TECHNISCHE UNIVERSITÄT WIEN

AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Swing-Up and Stabilization of a Spherical Inverted Pendulum on a Robot

DIPLOMA THESIS

Conducted in partial fulfillment of the requirements for the degree of a

Diplom-Ingenieur (Dipl.-Ing.)

supervised by

Univ.-Prof. Dr. techn. A. Kugi
Dipl.-Ing. C. Hartl-Nesic

submitted at the

TU Wien
Faculty of Electrical Engineering and Information Technology
Automation and Control Institute

by
Jens Kretschmer
Matriculation number 1025451
Hilschergasse 14/11
1120 Vienna
Austria

Vienna, January 2019

Preamble

I would like to express my very great appreciation to Univ.-Prof. Dr. techn. Andreas Kugi for giving me the opportunity to conduct this thesis. His lectures awakened my interest in control engineering and provided a solid foundation of knowledge, which I could utilize for this work. I would like to offer my special thanks to Dipl.-Ing. Christian Hartl-Nesic for sharing his knowledge and experience with me. I am grateful for the discussions we had and for the feedback he gave me. I would also like to thank him for providing a working robotic setup, so I could focus on the core topics of my thesis.

Advice on the mechanical design of the spherical pendulum given by Dipl.-Ing. Michael Schwegel and Dipl.-Ing. Gabriel Forstner was greatly appreciated. I wish to acknowledge Dr. techn. Tobias Glück for providing valuable insights to the swing-up of the triple inverted pendulum. Discussions with my fellow students, Dipl.-Ing. Ali Haddadi and Dipl.-Ing. Thomas Weingartshofer, were always insightful and a help in working with the robotic setup.

Finally, I would like to thank my parents and my brother for their support and encouragement throughout my study and while working on this thesis. I am grateful for all my friends, who always inspire me. This thesis would not have been possible without them.

Vienna, January 2019

Abstract

This thesis examines the swing-up and stabilization of a spherical inverted pendulum attached to a robot with seven degrees of freedom. A mechatronic design for a spherical inverted pendulum is proposed, which can be mounted on a KUKA LWR IV+ and other robots with a compatible end effector.

After deriving mathematical models for the spherical inverted pendulum, the robot and the complete system, an optimal control problem is solved to obtain the swing-up trajectory. The dynamic optimization problem is converted to a static optimization problem using a direct method. The swing-up trajectory is obtained in the course of a three-step process. First, the search for a swing-up trajectory is performed on a simpler model with fewer states and inputs. This trajectory is then used as an initial guess for the static optimization problem for the complete system, which is then solved to find a trajectory for the full problem. A time-variant LQR is designed to stabilize the system around the swing-up trajectory. After the successful swing-up, the controller switches to a stabilizing controller, which consists of two LQRs for stabilizing the pendulum and a Cartesian trajectory tracking controller.

The swing-up trajectory and the controllers are investigated in simulations as well as in a complete experimental setup using real-time capable industrial hardware. The robustness to model and parameter uncertainties, disturbances and calibration errors is verified.

Kurzzusammenfassung

Diese Arbeit befasst sich mit dem Aufschwingen und Stabilisieren eines sphärischen inversen Pendels, welches an einem Roboter mit sieben Freiheitsgraden befestigt ist. Ein mechatronisches Design für das sphärische inverse Pendel wird vorgestellt, welches die Befestigung an einem Kuka LWR IV+ und an anderen Robotern mit kompatiblem Endeffektor ermöglicht.

Es werden mathematische Modelle für das sphärische inverse Pendel, den Roboter und das komplette zusammengesetzte System hergeleitet. Die Aufschwingtrajektorie wird mittels eines Optimalsteuerungsproblems berechnet. Das dynamische Optimierungsproblem wird dabei mit einem direkten Verfahren in ein statisches Optimierungsproblem übergeführt. Die Aufschwingtrajetorie wird mithilfe eines dreistufigen Prozesses berechnet. Im ersten Schritt wird eine Aufschwingtrajektorie für ein einfacheres Modell mit weniger Zuständen und Eingängen ermittelt. Diese Lösung wird dann als Starttrajektorie für das statische Optimierungsproblem des kompletten zusammengesetzten Systems verwendet. Um das System um die Aufschwingtrajektorie zu stabilisieren, wird ein zeitvarianter LQR entworfen. Nach dem erfolgreichen Aufschwingen des Pendels wird auf einen Stabilisierungsregler umgeschaltet, welcher aus zwei LQRs, die das Pendel stabilisieren, und einem kartesischen Trajektorienfolgeregler besteht.

Die Aufschwingtrajektorie und die Regler werden in der Simulation und im experimentellen Aufbau, der echtzeitfähige Industriekomponenten verwendet, untersucht. Die Robustheit gegenüber Modellungenauigkeiten, Parameterschwankungen, Störgrößen und Kalibrierungsfehlern wird dabei verifiziert.

Contents

List of Figures

List of Tables

1 Introduction

Inverted pendulums are popular systems in control engineering, as they are under-actuated nonlinear systems with unstable equilibrium postures. Originally, inverted pendulums gained research interest as they are good models for designing attitude controllers for the vertical take-off of rockets [1, 2]. During take-off, rockets are extremely unstable as the airspeed is too small for aerodynamic stability and thus require a controller to stay upright, just as an inverted pendulum [3]. Inverted pendulums are also good models for automatic aircraft landing systems, aircraft stabilization in turbulent air-flow, stabilization of a cabin in a ship and humanoid walking control [1, 4]. In humanoid walking control, the robot can be approximated by an inverted pendulum on a cart. The pendulum is deflected from the equilibrium posture into the correct direction to enable the motion. Trajectory tracking control needs to be applied to stabilize the robot around the gait pattern [5].

1.1 Literature and Previous Works

There are many different experimental implementations of inverted pendulums, where the most common are the single arm rotary inverted pendulum, the inverted pendulum on a cart and the double inverted pendulum. An overview of inverted pendulum (IP) types is given in Table 1.1. The parallel type dual inverted pendulum, mentioned in this table, consists of two pendulums on carts and a bar connecting the top ends of the two inverted pendulums.

Name	Degree of under-actuation	Actuator	Paper
single arm rotary IP	1	rotary	[6]
IP on a cart	1	linear	[7]
double IP	2	linear	[8]
two-link rotary IP	2	rotary	[9]
parallel type dual IP	2	linear	[10]
spherical IP	2	linear or rotary	[11, 12]
3D pendulum	3	linear	[13]
triple IP	3	linear	[14]
quadruple IP	4	linear	[15]

Table 1.1: Inverted pendulum types.

The research topics of inverted pendulums can be divided into swing-up control, stabilization control, switching control, and trajectory tracking control. The purpose of swing-up control is to move the pendulum from the downward equilibrium posture to the upward equilibrium posture. Swing-up control is especially interesting and challenging as many standard techniques of nonlinear control are ineffective. The system is neither input-output linearizable, nor feedback linearizable and the controllability distribution does not have constant rank [16]. Stabilization control has the purpose of stabilizing the pendulum around the upper equilibrium posture. In many cases, the swing-up controller is switched to the stabilizing controller after a successful swing-up, which requires switching control. However, single controllers solving both problems have also been proposed [17]. Trajectory tracking control is used to move the base point of the pendulum along a trajectory, while the pendulum remains at the unstable position [18, 19].

Due to its properties, the inverted pendulum is a popular benchmarking system in control engineering. Therefore, many control concepts have been applied to inverted pendulums. For swing-up control, energy-based [6], fuzzy logic [20] and time optimal control [21] are often used. An extensive list of swing-up controllers is given in [18]. For energy-based swing-up control, a nonlinear feedback law is used to pump energy into the system until the upper equilibrium posture is reached. The asymptotic stability can be proven for energy-based swing-up control, as performed in [6, 22]. Fuzzy logic control uses linguistic rules to describe the control action. Stability is hard to confirm, but fuzzy logic control is easy to implement and computationally less demanding. Combined with optimization methods such as genetic algorithms and particle swarm optimization, it is robust against noise and disturbances and can also outperform energy-based controllers in terms of steady-state error, settling time, rise time, and maximum overshoot [23]. Compared to energy-based swing-up, the optimal swing-up control takes less time. Additionally, it is possible to choose a terminal state, as long as it is suitable. However, time optimal control is not robust to parameter perturbations [21].

In cases where a separate stabilizing controller is used, switching is performed to the stabilizing controller when the pendulum is near the upright equilibrium posture. These controllers are mostly based on a linear quadratic regulator (LQR) designed for the linearized system in the upright equilibrium posture or on a partially linearized system [12, 24]. Adaptive controllers [25], multiobjective integral sliding mode controllers [26], fuzzy logic regulators [27], pole placement techniques [28], and optimized PID controllers [29] have also been proposed in the literature.

The stabilization of a spherical inverted pendulum can be divided into approaches which use an LQR and other ones. A cascade controller consisting of an LQR for pendulum stabilization, inverse kinematics, and an inverse dynamics controller to compensate for the robot's dynamics is applied in [30]. A similar approach, but with a decentralized joint acceleration controller, is used in [31, 32] and another variation utilizing PID controllers to control the robot joints is presented in [33]. Other control concepts for stabilization of a spherical inverted pendulum include an LQR in combination with feedback linearization [12] and a plain LQR [34, 35]. A global stabilization method which uses the forwarding technique in combination with an LQR is proposed in [36]. An approach which does not use an LQR but an adaptive backstepping controller is presented in [37].

To the best of the author's knowledge, the swing-up of a spherical inverted pendulum has so far merely been performed in simulations with passivity-based control for a fully-actuated spherical inverted pendulum with rotary actuators [38, 39]. This system differs from the systems usually examined for swing-up control, as those are under-actuated. A further reference performs the swing-up in simulation and experiments where the controller design relies on passivity control and the forwarding technique for a spherical inverted pendulum mounted on a custom-built robot with three rotary joints [11]. Neither the combination of an industrial robot with a spherical inverted pendulum nor optimal control for swinging up a spherical inverted pendulum have been proposed in the literature so far.

A detailed overview of previous research results regarding the control of inverted pendulums can be found in [18, 19].

1.2 Aim of this Thesis

To date there have been experiments with a spherical inverted pendulum in combination with an omnidirectional mobile robot [37], a two-link (SCARA) robot [30], a three-link robot [11] and a redundant three-link robot [31, 32]. However, this work aims at using a robot with seven degrees of freedom (DOF) to operate a spherical inverted pendulum, which has not been proposed in literature so far, at least to the best of the author's knowledge. While passivity-based approaches for swing-up have been used before, this thesis proposes an optimal control approach for the swing-up of a spherical inverted pendulum. This approach has the benefit of incorporating the robot kinematics, dynamics, and the kinematic and dynamic limits, which were neglected in [11]. As the spherical inverted pendulum has two non-actuated DOF, the system has a degree of under-actuation of two. Therefore, this system constitutes a challenging combination of robotic control and the classical benchmarking system of an inverted pendulum. For stabilization, a cascade controller consisting of two LQRs for the pendulum and a Cartesian trajectory controller is used. It should be noted that [30–32, 40] use a similar approach for stabilization as proposed in this thesis.

1.3 Overview of this Thesis

First, a short overview of the system and the mechanical design of the pendulum is given in Chapter 2. Subsequently, the kinematics and dynamics of the spherical inverted pendulum with Cartesian input, the kinematics of the robot, as well as the kinematics and dynamics of the complete system with torque input are derived. The equilibrium postures of the pendulum are calculated and discussed.

The swing-up trajectory is calculated by solving an optimal control problem, which is described in Chapter 3. To ensure convergence to a feasible solution, the problem is solved in a three-step process. First, the swing-up trajectory is found for the model of the spherical inverted pendulum with Cartesian input, which has fewer DOF and is suitable to find the first physical solution. This trajectory is then converted into a trajectory of the complete system with torque input using inverse kinematics and forward dynamics. By employing the latter trajectory as an initial guess for the optimization problem of the complete system with torque input, a swing-up trajectory for the whole setup is found.

To ensure robustness to parameter perturbations, an LQR for the swing-up and a cascade controller are designed in Chapter 4. The controller is switched to a cascade controller as soon as the pendulum is near the upper equilibrium posture, which is also presented in this chapter.

Simulations of the setup with calibration errors and disturbances acting on the system are performed. The simulation results are discussed in Chapter 5, followed by the evaluation of the experimental results in Chapter 6.

2 Design and Mathematical Modeling

This chapter introduces the robotic system and the pendulum tool, which are used in this thesis. First, an overview of the system is given. Then, the design and the mathematical model of the spherical inverted pendulum are discussed, resulting in the equations of motion. These equations are linearized around an equilibrium posture, since the result is required for the controller design. Finally, the kinematics of the robot as well as the dynamics of the complete system are derived. The resulting models are used to calculate the swing-up trajectory for the spherical inverted pendulum separately and for the complete system. The equations of motion of the complete system are also used for the controller design in Chapter 4 and for the simulations in Chapter 5.

2.1 System Overview

The system consists of the 7-axis robot KUKA LWR IV+ with a custom-built spherical inverted pendulum mounted on its flange, which is depicted in Figure 2.1. The spherical inverted pendulum is composed of two axes, which intersect in one point to allow for motions of the pendulum tip on a sphere. Each of the axes of the spherical inverted pendulum is equipped with a magnetic angular encoder to measure the angle and the angular velocity. The combination of the robot and the spherical inverted pendulum results in a system with nine degrees of freedom and seven inputs.

2.2 Design and Model of the Spherical Inverted Pendulum

First, the mechanical design of the spherical inverted pendulum is introduced in this chapter and described in detail. Next, a mathematical model of the spherical inverted pendulum with Cartesian input is derived. After calculating the kinematics, the equations of motion are derived. The equilibrium postures of the model are calculated and then used to linearize the equations of motion.

2.2.1 Mechanical Design

The basic components of the mechanical design of the spherical inverted pendulum are a mounting enclosure and two perpendicular intersecting placed axes, which are shown in Figure 2.2.

The mounting enclosure (2) is connected to the flange of the robot with seven hexagon socket head cap screws (1). Axis 8 (8) and Axis 9 (9) are mounted to rotate freely using capped single row deep groove ball bearings, which are chosen for minimum friction [41]. A locating/non-locating bearing arrangement is used for both axes to compensate for axial

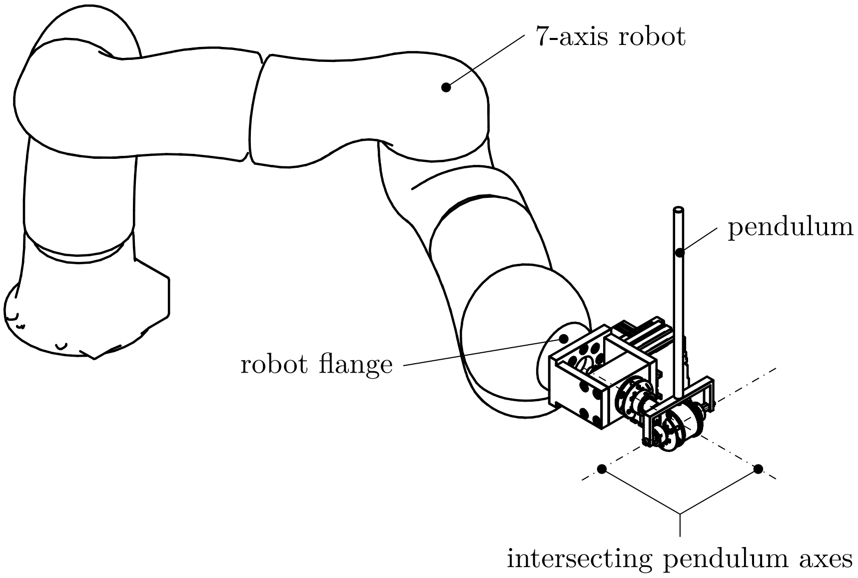

Figure 2.1: Robot with the spherical inverted pendulum.

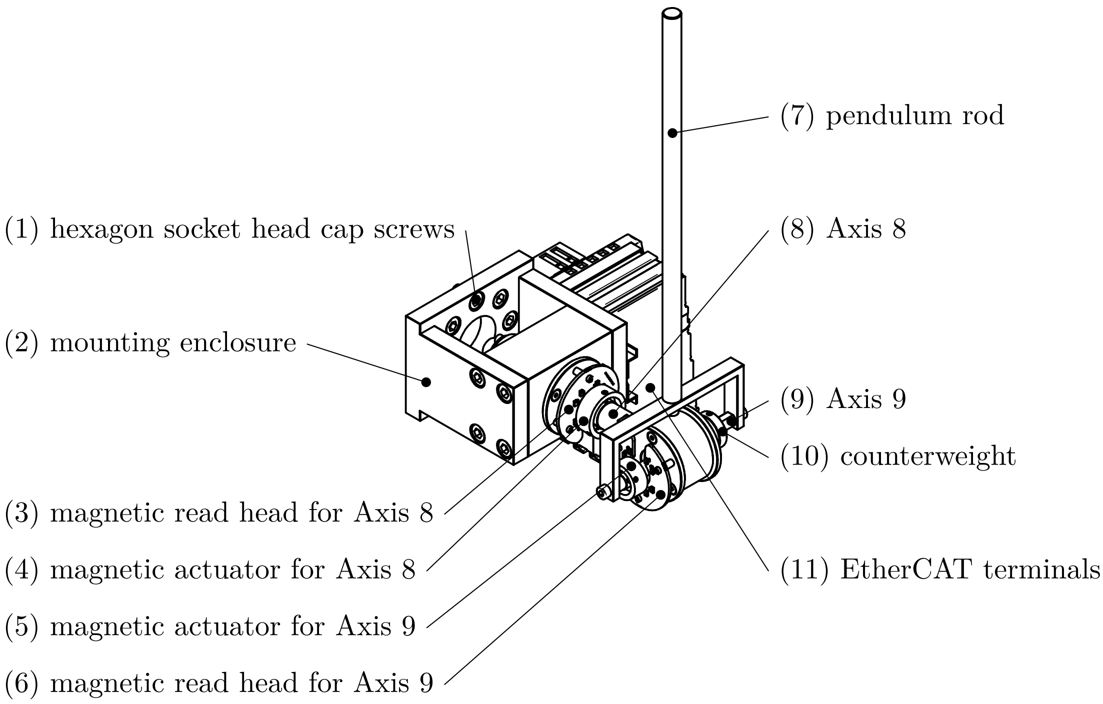

Figure 2.2: Components of the spherical inverted pendulum.

displacements and for the accumulation of tolerances of the components [41, 42]. Axis 8 (8) has a boring along its rotation axis to install the cables of the rotary encoder of Axis 9 (6). At the end of the boring, a slip ring is mounted to allow for infinite travel between the mounting enclosure and Axis 8. Axis 9 (9) including its bearing arrangement and rotary encoder is asymmetric. Therefore, a counterweight (10) is used to move the center of mass of Axis 9 onto the longitudinal axis of the rod (7) to ensure that the vertical position of the rod is an equilibrium posture.

Two magnetic rotary encoders are used to measure the angles of the magnetic actuators (4) and (5) relative to the respective read head (3) and (6). One magnetic actuator is mounted on each axis. The read head for Axis 8 is mounted on the mounting enclosure and the read head for Axis 9 is placed on Axis 8. Three EtherCAT terminals (11) are attached to the mounting enclosure, which read out the two rotary encoders.

2.2.2 Kinematics

System

The generalized coordinates of the system are the two degrees of freedom of the spherical inverted pendulum (subscripted by "p" for pendulum)

$$\mathbf{q}_p = \begin{bmatrix} q_8 \\ q_9 \end{bmatrix} . \tag{2.1}$$

The input of the system is composed of the Cartesian acceleration of the tool attachment point

$$\mathbf{u}_p = \ddot{\mathbf{r}} = \begin{bmatrix} \ddot{r}_x \\ \ddot{r}_y \\ \ddot{r}_z \end{bmatrix} , \tag{2.2}$$

which is located in the center of the plane of the mounting enclosure facing the robot. In the further text, results are only stated where they benefit the comprehension of the calculations. The calculations including the complete results can be found on the attached CD-ROM.

Coordinate Frames

The coordinate frames of the spherical inverted pendulum with Cartesian input are shown in Figure 2.3. The space-fixed coordinate frame is denoted by $(0_0 x_0 y_0 z_0)$, the frame of the tool attachment point is described by $(0_t x_t y_t z_t)$ with the origin $\mathbf{r}$. The orientation is fixed such that z_t is parallel to x_0, y_t is parallel to y_0 and x_t is antiparallel to z_0. Hence the transformation between $(0_0 x_0 y_0 z_0)$ and $(0_t x_t y_t z_t)$ is described by the homogeneous transformation matrix

$$\mathbf{T}^t_{0,p} = \begin{bmatrix} 0 & 0 & 1 & r_x \\ 0 & 1 & 0 & r_y \\ -1 & 0 & 0 & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{2.3}$$

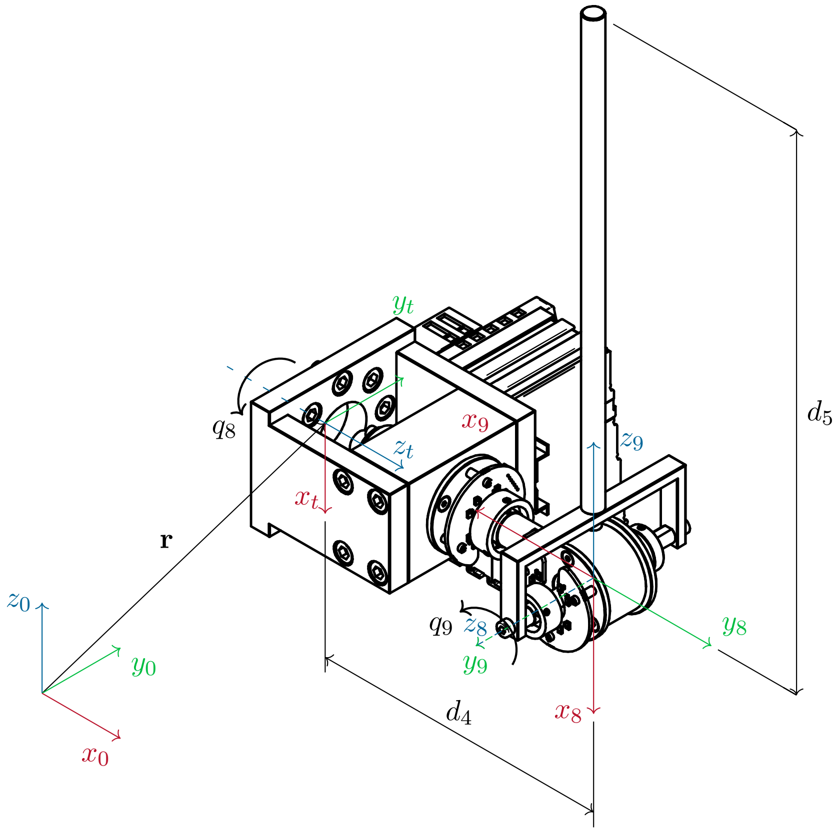

Figure 2.3: Coordinate frames of the spherical inverted pendulum with Cartesian input.

The kinematics of the pendulum is described by the Denavit-Hartenberg parameters [43] given in Table 2.1 and the parameterized homogeneous transformation matrix

$$\mathbf{T}(a, \alpha, b, \theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) & \cos(\theta)a \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) & \sin(\theta)a \\ 0 & \sin(\alpha) & \cos(\alpha) & b \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{2.4}$$

i	a_i	α_i	b_i	θ_i
8	0	$\frac{\pi}{2}$	d_4	q_8
9	0	$\frac{\pi}{2}$	0	$q_9 - \frac{\pi}{2}$

Table 2.1: Denavit-Hartenberg parameters of the spherical inverted pendulum.

The homogeneous transformation matrices between the coordinate frames $(0_0 x_0 y_0 z_0)$ and $(0_8 x_8 y_8 z_8)$, and $(0_0 x_0 y_0 z_0)$ and $(0_9 x_9 y_9 z_9)$ are calculated by the composition of the transformation matrices (2.3) and (2.4)

$$\mathbf{T}_{0,p}^{8} = \mathbf{T}_{0,p}^{t}\mathbf{T}(a_8, \alpha_8, b_8, \theta_8) \tag{2.5a}$$

$$\mathbf{T}_{0,p}^{9} = \mathbf{T}_{0,p}^{8}\mathbf{T}(a_9, \alpha_9, b_9, \theta_9) \; . \tag{2.5b}$$

Translation

The homogeneous vectors of the center of mass in the body-fixed coordinate frames

$$\mathbf{P}_i^{ic} = \begin{bmatrix} c_{ix} \\ c_{iy} \\ c_{iz} \\ 1 \end{bmatrix} , \qquad i = 8, 9 \tag{2.6}$$

are transformed using

$$\mathbf{P}_{0,p}^{ic} = \mathbf{T}_{0,p}^{i}\mathbf{P}_i^{ic} \, , \qquad i = 8, 9 \tag{2.7}$$

into the space-fixed coordinate frame. The vectors of the center of mass are extracted from the homogeneous vectors (2.7) by taking the first three entries

$$\mathbf{p}_{0,p}^{ic} = \mathbf{P}_{0,p}^{ic}[1 \ldots 3] \, , \qquad i = 8, 9 \; . \tag{2.8}$$

The linear velocities $\mathbf{v}$ are the time derivative of these position vectors

$$\mathbf{v}_{0,p}^{ic} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p}_{0,p}^{ic} \, , \qquad i = 8, 9 \; . \tag{2.9}$$

Rotation

Next, the rotation matrices are extracted from the homogeneous transformation matrices (2.5)

$$\mathbf{R}^i_{0,p} = \mathbf{T}^i_{0,p}[1\dots3, 1\dots3]\,, \qquad i = 8, 9\ . \tag{2.10}$$

To receive the angular velocity, the skew-symmetric operator $\mathbf{S}(\omega)$ is calculated by

$$\mathbf{S}\left(\omega^i_{0,p}\right) = \left(\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{R}^i_{0,p}\right)\left(\mathbf{R}^i_{0,p}\right)^{\mathrm{T}}, \qquad i = 8, 9\ . \tag{2.11}$$

As the operator $\mathbf{S}(\omega)$ has the form

$$\mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \tag{2.12}$$

the elements of the angular velocities are extracted from this equation.

