESISTANCE	Stator resistance	0.21Ω
61	MOTOR_STATOR_INDUCTANCE	Stator inductance	$4.7 \cdot 10^{-4}\text{H}$
62	MOTOR_INERTIA	Moment of inertia	$2.29 \cdot 10^{-4}\text{kgm}^2$
63	MOTOR_COMMUT_OFFSET	Commutation offset	2.55rad
74	MOTOR_WIND_TEMP_MAX	Limit temperature	125°C
866	MOTOR_TEST_MODE	Test Mode	7
872	MOTOR_CURR_ROT_DIR	Rotational direction of current	ncINVERSE

Table G.2: Technical Data of motor RD85x13

Table G.2 lists the motor datasheet from the motor supplier TQ-Systems GmbH. In practice, the motor is directly controlled by the servo drive. In order to access it, each motor parameter owns a writable ID in the servo drive. Because there are no temperature sensors in the motor, the values of the corresponding parameters such as *MOTOR_TEMPSENS_PARx* (Temperature sensor: Parameter x, $x = 1, \ldots, 10$) are set to zero. In our application, there are parameter settings of the motor brake, which works in case of emergency. Furthermore, *MOTOR_COMMUT_OFFSET* must be reconfigured after the motor is mounted in the test rig.

References

[1] O. J. Aldraihem, R. C. Wetherhold, and T. Singh. *Distributed control of laminated beams: Timoshenko theory vs. Euler-Bernoulli theory.* Journal of intelligent material systems and structures, 8(2):149–157, 1997.

[2] W. Bolton. *Programmable Logic Controllers.* Newnes, 2015.

[3] A. Brugnoli, D. Alazard, V. Pommier-Budinger, and D. Matignon. *Partitioned finite element method for the Mindlin plate as a port-Hamiltonian system.* IFAC-PapersOnLine, 52(2):88–95, 2019.

[4] F. L. Cardoso-Ribeiro, D. Matignon, and V. Pommier-Budinger. *Piezoelectric beam with distributed control ports: A power-preserving discretization using weak formulation.* IFAC-PapersOnLine, 49(8):290–297, 2016.

[5] D. Chen and B. Paden. *Stable inversion of nonlinear non-minimum phase systems.* International Journal of Control, 64(1):81–97, 1996.

[6] T. Courant and A. Weinstein. *Beyond Poisson structures.* Action hamiltoniennes de groupes. Troisieme théoreme de Lie (Lyon, 1986), 27:39–49, 1988.

[7] T. J. Courant. *Dirac manifolds.* Transactions of the American Mathematical Society, 319(2):631–661, 1990.

[8] L. B. da Veiga, C. Lovadina, and A. Reali. *Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods.* Computer methods in applied mechanics and engineering, 241:38–51, 2012.

[9] M. Dalsmo and A. J. van der Schaft. *On representations and integrability of mathematical structures in energy-conserving physical systems.* SIAM Journal on Control and Optimization, 37(1):54–91, 1998.

[10] J. Damerau. *Untersuchung der dynamischen Eigenschaften von Balken mit fraktionalen Stoffgesetzen.* Ph.D. dissertation, Helmut Schmidt University, Hamburg, 2008.

[11] A. De Luca, P. Lucibello, and A. G. Ulivi. *Inversion techniques for trajectory control of flexible robot arms.* Journal of Robotic Systems, 6(4):325–344, 1989.

[12] A. De Luca, S. Panzieri, and A. G. Ulivi. *Stable inversion control for flexible link manipulators.* In Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146), volume 1, pages 799–805. IEEE, 1998.

[13] J. Deutscher. *Zustandsregelung verteilt-parametrischer Systeme.* Springer-Verlag, 2012.

[14] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx. *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Springer Science & Business Media, 2009.

[15] H. M. Eigner. *Commissioning of the electric motors for a three-axis robot*. Master's thesis, Institute of Automatic Control, Technical University of Munich, Germany, 2017.

[16] S. Engell. *Optimale lineare Regelung: Grenzen der erreichbaren Regelgüte in linearen zeitinvarianten Regelkreisen*, volume 18. Springer-Verlag, 2013.

[17] H. Flanders. *Differential Forms with Applications to the Physical Sciences*, volume 11. Courier Corporation, 1963.

[18] M. Fliess, H. Mounier, P. Rouchon, and J. Rudolph. *A distributed parameter approach to the control of a tubular reactor: A multivariable case*. In Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171), volume 1, pages 439–442. IEEE, 1998.