Manipulator Jacobian

The Jacobian of the end of the rod of the spherical inverted pendulum referenced in the space-fixed coordinate frame is calculated by

$$\mathbf{J}^e_{0\mathbf{v},p} = \frac{\partial \mathbf{p}^e_{0,p}}{\partial \mathbf{q}_p} \tag{2.13}$$

with $\mathbf{p}^e_{0,p}$ being the position vector between the space-fixed coordinate frame $(0_0 x_0 y_0 z_0)$ and the end of the rod of the spherical inverted pendulum calculated by

$$\mathbf{P}^e_9 = \begin{bmatrix} 0 \\ 0 \\ d_5 \\ 1 \end{bmatrix} \tag{2.14a}$$

$$\mathbf{P}^e_{0,p} = \mathbf{T}^9_{0,p}\mathbf{P}^e_9 \tag{2.14b}$$

$$\mathbf{p}^e_{0,p} = \mathbf{P}^e_{0,p}[1\dots3]\ . \tag{2.14c}$$

2.2.3 Dynamics

The equations of motion of the spherical inverted pendulum with a Cartesian input are derived using the Euler-Lagrange equations [43]. Friction is neglected in this model.

The results from the previous chapter are used to calculate the kinetic and potential energies of the spherical inverted pendulum. The symmetric inertia tensors $\mathbf{I}_i$ for the individual rigid bodies are composed of

$$\mathbf{I}_i = \begin{bmatrix} I_{i,xx} & I_{i,xy} & I_{i,xz} \\ I_{i,xy} & I_{i,yy} & I_{i,yz} \\ I_{i,xz} & I_{i,yz} & I_{i,zz} \end{bmatrix}, \qquad i = 8, 9 \tag{2.15}$$

and are given in the body-fixed coordinate frames. Therefore, they have to be transformed into the space-fixed coordinate frame to calculate the rotatory kinetic energy. The mass of a link is denoted by m_i.

The kinetic energy T is the sum of the translatory and the rotatory kinetic energies of all links

$$T_p = \sum_{i=8,9} \frac{1}{2}\left(\boldsymbol{\omega}^i_{0,p}\right)^{\mathrm{T}} \mathbf{R}^i_{0,p}\mathbf{I}_i\left(\mathbf{R}^i_{0,p}\right)^{\mathrm{T}}\boldsymbol{\omega}^i_{0,p} + \frac{1}{2}m_i\left(\mathbf{v}^{ic}_{0,p}\right)^{\mathrm{T}}\mathbf{v}^{ic}_{0,p} \ . \tag{2.16}$$

Using the vectors of the center of mass (2.8), the potential energy is calculated by

$$V_p = \sum_{i=8,9} \begin{bmatrix} 0 & 0 & g \end{bmatrix} \mathbf{p}^{ic}_{0,p} m_i \ , \tag{2.17}$$

with the acceleration of gravity g.

Next, disturbance forces

$$\mathbf{F}_{ext} = \begin{bmatrix} F_x \\ F_y \\ 0 \end{bmatrix} \tag{2.18}$$

are introduced to the model of the spherical inverted pendulum. F_x and F_y are parallel to x_0 and y_0 respectively, as depicted in Figure 2.4. Using (2.13) the corresponding generalized force results in

$$\boldsymbol{\tau}_{ext,p} = \left(\mathbf{J}^e_{0\mathbf{v},p}\right)^{\mathrm{T}} \mathbf{F}_{ext} \ . \tag{2.19}$$

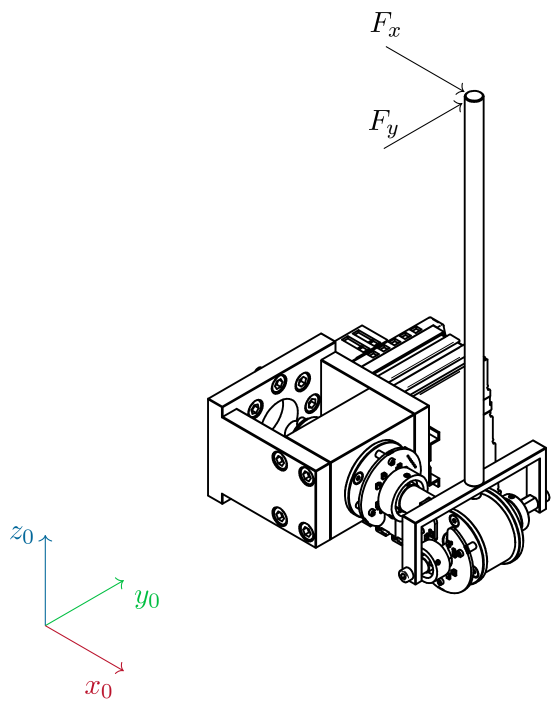

Figure 2.4: Disturbance force acting on the spherical inverted pendulum.

The equations of motion are finally calculated in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L_p}{\partial \dot{\mathbf{q}}_p}\right)^{\mathrm{T}} - \left(\frac{\partial L_p}{\partial \mathbf{q}_p}\right)^{\mathrm{T}} = \boldsymbol{\tau}_{ext,p} \tag{2.20}$$

with the Lagrangian

$$L_p = T_p - V_p \ . \tag{2.21}$$

2.2.4 Equilibrium Posture

By setting all time derivatives and the disturbance forces in (2.20) to zero, the equilibrium postures of the spherical inverted pendulum result from

$$\left.\left(\frac{\partial L_p}{\partial \mathbf{q}_p}\right)^{\mathrm{T}}\right|_{\substack{\ddot{\mathbf{q}}_p=\mathbf{0}\\ \dot{\mathbf{q}}_p=\mathbf{0}\\ \mathbf{q}_p=\mathbf{q}_{e,p}\\ \ddot{\mathbf{r}}=\mathbf{0}}} = \mathbf{0} \ , \tag{2.22}$$

which are two equations in two unknowns, q_{8e} and q_{9e}. The resulting six sets of equilibrium postures of the model are

$$q_{8e1} = \pi Z_1 \qquad q_{9e1} = 2\pi Z_2 \tag{2.23a}$$
$$q_{8e2} = \pi Z_3 \qquad q_{9e2} = \pi + 2\pi Z_4 \tag{2.23b}$$
$$q_{8e3} = \frac{\pi}{2} + 2\pi Z_5 \qquad q_{9e3} = \frac{\pi}{2} + 2\pi Z_6 \tag{2.23c}$$
$$q_{8e4} = \frac{\pi}{2} + 2\pi Z_7 \qquad q_{9e4} = -\frac{\pi}{2} + 2\pi Z_8 \tag{2.23d}$$
$$q_{8e5} = -\frac{\pi}{2} + 2\pi Z_9 \qquad q_{9e5} = \frac{\pi}{2} + 2\pi Z_{10} \tag{2.23e}$$
$$q_{8e6} = -\frac{\pi}{2} + 2\pi Z_{11} \qquad q_{9e6} = -\frac{\pi}{2} + 2\pi Z_{12} \ , \tag{2.23f}$$

with $Z_i \in \mathbb{Z}, i \in \{1, \ldots, 12\}$. Due to the periodicity of the trigonometric functions there are infinite solutions which are physically identical. Although a counterweight is used, the resulting equilibrium postures of the spherical inverted pendulum have small deviations from the exact vertical (max. $2 \cdot 10^{-8}$ rad) and exact horizontal position (max. $8 \cdot 10^{-4}$ rad), due to the asymmetric mechanical design of the spherical inverted pendulum. The equilibrium postures are given rounded here, for the sake of simplicity.

The equilibrium postures 3 to 6 describe postures where the rod of the spherical inverted pendulum is horizontal. Exemplarily, posture 6 is depicted in Figure 2.5. While postures 3 and 5 cannot be reached, as the rod of the spherical inverted pendulum would intersect Axis 8 and the mounting enclosure, postures 4 and 6 are actual equilibrium postures. The lever arm of the gravitational force is zero around Axis 8 and the gravitational force is parallel to Axis 9, resulting in zero momentum acting on the pendulum.

The remaining equilibrium postures 1 and 2 describe the vertical position of the rod. While both sets of postures include postures where the pendulum rod is both standing and hanging, they are not physically identical, due to the asymmetric design of the spherical inverted pendulum. Aside from this, the rotational direction of q_9 in those two different sets of postures differs. The solutions

$$\mathbf{q}_{eu} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.24a}$$

$$\mathbf{q}_{el} = \begin{bmatrix} \pi \\ 0 \end{bmatrix} \tag{2.24b}$$

for the standing and hanging equilibrium posture, respectively, are chosen and are further on used for swinging up the pendulum and stabilizing it.

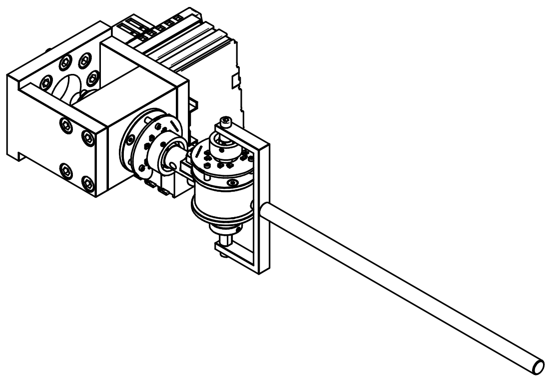

Figure 2.5: Equilibrium posture 6 of the spherical inverted pendulum.

2.2.5 Linearization of the Equations of Motion

The equations of motion (2.20) are rearranged in state-space form

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_p = \mathbf{f}_p(\mathbf{x}_p, \mathbf{u}_p, \mathbf{d}_p) \tag{2.25}$$

with the state

$$\mathbf{x}_p = \begin{bmatrix} \mathbf{q}_p \\ \dot{\mathbf{q}}_p \\ \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix}, \tag{2.26}$$

the input (2.2) and the disturbance input $\mathbf{d}_p = \mathbf{F}_{ext}$. The state-space representation (2.25) is completed by the equations $\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = \dot{\mathbf{r}}$ and $\frac{\mathrm{d}\dot{\mathbf{r}}}{\mathrm{d}t} = \ddot{\mathbf{r}}$. The equations (2.25) are linearized around an equilibrium posture from (2.23)

$$\Delta\dot{\mathbf{x}}_p = \mathbf{A}_p\Delta\mathbf{x}_p + \mathbf{B}_p\Delta\mathbf{u}_p + \mathbf{B}_{d,p}\Delta\mathbf{d}_p \tag{2.27a}$$

$$\mathbf{A}_p = \left.\frac{\partial\mathbf{f}(\mathbf{x}_p,\mathbf{u}_p,\mathbf{d}_p)}{\partial\mathbf{x}_p}\right|_{\mathbf{x}_p=\mathbf{x}_{e,p},\mathbf{u}_p=\mathbf{0},\mathbf{d}_p=\mathbf{0}} \tag{2.27b}$$

$$\mathbf{B}_p = \left.\frac{\partial\mathbf{f}(\mathbf{x}_p,\mathbf{u}_p,\mathbf{d}_p)}{\partial\mathbf{u}_p}\right|_{\mathbf{x}_p=\mathbf{x}_{e,p},\mathbf{u}_p=\mathbf{0},\mathbf{d}_p=\mathbf{0}} \tag{2.27c}$$

$$\mathbf{B}_{d,p} = \left.\frac{\partial\mathbf{f}(\mathbf{x}_p,\mathbf{u}_p,\mathbf{d}_p)}{\partial\mathbf{d}_p}\right|_{\mathbf{x}_p=\mathbf{x}_{e,p},\mathbf{u}_p=\mathbf{0},\mathbf{d}_p=\mathbf{0}}, \tag{2.27d}$$

with the deviations from the equilibrium posture

$$\Delta\mathbf{x}_p = \mathbf{x}_p - \mathbf{x}_{e,p} \tag{2.28a}$$

$$\Delta\mathbf{u}_p = \mathbf{u}_p \tag{2.28b}$$

$$\Delta\mathbf{d}_p = \mathbf{d}_p \ . \tag{2.28c}$$

2.3 Kinematics of the Robot

The coordinate frames of the robot are depicted in Figure 2.6. They are arranged according to the Denavit-Hartenberg convention, with the Denavit-Hartenberg parameters given in Table 2.2. The space-fixed coordinate frame located in the first joint and the coordinate frame of the tool attachment point of the robot are denoted by $(0_0x_0y_0z_0)$ and $(0_tx_ty_tz_t)$, respectively.

i	a_i	α_i	b_i	θ_i
1	0	$\frac{\pi}{2}$	0	q_1
2	0	$-\frac{\pi}{2}$	0	q_2
3	0	$-\frac{\pi}{2}$	d_1	q_3
4	0	$\frac{\pi}{2}$	0	q_4
5	0	$\frac{\pi}{2}$	d_2	q_5
6	0	$-\frac{\pi}{2}$	0	q_6
7	0	0	0	q_7
t	0	0	d_3	0

Table 2.2: Denavit-Hartenberg parameters of the robot.

To formulate the kinematic relationship between the space-fixed coordinate frame and the tool attachment point of the robot, the corresponding position vectors $\mathbf{p}^i_{0,r}$ (subscripted by "r" for robot), linear velocities $\mathbf{v}^i_{0,r}$ and angular velocities $\omega^i_{0,r}$ (2.6) to (2.9) with $i \in \{1,\ldots,7,t\}$ are calculated. The corresponding Jacobian of the tool attachment point

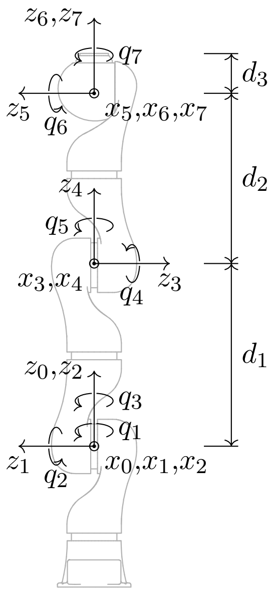

Figure 2.6: Coordinate frames of the robot for $\mathbf{q}_r = \mathbf{0}$.

with respect to the space-fixed coordinate frame results from

$$\mathbf{J}^t_{0,r} = \begin{bmatrix} \mathbf{J}^t_{0\mathbf{v},r} \\ \mathbf{J}^t_{0\boldsymbol{\omega},r} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{p}^t_{0,r}}{\partial \mathbf{q}_r} \\ \frac{\partial \boldsymbol{\omega}^t_{0,r}}{\partial \dot{\mathbf{q}}_r} \end{bmatrix} , \tag{2.29}$$

with the generalized coordinates of the robot

$$\mathbf{q}_r = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 \end{bmatrix}^{\mathrm{T}} . \tag{2.30}$$

2.4 Model of the Complete System

In this section, the model of the complete system is derived. The generalized coordinates $\mathbf{q}_s$ (subscripted by "s" for system) and the inputs $\mathbf{u}_s$ of the complete system read as

$$\mathbf{q}_s = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 & q_7 & q_8 & q_9 \end{bmatrix}^{\mathrm{T}} \tag{2.31}$$

and

$$\mathbf{u}_s = \begin{bmatrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 \end{bmatrix}^{\mathrm{T}} , \tag{2.32}$$

respectively.

2.4.1 Kinematics

The coordinate frames of the system are depicted in Figure 2.7 (compare Figure 2.3 and Figure 2.6). The Denavit-Hartenberg parameters of the complete system are given in Table 2.3, which is a composition of Table 2.1 and Table 2.2.

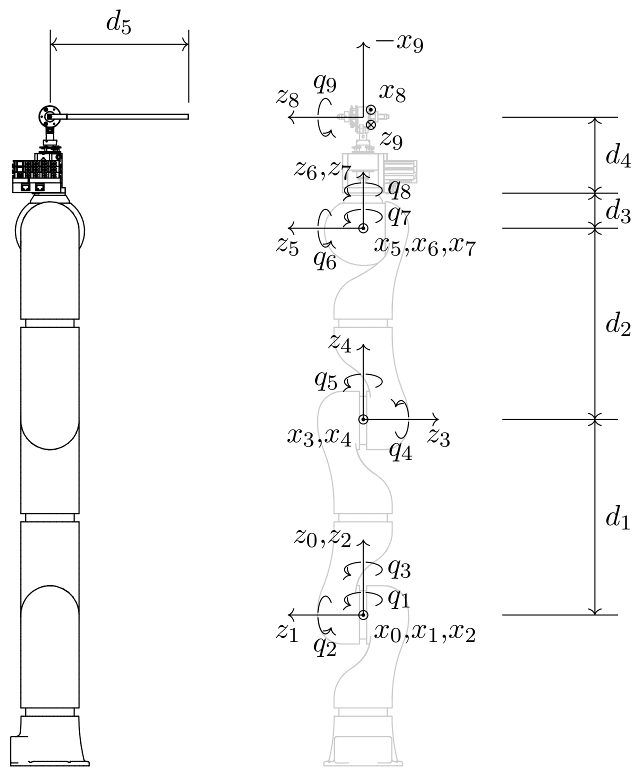

Figure 2.7: Coordinate frames of the complete system for $\mathbf{q}_s = \mathbf{0}$.

i	a_i	α_i	b_i	θ_i
1	0	$\frac{\pi}{2}$	0	q_1
2	0	$-\frac{\pi}{2}$	0	q_2
3	0	$-\frac{\pi}{2}$	d_1	q_3
4	0	$\frac{\pi}{2}$	0	q_4
5	0	$\frac{\pi}{2}$	d_2	q_5
6	0	$-\frac{\pi}{2}$	0	q_6
7	0	0	0	q_7
t	0	0	d_3	0
8	0	$\frac{\pi}{2}$	d_4	q_8
9	0	$\frac{\pi}{2}$	0	$q_9 - \frac{\pi}{2}$

Table 2.3: Denavit-Hartenberg parameters of the complete system.

Equations (2.6) to (2.9) with $i \in \mathcal{C}$, $\mathcal{C} = \{1, \ldots, 7, t, 8, 9\}$ give the position vectors of the center of mass $\mathbf{p}^{ic}_{0,s}$, the linear velocities of the center of mass $\mathbf{v}^{ic}_{0,s}$ and the angular velocities $\boldsymbol{\omega}^{i}_{0,s}$. Note that the center of mass $\mathbf{P}^{7c}_{7}$, mass m_7 and inertia tensor $\mathbf{I}_7$ are composed of the values of the robot and the mounting enclosure of the spherical inverted pendulum. The manipulator Jacobians are given by

$$\mathbf{J}^{ic}_{0\mathbf{v},s} = \frac{\partial \mathbf{p}^{ic}_{0,s}}{\partial \mathbf{q}_s} , \qquad i \in \mathcal{C} \tag{2.33}$$

$$\mathbf{J}^{i}_{0\boldsymbol{\omega},s} = \frac{\partial \boldsymbol{\omega}^{i}_{0,s}}{\partial \dot{\mathbf{q}}_s} , \qquad i \in \mathcal{C} \ . \tag{2.34}$$

2.4.2 Dynamics

This section derives the rigid-body dynamics using the results from the previous section to assemble the mass matrix $\mathbf{D}$, the Coriolis matrix $\mathbf{C}$ and the gravity vector $\mathbf{g}$ [43]. Friction is neglected in this model. By using the results from the previous section, the symmetric and positive definite mass matrix $\mathbf{D}_s$, which consists of translatory and rotatory components, is given by

$$\mathbf{D}_s(\mathbf{q}_s) = \sum_{i \in \mathcal{C}} m_i \left(\mathbf{J}^{ic}_{0\mathbf{v},s}\right)^{\mathrm{T}} \mathbf{J}^{ic}_{0\mathbf{v},s} + \left(\mathbf{J}^{i}_{0\boldsymbol{\omega},s}\right)^{\mathrm{T}} \mathbf{R}^{i}_{0,s} \mathbf{I}_i \left(\mathbf{R}^{i}_{0,s}\right)^{\mathrm{T}} \mathbf{J}^{i}_{0\boldsymbol{\omega},s} \ . \tag{2.35}$$

Next, the elements of the matrix $\mathbf{C}_s$ result from

$$\mathbf{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)[k,j] = \sum_{i=1}^{9} \frac{1}{2} \left(\frac{\partial \mathbf{D}_s(\mathbf{q}_s)[k,j]}{\partial q_i} + \frac{\partial \mathbf{D}_s(\mathbf{q}_s)[k,i]}{\partial q_j} - \frac{\partial \mathbf{D}_s(\mathbf{q}_s)[i,j]}{\partial q_k} \right) \dot{q}_i \ , \tag{2.36}$$

Using

$$V_s = \sum_{i=1}^{9} \begin{bmatrix} 0 & 0 & g \end{bmatrix} \mathbf{p}^{ic}_{0,s} m_i \tag{2.37}$$

to calculate the potential energy V_s, the gravity vector $\mathbf{g}_s$ is derived by

$$\mathbf{g}_s(\mathbf{q}_s) = \left(\frac{\partial V_s(\mathbf{q}_s)}{\partial \mathbf{q}_s} \right)^{\mathrm{T}} \ . \tag{2.38}$$

The torque vector of each link consists of the torques of the drives attached to both ends of every link (except for link 7)

$$\mathbf{M}_1 = \begin{bmatrix} 0 \\ M_1 \\ -M_2 \end{bmatrix} , \quad \mathbf{M}_2 = \begin{bmatrix} 0 \\ -M_2 \\ -M_3 \end{bmatrix} , \quad \mathbf{M}_3 = \begin{bmatrix} 0 \\ -M_3 \\ -M_4 \end{bmatrix} , \quad \mathbf{M}_4 = \begin{bmatrix} 0 \\ M_4 \\ -M_5 \end{bmatrix} \tag{2.39a}$$

$$\mathbf{M}_5 = \begin{bmatrix} 0 \\ M_5 \\ -M_6 \end{bmatrix} , \quad \mathbf{M}_6 = \begin{bmatrix} 0 \\ -M_6 \\ -M_7 \end{bmatrix} , \quad \mathbf{M}_7 = \begin{bmatrix} 0 \\ 0 \\ M_7 \end{bmatrix} , \tag{2.39b}$$

which are then mapped to the corresponding generalized force by

$$\boldsymbol{\tau}_s = \sum_{i=1}^{7} \left(\mathbf{J}^i_{0\boldsymbol{\omega},s}\right)^{\mathrm{T}} \mathbf{R}^i_{0,s} \mathbf{M}_i = \begin{bmatrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 & M_7 & 0 & 0 \end{bmatrix}^{\mathrm{T}} . \tag{2.40}$$

The disturbance force $\mathbf{F}_{ext}$ as defined in (2.18) is also used for this system as disturbance input. Analogous to (2.19), the generalized force is calculated as

$$\boldsymbol{\tau}_{ext,s} = \left(\mathbf{J}^e_{0\mathbf{v},s}\right)^{\mathrm{T}} \mathbf{F}_{ext} \ . \tag{2.41}$$

Finally, the equations of motion are composed of

$$\mathbf{D}_s(\mathbf{q}_s)\ddot{\mathbf{q}}_s + \mathbf{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s + \mathbf{g}_s(\mathbf{q}_s) = \boldsymbol{\tau}_s + \boldsymbol{\tau}_{ext,s} \ . \tag{2.42}$$

The rigid-body equation (2.42) can be rearranged in state-space form by multiplying the equation with the inverse of the mass matrix $\mathbf{D}_s$, which is always invertible due to its positive definiteness, resulting in

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \mathbf{q}_s \\ \dot{\mathbf{q}}_s \end{bmatrix} = \dot{\mathbf{x}}_s = \mathbf{f}_s = \begin{bmatrix} \dot{\mathbf{q}}_s \\ (\mathbf{D}_s(\mathbf{q}_s))^{-1}(-\mathbf{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s - \mathbf{g}_s(\mathbf{q}_s) + \boldsymbol{\tau}_s + \boldsymbol{\tau}_{ext,s}) \end{bmatrix} . \tag{2.43}$$

2.4.3 Equilibrium Postures

The equilibrium postures are calculated from (2.42), by setting all time derivatives and disturbance forces to zero, which results in

$$\mathbf{g}_s(\mathbf{q}_{e,s}) = \boldsymbol{\tau}_{e,s} \ . \tag{2.44}$$

As the posture of the robot can be chosen freely, there is an infinite number of operating points in addition to the infinite number of equilibrium postures of the spherical inverted pendulum. If the robot adopts a posture such that the tool attachment point has the same orientation as in (2.3), the equilibrium postures of q_8 and q_9 are the same as in (2.23).

3 Swing-Up Trajectory

The swing-up trajectory of the spherical inverted pendulum is calculated in a three-step process by sequentially solving multiple optimization problems, using each solution as the initial guess for the next problem. This division into multiple optimization problems is necessary in order to obtain a feasible solution using a simpler model first and, further on, increase the problem complexity until a viable solution for the complete model is found. The first feasible solution is calculated for the model of the spherical inverted pendulum with Cartesian acceleration input from Section 2.2. This solution trajectory is converted into a trajectory for the model of the complete system with torque input from Section 2.4, using inverse kinematics and forward dynamics. This serves as an initial guess for the optimization problem of the complete model. The solution of the latter optimization problem is used for the simulations in Chapter 5 and the experiments in Chapter 6.

3.1 Spherical Inverted Pendulum with Cartesian Acceleration Input

The purpose of the optimization problem discussed in this section is to first find a feasible swing-up trajectory for a simpler problem. The optimization problem uses the model of the spherical inverted pendulum with Cartesian acceleration input, which is derived in Section 2.2. Compared to the model of the complete system, the model of the spherical inverted pendulum with a Cartesian acceleration input has a smaller number of generalized coordinates and inputs, which simplifies the search for a feasible solution when starting from an infeasible starting point. The resulting trajectory can be converted into a suitable initial guess for the subsequent optimization problem, as described in Section 3.3.

The search for a feasible swing-up trajectory is formulated as a dynamic optimization problem

$$
\begin{aligned}
\min_{\mathbf{u}(\cdot)} \quad & J(\mathbf{u}(\cdot)) = \int_{t_0}^{t_1} l(t, \mathbf{x}(t), \mathbf{u}(t)) \, \mathrm{d}t && (3.1a) \\
\text{s.t.} \quad & \dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}, \mathbf{u}) \qquad \mathbf{x}(t_0) = \mathbf{x}_0 && (3.1b) \\
& \phi(t_1, \mathbf{x}(t_1)) = \mathbf{0} && (3.1c) \\
& \mathbf{x}_{lb} \le \mathbf{x}(t) \le \mathbf{x}_{ub}, \qquad \forall t \in [t_0, t_1] && (3.1d) \\
& \mathbf{u}_{lb} \le \mathbf{u}(t) \le \mathbf{u}_{ub}, \qquad \forall t \in [t_0, t_1] \ , && (3.1e)
\end{aligned}
$$

which leads to the optimal input $\mathbf{u}(t)$ with respect to the cost functional $J(\mathbf{u}(\cdot))$, the dynamic equations (3.1b), the final condition (3.1c) and further inequality constraints (3.1d) and (3.1e). Instead of solving the infinite-dimensional optimal control problem (3.1), the direct method is used to derive a finite-dimensional static optimization problem. Therefore, a time grid t^i with N collocation points is introduced, which splits the optimization horizon $[t_0, t_1]$ into $N-1$ equidistant intervals

$$t^i = t_0 + i\frac{t_1 - t_0}{N-1} , \qquad i = 0, \dots, N-1 . \tag{3.2}$$

The states and inputs are only evaluated at the grid points and constitute the vector of optimization variables $\mathbf{y}_p \in \mathbb{R}^{(10+3)N}$,

$$\mathbf{y}_p = \left[\left(\mathbf{x}_p^0\right)^{\mathrm{T}} \quad \left(\mathbf{u}_p^0\right)^{\mathrm{T}} \quad \left(\mathbf{x}_p^1\right)^{\mathrm{T}} \quad \left(\mathbf{u}_p^1\right)^{\mathrm{T}} \quad \cdots \quad \left(\mathbf{x}_p^{N-1}\right)^{\mathrm{T}} \quad \left(\mathbf{u}_p^{N-1}\right)^{\mathrm{T}} \right]^{\mathrm{T}} , \tag{3.3}$$

with the state $\mathbf{x}_p^i = \mathbf{x}_p(t^i)$ and the input $\mathbf{u}_p^i = \mathbf{u}_p(t^i)$ from the system (2.25). The cost function and constraints of the dynamic optimization problem (3.1) are discretized accordingly, which transforms (3.1) to the static optimization problem

$$\min_{\mathbf{y}_p} \quad J_p(\mathbf{y}_p) \tag{3.4a}$$

$$\text{s.t.} \quad \mathbf{c}_{eq,p}(\mathbf{y}_p) = \mathbf{0} . \tag{3.4b}$$

This discretization is discussed in the following subsections.

While indirect methods have the benefit of providing solutions with high accuracy, it is difficult to initialize them and the domain of convergence is relatively small. Therefore, indirect methods are often initialized with a solution obtained from a direct method. Adding equality and inequality constraints is usually more difficult with indirect methods than with direct methods. Although direct methods only provide suboptimal solutions, it has been shown that the solution converges to the exact solution of the optimization problem if the number of collocation points is increased. Due to the aforementioned drawbacks of indirect methods, the direct method is applied for solving the optimal control problem. As the approach used in this thesis discretizes the states *and* inputs, it is called a direct simultaneous method. Compared to direct sequential methods, which discretize inputs alone, inequality constraints are handled more easily and the computational effort is lower, as no computational time is wasted with obtaining accurate state values for inputs/parameters which are far from their optimal values. This is especially beneficial for finding solutions where instabilities occur in the range of inputs or no solution exists for certain inputs, see, e. g., [44, 45].

3.1.1 Cost Function

By applying the trapezoidal rule to the cost functional (3.1a) the integral is discretized, thus the discretized cost function results in

$$J_p(\mathbf{y}_p) = \sum_{i=0}^{N-2} \frac{1}{2}\left(t^{i+1} - t^i\right)\left[l_p\left(t^i, \mathbf{x}_p^i, \mathbf{u}_p^i\right) + l_p\left(t^{i+1}, \mathbf{x}_p^{i+1}, \mathbf{u}_p^{i+1}\right)\right] . \tag{3.5}$$

The Lagrange function is chosen to be of quadratic form

$$\begin{aligned} l_p\left(t^i, \mathbf{x}_p^i, \mathbf{u}_p^i\right) =& \frac{1}{2}\Bigg(\left(\mathbf{x}_p^i - \mathbf{x}_{r,p}\right)^{\mathrm{T}}\mathbf{Q}_p\left(\mathbf{x}_p^i - \mathbf{x}_{r,p}\right) \\ &+ \left(\mathbf{u}_p^i - \mathbf{u}_{r,p}\right)^{\mathrm{T}}\mathbf{R}_p^i\left(\mathbf{u}_p^i - \mathbf{u}_{r,p}\right)\Bigg) , \end{aligned} \tag{3.6}$$

where the weighting matrices $\mathbf{Q}_p$ and $\mathbf{R}_p^i$ penalize the deviation from the reference values $\mathbf{x}_{r,p}$ and $\mathbf{u}_{r,p}$, respectively. Additionally, the time-dependent weighting matrix for the inputs $\mathbf{R}_p^i$ allows to adjust the weights for each instant of time. To decrease the search time and increase the accuracy, the gradient of the cost function is provided to the optimization algorithm, which is computed from (3.5) and leads to

$$
\begin{aligned}
\frac{\partial J_p}{\partial \mathbf{y}_p}(\mathbf{y}_p) = \Big[& \tfrac{1}{2}\left(t^1 - t^0\right) \tfrac{\partial l_p}{\partial \mathbf{x}_p}\left(t^0, \mathbf{x}_p^0, \mathbf{u}_p^0\right), \\
& \tfrac{1}{2}\left(t^1 - t^0\right) \tfrac{\partial l_p}{\partial \mathbf{u}_p}\left(t^0, \mathbf{x}_p^0, \mathbf{u}_p^0\right), \\
& \tfrac{1}{2}\left(t^2 - t^0\right) \tfrac{\partial l_p}{\partial \mathbf{x}_p}\left(t^1, \mathbf{x}_p^1, \mathbf{u}_p^1\right), \\
& \tfrac{1}{2}\left(t^2 - t^0\right) \tfrac{\partial l_p}{\partial \mathbf{u}_p}\left(t^1, \mathbf{x}_p^1, \mathbf{u}_p^1\right), \quad \cdots \\
& \tfrac{1}{2}\left(t^{N-1} - t^{N-3}\right) \tfrac{\partial l_p}{\partial \mathbf{x}_p}\left(t^{N-2}, \mathbf{x}_p^{N-2}, \mathbf{u}_p^{N-2}\right), \\
& \tfrac{1}{2}\left(t^{N-1} - t^{N-3}\right) \tfrac{\partial l_p}{\partial \mathbf{u}_p}\left(t^{N-2}, \mathbf{x}_p^{N-2}, \mathbf{u}_p^{N-2}\right), \\
& \tfrac{1}{2}\left(t^{N-1} - t^{N-2}\right) \tfrac{\partial l_p}{\partial \mathbf{x}_p}\left(t^{N-1}, \mathbf{x}_p^{N-1}, \mathbf{u}_p^{N-1}\right), \\
& \tfrac{1}{2}\left(t^{N-1} - t^{N-2}\right) \tfrac{\partial l_p}{\partial \mathbf{u}_p}\left(t^{N-1}, \mathbf{x}_p^{N-1}, \mathbf{u}_p^{N-1}\right)\Big] \; .
\end{aligned}
\tag{3.7}
$$

with

$$
\frac{\partial l_p}{\partial \mathbf{x}_p}\left(t^i, \mathbf{x}_p^i, \mathbf{u}_p^i\right) = \left(\mathbf{x}_p^i - \mathbf{x}_{r,p}\right)^{\mathrm{T}} \mathbf{Q}_p \tag{3.8a}
$$

$$
\frac{\partial l_p}{\partial \mathbf{u}_p}\left(t^i, \mathbf{x}_p^i, \mathbf{u}_p^i\right) = \left(\mathbf{u}_p^i - \mathbf{u}_{r,p}\right)^{\mathrm{T}} \mathbf{R}_p^i \; . \tag{3.8b}
$$

3.1.2 Constraints

To find a viable swing-up trajectory, two constraints are introduced, which specify the state vector at the beginning and at the end of the trajectory with $\mathbf{x}_p^0$ and $\mathbf{x}_p^{N-1}$, respectively. For $t = t_0$, the spherical inverted pendulum needs to be in the lower equilibrium posture (2.24b), the tool attachment point needs to be in a defined start position $\mathbf{r}_0$. Although the actual value for $\mathbf{r}_0$ is irrelevant for the swing-up trajectory in this optimization problem, it is set to a suitable and reachable start position for the robot to ease the conversion of the resulting trajectory for the complete system in the next chapter. The initial condition is therefore

$$
\mathbf{x}_p(t_0) = \begin{bmatrix} \mathbf{q}_{el} \\ \mathbf{0} \\ \mathbf{r}_0 \\ \mathbf{0} \end{bmatrix} \; . \tag{3.9}
$$

As with the initial condition, the final condition of the state is composed of the requirement that the spherical inverted pendulum is in the upper equilibrium posture (2.24a) and the tool attachment point of the robot has returned to the start position $\mathbf{r}_0$, i. e.

$$\mathbf{x}_p(t_1) = \begin{bmatrix} \mathbf{q}_{eu} \\ \mathbf{0} \\ \mathbf{r}_0 \\ \mathbf{0} \end{bmatrix} . \tag{3.10}$$

Since (3.1b) is discretized using the trapezoidal rule, the system equation constraints are only fulfilled at the collocation points. The constraints resulting from the equations of motion, as well as the initial and final condition, are combined to the vector of equality constraints

$$\mathbf{c}_{eq,p}(\mathbf{y}_p) = \begin{bmatrix} \mathbf{x}_p^1 - \mathbf{x}_p^0 - \frac{t^1-t^0}{2}\left(\mathbf{f}_p\left(t^0, \mathbf{x}_p^0, \mathbf{u}_p^0\right) + \mathbf{f}_p\left(t^1, \mathbf{x}_p^1, \mathbf{u}_p^1\right)\right) \\ \vdots \\ \mathbf{x}_p^{N-1} - \mathbf{x}_p^{N-2} - \frac{t^{N-1}-t^{N-2}}{2}\left(\mathbf{f}_p\left(t^{N-2}, \mathbf{x}_p^{N-2}, \mathbf{u}_p^{N-2}\right) + \mathbf{f}_p\left(t^{N-1}, \mathbf{x}_p^{N-1}, \mathbf{u}_p^{N-1}\right)\right) \\ \mathbf{x}_p^0 - \mathbf{x}_p(t_0) \\ \mathbf{x}_p^{N-1} - \mathbf{x}_p(t_1) \end{bmatrix} . \tag{3.11}$$

The gradient of the vector of equality constraints is calculated as

$$\frac{\partial \mathbf{c}_{eq,p}(\mathbf{y}_p)}{\partial \mathbf{y}_p} = \begin{bmatrix} \mathbf{A}_{eq,p}^0 & \mathbf{B}_{eq,p}^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{eq,p}^1 & \mathbf{B}_{eq,p}^2 & \cdots & \mathbf{0} \\ \vdots & & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & & \mathbf{A}_{eq,p}^{N-2} & \mathbf{B}_{eq,p}^{N-1} \\ \begin{bmatrix}\mathbf{I}_{10} & \mathbf{0}\end{bmatrix} & \mathbf{0} & \cdots & & \mathbf{0} \\ \mathbf{0} & & \cdots & & \begin{bmatrix}\mathbf{I}_{10} & \mathbf{0}\end{bmatrix} \end{bmatrix} \tag{3.12a}$$

$$\mathbf{A}_{eq,p}^i = \begin{bmatrix} -\mathbf{I}_{10} - \frac{t^{i+1}-t^i}{2}\frac{\partial \mathbf{f}_p}{\partial \mathbf{x}_p}\left(t^i, \mathbf{x}^i, \mathbf{u}^i\right) & -\frac{t^{i+1}-t^i}{2}\frac{\partial \mathbf{f}_p}{\partial \mathbf{u}_p}\left(t^i, \mathbf{x}^i, \mathbf{u}^i\right) \end{bmatrix} \tag{3.12b}$$

$$\mathbf{B}_{eq,p}^i = \begin{bmatrix} \mathbf{I}_{10} - \frac{t^{i+1}-t^i}{2}\frac{\partial \mathbf{f}_p}{\partial \mathbf{x}_p}\left(t^i, \mathbf{x}^i, \mathbf{u}^i\right) & -\frac{t^{i+1}-t^i}{2}\frac{\partial \mathbf{f}_p}{\partial \mathbf{u}_p}\left(t^i, \mathbf{x}^i, \mathbf{u}^i\right) \end{bmatrix} , \tag{3.12c}$$

with the identity matrix $\mathbf{I}_n$ of size n.

3.1.3 Algorithm and Parameters

The search for a minimum of the optimization problem (3.4) with (3.5) to (3.12) is performed using `fmincon` from the MATLAB Optimization Toolbox [46]. The parameters used for the `fmincon` solver are shown in Table 3.1. The interior-point algorithm is used because it is the recommended algorithm for nonlinear problems in [46]. The usage of the gradients of the objective function and of the constraints is enabled and the number of iterations and function evaluations are increased to avoid a premature end of the search. Additionally, the use of parallel processing is activated to decrease the search time for a minimum.

Parameter	Value
Algorithm	interior-point
SpecifyConstraintGradient	true
SpecifyObjectiveGradient	true
MaxFunctionEvaluations	10^9
MaxIterations	10^9
UseParallel	true

Table 3.1: `fmincon` options for the optimization problem of the spherical inverted pendulum with Cartesian acceleration input.

The matrices used in the Lagrange function (3.6) are chosen as

$$\mathbf{Q}_p = \mathrm{diag}(0, 0, 1, 1, 1, 1, 1, 1, 100, 100) \tag{3.13a}$$

$$\mathbf{R}_p^i = \mathrm{diag}(100, 1, 1)\,, \qquad i = 0, \ldots, N-1\,, \tag{3.13b}$$

with $\mathrm{diag}(d_1, d_2, \ldots, d_n)$ denoting a diagonal matrix whose diagonal elements starting in the upper left corner are $d_1, d_2, \ldots, d_n$. The weights of q_8 and q_9 in $\mathbf{Q}_p$ are chosen to be zero, but tests have shown that setting them to one results in satisfying solutions too. The last two elements of the diagonal of the matrix $\mathbf{Q}_p$ are chosen to be 100 to retain the velocities $\dot{r}_y$ and $\dot{r}_z$. The weight of the acceleration $\ddot{r}_x$ in $\mathbf{R}_p$ is increased to avoid any movements in the direction of the x_0 axis, which are not needed for the swing-up of the spherical inverted pendulum (see Figure 2.3). The reference values used in the Lagrange function (3.6) are set to

$$\mathbf{x}_{r,p} = \mathbf{x}_p(t_1) \tag{3.14a}$$

$$\mathbf{u}_{r,p} = \mathbf{0}\ . \tag{3.14b}$$

Furthermore, the initial value of $\mathbf{y}_p$ (3.3) is obtained by assigning the linear interpolation between (3.9) and (3.10) to $\mathbf{x}_p^i$ and setting $\mathbf{u}_p^i$ to zero.

The number of collocation points N is set to 80 with an optimization horizon of $t_1 = 2\,\mathrm{s}$, as this has proven to be a good compromise between accuracy and calculation time for solving the optimization problem.

3.1.4 Results

The `fmincon` solver found a solution with the characteristic values of the optimization process shown in Table 3.2. The resulting trajectory is shown in Figure 3.1. In this thesis, discrete-time signals are shown as continuous-time signals for reasons of clarity. Both the initial condition (3.9) and the final condition (3.10) are satisfied and the position $\mathbf{r}$ and the velocity $\dot{\mathbf{r}}$ of the tool attachment point have small deviations from their initial values of about $0.1\,\mathrm{m}$ and $0.5\,\mathrm{m/s}$, respectively. Small values are beneficial since more reserve is available in view of the kinematic and dynamic limits of the robot, which is in the next section used to perform the dynamic motion of the tool attachment point. The angle q_9 remains very close to zero and is not used for the swing-up, as was intended by the increased weight of $\ddot{r}_x$.

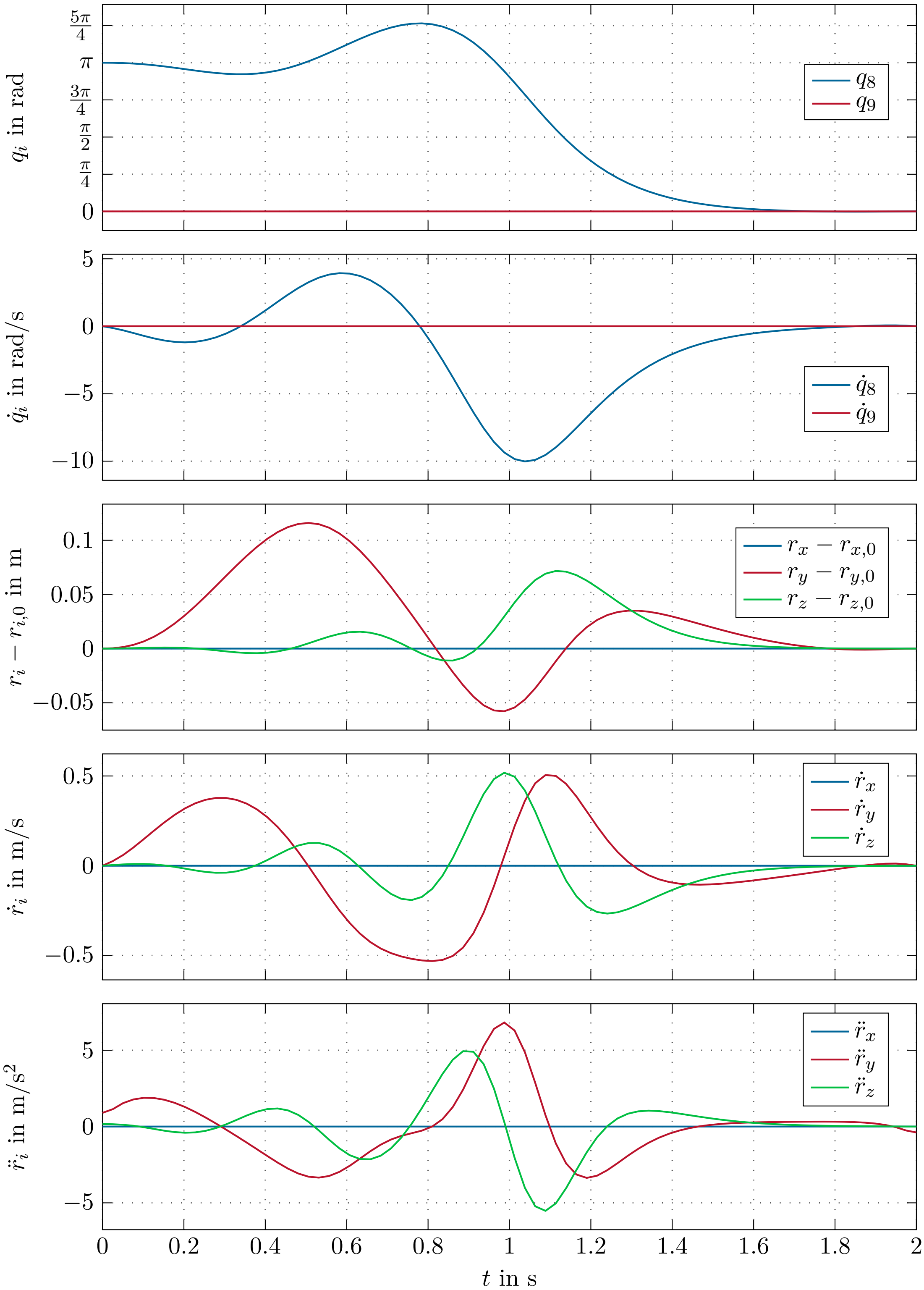

Figure 3.1: Swing-up trajectory of the spherical inverted pendulum with Cartesian input.

Name	Value
iterations	1062
function evaluations	9093
constraint violation	$2.2482 \cdot 10^{-15}$
first order optimality	$1.2746 \cdot 10^{-6}$
$J_p(\mathbf{y}_{\mathbf{p,min}})$	34.0357
optimization variables	1040

Table 3.2: Characteristic values of the optimization problem of the spherical inverted pendulum with Cartesian acceleration input.

3.2 Inverse Kinematics and Forward Dynamics

As the resulting trajectory of the previous section only describes the pendulum and the tool attachment point, this solution is now transformed into a swing-up trajectory of the complete system with torque input. This transformed trajectory serves as an initial guess for subsequent optimizations. For this purpose, the relationship between the linear and angular velocity of the tool attachment point $\dot{\mathbf{z}}$ and the joint velocities of the robot $\dot{\mathbf{q}}_r$, described by (2.29)

$$\dot{\mathbf{z}} = \mathbf{J}^t_{0,r}\dot{\mathbf{q}}_r \ , \tag{3.15}$$

is used [43]. The time derivative of (3.15) is

$$\ddot{\mathbf{z}} = \dot{\mathbf{J}}^t_{0,r}\dot{\mathbf{q}}_r + \mathbf{J}^t_{0,r}\ddot{\mathbf{q}}_r \ , \tag{3.16}$$

which involves the linear and angular acceleration of the tool attachment point $\ddot{\mathbf{z}}$ and the joint accelerations $\ddot{\mathbf{q}}_r$. While the linear velocities and accelerations are taken from the results of the optimization problem for the spherical inverted pendulum with Cartesian acceleration input, the angular velocities and accelerations are set to zero, to reflect that the orientation of the coordinate frame of the tool attachment point remains fixed

$$\dot{\mathbf{z}}^i = \begin{bmatrix} \dot{\mathbf{r}}^i \\ \mathbf{0}_{3\times 1} \end{bmatrix} , \qquad \ddot{\mathbf{z}}^i = \begin{bmatrix} \ddot{\mathbf{r}}^i \\ \mathbf{0}_{3\times 1} \end{bmatrix} , \qquad i = 0, \dots, N-1 \ . \tag{3.17}$$

Using the pseudo-inverse, denoted by †, (3.15) and (3.16) are rearranged to calculate the joint velocities and accelerations for each instant of time t^i as

$$\dot{\mathbf{q}}^i_r = \left(\mathbf{J}^{t,i}_{0,r}\right)^\dagger \dot{\mathbf{z}}^i \ , \qquad \ddot{\mathbf{q}}^i_r = \left(\mathbf{J}^{t,i}_{0,r}\right)^\dagger \left(\ddot{\mathbf{z}}^i - \dot{\mathbf{J}}^{t,i}_{0,r}\dot{\mathbf{q}}^i_r\right) , \qquad i = 0, \dots, N-1 \ . \tag{3.18}$$

The joint angles result from integrating the joint velocities using the Euler method

$$\mathbf{q}^{i+1}_r = \mathbf{q}^i_r + \left(t^{i+1} - t^i\right)\dot{\mathbf{q}}^i_r \ , \qquad i = 0, \dots, N-2 \ . \tag{3.19}$$

As a result, this inverse kinematics process deduces a joint-based trajectory for the robot $\mathbf{q}^i_r$, $\dot{\mathbf{q}}^i_r$ and $\ddot{\mathbf{q}}^i_r$ from the desired tool attachment point velocities $\dot{\mathbf{r}}^i$ and accelerations $\ddot{\mathbf{r}}^i$.

Based on this trajectory, the dynamics of the complete system (2.42) can be simulated forward, to calculate the trajectory of the complete system. The equations to calculate the joint angles, velocities, and accelerations of the complete system with torque input read as

$$\mathbf{q}_s^i = \begin{bmatrix} \mathbf{q}_r^i \\ q_8^i \\ q_9^i \end{bmatrix} , \quad \dot{\mathbf{q}}_s^i = \begin{bmatrix} \dot{\mathbf{q}}_r^i \\ \dot{q}_8^i \\ \dot{q}_9^i \end{bmatrix} , \quad \ddot{\mathbf{q}}_s^i = \begin{bmatrix} \ddot{\mathbf{q}}_r^i \\ \begin{bmatrix} \mathbf{0}_{2\times 2} & \mathbf{I}_2 & \mathbf{0}_{2\times 6} \end{bmatrix} \mathbf{f}_p^i \end{bmatrix} , \qquad i = 0, \ldots, N-1 \ , \tag{3.20}$$

with $\mathbf{f}_p$ from (2.25). The input of the complete system with torque input is calculated from the equations of motion (2.42) with $\boldsymbol{\tau}_{ext,s} = \mathbf{0}$

$$\boldsymbol{\tau}_s^i = \mathbf{D}_s\left(\mathbf{q}_s^i\right)\ddot{\mathbf{q}}_s^i + \mathbf{C}_s\left(\mathbf{q}_s^i, \dot{\mathbf{q}}_s^i\right)\dot{\mathbf{q}}_s^i + \mathbf{g}_s\left(\mathbf{q}_s^i\right) \tag{3.21a}$$

$$\mathbf{u}_s^i = \begin{bmatrix} \mathbf{I}_7 & \mathbf{0}_{7\times 2} \end{bmatrix} \boldsymbol{\tau}_s^i \, , \qquad i = 0, \ldots, N-1 \ . \tag{3.21b}$$

As start value for the angles of the robot

$$\mathbf{q}_r^0 = \begin{bmatrix} \frac{\pi}{6} & -\frac{\pi}{2} & -\frac{\pi}{2} & \frac{\pi}{3} & 0 & \frac{\pi}{6} & \frac{\pi}{2} \end{bmatrix}^{\mathrm{T}} \tag{3.22}$$

is used. This configuration has the benefit that three axes of the robot are parallel to the z_0 axis of the space-fixed coordinate frame. The robot has therefore a large workspace, although the tool attachment point is restricted to translatory motions along the x_0 and y_0 axes and rotatory motions around z_0 axis. The coordinate frame of the tool attachment point $(0_t x_t y_t z_t)$ has the same orientation as the one used in the model for the pendulum with Cartesian input. This allows direct conversion of the trajectory obtained in Section 3.1 to a trajectory for the complete system with torque input.

The robot kinematics and dynamics are restricted by physical limits in terms of limiting angles $\mathbf{q}_{l,r}$ (subscripted by "l" for limit), joint velocities $\dot{\mathbf{q}}_{l,r}$, and torques $\mathbf{M}_{l,r}$, which are shown in Table A.1. These limits need to be satisfied at all times [47]. The chosen start value (3.22) provides a sufficient distance to the angle limits $\mathbf{q}_{l,r}$ of the robot. In particular, the three axes parallel to z_0 have at least 50 % reserve to the angle limits of the robot. The results of the trajectory conversion (3.17) to (3.21) is shown in Figure 3.2. Indices of vectors are denoted by the subscript "i". The values in the figure are scaled with

$$\mathbf{q}_{s,s} = \begin{bmatrix} \mathbf{q}_{l,r}^{\mathrm{T}} & 360° & 360° \end{bmatrix}^{\mathrm{T}} \tag{3.23a}$$

$$\dot{\mathbf{q}}_{s,s} = \begin{bmatrix} \dot{\mathbf{q}}_{l,r}^{\mathrm{T}} & 720\,°/\mathrm{s} & 720\,°/\mathrm{s} \end{bmatrix}^{\mathrm{T}} \tag{3.23b}$$

$$\mathbf{M}_{s,s} = \mathbf{M}_{l,r} \ , \tag{3.23c}$$

respectively. The limits of the robot are satisfied with a reserve of at least 24 % to the limits in all axes, which is sufficient for additional controller input occurring during the swing-up. This is ensured by the chosen start value for the angles and the chosen weights of the optimization problem in the previous section.