[19] O. Föllinger, U. Konigorski, B. Lohmann, G. Roppenecker, and A. Trächtler. *Regelungstechnik*. VDE-Verlag, 11. edition, 2013.

[20] B. Fornberg. *A Practical Guide to Pseudospectral Methods*. Number 1. Cambridge University Press, 1998.

[21] K. S. Fu, R. Gonzalez, and C. G. Lee. *Robotics: Control, Sensing, Vision and Intelligence*. Tata McGraw-Hill Education, 1987.

[22] Y. Q. Gao, F. Y. Wang, and Z. Q. Xiao. *Flexible Manipulators: Modeling, Analysis and Optimum Design*. Academic Press, 2012.

[23] D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington. *An overview of three pseudospectral methods for the numerical solution of optimal control problems*. ances in the Astronautical Sciences, 135(1):475–487, 2009.

[24] H. Gattringer. *Starr-elastische Robotersysteme: Theorie und Anwendungen*. Springer-Verlag, 2011.

[25] H. Gattringer and J. Gerstmayr. *Multibody System Dynamics, Robotics and Control*. Springer Science & Business Media, 2013.

[26] B. Gebler. *Modellbildung, Steuerung und Regelung für elastische Industrieroboter: (Rückentitel: Dynamik elastischer Roboter)*. VDI-Verlag, 1987.

[27] M. Giftthaler, T. Wolf, H. K. Panzer, and B. Lohmann. *Parametric model order reduction of port-Hamiltonian systems by matrix interpolation*. at-Automatisierungstechnik, 62(9):619–628, 2014.

[28] G. Golo, V. Talasila, A. J. van der Schaft, and B. Maschke. *Hamiltonian discretization of boundary control systems*. Automatica, 40(5):757–771, 2004.

[29] K. Graichen, V. Hagenmeyer, and M. Zeitz. *A new approach to inversion-based feedforward control design for nonlinear systems*. Automatica, 41(12):2033–2041, 2005.

[30] S. Gugercin, R. V. Polyuga, C. Beattie, and A. J. van der Schaft. *Structure-preserving tangential interpolation for model reduction of port-Hamiltonian systems.* Automatica, 48(9):1963–1974, 2012.

[31] C. M. Hackl. *Dynamische Reibungsmodellierung: Das Lund-Grenoble (LuGre) Reibmodell.* In Elektrische Antriebe-Regelung von Antriebssystemen, pages 1615–1657. Springer, 2015.

[32] C. Heicks. *Simulationsmodell eines Roboters mit elastischen Balken.* Master's thesis, Institute of Automatic Control, Technical University of Munich, Germany, 2017.

[33] A. Hermann, P. Kotyczka, and R. Banavar. *Coordinate-invariant modeling and control of a three-DOF robot manipulator.* 7th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, Berlin, 2021.

[34] J. S. Hesthaven. *From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex.* SIAM Journal on Numerical Analysis, 35(2):655–676, 1998.

[35] A. Isidori. *Nonlinear Control Systems.* Springer Science & Business Media, 3. edition, 2013.

[36] B. Jacob and H. Zwart. *Linear Port-Hamiltonian Systems on Infinite-dimensional Spaces*, volume 223. Springer Science & Business Media, 2012.

[37] U. Kleemann. *Regelung elastischer Roboter.* VDI-Verlag, 1989.

[38] P. Kotyczka. *Finite volume structure-preserving discretization of 1D distributed-parameter port-Hamiltonian systems.* IFAC-PapersOnLine, 49(8):298–303, 2016.

[39] P. Kotyczka. *Numerical Methods for Distributed Parameter Port-Hamiltonian Systems.* Habilitation, TUM.University Press, 2019.

[40] P. Kotyczka and S. Brandstäter. *Inversion-based feedforward control for discretized port-Hamiltonian systems.* In Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems, pages 729–735, 2014.

[41] P. Kotyczka and T. Thoma. *Symplectic discrete-time energy-based control for nonlinear mechanical systems.* Automatica, 133:109842, 2021.

[42] A. Krämer and J. Kempkes. *Modellierung und Simulation von nichtlinearen Reibungseffekten bei der Lageregelung von Servomotoren.* FHWS Science Journal, 1 (2):47–57, 2014.