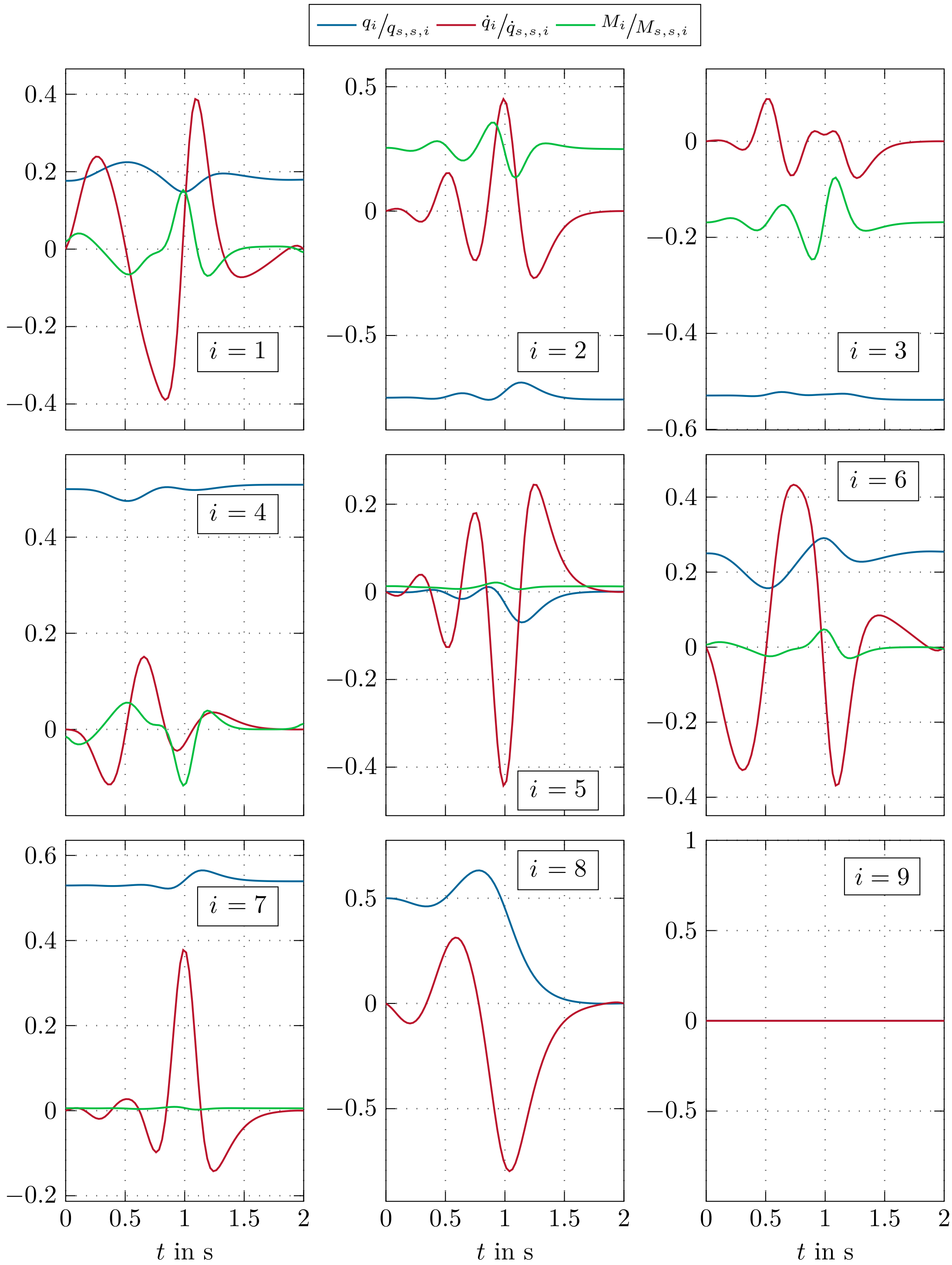

Figure 3.2: Scaled swing-up trajectory of the complete system with torque input, which is used as initial guess for the optimization problem. The graphs are scaled with the scaling vectors from (3.23).

3.3 Complete System with Torque Input

The converted trajectory from the previous section is now used as an initial guess for the optimization problem of the complete system with torque input. This optimization problem is used to calculate the swing-up trajectory of the spherical inverted pendulum, utilized during the simulation and the experiments. As this optimization problem is based on the model of the complete system from Section 2.4, the weights in the cost function are applied directly to the axes, which allows for more direct tuning of the final trajectory. Additionally, this approach also allows to constrain the robot axes directly according to the kinematic and dynamic limits of the robot listed in Table A.1. As a trajectory with a time resolution of 1 ms is needed for the experimental setup, the first solution with a lower resolution is then successively refined to reach the desired time resolution.

The vector of variables $\mathbf{y}_s \in \mathbb{R}^{(18+7)N}$ is composed of the state $\mathbf{x}_s = \begin{bmatrix}\mathbf{q}_s^{\mathrm{T}} & \dot{\mathbf{q}}_s^{\mathrm{T}}\end{bmatrix}$ and the input $\mathbf{u}_s$ from (2.32) evaluated on the time grid (3.2)

$$\mathbf{y}_s = \begin{bmatrix}(\mathbf{x}_s^0)^{\mathrm{T}} & (\mathbf{u}_s^0)^{\mathrm{T}} & (\mathbf{x}_s^1)^{\mathrm{T}} & (\mathbf{u}_s^1)^{\mathrm{T}} & \cdots & \left(\mathbf{x}_s^{N-1}\right)^{\mathrm{T}} & \left(\mathbf{u}_s^{N-1}\right)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} . \tag{3.24}$$

Similar to the static optimization problem (3.4), the static optimization problem consisting of

$$\min_{\mathbf{y}_s} \quad J_s(\mathbf{y}_s) \tag{3.25a}$$

$$\text{s.t.} \quad \mathbf{c}_{eq,s}(\mathbf{y}_s) = \mathbf{0} \tag{3.25b}$$

$$\mathbf{y}_{lb,s} \leq \mathbf{y}_s \leq \mathbf{y}_{ub,s} \tag{3.25c}$$

is solved.

3.3.1 Cost Function

The same cost function and gradient of the cost function as in (3.5) to (3.8) (with subscript "s") are used for this optimization problem. The optimization algorithm has a tendency to find solutions, which incorporate a jerky motion towards the end of the optimization horizon $[t_0, t_1]$ to meet the constraint of the final value. To counteract this behavior, the time-variant matrix $\mathbf{R}_s^i$ is used to increase the weights of the input towards the end the optimization horizon. This ensures a smooth motion of the complete system into the final position.

3.3.2 Constraints

The initial and final condition of the optimization problem need to reflect the requirement that, initially, the robot is in its start position $\mathbf{q}_r^0$ (3.22) and the pendulum is hanging down, and, when the swing-up maneuver is finished the robot should return to its start position and the pendulum is standing upright. Thus, the initial condition consists of the start position of the robot and the lower equilibrium posture of the pendulum (2.24b) and the final condition consists of the start position of the robot and the upper equilibrium posture of the pendulum (2.24a), i. e.

$$\mathbf{x}_s(t_0) = \begin{bmatrix} \mathbf{q}_r^0 \\ \mathbf{q}_{el} \\ \mathbf{0}_{9\times 1} \end{bmatrix} \tag{3.26a}$$

$$\mathbf{x}_s(t_1) = \begin{bmatrix} \mathbf{q}_r^0 \\ \mathbf{q}_{eu} \\ \mathbf{0}_{9\times 1} \end{bmatrix} . \tag{3.26b}$$

The equality constraint vector and its gradient have the same structure as in (3.11) and (3.12), respectively, whereby the model equations in state-space form $\mathbf{f}_s$ are taken from (2.43). To receive the gradient of the equality constraints, the partial derivative of (2.42) with respect to $\mathbf{x}_s$ and $\mathbf{u}_s$ is calculated and then rearranged to

$$\frac{\partial \mathbf{f}_s}{\partial \mathbf{x}_s} = \begin{bmatrix} \begin{bmatrix} \mathbf{0}_{9\times 9} & \mathbf{I}_9 \end{bmatrix} \\ (\mathbf{D}_s(\mathbf{q}_s))^{-1}\Big(-\Big(\sum_{i=1}^{9} \frac{\partial \mathbf{d}_{i,s}(\mathbf{q}_s)}{\partial \mathbf{x}_s}\ddot{q}_{i,s}\Big) - \frac{\partial (\mathbf{C}_s(\mathbf{q}_s,\dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s)}{\partial \mathbf{x}_s} - \frac{\partial \mathbf{g}_s(\mathbf{q}_s)}{\partial \mathbf{x}_s} + \frac{\partial \boldsymbol{\tau}_s}{\partial \mathbf{x}_s}\Big) \end{bmatrix} \tag{3.27a}$$

$$\frac{\partial \mathbf{f}_s}{\partial \mathbf{u}_s} = \begin{bmatrix} \mathbf{0}_{18\times 9} \\ (\mathbf{D}_s(\mathbf{q}_s))^{-1}\frac{\partial \boldsymbol{\tau}_s}{\partial \mathbf{u}_s} \end{bmatrix} . \tag{3.27b}$$

The term $\sum_{i=1}^{9} \frac{\partial \mathbf{d}_{i,s}(\mathbf{q}_s)}{\partial \mathbf{x}_s}\ddot{q}_{i,s}$ results from the product rule, with $\mathbf{d}_{i,s}(\mathbf{q}_s)$ denoting the columns of $\mathbf{D}_s(\mathbf{q}_s)$.

Next, the upper bound $\mathbf{y}_{ub,s}$ and the lower bound $\mathbf{y}_{lb,s}$ of the vector of optimization variables $\mathbf{y}_s$ are used to incorporate the kinematic and dynamic limits of the states (2.31) and the inputs (2.32) of the complete system. The vectors of kinematic and dynamic limits of the robot (see Table A.1) are extended with limits for $q_8, q_9, \dot{q}_8$, and $\dot{q}_9$. As both axes allow for an infinite travel and their joint velocity is not constrained, this leads to

$$\mathbf{q}_{l,s} = \begin{bmatrix} \mathbf{q}_{l,r}^{\mathrm{T}} & \infty & \infty \end{bmatrix}^{\mathrm{T}} \tag{3.28a}$$

$$\dot{\mathbf{q}}_{l,s} = \begin{bmatrix} \dot{\mathbf{q}}_{l,r}^{\mathrm{T}} & \infty & \infty \end{bmatrix}^{\mathrm{T}} \tag{3.28b}$$

$$\mathbf{M}_{l,s} = \mathbf{M}_{l,r} . \tag{3.28c}$$

The possibility of a collision between the pendulum rod and the mounting enclosure (compare Figure 2.3) is neglected here, but is taken into consideration while choosing the weights of the cost function in the following subsection. The vectors (3.28) constitute the upper bound $\mathbf{y}_{ub,s}$ of the vector of optimization variables $\mathbf{y}_s$, which are composed to a vector of the same dimension as $\mathbf{y}_s \in \mathbb{R}^{(18+7)N}$. Furthermore, the lower bound $\mathbf{y}_{lb,s}$ is equal to the negative upper bound, i. e.

$$\mathbf{y}_{ub,s} = \big[\underbrace{\mathbf{q}_{l,s}^{\mathrm{T}} \quad \dot{\mathbf{q}}_{l,s}^{\mathrm{T}} \quad \mathbf{M}_{l,s}^{\mathrm{T}} \quad \cdots \quad \mathbf{q}_{l,s}^{\mathrm{T}} \quad \dot{\mathbf{q}}_{l,s}^{\mathrm{T}} \quad \mathbf{M}_{l,s}^{\mathrm{T}}}_{N \text{ times}}\big]^{\mathrm{T}} \tag{3.29a}$$

$$\mathbf{y}_{lb,s} = -\mathbf{y}_{ub,s} . \tag{3.29b}$$

3.3.3 Algorithm and Parameters

Just as with the optimization problem for the spherical inverted pendulum with Cartesian input, the `fmincon` solver is used to solve the optimization problem for the complete system. The algorithm is switched to SQP, as this has proven to reduce the search time. The remaining options are kept unchanged and are equal to the ones listed in Table 3.1. The matrices used in the cost function are chosen as

$$\mathbf{Q}_s = \operatorname{diag}\Big(1,1,1,1,1,1,1,0,10^3,1,1,10^2,10^1,5\cdot 10^1,10^2,10^1,0,10^3\Big) \tag{3.30a}$$

$$\mathbf{R}_s^i = \begin{cases} \mathbf{I}_7, & \text{for } t^i \leq 1.95\,\mathrm{s} \\ 10^3 \cdot \mathbf{I}_7, & \text{for } t^i > 1.95\,\mathrm{s}\ . \end{cases} \tag{3.30b}$$

The weights of q_8 and $\dot{q}_8$ are chosen to be zero, as the trajectory is not important, as long as the initial and final condition are satisfied. The largest weight is applied to q_9 and $\dot{q}_9$ to avoid any motions in this coordinate. Therefore, the swing-up is primarily performed using Axis 8. This also benefits the convergence during the search for a minimum, as otherwise non-physical solutions are found. The weights of the joint velocities are adjusted such that the solution trajectory has approximately 50 % reserve to the limits of the robot. The weights of the inputs are increased towards the end of the optimization horizon, to avoid jerky motions. The reference values of the states and inputs are set to

$$\mathbf{x}_{r,s} = \mathbf{x}_s(t_1) \tag{3.31}$$

and

$$\mathbf{u}_{r,s} = \mathbf{g}_s\left(\begin{bmatrix}\mathbf{q}_r^0 \\ \mathbf{q}_{eu}\end{bmatrix}\right)\ , \tag{3.32}$$

respectively, which compensates the gravitational force in the final posture.

The number of collocation points is again chosen to be $N = 80$. As the experimental setup runs with a sampling time of 1 ms, the final trajectory needs to have a larger number of sampling points $N = 2001$ for the optimization horizon of $t_1 = 2\,\mathrm{s}$. Therefore, the optimization problem (3.25) is solved repeatedly, while increasing N by 10 % at every iteration. The solution of the previous iteration is interpolated on the new time grid and used as initial guess for the next iteration. Due to memory limit of the computer performing the calculation, N could only be increased to 1282 in this way. The final solution with $N = 2001$ is calculated using a cubic spline interpolation.

3.3.4 Results

The result of the optimization using the model (2.42) with torque input is shown in Figure 3.3. The characteristic values of the optimization problem are listed in Table 3.3. A small difference between the first solution with $N = 80$ and $N = 2001$ can be observed. The kinematic and dynamic limits of the robot are satisfied and the system moves from the start position (3.26a) to the final position (3.26b). As intended by the choice of the weights, the joint velocities have more than 50 % reserve to the limits. The largest joint velocity of 37 % is needed in Axis 3. The maximum torque is needed in Axis 2 (54.4 N m or 27 %). This is due to the choice of the initial posture of the robot, which causes a large lever arm between the joint of Axis 2 and the weight forces of Axis 2 to 7 as well as the weight forces of the spherical inverted pendulum. Large input torques towards the end of the optimization horizon are not observed and the system moves smoothly into the final position.

	Name	Value
$N = 80$	iterations	273
	function evaluations	955
	constraint violation	$5.3291 \cdot 10^{-15}$
	first order optimality	$1.9597 \cdot 10^{-7}$
	$J_p(\mathbf{y}_{p,min})$	49.7188
	optimization variables	2000
$N = 1282$	iterations	280
	function evaluations	561
	constraint violation	$1.9984 \cdot 10^{-15}$
	first order optimality	$9.2861 \cdot 10^{-7}$
	$J_p(\mathbf{y}_{p,min})$	49.6803
	optimization variables	32050

Table 3.3: Characteristic values of the optimization problem of the complete system with torque input.

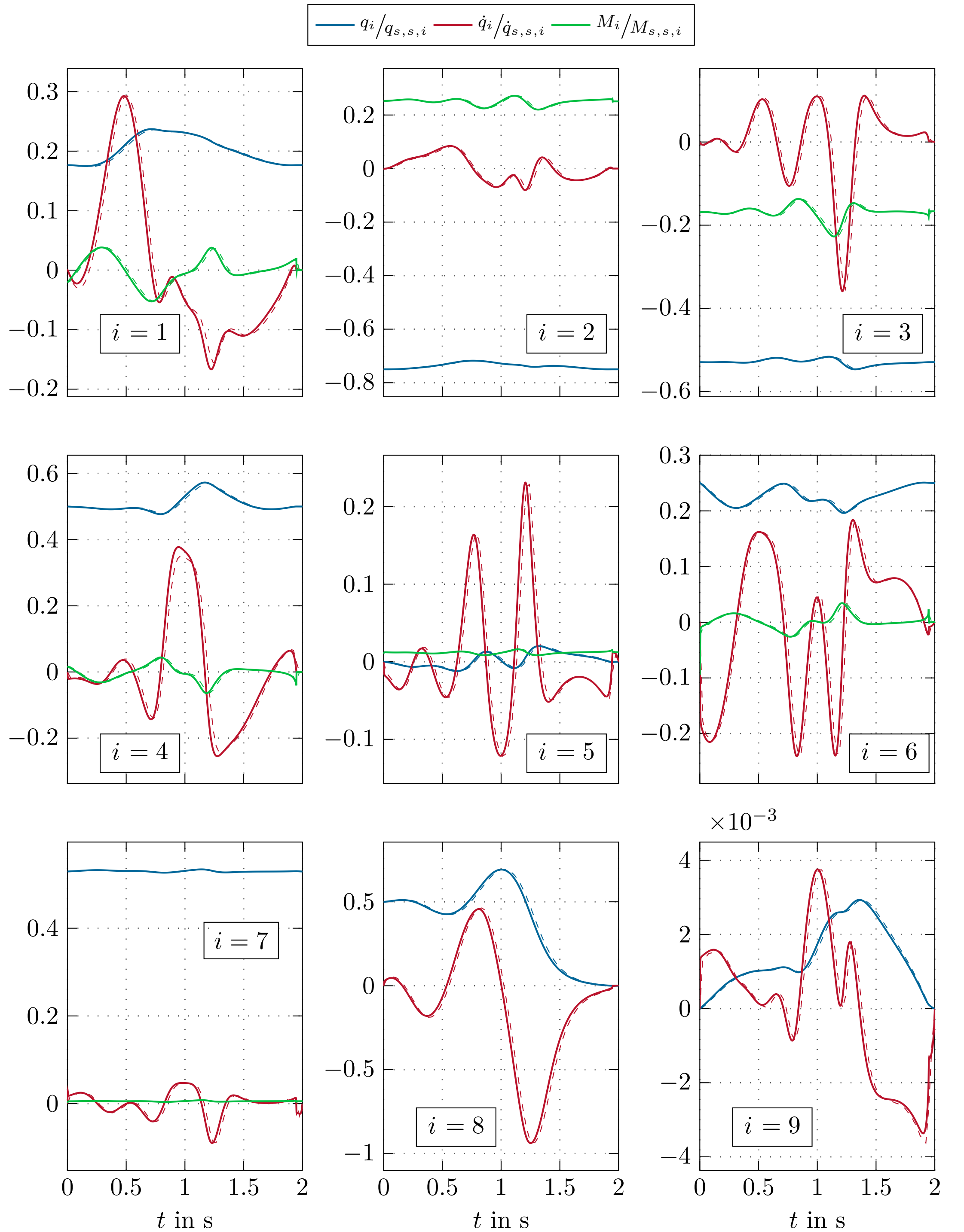

Figure 3.3: Scaled result of the optimization problem of the swing-up trajectory for the complete system with torque input. The trajectory with $N = 2001$ is shown with a solid line, the trajectory with $N = 80$ is shown with a thin, dashed line. The graphs are scaled with the scaling vectors from (3.23).

4 Controller Design

This chapter describes the design of the controller used during the swing-up of the spherical inverted pendulum and the controller used for stabilizing the pendulum. A time-variant LQR is used to perform the swing-up motion. Once the system is close to the operating point where the pendulum is in the unstable upright position, the controller is switched to a cascaded controller structure, which is used for stabilizing the pendulum. The cascade controller structure consists of an inner and an outer loop. The outer loop is composed of an LQR, which stabilizes the pendulum and the pendulum with Cartesian input. This LQR provides the desired Cartesian position, velocity, and acceleration of the tool attachment point, which are fed as desired trajectory to the Cartesian trajectory tracking controller moving the robot.

4.1 Time-variant LQR for the Swing-Up Trajectory

In Section 3.3, a swing-up trajectory for the complete system with torque input (2.42) was obtained, which consists of desired values for the joint angles $\mathbf{q}_{d,s}$, the joint velocities $\dot{\mathbf{q}}_{d,s}$, and the inputs $\mathbf{u}_{d,s}$. Due to model uncertainties and the fact that friction was neglected in the system (2.42), the system needs to be stabilized around this trajectory to ensure a successful swing-up of the spherical inverted pendulum. Therefore, a time-variant LQR is utilized in this section, which is easier to be designed for MIMO systems than using the pole-placement approach [48].

An LQR is a state-space controller for a linear time-variant system $\left(\mathbf{\Phi}^k, \mathbf{\Gamma}^k\right)$, which is derived from the optimal solution to the optimization problem

$$\min_{(\mathbf{u}^0,\ldots,\mathbf{u}^{N-1})} \quad J\Big((\mathbf{u}^0, \ldots, \mathbf{u}^{N-1})\Big) = \frac{1}{2}\left(\mathbf{x}^N\right)^{\mathrm{T}} \mathbf{S}\mathbf{x}^N + \frac{1}{2}\sum_{k=0}^{N-1} \left(\mathbf{x}^k\right)^{\mathrm{T}} \mathbf{Q}\mathbf{x}^k + \left(\mathbf{u}^k\right)^{\mathrm{T}} \mathbf{R}\mathbf{u}^k \tag{4.1a}$$

$$\text{s.t.} \quad \mathbf{x}^{k+1} = \mathbf{\Phi}^k \mathbf{x}^k + \mathbf{\Gamma}^k \mathbf{u}^k \ , \tag{4.1b}$$

with superscript “k” denoting the time step. This optimal solution can be given in closed form as

$$\mathbf{u}^k = \mathbf{K}^k \mathbf{x}^k \tag{4.2a}$$

$$\mathbf{K}^k = -\left(\mathbf{R} + \mathbf{\Gamma}^{\mathrm{T}} \mathbf{P}^{k+1} \mathbf{\Gamma}\right)^{-1} \left(\mathbf{\Gamma}^{\mathrm{T}} \mathbf{P}^{k+1} \mathbf{\Phi}\right) \tag{4.2b}$$

$$\mathbf{P}^k = \left(\mathbf{Q} + \mathbf{\Phi}^{\mathrm{T}} \mathbf{P}^{k+1} \mathbf{\Phi}\right) - \left(\mathbf{\Gamma}^{\mathrm{T}} \mathbf{P}^{k+1} \mathbf{\Phi}\right)^{\mathrm{T}} \left(\mathbf{R} + \mathbf{\Gamma}^{\mathrm{T}} \mathbf{P}^{k+1} \mathbf{\Gamma}\right)^{-1} \left(\mathbf{\Gamma}^{\mathrm{T}} \mathbf{P}^{k+1} \mathbf{\Phi}\right) \ , \tag{4.2c}$$

with the time-variant feedback matrix $\mathbf{K}^k$ and the symmetric, positive (semi)definite solution $\mathbf{P}^k$, $\mathbf{P}^N = \mathbf{S}$, of the Riccati equation (4.2c). Both matrices, $\mathbf{K}^k$ and $\mathbf{P}^k$, are independent of the initial state $\mathbf{x}^0$ and can be calculated off-line.

For a time-invariant system

$$\mathbf{x}^{k+1} = \mathbf{\Phi}\mathbf{x}^k + \mathbf{\Gamma}\mathbf{u}^k \ , \tag{4.3}$$

a stationary solution ($k \to \infty$) can be calculated by using

$$\mathbf{u}^k = \mathbf{K}^\infty \mathbf{x}^k \tag{4.4a}$$

$$\mathbf{K}^\infty = -\left(\mathbf{R} + \mathbf{\Gamma}^{\mathrm{T}}\mathbf{P}^\infty\mathbf{\Gamma}\right)^{-1}\left(\mathbf{\Gamma}^{\mathrm{T}}\mathbf{P}^\infty\mathbf{\Phi}\right) \tag{4.4b}$$

$$\mathbf{P}^\infty = \left(\mathbf{Q} + \mathbf{\Phi}^{\mathrm{T}}\mathbf{P}^\infty\mathbf{\Phi}\right) - \left(\mathbf{\Gamma}^{\mathrm{T}}\mathbf{P}^\infty\mathbf{\Phi}\right)^{\mathrm{T}}\left(\mathbf{R} + \mathbf{\Gamma}^{\mathrm{T}}\mathbf{P}^\infty\mathbf{\Gamma}\right)^{-1}\left(\mathbf{\Gamma}^{\mathrm{T}}\mathbf{P}^\infty\mathbf{\Phi}\right) \ . \tag{4.4c}$$

The equation for calculating $\mathbf{P}^\infty$ is also called algebraic Riccati equation, see, e. g., [45].

As the LQR can only be applied to linear systems, the equations of motion (2.42) are linearized around the swing-up trajectory $\mathbf{x}_s^i$, $\mathbf{u}_s^i$ with $i = 0, \dots, N-1$. The states and inputs can be written in the form

$$\mathbf{x}_s\left(t^k\right) = \mathbf{x}_s^i + \Delta\mathbf{x}_s\left(t^k\right), \qquad k = i \ , \qquad i = 0, \dots, N-1 \tag{4.5a}$$

$$\mathbf{u}_s\left(t^k\right) = \mathbf{u}_s^i + \Delta\mathbf{u}_s\left(t^k\right), \qquad k = i \ , \qquad i = 0, \dots, N-1 \ , \tag{4.5b}$$

with the deviations from the trajectory $\Delta\mathbf{x}_s^k$ and $\Delta\mathbf{u}_s^k$. For small deviations, a linear time-variant system is obtained by using the already derived gradients (3.27) resulting in

$$\Delta\dot{\mathbf{x}}_s\left(t^k\right) = \mathbf{A}_s^k\Delta\mathbf{x}_s\left(t^k\right) + \mathbf{B}_s^k\Delta\mathbf{u}_s\left(t^k\right) \tag{4.6a}$$

$$\mathbf{A}_s^k = \left.\frac{\partial\mathbf{f}_s}{\partial\mathbf{x}_s}\right|_{\substack{\mathbf{x}_s=\mathbf{x}_s^i\\ \mathbf{u}_s=\mathbf{u}_s^i}} , \qquad k = i \ , \quad i = 0, \dots, N-1 \tag{4.6b}$$

$$\mathbf{B}_s^k = \left.\frac{\partial\mathbf{f}_s}{\partial\mathbf{u}_s}\right|_{\substack{\mathbf{x}_s=\mathbf{x}_s^i\\ \mathbf{u}_s=\mathbf{u}_s^i}} , \qquad k = i \ , \quad i = 0, \dots, N-1 \ . \tag{4.6c}$$

This linearized system is discretized using zero-order hold and a sampling time of $T_s = 1\,\mathrm{ms}$

$$\mathbf{x}_s^{k+1} = \mathbf{\Phi}_s^k\mathbf{x}_s^k + \mathbf{\Gamma}_s^k\mathbf{u}_s^k \ . \tag{4.7}$$

To receive a usable start value for $\mathbf{K}_{su}^{N-1}$ and $\mathbf{P}_{su}^{N-1}$ (subscript "su" for swing-up), the stationary solution (4.4) is calculated. These matrices are then used to calculate $\mathbf{K}_{su}^k$ and $\mathbf{P}_{su}^k$ using (4.2) to iterate backward from $k = N-2, \dots, 0$. The control law consists of the desired input $\mathbf{u}_{d,su}^k = \mathbf{u}_s^i, k = i$ obtained in Section 3.3 and the control input of the LQR $\mathbf{u}_{LQR,su}$

$$\mathbf{u}_{su}^k = \mathbf{u}_{d,su}^k + \mathbf{u}_{LQR,su}^k \ , \qquad k = 0, \dots, N-1 \tag{4.8a}$$

$$\mathbf{u}_{LQR,su}^k = \mathbf{K}_{su}^k\mathbf{x}_s^k \ , \qquad k = 0, \dots, N-1 \ . \tag{4.8b}$$

4.2 Cascade Controller for Pendulum Stabilization

After the swing-up of the spherical inverted pendulum is finished and the tool attachment point is close to the end position, the controller is switched to a cascade controller for pendulum stabilization. This controller has the benefit of being easier to tune and performs better if disturbances are applied to the closed-loop system. The cascade controller consists of an inner and an outer loop. The outer loop assumes that Cartesian movements of the tool attachment point of the robot can be commanded freely. This is used to stabilize the pendulum on a plane parallel to the x_0-y_0-plane of the space-fixed coordinate frame, with a fixed orientation of the frame of the tool attachment point. The inner loop is a Cartesian trajectory tracking controller, which stabilizes the robot motion on the trajectory given by the outer loop.

4.2.1 LQR for the Stabilization of the Spherical Inverted Pendulum

The outer loop is designed using the linearized equations of motion (2.27) evaluated at the upper equilibrium posture (2.24a). As the state $\mathbf{x}_p$ also includes the position $\mathbf{r}$ and velocity $\dot{\mathbf{r}}$ of the tool attachment point, it is possible to stabilize the pendulum and move the robot back into its initial position. The state input pairs (q_8,$\ddot{r}_y$) and (q_9,$\ddot{r}_x$) are decoupled in the equilibrium posture, due to the symmetry of the mass distribution and the fact that Axis 8 and 9 are parallel to the x_0 axis and y_0 axis, respectively. This can be deduced from the linearized equations in

$$\frac{\mathrm{d}}{\mathrm{d}t}\Delta\mathbf{x}_p = \mathbf{A}_p\Delta\mathbf{x}_p + \mathbf{B}_p\Delta\mathbf{u}_p \ , \tag{4.9}$$

with

$$\mathbf{A}_p = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \end{bmatrix} & \mathbf{0}_{4\times 3} & \mathbf{0}_{4\times 3} \\ \mathbf{0}_{3\times 4} & \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \\ \mathbf{0}_{3\times 4} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \end{bmatrix} \tag{4.10a}$$

$$\mathbf{B}_p = \begin{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & * & 0 \\ * & 0 & 0 \end{bmatrix} \\ \mathbf{0}_{3\times 3} \\ \mathbf{I}_{3\times 3} \end{bmatrix} , \tag{4.10b}$$

whereby the elements marked with $*$ are skipped for brevity. The linearized equations of motion can be split up into

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\Delta\mathbf{x}_{p,q_8} &= \mathbf{A}_{p,q_8}\Delta\mathbf{x}_{p,q_8} + \mathbf{B}_{p,q_8}\Delta\ddot{r}_y && \text{(4.11a)}\\
\mathbf{A}_{p,q_8} &= \mathbf{V}_{q_8}^{\mathrm{T}}\mathbf{A}_p\mathbf{V}_{q_8} && \text{(4.11b)}\\
\mathbf{B}_{p,q_8} &= \mathbf{V}_{q_8}^{\mathrm{T}}\mathbf{B}_p\mathbf{V}_{r_y} && \text{(4.11c)}\\
\Delta\mathbf{x}_{p,q_8} &= \mathbf{V}_{q_8}^{\mathrm{T}}\Delta\mathbf{x}_p = \begin{bmatrix}\Delta q_8\\ \dot{q}_8\\ \Delta r_y\\ \dot{r}_y\end{bmatrix}, && \text{(4.11d)}
\end{aligned}
$$

with

$$
\mathbf{V}_{q_8}^{\mathrm{T}} = \begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&1&0\end{bmatrix}, \qquad \mathbf{V}_{r_y} = \begin{bmatrix}0\\1\\0\end{bmatrix}, \tag{4.12}
$$

and

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\Delta\mathbf{x}_{p,q_9} &= \mathbf{A}_{p,q_9}\Delta\mathbf{x}_{p,q_9} + \mathbf{B}_{p,q_9}\Delta\ddot{r}_x && \text{(4.13a)}\\
\mathbf{A}_{p,q_9} &= \mathbf{V}_{q_9}^{\mathrm{T}}\mathbf{A}_p\mathbf{V}_{q_9} && \text{(4.13b)}\\
\mathbf{B}_{p,q_9} &= \mathbf{V}_{q_9}^{\mathrm{T}}\mathbf{B}_p\mathbf{V}_{r_x} && \text{(4.13c)}\\
\Delta\mathbf{x}_{p,q_9} &= \mathbf{V}_{q_9}^{\mathrm{T}}\Delta\mathbf{x}_p = \begin{bmatrix}\Delta q_9\\ \dot{q}_9\\ \Delta r_x\\ \dot{r}_x\end{bmatrix}, && \text{(4.13d)}
\end{aligned}
$$

with

$$
\mathbf{V}_{q_9}^{\mathrm{T}} = \begin{bmatrix}0&1&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0\end{bmatrix}, \qquad \mathbf{V}_{r_x} = \begin{bmatrix}1\\0\\0\end{bmatrix}. \tag{4.14}
$$

The matrices $\mathbf{V}_{q_8}$, $\mathbf{V}_{r_y}$, $\mathbf{V}_{q_9}$, and $\mathbf{V}_{r_x}$ select the rows/columns from $\mathbf{A}_p$, $\mathbf{B}_p$ and $\Delta\mathbf{x}_p$, which are part of the subsystems (q_8,$\ddot{r}_y$) and (q_9,$\ddot{r}_x$), respectively. These two systems are then discretized using zero-order hold and a sampling time $T_s = 1\,\mathrm{ms}$. Additionally, an integrator state is added to each system to allow for the design of an LQR with integrator, which is performed with the MATLAB function `dlqr`. The resulting systems read as

$$\Delta\mathbf{x}_{p,q_8}^{k+1} = \mathbf{\Phi}_{p,q_8}\Delta\mathbf{x}_{p,q_8}^{k} + \mathbf{\Gamma}_{p,q_8}\Delta\ddot{r}_y^k \tag{4.15a}$$

$$\Delta\mathbf{x}_{p,q_9}^{k+1} = \mathbf{\Phi}_{p,q_9}\Delta\mathbf{x}_{p,q_9}^{k} + \mathbf{\Gamma}_{p,q_9}\Delta\ddot{r}_x^k \tag{4.15b}$$

$$\begin{bmatrix}\Delta\mathbf{x}_{p,q_8}^{k+1}\\ \Delta\mathbf{r}_{y,I}^{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_{p,q_8} & \mathbf{0}_{4\times1}\\ \begin{bmatrix}0 & 0 & T_s & 0\end{bmatrix} & 1\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}_{p,q_8}^{k}\\ \Delta\mathbf{r}_{y,I}^{k}\end{bmatrix} + \begin{bmatrix}\mathbf{\Gamma}_{p,q_8}\\ 0\end{bmatrix}\Delta\ddot{r}_y^k \tag{4.15c}$$

$$\begin{bmatrix}\Delta\mathbf{x}_{p,q_9}^{k+1}\\ \Delta\mathbf{r}_{x,I}^{k+1}\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi}_{p,q_9} & \mathbf{0}_{4\times1}\\ \begin{bmatrix}0 & 0 & T_s & 0\end{bmatrix} & 1\end{bmatrix}\begin{bmatrix}\Delta\mathbf{x}_{p,q_9}^{k}\\ \Delta\mathbf{r}_{x,I}^{k}\end{bmatrix} + \begin{bmatrix}\mathbf{\Gamma}_{p,q_9}\\ 0\end{bmatrix}\Delta\ddot{r}_x^k\ . \tag{4.15d}$$

The desired accelerations of the tool attachment point, i. e. the control laws of the LQRs, are calculated using

$$\ddot{r}_{x,d}^k = \Delta\ddot{r}_{x,d}^k = \mathbf{k}_{q_9}^{\mathrm{T}}\begin{bmatrix}\Delta\mathbf{x}_{p,q_9}^{k}\\ \Delta\mathbf{r}_{x,I}^{k}\end{bmatrix} \tag{4.16a}$$

$$\ddot{r}_{y,d}^k = \Delta\ddot{r}_{y,d}^k = \mathbf{k}_{q_8}^{\mathrm{T}}\begin{bmatrix}\Delta\mathbf{x}_{p,q_8}^{k}\\ \Delta\mathbf{r}_{y,I}^{k}\end{bmatrix}\ , \tag{4.16b}$$

where the desired velocities $\dot{r}_{x,d}^k$ and $\dot{r}_{y,d}^k$ and positions $r_{x,d}^k$ and $r_{y,d}^k$ result from discrete-time integrators. These values are then given to the inner control loop as a desired trajectory for the Cartesian motion of the tool attachment point.

4.2.2 Cartesian Trajectory Tracking Controller for the Robot

The Cartesian trajectory tracking controller for the robot receives desired values for the position and orientation of the tool attachment point including the first and second order time derivative and has the objective to asymptotically stabilize the robot around this trajectory. For this purpose, the inverse dynamics control law

$$\boldsymbol{\tau}_s = \mathbf{D}_s(\mathbf{q}_s)\mathbf{v} + \mathbf{C}_s(\mathbf{q}_s,\dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s + \mathbf{g}_s(\mathbf{q}_s) \tag{4.17}$$

is used for the system (2.42) with $\boldsymbol{\tau}_{ext,s} = \mathbf{0}$. Due to the positive definiteness of the mass matrix $\mathbf{D}(\mathbf{q}_s)$, it is always invertible and the closed-loop system results in

$$\ddot{\mathbf{q}}_s = \mathbf{v}\ , \tag{4.18}$$

with the virtual input $\mathbf{v}$, which is an exact linearization of the system dynamics by means of a nonlinear state feedback [43]. As the outer loop provides the desired trajectory in Cartesian coordinates, (3.15) and (3.16) are used to establish a relationship between Cartesian and joint acceleration for the complete system

$$\dot{\mathbf{z}} = \mathbf{J}_{0,s}^t\dot{\mathbf{q}}_s \tag{4.19a}$$

$$\ddot{\mathbf{z}} = \dot{\mathbf{J}}_{0,s}^t\dot{\mathbf{q}}_s + \mathbf{J}_{0,s}^t\ddot{\mathbf{q}}_s\ . \tag{4.19b}$$

The desired orientation is denoted as $\mathbf{R}_{0,d}^t$, the actual orientation is computed via the forward kinematic and denoted as $\mathbf{R}_0^t$. The orientation error $\mathbf{R}_0^t\left(\mathbf{R}_{0,d}^t\right)^{\mathrm{T}}$ is expressed as a unit quaternion $\Delta\mathcal{Q} = \{\Delta\eta, \Delta\boldsymbol{\epsilon}\}$ [43]. This unit quaternion is then used in the position and orientation error

$$\Delta\mathbf{z} = \begin{bmatrix} \mathbf{r} - \mathbf{r}_d \\ \Delta\boldsymbol{\epsilon} \end{bmatrix} . \tag{4.20}$$

Using (4.18), (4.19b) and (4.20) with a new virtual input

$$\mathbf{v} = \left(\mathbf{J}_{0,s}^t\right)^{\dagger}\left(\ddot{\mathbf{z}}_d - \mathbf{K}_1(\dot{\mathbf{z}} - \dot{\mathbf{z}}_d) - \mathbf{K}_0\Delta\mathbf{z} - \dot{\mathbf{J}}_{0,s}^t\dot{\mathbf{q}}_s\right) \tag{4.21}$$

leads to the nonlinear error system

$$\ddot{\mathbf{z}} - \ddot{\mathbf{z}}_d + \mathbf{K}_1(\dot{\mathbf{z}} - \dot{\mathbf{z}}_d) + \mathbf{K}_0\Delta\mathbf{z} = \mathbf{0} . \tag{4.22}$$

If the matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ are chosen to be diagonal, the rows in (4.22) are decoupled. In this case, the first three rows describe the position error, the remaining three rows the orientation error. If the matrices are positive definite, the asymptotic stability around a desired trajectory can be shown using a Lyapunov argument [43].

The virtual input (4.21) only controls the six degrees of freedom of the three dimensional space. As the robot is redundant and has seven links, the remaining degree of freedom needs to be stabilized with a null-space controller. Therefore, (4.21) is extended to

$$\mathbf{v} = \left(\mathbf{J}_{0,s}^t\right)^{\dagger}\left(\ddot{\mathbf{z}}_d - \mathbf{K}_1(\dot{\mathbf{z}} - \dot{\mathbf{z}}_d) - \mathbf{K}_0\Delta\mathbf{z} - \dot{\mathbf{J}}_{0,s}^t\dot{\mathbf{q}}_s\right) + \left(\mathbf{I}_9 - \left(\mathbf{J}_{0,s}^t\right)^{\dagger}\mathbf{J}_{0,s}^t\right)\ddot{\mathbf{z}}_0 \tag{4.23}$$

whereby the term $\left(\mathbf{I}_9 - \left(\mathbf{J}_{0,s}^t\right)^{\dagger}\mathbf{J}_{0,s}^t\right)$ represents a projection of $\ddot{\mathbf{z}}_0$ onto the null-space of $\mathbf{J}_{0,s}^t$, along the range-space of $\left(\mathbf{J}_{0,s}^t\right)^{\dagger}\mathbf{J}_{0,s}^t$ [49, 50]. This representation is used to implement a null-space controller in the form

$$\ddot{\mathbf{z}}_0 = -\mathbf{K}_{1,n}\dot{\mathbf{q}} - \mathbf{K}_{0,n}(\mathbf{q} - \mathbf{q}_{d,0}) , \tag{4.24}$$

with a properly chosen value for $\mathbf{q}_{d,0}$ to be consistent with the virtual Cartesian equilibrium position $\mathbf{z}_d$ [51]. The value of $\mathbf{q}_{d,0}$ is set to $\begin{bmatrix}(\mathbf{q}_r^0)^{\mathrm{T}} & \mathbf{0}_{1\times 2}\end{bmatrix}^{\mathrm{T}}$ which is consistent in the initial position. This null-space controller is then substituted into (4.23), which is then inserted into the control law (4.17) to receive the final control law

$$\begin{aligned}\boldsymbol{\tau}_s = \mathbf{D}_s(\mathbf{q}_s)\Bigg(&\left(\mathbf{J}_{0,s}^t\right)^{\dagger}\left(\ddot{\mathbf{z}}_d - \mathbf{K}_1(\dot{\mathbf{z}} - \dot{\mathbf{z}}_d) - \mathbf{K}_0\Delta\mathbf{z} - \dot{\mathbf{J}}_{0,s}^t\dot{\mathbf{q}}_s\right) \\ &+\left(\mathbf{I}_9 - \left(\mathbf{J}_{0,s}^t\right)^{\dagger}\mathbf{J}_{0,s}^t\right)(-\mathbf{K}_{1,n}\dot{\mathbf{q}} - \mathbf{K}_{0,n}(\mathbf{q} - \mathbf{q}_{d,0}))\Bigg) + \mathbf{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s + \mathbf{g}_s(\mathbf{q}_s) .\end{aligned} \tag{4.25}$$

The control law is discretized by $\boldsymbol{\tau}_s^k = \boldsymbol{\tau}_s\left(t^k\right)$ and the robot inputs are extracted by $\mathbf{u}_{cart}^k = \begin{bmatrix}\mathbf{I}_7 & \mathbf{0}_{7\times 2}\end{bmatrix}\boldsymbol{\tau}_s^k$.

For the experimental setup, the elasticity of the drive trains of the robot joints needs to be considered. These are modeled as linear springs located between the motors and the subsequent links. For this purpose, the equations of motion (2.42) are extended to

$$\mathbf{D}_s(\mathbf{q}_s)\ddot{\mathbf{q}}_s + \mathbf{C}_s(\mathbf{q}_s, \dot{\mathbf{q}}_s)\dot{\mathbf{q}}_s + \mathbf{g}_s(\mathbf{q}_s) = \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}_s) + \boldsymbol{\tau}_{ext,s} \tag{4.26a}$$

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \mathbf{K}(\boldsymbol{\theta} - \mathbf{q}_s) = \boldsymbol{\tau}_{m,s} . \tag{4.26b}$$

Here, the rotor angles of the motors divided by the transmission ratios of the gears are denoted as $\boldsymbol{\theta}$, the motor torques multiplied by the transmission ratios as $\boldsymbol{\tau}_{m,s}$, the spring stiffnesses as $\mathbf{K}$ and the inertias of the rotors as $\mathbf{B}$. This system consists of a slow part (4.26a) and a fast part (4.26b). Using the singular perturbation approach and a new control law

$$\boldsymbol{\tau}_{m,s} = \boldsymbol{\tau}_d - \mathbf{K}_\tau(\mathbf{K}(\boldsymbol{\theta} - \mathbf{q}) - \boldsymbol{\tau}_d) - \epsilon\mathbf{D}_\tau\left(\dot{\boldsymbol{\theta}} - \dot{\mathbf{q}}\right) \tag{4.27}$$

with the new control input $\boldsymbol{\tau}_d$, the positive definite controller gain matrices $\mathbf{K}_\tau$, $\mathbf{D}_\tau$ and ϵ as the singular perturbation parameter, the quasi-static closed-loop system results in

$$\underbrace{\left(\mathbf{D}_s(\bar{\mathbf{q}}_s) + \underbrace{(\mathbf{I} + \mathbf{K}_\tau)^{-1}\mathbf{B}}_{\tilde{\mathbf{B}}}\right)}_{\tilde{\mathbf{D}}_s(\bar{\mathbf{q}}_s)} \ddot{\bar{\mathbf{q}}}_s + \mathbf{C}_s(\bar{\mathbf{q}}_s, \dot{\bar{\mathbf{q}}}_s)\dot{\bar{\mathbf{q}}}_s + \mathbf{g}_s(\bar{\mathbf{q}}_s) = \boldsymbol{\tau}_d\ . \tag{4.28}$$

Thus, (4.28) has the form of a rigid-body model with the mass matrix $\tilde{\mathbf{D}}_s(\bar{\mathbf{q}}_s)$. For the experimental setup, the mass matrix $\mathbf{D}_s(\mathbf{q}_s)$ in (4.25) is replaced by $\tilde{\mathbf{D}}_s(\mathbf{q}_s)$. The matrix $\tilde{\mathbf{B}}$ is empirically chosen to be $\mathbf{I}$. The detailed steps of the singular perturbation approach are presented in [51].

4.2.3 Connection of the Inner and Outer Loop

The connection between inner and outer control loop is shown in Figure 4.1. The controller for stabilizing the pendulum calculates desired Cartesian accelerations $\ddot{r}_{x,d}$ and $\ddot{r}_{y,d}$, which are then integrated twice to receive the desired Cartesian velocities $\dot{r}_{x,d}$ as well as $\dot{r}_{y,d}$, and positions $r_{x,d}$ and $r_{y,d}$. The value of $r_{z,d}$ is fixed to $r_{z,d} = \begin{bmatrix}0 & 0 & 1\end{bmatrix}\mathbf{p}^t_{0,r}$, with the position of the tool attachment point $\mathbf{p}^t_{0,r}$ from the initial position $\mathbf{q}_{0,r}$. The desired orientation of the tool attachment point is chosen to be the same as the orientation of the tool attachment point used in the model of the pendulum with Cartesian input (2.3). As the orientation of the tool attachment point is chosen to be fixed during the stabilization phase, the vectors for desired Cartesian and angular velocity and acceleration result in, respectively,

$$\dot{\mathbf{z}}_d = \begin{bmatrix}\dot{\mathbf{r}}_d \\ \mathbf{0}_{3\times1}\end{bmatrix} \qquad \ddot{\mathbf{z}}_d = \begin{bmatrix}\ddot{\mathbf{r}}_d \\ \mathbf{0}_{3\times1}\end{bmatrix}\ . \tag{4.29}$$

4.3 Transition between Controllers

After the swing-up is finished, the active controller is switched from the time-variant LQR for swing-up to the cascade controller for stabilization. The connections between the controllers, the switching and the system are shown in Figure 4.2. The LQR for swing-up passes a signal to the switching block as soon as the swing-up is finished. The switching block waits for the tool attachment point to reach the zero-crossing point of the coordinate

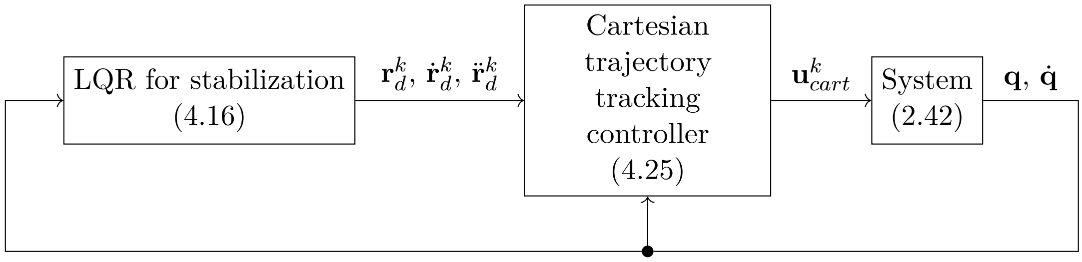

Figure 4.1: Connection of the inner and outer loop.

r_y and for

$$|q_8| < 2° \tag{4.30a}$$

$$|q_9| < 2° \tag{4.30b}$$

to be satisfied, before the switching is initiated. These measures have proven to reduce displacements during the switching. As the two controllers have different structures, the inputs of the Cartesian controller $\mathbf{u}_{cart}^k$ and the LQR for swing-up $\mathbf{u}_{su}^k$ are linearly faded with

$$\mathbf{u}_s^k = \begin{cases} \mathbf{u}_{su}^k & \text{for } k < k_f \\ \frac{T_s\left(k-k_f\right)}{T_f}\mathbf{u}_{cart}^k + \left(1 - \frac{T_s\left(k-k_f\right)}{T_f}\right)\mathbf{u}_{su}^k & \text{for } k = k_f, \ldots, k_f + \frac{T_f}{T_s} - 1 \\ \mathbf{u}_{cart}^k & \text{for } k \geq k_f + \frac{T_f}{T_s}\ , \end{cases} \tag{4.31}$$

where the swing-up finishes at $k = k_f$, with the sampling time $T_s = 1\,\mathrm{ms}$ and the fade time $T_f = 100\,\mathrm{ms}$, which is chosen to reduce displacements during the switching. To avoid wind-up of the integrators of the cascade controller, this controller is disabled during the swing-up phase and is enabled by a signal from the switch at $k = k_f + 1$.

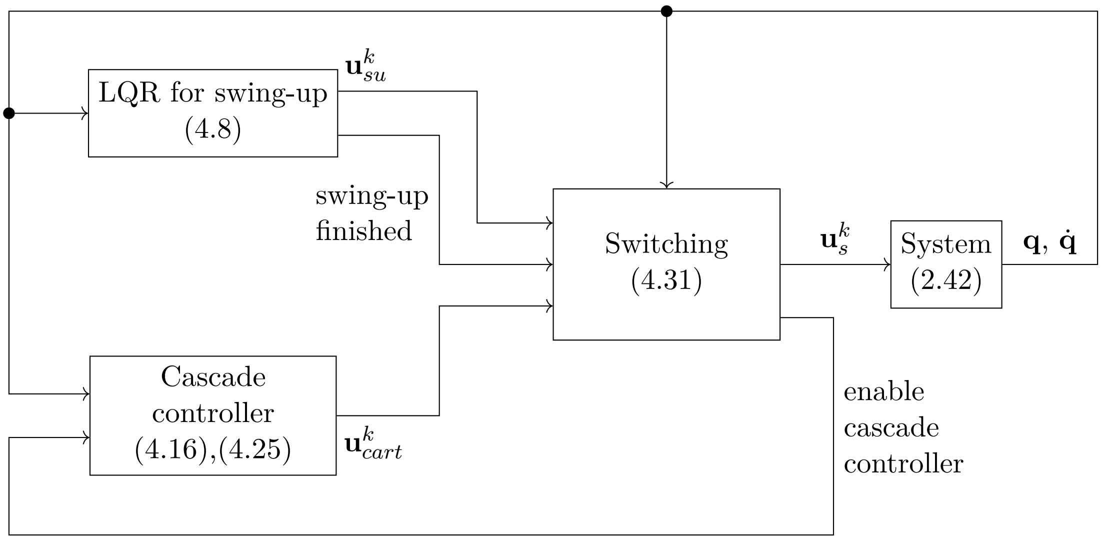

Figure 4.2: Switching between the LQR for the swing-up trajectory and the cascade controller for stabilization.

5 Simulation

In this chapter, the swing-up trajectory obtained in Chapter 3 and the designed controllers from Chapter 4 are used for simulations. The controller parameters used in the simulations are also discussed. First, an open-loop swing-up experiment is performed, which fails and shows the need for a trajectory tracking controller. Subsequently, successful simulation results of the swing-up with the LQR from Section 4.1 are shown and discussed. Additionally, the influence of non-measurable states is discussed and, finally, the transition to the cascade controller from Section 4.2 and the impact of disturbances and calibration errors are examined. The simulations are performed using MATLAB/SIMULINK.

5.1 Swing-Up Phase

First, the simulation of the open-loop system, where only the trajectory obtained in Section 3.3 is used as input for the system (2.42), is performed. The results of the simulation are shown in Figure 5.1. The states of the system diverge from the desired trajectory and the states reach values greater than the limits of the robot (see Table A.1) after less than 0.5 s, which is caused by the discretization of the swing-up trajectory in the optimization problem. Consequently, a controller which stabilizes the system around the desired trajectory is already needed for the simulations.