[43] G. Kreisselmeier. *Struktur mit zwei Freiheitsgraden/Two-Degree-of-Freedom Control Structure.* at-Automatisierungstechnik, 47(6):266–269, 1999.

[44] M. Kubisch. *Modellierung und Simulation nichtlinearer Motoreigenschaften.* Master's thesis, Humboldt-Universität zu Berlin, Germany, 2008.

[45] U. Kurz, H. Hintzen, and H. Laufenberg. *Konstruieren, Gestalten, Entwerfen: Ein Lehr-und Arbeitsbuch für das Studium der Konstruktionstechnik.* Springer-Verlag, 2009.

[46] Y. Le Gorrec, H. Zwart, and B. Maschke. *Dirac structures and boundary control systems associated with skew-symmetric differential operators.* SIAM Journal on Control and Optimization, 44(5):1864–1892, 2005.

[47] J. Lee and W. W. Schultz. *Eigenvalue analysis of Timoshenko beams and axisymmetric Mindlin plates by the pseudospectral method.* Journal of Sound and Vibration, 269(3):609–621, 2004.

[48] J. Lessard, P. Bigras, Z. Liu, and B. Hazel. *Characterization, modeling and vibration control of a flexible joint for a robotic system.* Journal of Vibration and Control, 20(6):943–960, 2014.

[49] A. D. Luca and B. Siciliano. *Trajectory control of a non-linear one-link flexible arm.* International Journal of Control, 50(5):1699–1715, 1989.

[50] J. Lunze. *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung.* Springer-Verlag, 5. edition, 2014.

[51] A. Macchelli and C. Melchiorri. *Control by interconnection and energy shaping of the Timoshenko beam.* Mathematical and Computer Modelling of Dynamical Systems, 10(3-4):231–251, 2004.

[52] A. Macchelli and C. Melchiorri. *Modeling and control of the Timoshenko beam. The distributed port Hamiltonian approach.* SIAM Journal on Control and Optimization, 43(2):743–767, 2004.

[53] A. Macchelli, C. Melchiorri, and S. Stramigioli. *Port-based modeling of a flexible link.* IEEE transactions on robotics, 23(4):650–660, 2007.

[54] A. Macchelli, C. Melchiorri, and S. Stramigioli. *Port-based modeling and simulation of mechanical systems with rigid and flexible links.* IEEE Transactions on Robotics, 25(5):1016–1029, 2009.

[55] N. Macon and D. Dupree. *Existence and uniqueness of interpolating rational functions.* The American Mathematical Monthly, 69(8):751–759, 1962.

[56] L. Majkut. *Free and forced vibrations of Timoshenko beams described by single difference equation.* Journal of Theoretical and Applied Mechanics, 47(1):193–210, 2009.

[57] B. M. Maschke, A. J. van der Schaft, and P. C. Breedveld. *An intrinsic Hamiltonian formulation of network dynamics: Non-standard Poisson structures and gyrators.* Journal of the Franklin institute, 329(5):923–966, 1992.

[58] X. Meng. *Implementation and investigation of control concepts for an elastic robot.* Master's thesis, Institute of Automatic Control, Technical University of Munich, Germany, 2018.

[59] T. Meurer and M. Zeitz. *Flachheitsbasierte Steuerung und Regelung eines Wärmeleitungssystems (Flatness-based Feedforward and Feedback Control of a Heat Conduction System).* at-Automatisierungstechnik, 52(9):411–420, 2004.

[60] R. Moulla, L. Lefèvre, and B. Maschke. *Pseudo-spectral methods for the spatial symplectic reduction of open systems of conservation laws.* Journal of Computational Physics, 231(4):1272–1292, 2012.

[61] P. Münch. *Konzeption und Entwurf integrierter Regelungen für Modulare Multilevel Umrichter*, volume 3. Logos Verlag Berlin GmbH, 2011.

[62] R. Ortega and E. Garcia-Canseco. *Interconnection and damping assignment passivity-based control: A Survey.* European Journal of control, 10(5):432–450, 2004.

[63] R. Ortega, A. J. van der Schaft, I. Mareels, and B. Maschke. *Putting energy back in control.* IEEE Control Systems Magazine, 21(2):18–33, 2001.

[64] R. Ortega, A. J. van der Schaft, B. Maschke, and G. Escobar. *Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems.* Automatica, 38(4):585–596, 2002.