Next, the simulation of the closed-loop system is performed, where in addition the LQR designed in Section 4.1 is used. The weighting matrices of the LQR are chosen as

$$\begin{aligned}\mathbf{Q}_{su} &= \mathrm{diag}\Big(1,1,1,1,1,1,1,10^2,10^2,\\ &\qquad 10^{-1},10^{-1},10^{-1},10^{-1},10^{-1},10^{-1},10^{-1},10^{-1},10^{-1}\Big) \qquad (5.1\text{a})\\ \mathbf{R}_{su} &= 10^{-2}\cdot\mathbf{I}_7\ . \qquad (5.1\text{b})\end{aligned}$$

To decrease the deviations from the trajectory of the pendulum angles q_8 and q_9, their weights are increased to 10^2. As the velocities cannot be measured directly, but have to be derived from the position signal (see Section 6.1.3), the corresponding weights are decreased to 10^{-1}. This reduces oscillations and noise in the states and inputs. The weights of the inputs are decreased to 10^{-2} to allow for larger control inputs. Decreasing the weights further would cause strong oscillations when steps occur in the desired trajectory.

The results of the simulation of the closed-loop system are shown in Figure 5.2, the depicted graphs are scaled with (3.23). The LQR stabilizes the system around the desired swing-up trajectory and there is no significant difference between the desired trajectory and the simulated time evolution of the states visible. The errors of the angles $e_{q_i} = q_{d,i} - q_i$ and joint velocities $e_{\dot{q}_i} = \dot{q}_{d,i} - \dot{q}_i$ and the deviation from the desired input $\Delta u_{s,i} = u_{s,i} - u_{d,su,i}$, which is equal to the LQR input $\mathbf{u}_{LQR,su}$ until the switching between

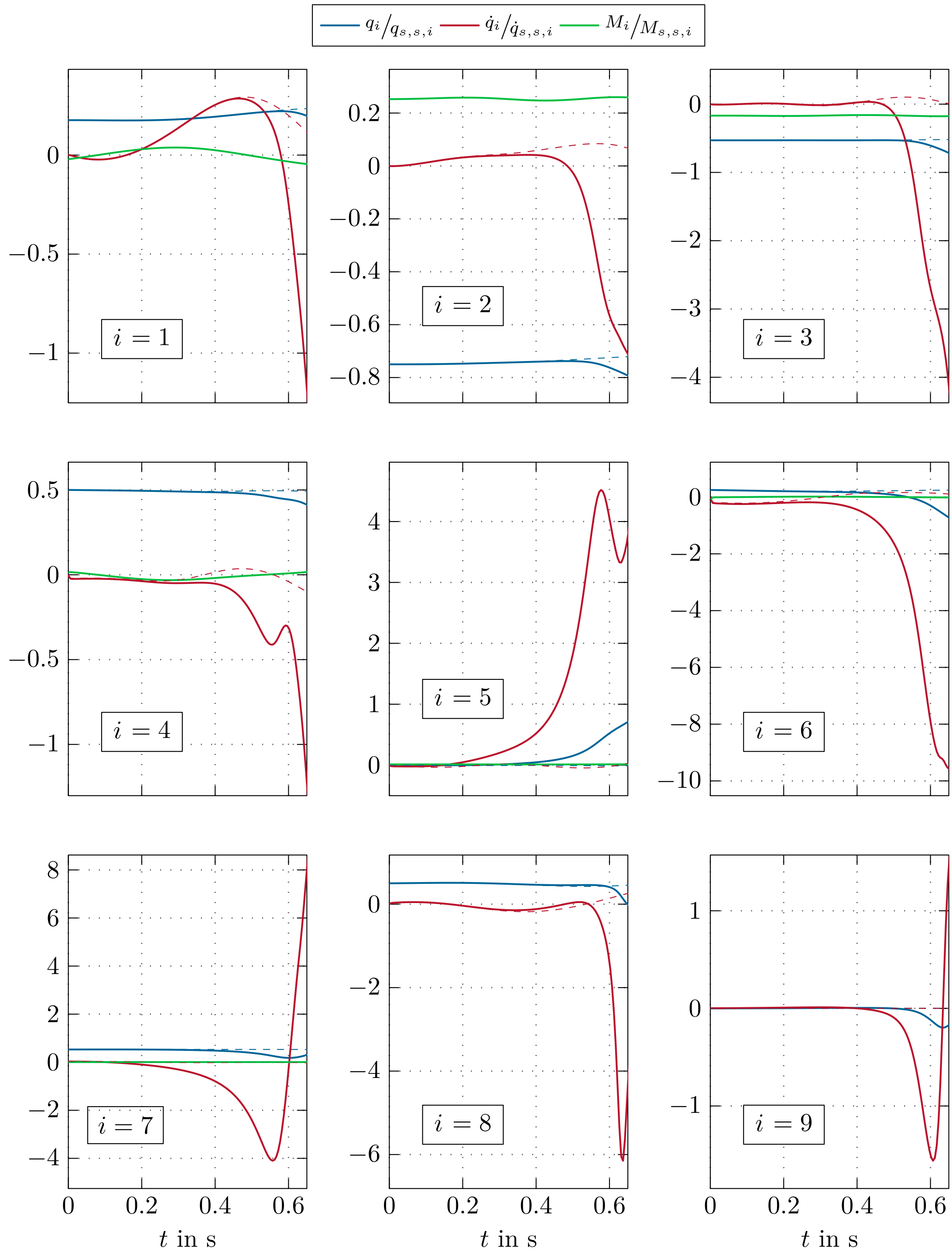

Figure 5.1: Simulation results for the states and the inputs when applying the trajectory input for the swing-up to the system without feedback control. The simulation results are shown as a solid line, the desired trajectory is shown as a thin, dashed line. The graphs are scaled with the scaling vectors from (3.23).

the controllers occurs, are depicted in Figure 5.3. The largest deviations in the joint angles can be observed in Axis 5 and Axis 7 with 0.021° and 0.022°, respectively. There are spikes visible in the joint velocity errors at the beginning and the end of the swing-up, which are caused by the rapid change in the desired velocity trajectory. However, these spikes are negligible, as the largest spike is less than 1.9 °/s for Axis 6. The necessary control input of the LQR is less than 0.2 N m in all axes. Small deviations from the operating point can be observed at around 3.7 s as the switching between the controllers is performed.

As there are non-measurable states in the experimental setup, i. e. the joint velocities, the influence of these states is examined by simulation. For this purpose, the angles of q_8 and q_9 are quantized with $\frac{2\pi}{2^{14}}$ to simulate a digital incremental encoder with 14 bit resolution per revolution. All joint velocities are calculated from the joint angles using a zero-order hold discretized D-T$_1$ filter, which is derived from the continuous-time filter

$$G(s) = \frac{s}{T_1 s + 1} , \tag{5.2}$$

with $T_1 = 15\,\mathrm{ms}$. The results from the simulation are shown in Figure 5.4. Compared to the simulation shown in Figure 5.2, large deviations from the desired trajectory can be observed, which are caused by the phase lag introduced by the D-T$_1$ filter. Additionally, noise caused by the quantization and filtering is visible. The errors of the angles and the joint velocities, as well as $\Delta\mathbf{u}_s$ are depicted in Figure 5.5. The influence of the weights of the LQR are visible in the figure, as the maximum errors of the angles of q_8 and q_9 increase to 1.9° and 1.2°, respectively, while the maximum errors of the angles of the robot joints increase to 9.7° in Axis 6. The errors of the joint velocities also increase significantly. The maximum error of Axis 5 increases from 0.41 °/s to 51.0 °/s. The input generated by the LQR is less than 3 N m in all axes, the noise amplitude is less than 0.4 N m in all axes. The switching between the controllers is visible at around 4.6 s.

5.2 Pendulum Stabilization Phase

The stabilization phase starts after the switching between the controllers (see Section 4.3). During this phase, the cascade controller from Section 4.2 is used. To test the closed-loop behavior of the cascade controller, the forces

$$F_x = 0.1\,\mathrm{N}(\sigma(t - 5\,\mathrm{s}) - \sigma(t - 5.05\,\mathrm{s})) \tag{5.3a}$$
$$F_y = 0.1\,\mathrm{N}(\sigma(t - 8\,\mathrm{s}) - \sigma(t - 8.05\,\mathrm{s})) , \tag{5.3b}$$

with the step function $\sigma(t)$, are applied as disturbances to the system (2.42) via (2.18) and (2.41). The aforementioned quantization and approximate differentiation are neglected again for this simulation.

The weighting matrices of the LQR stabilizing the pendulum (4.16) are chosen as

$$\mathbf{Q}_{q_8} = \mathrm{diag}(1, 1, 15, 1, 5) \tag{5.4a}$$
$$\mathbf{R}_{q_8} = 1 \tag{5.4b}$$
$$\mathbf{Q}_{q_9} = \mathrm{diag}(1, 1, 15, 1, 5) \tag{5.4c}$$
$$\mathbf{R}_{q_9} = 1 , \tag{5.4d}$$

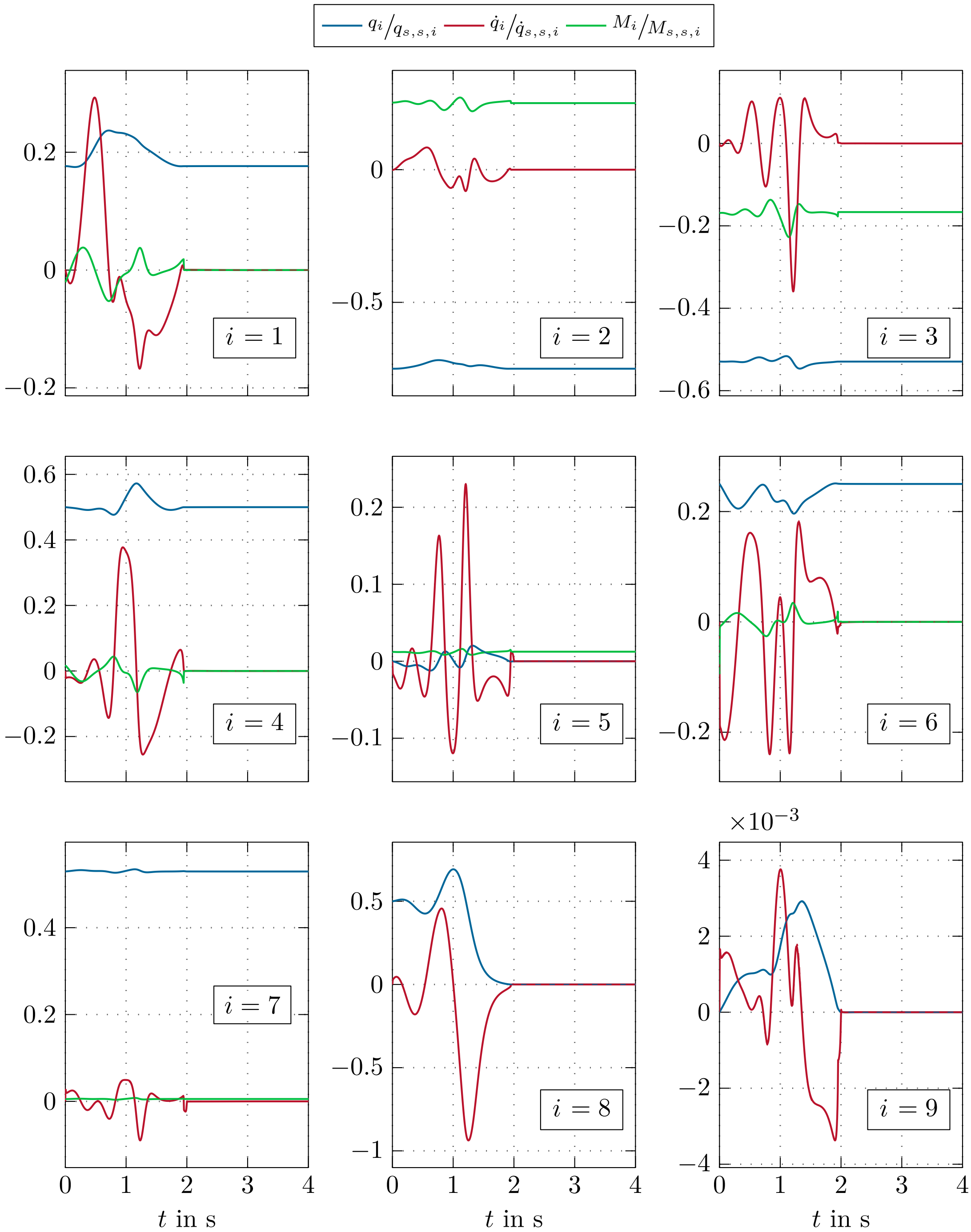

Figure 5.2: Simulation results for the states and the input of the swing-up with LQR feedback control. The simulation results are shown as a solid line, the desired trajectory is shown as a thin, dashed line. Due to the almost perfect alignment between simulation results and the desired trajectory, the thin, dashed line is barely visible. The graphs are scaled with the scaling vectors from (3.23).

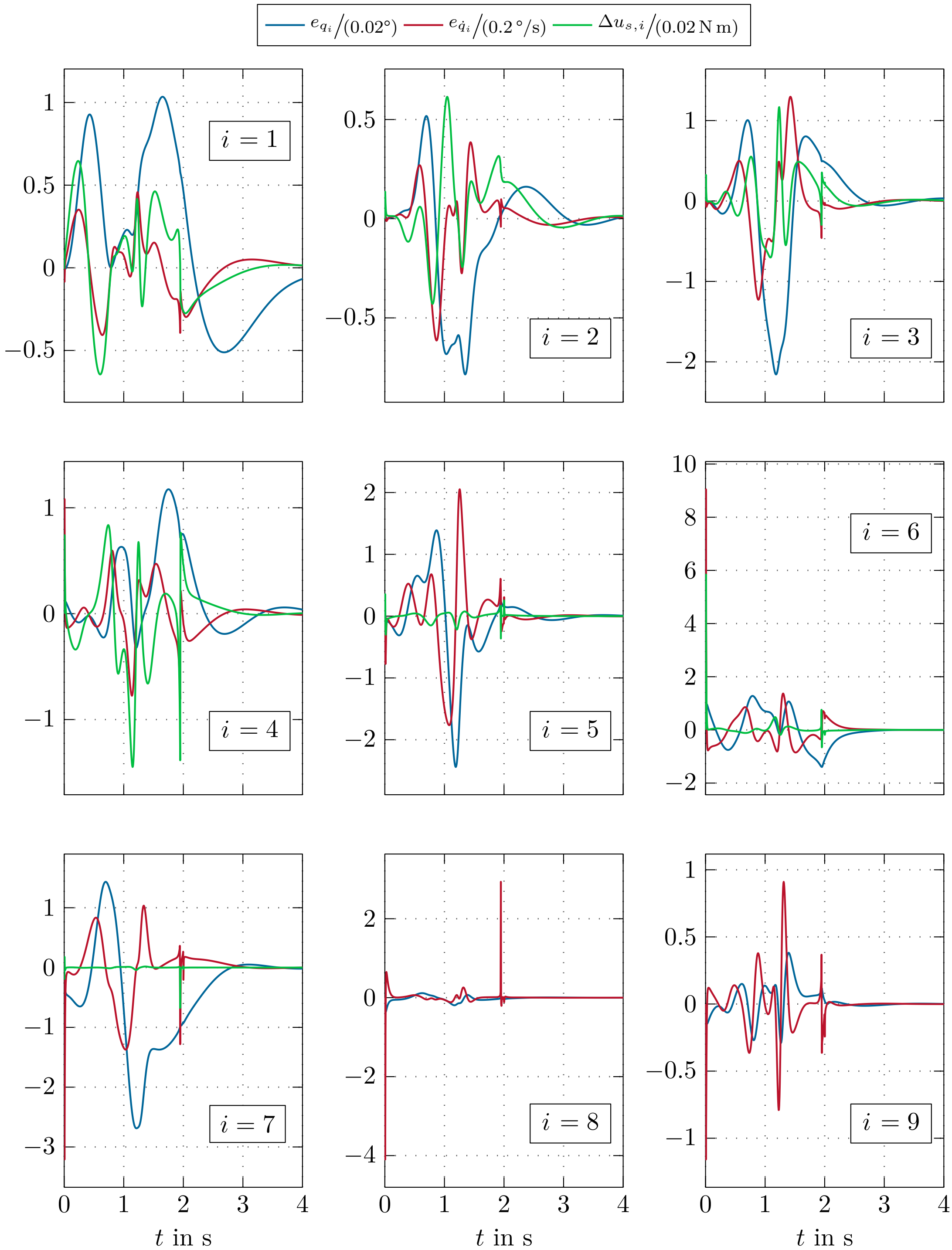

Figure 5.3: Simulation results for the errors of the states and the control input of the swing-up with LQR feedback control.

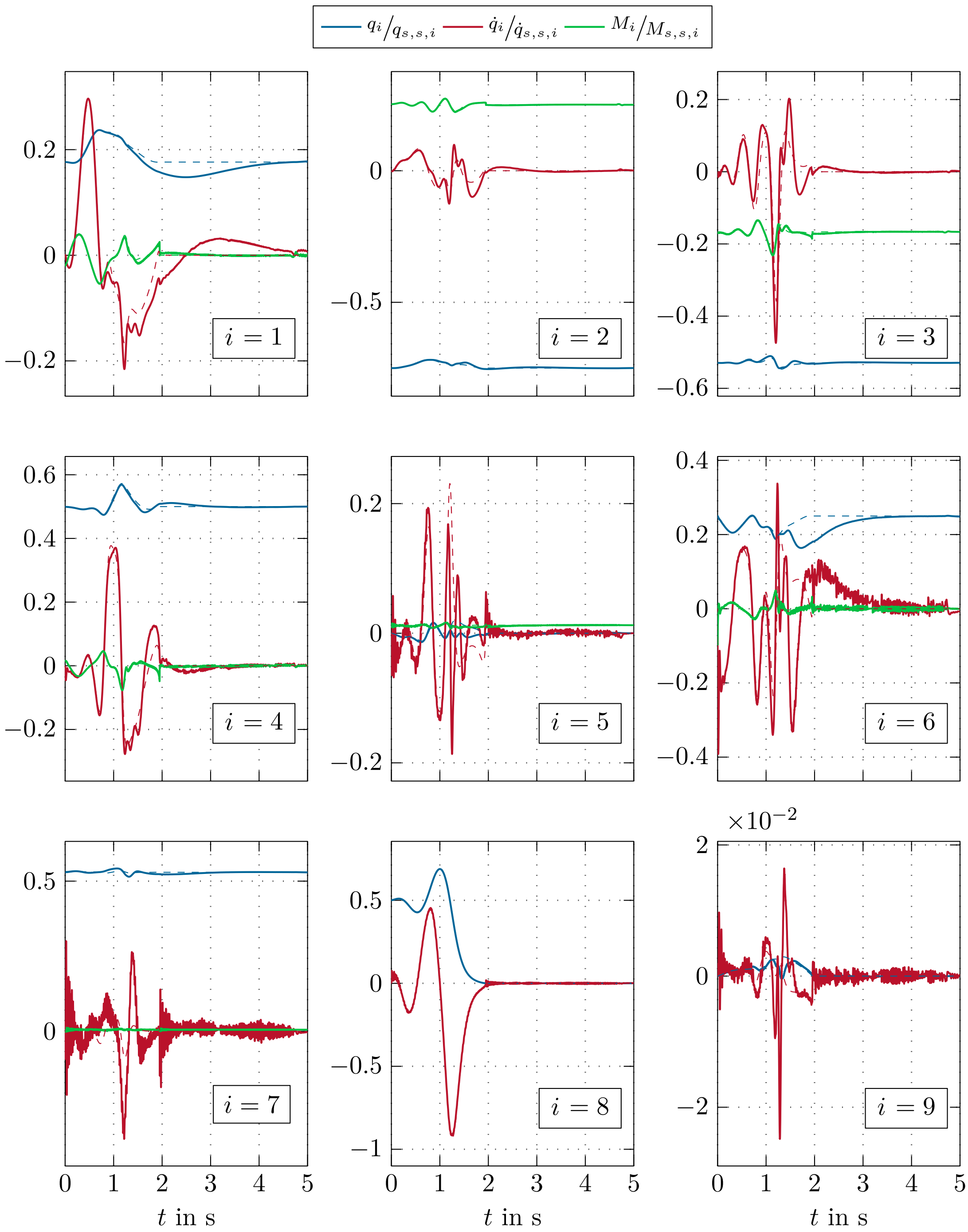

Figure 5.4: Simulation results for the states and the input of the swing-up with LQR feedback control including quantization and calculation of the velocities based on approximate differentiation. The simulation results are shown as a solid line, the desired trajectory is shown as a thin, dashed line. The graphs are scaled with the scaling vectors from (3.23).

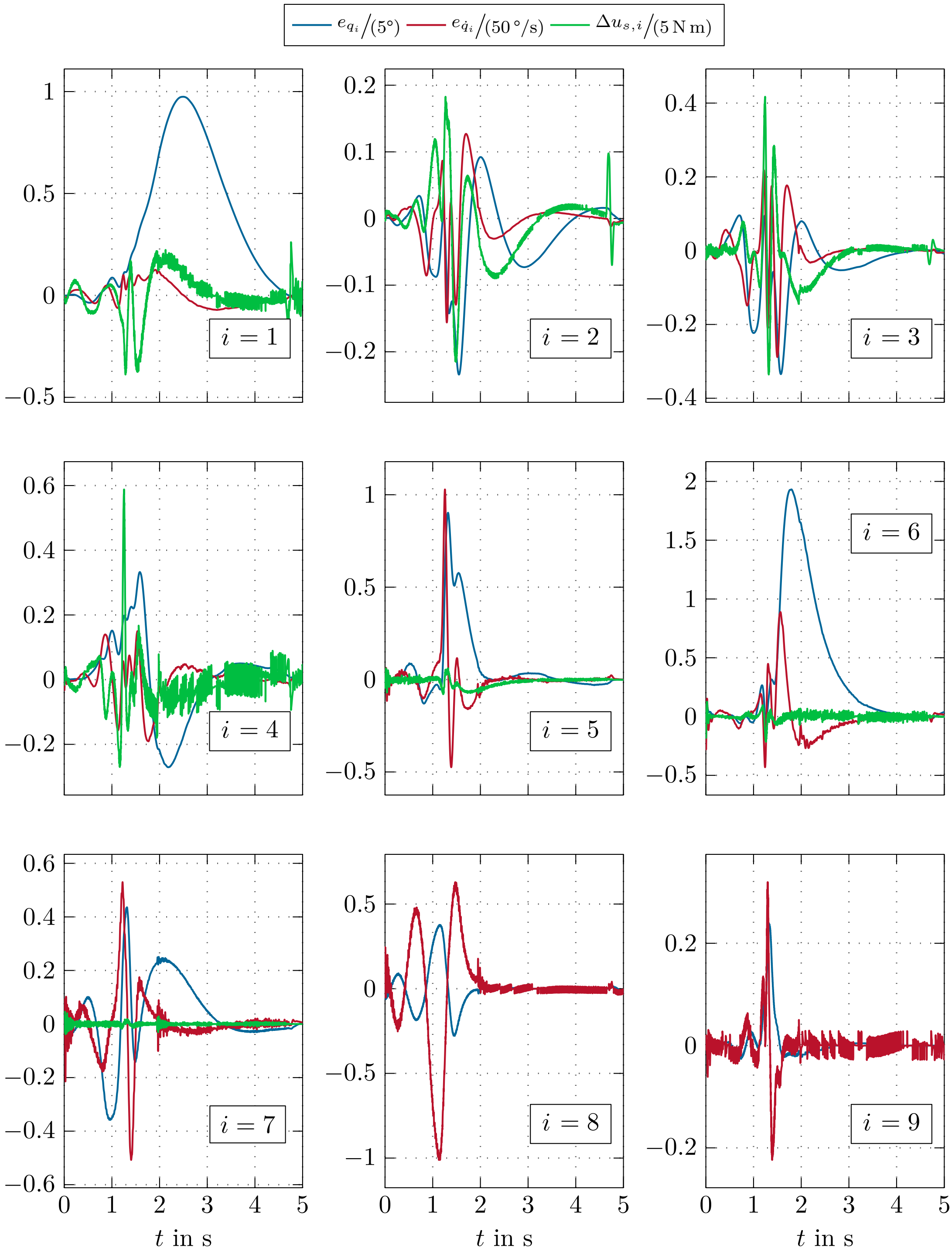

Figure 5.5: Simulation results for the errors of the states and the control input of the swing-up with LQR feedback control including quantization and calculation of the velocities based on approximate differentiation.

which results in a stable closed-loop system in the simulation using the nonlinear model. The weights of r_x and r_y are increased to 15 to retain the displacements occurring when disturbances are applied to the system. To decrease the time needed for compensating model inaccuracies and calibration errors, the weights of the integrator states are increased to 5.

The control gain matrices of the Cartesian trajectory tracking controller (4.25) are chosen as

$$\mathbf{K}_0 = 484 \cdot \mathrm{diag}(1,1,1,1,1,1) \tag{5.5a}$$
$$\mathbf{K}_1 = 44 \cdot \mathrm{diag}(1,1,1,1,1,1) \tag{5.5b}$$
$$\mathbf{K}_{0,n} = 10 \cdot \mathrm{diag}(1,1,1,1,1,1,1,0,0) \tag{5.5c}$$
$$\mathbf{K}_{1,n} = 3 \cdot \mathrm{diag}(1,1,1,1,1,1,1,0,0)\ . \tag{5.5d}$$

The matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ result from placing all poles of the error system (4.22) to $\lambda_i = -22$. These values originate from the experiments in Chapter 6 and are the fastest poles which could be used on the experimental setup. Therefore, these poles are also used in the simulation. The matrices for the null-space controller $\mathbf{K}_{0,n}$ and $\mathbf{K}_{1,n}$ have the dimension 9×9. However, as the null-space controller only controls the seven joints of the robot, the last two elements on the diagonal are set to zero. The gains of the matrices have proven to stabilize the null space in the simulation.

The joint angles, joint velocities and the control input for the stabilizing phase are shown in Figure 5.6. Due to the pulse-shaped disturbances, the joint angles of the robot are deflected up to 35° (Axis 4), while the angles of the pendulum are deflected to a maximum of 3.1° (Axis 8). The deviation of the control input from the operating point is less than 10.1 N m. The error of the Cartesian controller is shown in Figure 5.7. This figure depicts the position and orientation error

$$\Delta\mathbf{z} = \begin{bmatrix} e_{r_x} & e_{r_y} & e_{r_z} & e_{\epsilon_1} & e_{\epsilon_2} & e_{\epsilon_3} \end{bmatrix}^{\mathrm{T}} \tag{5.6}$$

and velocity error

$$\dot{\mathbf{z}} - \dot{\mathbf{z}}_d = \begin{bmatrix} e_{\dot{r}_x} & e_{\dot{r}_y} & e_{\dot{r}_z} & e_{\omega^7_{0,x_7}} & e_{\omega^7_{0,y_7}} & e_{\omega^7_{0,z_7}} \end{bmatrix}^{\mathrm{T}} \tag{5.7}$$

from (4.20) and (4.22). The Cartesian trajectory tracking controller achieves errors in the Cartesian position and velocity of the tool attachment point of less than 0.085 mm and 0.71 mm/s, respectively. The orientation error (4.20) is smaller than $3.4 \cdot 10^{-3}$ and the angular velocity error of the tool attachment point is below 0.095 rad/s.