[65] R. Ortega, A. J. van der Schaft, F. Castanos, and A. Astolfi. *Control by interconnection and standard passivity-based control of port-Hamiltonian systems.* IEEE Transactions on Automatic control, 53(11):2527–2542, 2008.

[66] R. V. Polyuga and A. J. van der Schaft. *Structure preserving model reduction of port-Hamiltonian systems by moment matching at infinity.* Automatica, 46(4): 665–672, 2010.

[67] J. N. Reddy. *Energy Principles and Variational Methods in Applied Mechanics.* John Wiley & Sons, 2017.

[68] C. A. Roduner. *H_∞-Regelung linearer Systeme mit Totzeiten.* Ph.D. dissertation, ETH Zurich, 1997.

[69] M. Ruderman. *Zur Modellierung und Kompensation dynamischer Reibung in Aktuatorsystemen.* Ph.D. dissertation, Technische Universität Dortmund, Germany, 2012.

[70] J. Rudolph and F. Woittennek. *Flachheitsbasierte Steuerung eines Timoshenko-Balkens.* ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics, 83 (2):119–127, 2003.

[71] M. Schöberl. *Differentialgeometrische Beschreibung und Analyse Tor-basierter Hamilton'scher Systeme.* at-Automatisierungstechnik, 63(9):672–683, 2015.

[72] M. Schöberl and A. Siuka. *On the port-Hamiltonian representation of systems described by partial differential equations.* IFAC Proceedings Volumes, 45(19):1–6, 2012.

[73] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators.* Springer Science & Business Media, 2012.

[74] R. Seifried, M. Burkhardt, and A. Held. *Trajectory control of flexible manipulators using model inversion.* In Proceedings of the ECCOMAS Thematic Conference on Multibody Dynamics, 2011.

[75] M. Seslija, A. J. van der Schaft, and J. M. Scherpen. *Structure preserving spatial discretization of a 1-D piezoelectric Timoshenko beam.* Multiscale Modeling & Simulation, 9(1):129–154, 2011.

[76] B. Simeon. *Computational flexible multibody dynamics.* In A Differential-Algebraic Approach. Differential-Algebraic Equations Forum. Springer, Heidelberg, 2013.

[77] A. Siuka, M. Schöberl, K. Rieger, and K. Schlacher. *Regelung verteilt-parametrischer Hamiltonscher Systeme auf Basis struktureller Invarianten.* at-Automatisierungstechnik, 59(8):465–478, 2011.

[78] A. Siuka, M. Schöberl, and K. Schlacher. *Port-Hamiltonian modelling and energy-based control of the Timoshenko beam.* Acta mechanica, 222(1-2):69–89, 2011.

[79] P. Staufer and H. Gattringer. *Passivity-based tracking control of a flexible link robot.* In Multibody System Dynamics, Robotics and Control, pages 95–112. Springer, 2013.

[80] F. Svaricek. *Nulldynamik linearer und nichtlinearer Systeme: Definitionen, Eigenschaften und Anwendungen (Zero Dynamics of Linear and Nonlinear Systems: Definitions, Properties and Applications).* at-Automatisierungstechnik, 54(7):310–322, 2006.

[81] H. D. Taghirad and P. R. Bélanger. *Modeling and parameter identification of Harmonic Drive systems.* ASME Journal of Dynamic Systems, Measurement and Control, 120(4):439–444, 1998.

[82] T. Thoma, X. Wu, A. Dietrich, and P. Kotyczka. *Symplectic discrete-time control of flexible-joint robots: Experiments with two links.* 7th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, Berlin, 2021.

[83] M. Tiegelkamp and K.-H. John. *IEC 61131-3: Programming Industrial Automation Systems.* Springer, 1995.

[84] S. P. Timoshenko and J. M. Gere. *Theory of Elastic Stability.* Courier Corporation, 2009.

[85] M. V. Trivedi, R. N. Banavar, and P. Kotyczka. *Hamiltonian modelling and buckling analysis of a nonlinear flexible beam with actuation at the bottom.* Mathematical and Computer Modelling of Dynamical Systems, 22(5):475–492, 2016.

[86] A. J. van der Schaft and D. Jeltsema. *Port-Hamiltonian systems theory: An introductory overview.* Foundations and Trends in Systems and Control, 1(2-3): 173–378, 2014.

[87] A. J. van der Schaft and B. Maschke. *Hamiltonian formulation of distributed-parameter systems with boundary energy flow.* Journal of Geometry and Physics, 42(1):166–194, 2002.