Next, the impact of offsets in the measurements of joint angles of the robot and the pendulum axes is examined. These offsets can occur if the sensors of the robot and the pendulum are not properly calibrated. In the simulation, the offset vector

$$\mathbf{q}_{o,s} = \begin{bmatrix} -0.2 & 0.3 & 0.1 & -0.3 & 0.2 & 0.1 & 0.2 & 0.1 & -0.2 \end{bmatrix}^{\mathrm{T}} \cdot 1° \tag{5.8}$$

is added to the measurement of $\mathbf{q}_s$. The position $\mathbf{r}$ of the tool attachment point is shown in Figure 5.8 and compared to the position without the added offset. The figure depicts

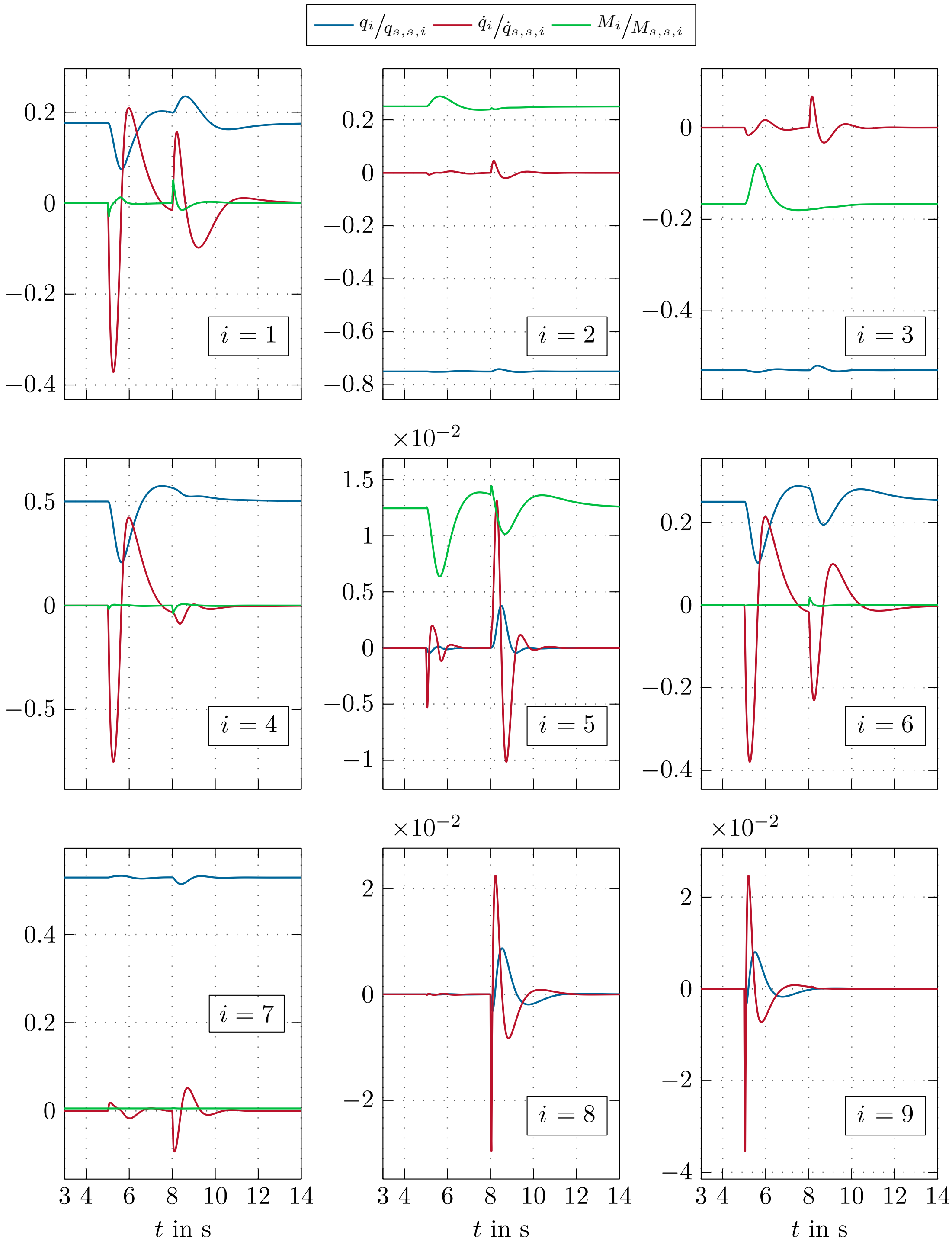

Figure 5.6: Simulation results for the states and the input of the stabilization phase using the cascade controller. The graphs are scaled with the scaling vectors from (3.23). Disturbances F_x and F_y are acting on the system at $t = 5\,\mathrm{s}$ and $t = 8\,\mathrm{s}$, respectively.

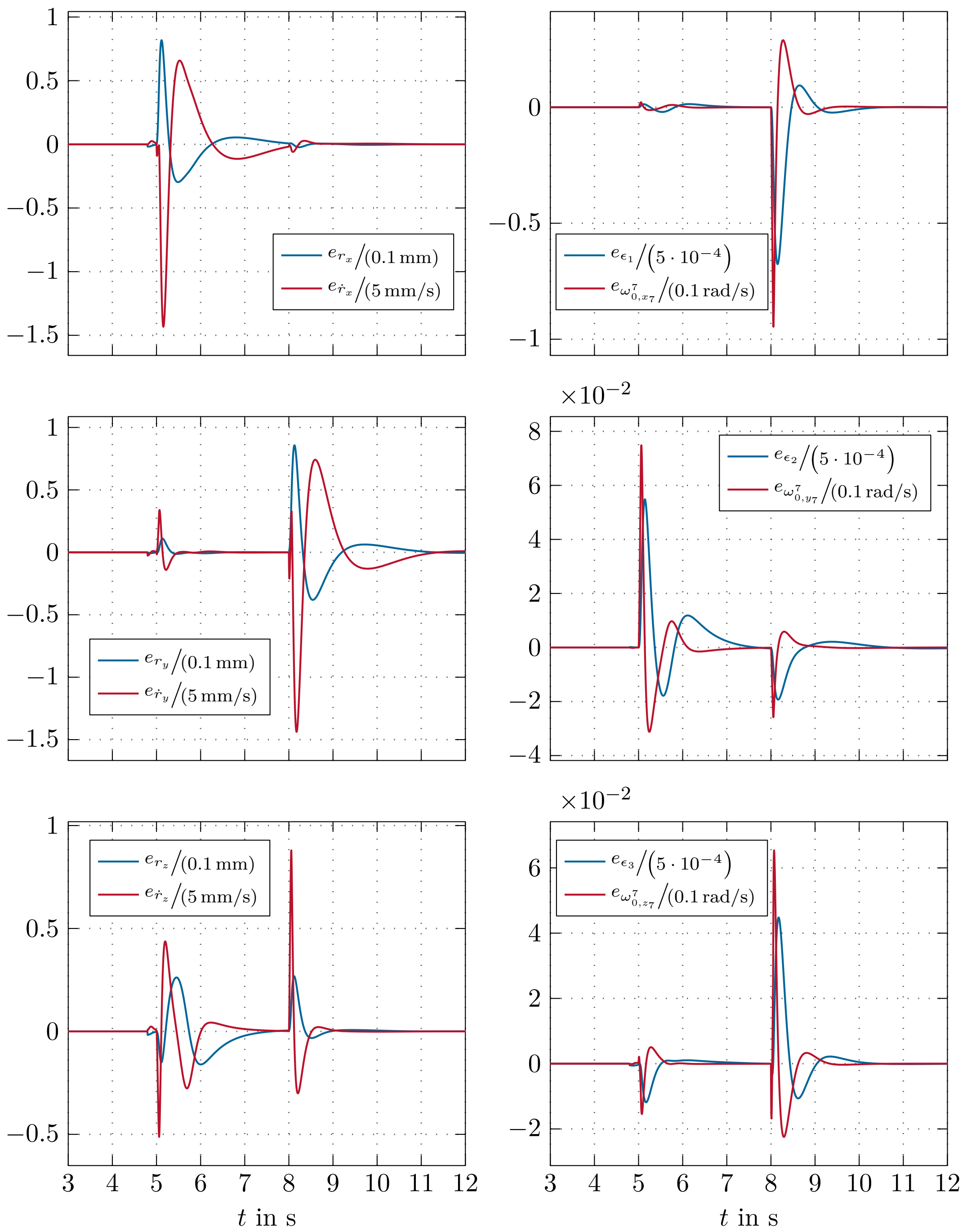

Figure 5.7: Simulation results for the errors of the Cartesian trajectory tracking controller during the stabilization phase using the cascade controller. Disturbances F_x and F_y are acting on the system at $t = 5\,\mathrm{s}$ and $t = 8\,\mathrm{s}$, respectively.

the swing-up phase and the stabilization phase. The depicted positions are calculated from the angle measurements, which include the calibration offsets (5.8). Hence, there is no offset at the beginning of the simulation. The switching between the controllers is forced at 2.5 s as the transition condition is not naturally met. After the transition, the deflection from the desired position $\mathbf{r}_0$ can be observed. The controllers for stabilizing the pendulum manage to move the tool attachment point back to the desired position even if disturbances occur.

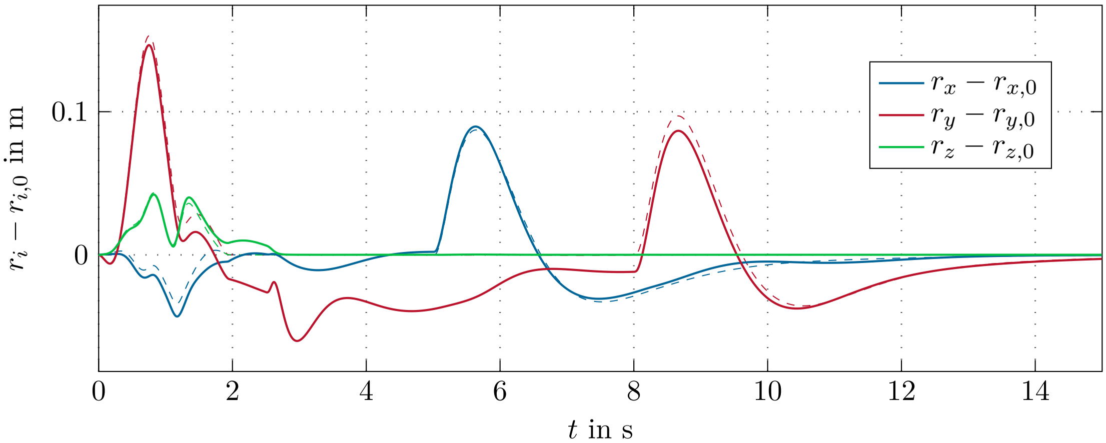

Figure 5.8: Simulation results for the position of the tool attachment point $\mathbf{r}$. The simulation results with an offset $\mathbf{q}_{o,s}$ from (5.8) added to the measurement of the joint angles $\mathbf{q}$ are shown as a solid line, the simulation results without offset are shown as a thin, dashed line. The swing-up can be seen between $t = 0\,\mathrm{s}$ and $t = 2\,\mathrm{s}$, the controller switching is performed at 2.5 s and the disturbances F_x and F_y are acting on the system at $t = 5\,\mathrm{s}$ and $t = 8\,\mathrm{s}$, respectively.

6 Experiments

In this chapter, the experimental setup and the implementation of the controllers are described and the results from the experiments are discussed. First, an overview of the experimental setup is given and the data exchange between the components of the setup is introduced. Moreover, the results from the experimental swing-up using the trajectory obtained in Section 3.3 and the LQR from Section 4.1 are presented. Finally, the impact of disturbances on the closed-loop system using the cascade controller from Section 4.2 during the stabilization phase is shown.

6.1 Experimental Setup and Implementation

The experimental setup can be roughly divided into four parts: the computer running the controller, the robot controller Kuka KR C2 lr, the robot Kuka LWR IV+ and the spherical inverted pendulum including its sensors and communication interface. After giving an overview, the connections between the components are discussed. Subsequently, the robot interface, the sensors of the spherical inverted pendulum, and the velocity filters are shortly presented.

6.1.1 Overview of the Setup

The overview of the experimental setup is shown in Figure 6.1. The computer is running the controller used for the experimental setup. The calculations of the controller are performed with the real-time software TwinCAT. Due to the availability of a communication path between Matlab/Simulink and TwinCAT, the experimental setup can be controlled directly with the help of Matlab/Simulink. The computer is equipped with two network interface cards (NIC). One is connected to the sensors of the pendulum and receives the measurements from the joint angles of the pendulum. The second NIC is connected via an Ethernet cable to the robot controller Kuka KR C2 lr. The computer and the robot controller exchange measurement values of the joint angles of the robot and the calculated torque control input.

The robot controller Kuka KR C2 lr is connected to the robot Kuka LWR IV+ to receive the measurements of the joint angles of the robot and to send the control inputs to the robot. The robot controller also acts as a power source for the robot.

6.1.2 Robot Interface

The computer communicates with the robot controller using the Fast Research Interface (FRI). This interface provides the measurement values of the joint angle of the robot, allows the configuration of the sampling time, which is set to $T_s = 1\,\mathrm{ms}$, and provides

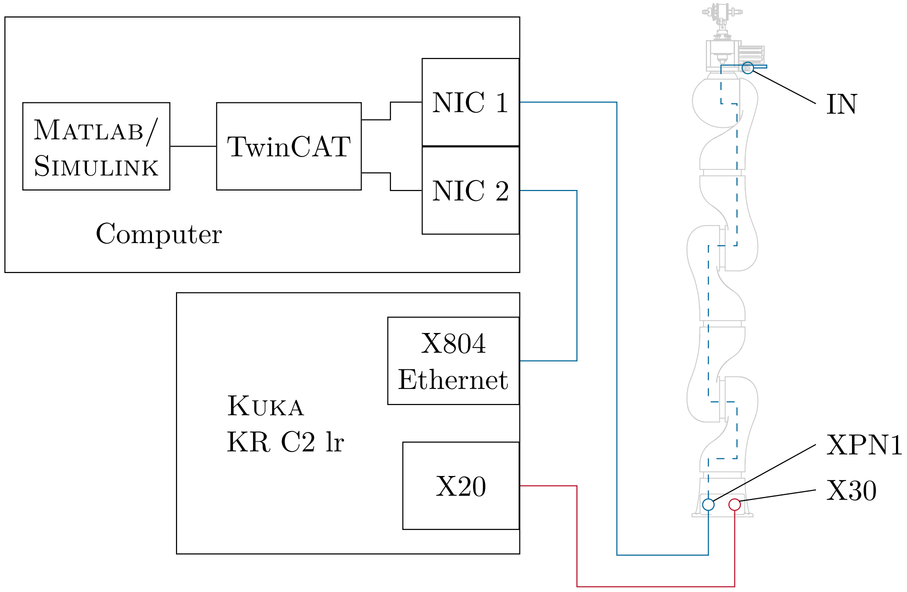

Figure 6.1: Overview of the experimental setup. The ethernet cables are shown in blue, the connection between the robot controller and the robot is shown in red. Dashed lines depict internal cables [47, 52].

different control strategies. In this thesis, the control strategy 30 (joint specific impedance control) is used [53], which utilizes the control law

$$\boldsymbol{\tau}_{cmd} = \mathbf{k}_j(\mathbf{q}_{FRI} - \mathbf{q}_{msr}) + \mathbf{D}(\mathbf{d}_j) + \boldsymbol{\tau}_{FRI} + \mathbf{f}_{dynamics}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \ . \tag{6.1}$$

This control law is adapted to allow for a direct torque input to implement the controllers derived in Chapter 4. Therefore, the first two terms in (6.1) are eliminated. By setting $\mathbf{k}_j$ to zero, the stiffness term $\mathbf{k}_j(\mathbf{q}_{FRI} - \mathbf{q}_{msr})$ is eliminated and by setting $\mathbf{d}_j$ to zero the damping term $\mathbf{D}(\mathbf{d}_j)$ is also eliminated. The term $\mathbf{f}_{dynamics}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ in the control law (6.1) is not documented well and is unknown, but is assumed to compensate the friction in the joints, the gravity acting on the robot and possibly also Coriolis terms. Hence, the robot gravity vector $\mathbf{g}_r$ is subtracted from $\mathbf{g}_s$ in (4.8) and $\mathbf{u}^k_{d,su}$ in (4.25). Additionally, the matrix $\mathbf{C}_s$ (2.36) is set to zero in (4.25) as the Coriolis terms are negligible compared to the terms which stem from the mass matrix during the stabilization phase. Additionally, this simplification also reduces the compilation time of the controller.

6.1.3 Sensors

Two RLS Orbis rotary encoders [54] are used to measure the pendulum angles q_8 and q_9. The rotary encoders consist of diametrically magnetized permanent ring magnets and printed circuit boards with Hall sensors and a communication interface. As the rotary encoders are based on a non-contact measurement principle, no additional friction is introduced to the system. The resolution of the rotary encoders is 14 bit. The rotary encoders provide relative angles, whereby calibration is required before each experiment.

The encoders are interfaced by a BECKHOFF SSI terminal EL5002 [55] via the synchronous serial interface (SSI). This terminal is connected to a BECKHOFF EtherCAT coupler EK1100 [56], communicating via the EtherCAT protocol with the computer running TwinCAT and MATLAB. The rotary encoders are powered by a BECKHOFF EL9505 [57] 5 V power supply. All three BECKHOFF terminals are mounted to the mounting enclosure of the spherical inverted pendulum (compare Figure 2.2).

6.1.4 Velocity Filters

As the joint velocities of the robot and the pendulum cannot be measured directly, a zero-order hold discretized D-T_1 filter (5.2) is used to numerically form the time derivative of the angles in order to obtain the joint velocities. The discretized equations of the filter are

$$\dot{\mathbf{q}}^k = \mathbf{x}_{\dot{\mathbf{q}}}^k + \frac{1}{T_s}\left(1 - \mathrm{e}^{\left(-\frac{T_s}{T_1}\right)}\right)\mathbf{q}^k \tag{6.2a}$$

$$\mathbf{x}_{\dot{\mathbf{q}}}^{k+1} = \mathrm{e}^{\left(-\frac{T_s}{T_1}\right)}\mathbf{x}_{\dot{\mathbf{q}}}^k - \frac{1}{T_s}\left(1 - \mathrm{e}^{\left(-\frac{T_s}{T_1}\right)}\right)^2\mathbf{q}^k \ , \tag{6.2b}$$

with the output of the filter $\dot{\mathbf{q}}^k$, the internal state $\mathbf{x}_{\dot{\mathbf{q}}}^k$ and the initial condition

$$\mathbf{x}_{\dot{\mathbf{q}}}^0 = -\frac{1}{T_s}\left(1 - \mathrm{e}^{\left(-\frac{T_s}{T_1}\right)}\right)\mathbf{q}^0 \tag{6.3a}$$

$$\dot{\mathbf{q}}^0 = \mathbf{0} \ . \tag{6.3b}$$

The sampling time is $T_s = 1\,\mathrm{ms}$ and the time constant is chosen as $T_1 = 15\,\mathrm{ms}$, which has proven to allow the fastest poles in the closed-loop system for the Cartesian trajectory tracking controller.

6.2 Swing-Up Phase

The swing-up is performed using the setup described in Section 6.1, the swing-up trajectory obtained in Section 3.3 and the LQR designed in Section 4.1. For the experimental swing-up, the weights of the LQR are adapted from (5.1) to

$$\begin{aligned}\mathbf{Q}_{su} = \mathrm{diag}\Big(&1, 1, 1, 1, 1, 1, 1, 10^{-5}, 10^{-2}, \\ &10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-1}, 10^{-6}, 10^{-4}\Big)\end{aligned} \tag{6.4a}$$

$$\mathbf{R}_{su} = 10^{-3} \cdot \mathbf{I}_7 \ . \tag{6.4b}$$

To reduce oscillations in the closed-loop system, the weights of q_8 and $\dot{q}_8$ are decreased from 10^2 to 10^{-5} and from 10^{-1} to 10^{-6}, respectively, and the weights of q_9 and $\dot{q}_9$ are decreased from 10^2 to 10^{-5} and from 10^{-1} to 10^{-4}, respectively. This selection also has the benefit that the controller transition condition in Section 4.3 is naturally met and a forced switching in an unfavorable position, as performed in Section 5.2, can be avoided. Additionally, the weights of the input are decreased from 10^{-2} to 10^{-3}, whereby oscillations in the closed-loop system are also reduced.

The joint angles, joint velocities and the input of the swing-up performed on the experimental setup are shown in Figure 6.2. The depicted graphs are scaled with the scaling vectors from (3.23). The LQR stabilizes the system around the desired trajectory, but deviations from the desired trajectory are visible. Additionally, noise caused by the quantization and filtering is also visible, especially in the inputs and $\dot{q}_9$. Compared to the simulation, larger deviations from the desired trajectory of q_8, $\dot{q}_8$, q_9 and $\dot{q}_9$ can be observed. This roots in the fact that the weights in the LQR were significantly reduced, the parameters of the experimental setup are not exactly known and friction is neglected in the simulation. The joint angle of Axis 7 deflects less than 0.007°, as the control input for Axis 7 is too small to overcome the static friction, while the desired trajectory has deviations from the start point of approximately 0.82°. The switching between the controllers can be observed at around 3.0 s. Oscillations during and after the transition can be observed. These oscillations originate from the poorly calibrated robot joint and pendulum rotary encoders, thus the swing-up controller and the stabilizing controller have very different actual operating points in the workspace. This means that the stabilizing controller needs to compensate for a large initial deflection when the transition is performed. The fade time $T_f = 100\,\mathrm{ms}$ in (4.31) is chosen to minimize these oscillations.

The errors of the angles and the joint velocities, as well as $\Delta\mathbf{u}_s$ are depicted in Figure 6.3. Noise caused by the quantization and filtering is visible. The maximum error in the joint angles of the robot can be observed in Axis 1, compared to Figure 5.5 it increases from 4.9° to 7.1°. Conversely, the maximum error in Axis 6 decreases from 9.7° to 6.1°. The errors of the pendulum angles q_8 and q_9 increase from 1.9° and 1.2° to 13.6° and 4.0°, respectively. The maximum joint velocity error can be observed in Axis 8 with 147.2 °/s. This significant increase of errors in Axis 8 and 9 is caused by the reduced weights of q_8, $\dot{q}_8$, q_9, and $\dot{q}_9$. A simulation with the same weights for the swing-up controller shows smaller errors, which indicates that effects such as parameter uncertainties and friction, which are neglected in the simulation, increase the errors. The control input increases in all axes and is largest in Axis 4 with 9.9 N m. This is caused by the reduced weights of the inputs and the overall larger errors while performing the experiment. After the swing-up is finished at $t = 2\,\mathrm{s}$, constant errors for the joint angles and joint velocities of zero in Axis 2, 3, 5 and 7 can be observed, as the swing-up controller has too small gains to overcome the static friction in those joints. During the transition between controllers large deviations and oscillations can be observed in all axes. As mentioned they are caused by the poorly calibrated robot joint and pendulum rotary encoders. The most significant deviations of the operating point of the control input $\Delta u_{s,i}$ can be observed in Axis 1 with 42.4 N m.

6.3 Pendulum Stabilization Phase

The stabilization phase starts after the switching from the swing-up controller to the stabilizing controller (see Section 4.3). During this phase, the cascade controller from

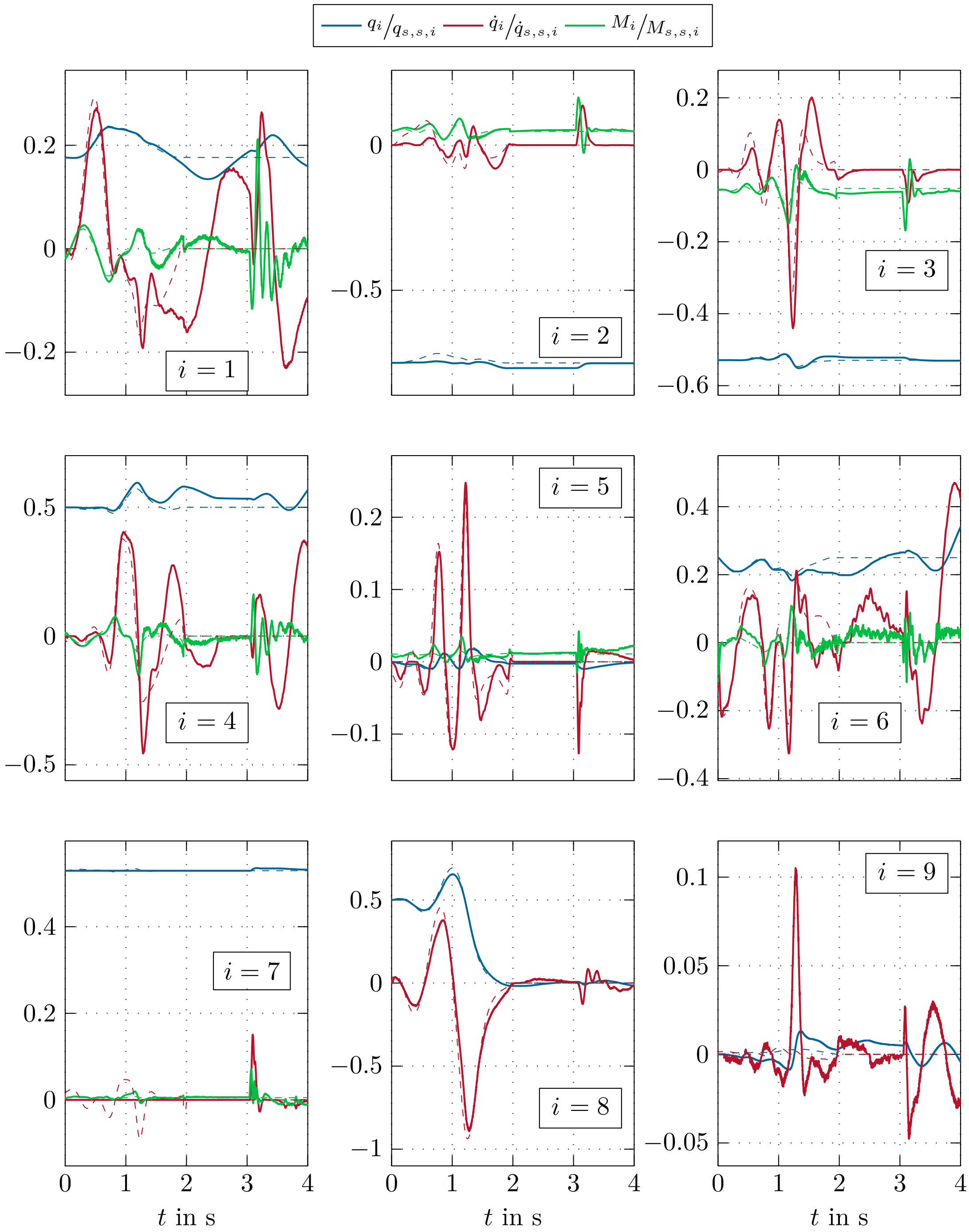

Figure 6.2: Experimental results for the states and the input of the swing-up with LQR feedback control. The measured signals are shown as a solid line, the desired trajectory is shown as a thin, dashed line. The graphs are scaled with the scaling vectors from (3.23). The switching between the controllers is performed at around 3.0 s.