[88] T. Voß. *Port-Hamiltonian modeling and control of piezoelectric beams and plates: Application to inflatable space structures.* Ph.D. dissertation, University of Groningen, Netherlands, 2010.

[89] N. M. T. Vu, L. Lefevre, R. Nouailletas, and S. Brémond. *Geometric discretization for a plasma control model.* IFAC Proceedings Volumes, 46(2):755–760, 2013.

[90] N. M. T. Vu, L. Lefèvre, R. Nouailletas, and S. Brémond. *Symplectic spatial integration schemes for systems of balance equations.* Journal of Process Control, 51:1–17, 2017.

[91] D. Wang and M. Vidyasagar. *Transfer functions for a single flexible link.* The International journal of robotics research, 10(5):540–549, 1991.

[92] F.-Y. Wang, P. Zhou, and P. Lever. *Dynamic effects of rotatory inertia and shear deformation on flexible manipulators.* In 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929), volume 3, pages 2315–2320. IEEE, 1996.

[93] M. Wang and P. Kotyczka. *Port-Hamiltonian model of a flexible manipulator test rig.* In Methoden und Anwendungen der Regelungstechnik. Shaker Verlag, Aachen, Jul 2017.

[94] M. Wang and P. Kotyczka. *Trajektorienfolgeregelung auf Basis diskretisierter Port-Hamiltonscher Modelle am Beispiel eines elastischen Roboterarms.* In GMA Fachausschuss 1.40 "Theoretische Verfahren der Regelungstechnik", Anif, Austria, 2018. URL `https://www.control.tf.uni-kiel.de/en/gma-fa-1.40/archiv-1/workshop-2018-fa-1.40`.

[95] M. Wang and P. Kotyczka. *Trajectory control of an elastic manipulator based on the discretized port-Hamiltonian model.* In 90th GAMM Annual Meeting: Presentation at "Young Researchers' Minisymposia", Vienna, Austria, 2019.

[96] M. Wang and P. Kotyczka. *Trajectory control of an elastic beam based on port-Hamiltonian numerical models.* at-Automatisierungstechnik, 69(6):459–473, 2021.

[97] M. Wang, A. Bestler, and P. Kotyczka. *Modeling, discretization and motion control of a flexible beam in the port-Hamiltonian framework.* IFAC-PapersOnLine, 50(1): 6799–6806, 2017.

[98] J. Weidauer. *Elektrische Antriebstechnik: Grundlagen, Auslegung, Anwendungen, Lösungen.* John Wiley & Sons, 2019.

[99] M. Wenz. *Automatische Konfiguration der Bewegungssteuerung von Industrierobotern.* Logos Verlag Berlin GmbH, 2008.

[100] T. Wolf, B. Lohmann, R. Eid, and P. Kotyczka. *Passivity and structure preserving order reduction of linear port-Hamiltonian systems using Krylov subspaces.* European Journal of Control, 16(4):401–406, 2010.

[101] Y. Wu, B. Hamroun, Y. Le Gorrec, and B. Maschke. *Port Hamiltonian system in descriptor form for balanced reduction: Application to a nanotweezer.* IFAC Proceedings Volumes, 47(3):11404–11409, 2014.

[102] Y. Wu, B. Hamroun, Y. Le Gorrec, and B. Maschke. *Reduced order controller design for Timoshenko beam: A port Hamiltonian approach.* IFAC-PapersOnLine, 50(1):7121–7126, 2017.

[103] P. Zhang. *Industrial Control Technology: A Handbook for Engineers and Researchers.* William Andrew, 2008.

[104] T. Zhang, Y. Liu, S. Yan, Q. Zhang, and H. Zhang. *Comparative study on the acceleration feedback and the strain feedback of a flexible manipulator.* In 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929), volume 2, pages 1031–1035. IEEE, 1996.

[105] S. Zollitsch. *Simulative Untersuchung eines Laborroboters mit elastischen Gelenken.* Master's thesis, Institute of Automatic Control, Technical University of Munich, Germany, 2016.

[106] Q. Zou. *Optimal preview-based stable-inversion for output tracking of nonminimum-phase linear systems.* Automatica, 45(1):230–237, 2009.

[107] Q. Zou and S. Devasia. *Preview-based stable-inversion for output tracking.* In American Control Conference, 1999. Proceedings of the 1999, volume 5, pages 3544–3548. IEEE, 1999.