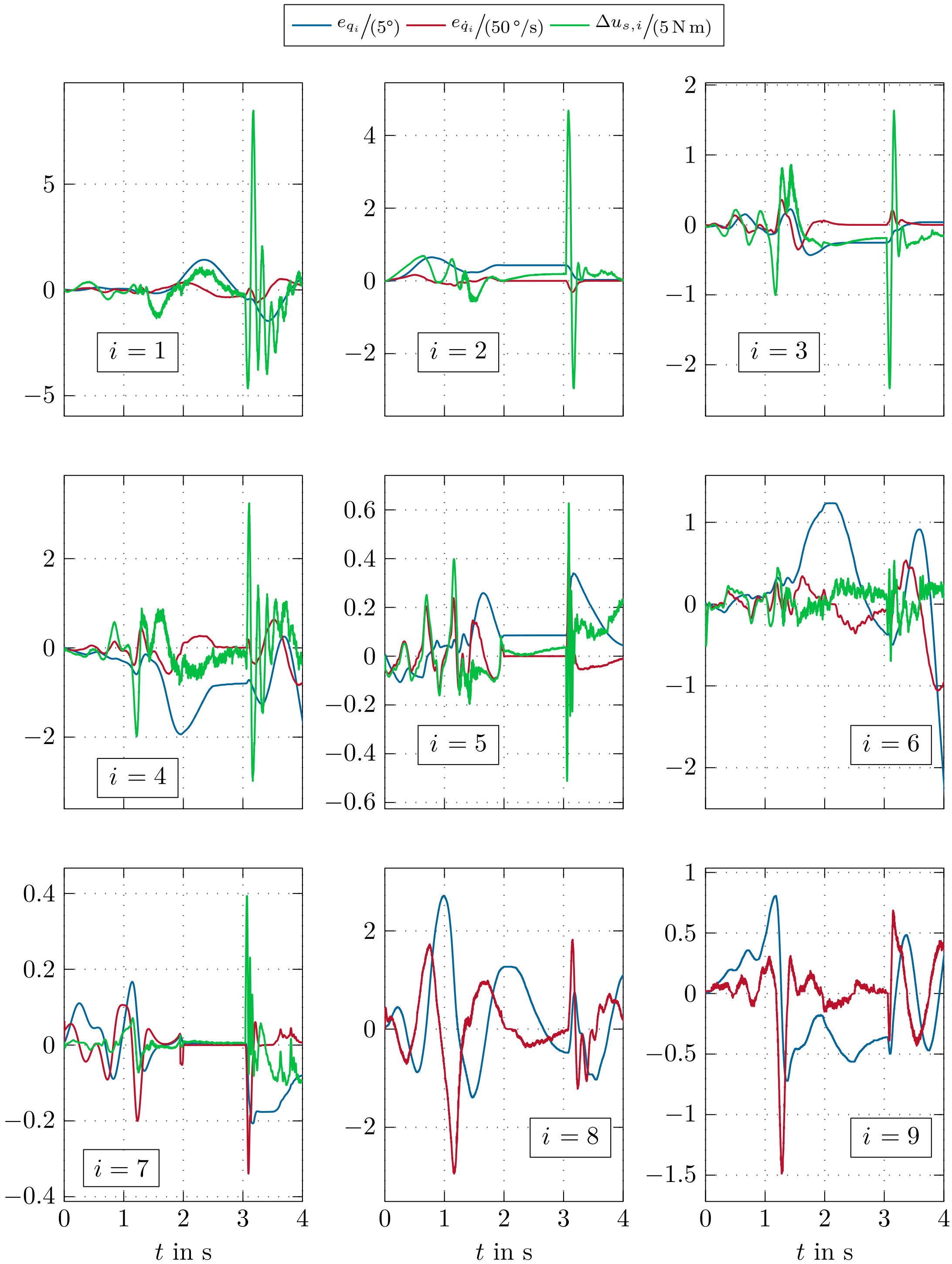

Figure 6.3: Results for the errors of the states and the control input of the pendulum swing-up on the experimental setup with LQR feedback control.

Section 4.2 is used. The weights of the LQR stabilizing the pendulum (4.16) for the experiments are

$$\mathbf{Q}_{q_8} = \mathrm{diag}(1, 1, 15, 1, 0.1) \tag{6.5a}$$

$$\mathbf{R}_{q_8} = 1 \tag{6.5b}$$

$$\mathbf{Q}_{q_9} = \mathrm{diag}(1, 1, 15, 1, 0.1) \tag{6.5c}$$

$$\mathbf{R}_{q_9} = 1 \ . \tag{6.5d}$$

The weights of the integrator states are decreased from 5 to 0.1 to avoid strong impacts of disturbances on the integrator state. While the resulting gains provide a stable closed-loop system, the occurring oscillations have a high enough frequency and a large enough amplitude to damage the experimental setup including the robot. Therefore, the resulting gains are further tuned to decrease the oscillation frequency and to thereby avoid any damage of the experimental setup. The new gain vectors used during the experiments are chosen as

$$\mathbf{k}_{q_8} = \mathbf{k}_{q_9} = \begin{bmatrix} 25 & 2 & -4 & 4.8 & -0.3 \end{bmatrix}^{\mathrm{T}} \ . \tag{6.6}$$

The gains of the pendulum angle errors and the pendulum joint velocity errors are decreased from around 30 to 25 and from around 4 to 2, respectively, to avoid quivering motions of the overall system. The gains for the position and velocity error remain approximately the same, but are fine tuned to reduce the amplitude of the oscillations. The gain matrices $\mathbf{K}_0$, $\mathbf{K}_1$ and $\mathbf{K}_{0,n}$ of the Cartesian trajectory tracking controller (4.25) remain unchanged from (5.5a), (5.5b) and (5.5c), respectively. The gain matrix $\mathbf{K}_{1,n}$ is adapted to

$$\mathbf{K}_{1,n} = 1 \cdot \mathrm{diag}(1, 1, 1, 1, 1, 1, 1, 0, 0) \tag{6.7}$$

due to the damping caused by the friction of the experimental setup.

The position of the tool attachment point is shown in Figure 6.4, the swing-up is observable between 0 s and 2 s. After the swing-up is finished, the swing-up controller stabilizes the pendulum and moves the joints of the robot back to their desired position, such that the transition condition is eventually met. This condition is met at around 3.0 s and the switching between the controllers is performed. In the seconds following the transition, large deviations of up to 0.23 m in r_y from the initial position of the tool attachment point $\mathbf{r}_0$ occur, which are caused by the inaccurate calibration of the robot joints and the pendulum angles. The accuracy of the robot joint calibration is critical for the calibration of the rotary encoders of the pendulum, as calibration errors in the robot joints also cause calibration errors in the two rotary encoders of the pendulum. Due to the offset in the pendulum angle measurements added by the poor calibration, the stabilizing controller finds a new working point, where this angle offset is compensated by a position offset. This results in a mean desired acceleration of the tool attachment point of zero at the shifted working point. The integrators of the stabilizing controllers manage to move the tool attachment point back to the initial position $\mathbf{r}_0$ even though disturbances act on the pendulum, as can be seen in the figure. After around 20 s, the position offset is very close to zero.

The aforementioned oscillations, which can be associated with a limit cycle, can also be seen in the figure. They are presumably caused by friction in the two pendulum bearings or

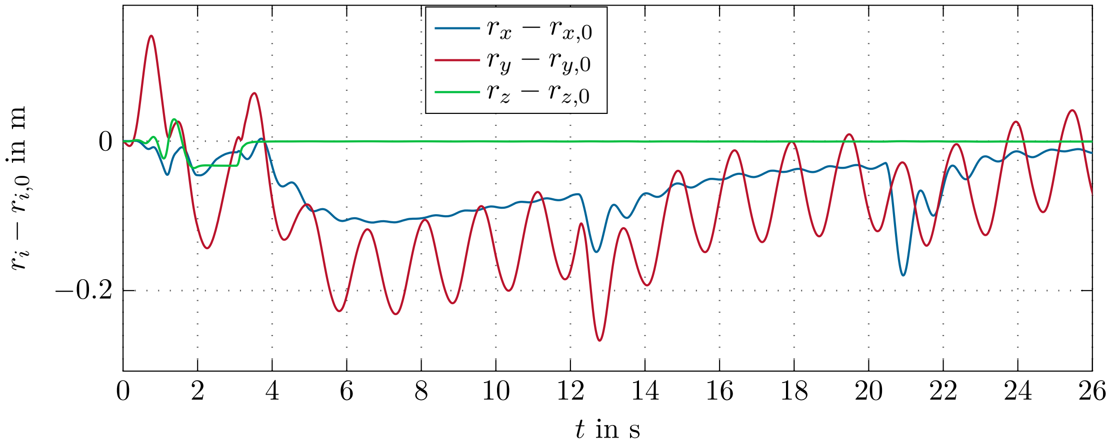

Figure 6.4: Experimental results for the position of the tool attachment point $\mathbf{r}$. The pendulum swing-up can be seen between $t = 0\,\mathrm{s}$ and $t = 2\,\mathrm{s}$. After the swing-up is finished, the swing-up controller stabilizes the pendulum until the controller switching is performed at 3.0 s and disturbances are acting on the system at $t = 12\,\mathrm{s}$ and $t = 20.5\,\mathrm{s}$.

dead time in the communication path. Multiple attempts to eliminate the oscillations were performed. The bearing arrangement of Axis 9, which is part of Axis 8, was redesigned to eliminate the backlash in the original design. Aside from this, decentralized PI velocity controllers for the joint velocities of the robot were designed and tested on the experimental setup. However, the velocity controllers showed tracking errors of at least an order of magnitude larger than the used Cartesian trajectory tracking controller from Section 4.2.2, hence the velocity controllers were not able to stabilize the pendulum. Additionally, the internal position controller of the robot controller Kuka KR C2 lr was tested, which, apart from having smaller tracking errors than the Cartesian trajectory tracking controller from Section 4.2.2, did not reduce the limit cycle of the closed-loop system. An additional insight was gained during the tuning of the Cartesian trajectory tracking controller. Using faster poles than the proposed ones, causes the closed-loop system to oscillate with a frequency of approximately 60 Hz. While the oscillations are visible in all states and inputs and are clearly audible, it eliminates the limit cycle, which strengthens the theory of friction occurring in the pendulum joints causing the limit cycle.

At around 12 s and 20.5 s, disturbance forces in the negative y_0 and negative x_0 direction, respectively, are applied to the system. The cascade controller is able to compensate for these disturbances.

The joint angles, joint velocities and the inputs are shown in Figure 6.5. The depicted graphs are scaled with (3.23). Due to the fixed orientation of the tool attachment point and the fixed value of r_z, Axis 2, 3, 5 and 7 are not actively used for stabilizing the spherical inverted pendulum, but only for satisfying these desired values. The oscillations are indeed observable in all axes, but due to the aforementioned reasons, the amplitude of the oscillation is significantly larger in Axis 1, 4 and 6. The amplitudes of the oscillation in q_8 and q_9 are 3.6° and 0.32°, respectively. While both axes have two bearings, Axis 8 is connected to the slip ring, which causes additional friction, which presumably explains the larger amplitude of the oscillations in q_8 and r_y. After the first disturbance occurs at 12 s in the negative y_0 direction, the joint angles q_8 and q_9 are deflected, as the force is not perfectly oriented in the negative y_0 direction, while after the second disturbance at 20.5 s in the negative x_0 direction, mainly q_9 is deflected. Deflections in the robot joints are mainly observed in Axis 1, 4 and 6, as they are primarily used for stabilizing the pendulum, as mentioned previously.

The error of the Cartesian controller is depicted in Figure 6.6. Oscillations with the same frequency as in the other figures from the stabilization phase can be observed. As the oscillations have a larger amplitude in r_y (see Figure 6.4), the errors are also larger in e_{r_y}. Larger deflection in e_{r_x} and e_{r_y} can be observed at 12.5 s, when a force in approximately the negative y_0 direction is applied. As the force at 20 s is well aligned in the negative x_0 direction, an increase is only seen in e_{r_x}. Deflections of the remaining four error variables can also be observed when a force is applied, which is in agreement with the simulation. Compared to the simulation including quantization and velocity filtering, the maximum position error increases from 2.0 mm to 7.2 mm in e_{r_x}, the maximum velocity error can be observed in $e_{\dot{r}_y}$ with 62.5 mm/s, which is an increase compared to the maximum of 20.2 mm/s. The largest orientation and angular velocity error occur in e_{ϵ_3} and $e_{\omega^7_{0,z_7}}$ with $6.0 \cdot 10^{-3}$ and 0.16 rad/s, respectively. In the simulation including quantization and velocity filtering, the maximum orientation and angular velocity error are $3.5 \cdot 10^{-3}$ in e_{ϵ_1} and 0.13 rad/s in $e_{\omega^7_{0,x_7}}$. The increase of the errors compared to the simulation can be attributed to the aforementioned parameter uncertainties and friction in the experimental setup.

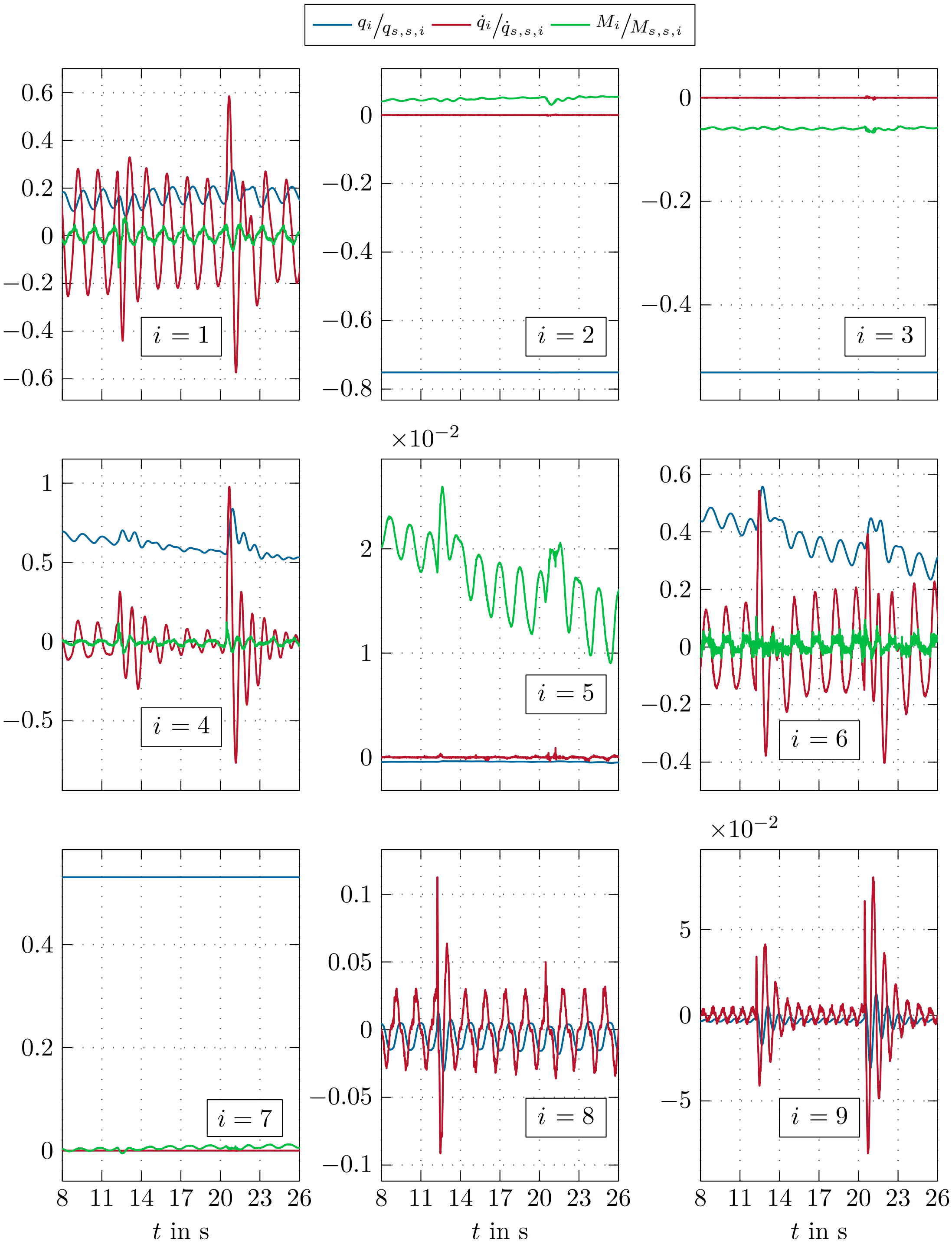

Figure 6.5: Experimental results for the states and the input during the stabilization phase using the cascade controller. The graphs are scaled with the scaling vectors from (3.23). Disturbances are acting on the system at $t = 12\,\mathrm{s}$ and $t = 20.5\,\mathrm{s}$.

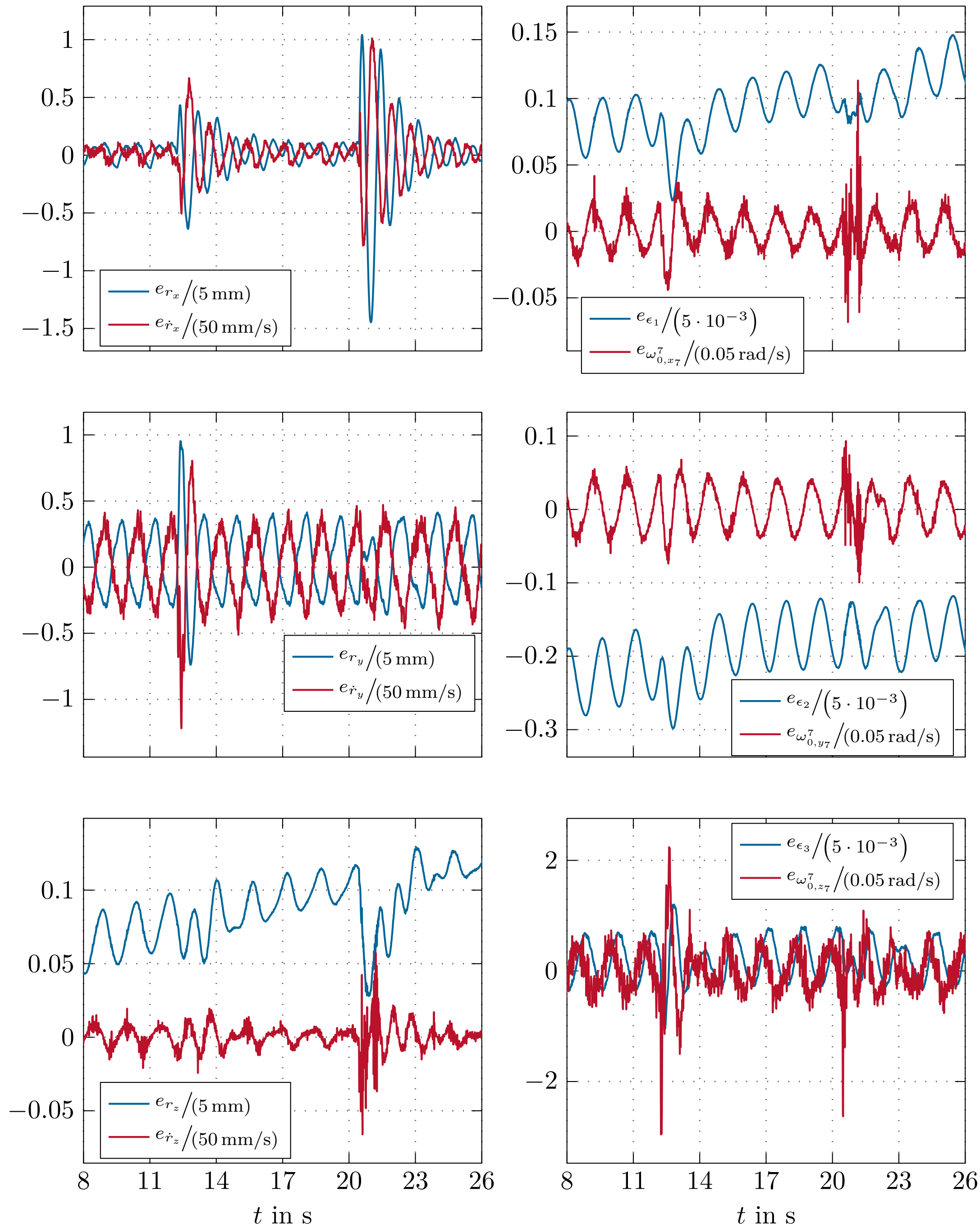

Figure 6.6: Experimental results for the errors of the Cartesian trajectory tracking controller during the stabilization phase. Disturbances are acting on the system at $t = 12\,\mathrm{s}$ and $t = 20.5\,\mathrm{s}$.

7 Summary and Conclusions

In this thesis, a swing-up trajectory and a control strategy for swing-up and stabilization of a spherical inverted pendulum with a 7-DOF robot were presented and validated in simulation and experiments. A mechatronic design for a spherical inverted pendulum is proposed, which can be mounted on a Kuka LWR IV+ and other robots with a compatible end effector.

First, the kinematics and dynamics of the spherical inverted pendulum with Cartesian input, the kinematics of the robot and the kinematics and dynamics of the complete system with torque input were derived in Chapter 2. The coordinate frames are placed to allow for an easy conversion between the kinematic quantities of the different models.

The swing-up trajectory was obtained by solving an optimal control problem. Due to its benefits in adding equality and inequality constraints and better convergence, a direct method was used to convert the dynamic optimization problem to a static one. The equality constraints were used to incorporate the model equations and the initial and final posture. The kinematic and dynamic limits of the robot were considered through inequality constraints. A start value for the angles of the robot is found, which provides sufficient reserve to the angle limits of the robot. A three-step process was utilized, whereby multiple optimization problems were solved sequentially, while using each solution as the initial guess of the next problem. This process was necessary as the optimization problem to find a swing-up trajectory for the complete system with torque input did not converge without a feasible initial guess. Therefore, the swing-up trajectory was calculated for the spherical inverted pendulum with Cartesian input, as this simpler problem also converged for trivial infeasible initial values. This first result was tuned such that the constraints of the robot were not exceeded during the execution of the trajectory. Using the inverse kinematics of the robot and the forward dynamics of the complete system with torque input, the solution was converted to a trajectory for the complete system. The resulting trajectory was used as the initial value for the optimization problem of the complete system with torque input. To decrease the sampling time, this optimization problem was repeatedly solved, while increasing the number of sampling points with each iteration and using the interpolated trajectory of the previous iteration as initial guess. This decrease of the sampling time was necessary to receive a trajectory with the same sampling time that is used by the real-time industrial hardware used in the experimental setup.

To ensure a successful swing-up, a time-variant LQR was designed in Chapter 4. This control concept can be utilized for the swing-up trajectory, around which the system is linearized, is known beforehand and the feedback matrices can be calculated off-line. For choosing the weights of the LQR, not only the errors, but also the offsets in the angle measurements have to be considered, to avoid the tool attachment point drifting off and to avoid that the transition conditions are thereby not naturally met. As soon as the pendulum was close to the upright equilibrium posture, a switching to the stabilization phase was performed. An additional transition condition concerning the position of the tool attachment point improved the smoothness of the switching. The start value for the angles of the robot in the optimization problem provides a large workspace during the stabilization phase when additional constraints for the motion of the robot are introduced and is therefore used as the operating point during the stabilization phase. During the stabilization phase, a cascade controller was used. The outer loop consists of two LQRs with integrators designed for the linearized equations of motion of the spherical inverted pendulum with Cartesian input. The integrators were added to the LQRs to compensate for the poorly calibrated robot and pendulum rotary encoders. The two controllers provide a desired trajectory for the inner loop consisting of a Cartesian trajectory tracking controller that controls the motion of the tool attachment point of the robot. As the robot is redundant, a null-space controller was additionally designed.

The swing-up and stabilization phase were simulated in Chapter 5 and the influence of disturbances, non-measurable states and poorly calibrated joint angles was examined. The closed-loop system was stable in the simulation, even with the aforementioned non-ideal influences.

The results of the experiments were presented in Chapter 6. Successful swing-up and stabilization was performed with the experimental setup, which includes the custom-built pendulum and the robot Kuka LWR IV+. To eliminate the backlash introduced by the single bearing arrangement of the original design, the pendulum design was reworked once. Additional controller concepts were tested to eliminate the limit cycle, which is observable in the experimental results. A decentralized PI velocity controller for the joint velocities of the robot was tested, whereas the Cartesian trajectory tracking controller showed better performance. Neither the PI velocity controller, nor the internal position controller of the robot controller, which was also tested, were able to eliminate the limit cycle. It was possible to isolate the root cause of the limit cycle to be the friction introduced by the bearings and the slip ring. The proposed controller structure turns out to be robust to model and parameter uncertainties, disturbances and poorly calibrated joint angles.

This thesis presents the swing-up and stabilization of a spherical inverted pendulum on a 7-DOF robot. For future work, the optimal control problem can be extended to incorporate time optimality, which would reduce the swing-up time. Friction can be added to the model and the friction parameters could be identified, e. g., with a nonlinear observer as in [58], which has the potential to eliminate the limit cycle occurring during the experiments. Further ideas to extend the control concept could be an implementation of a trajectory tracking controller for the base point of the spherical inverted pendulum.

A System Parameters

The kinematic and dynamic limits of the robot are shown in Table A.1. The parameters of the system are listed in Table A.2. The geometric parameters are obtained from [47], the remaining robot parameters are identified in [59, 60]. The masses of the inverted pendulum are measured values from the experimental setup. The inertias are extracted from Solid Edge, after adjusting the masses to the measured values.

	Joint	Value
angle limits $\mathbf{q}_{l,r}$	1	170°
	2	120°
	3	170°
	4	120°
	5	170°
	6	120°
	7	170°
joint velocity limits $\dot{\mathbf{q}}_{l,r}$	1	112.5 °/s
	2	112.5 °/s
	3	112.5 °/s
	4	112.5 °/s
	5	180 °/s
	6	112.5 °/s
	7	112.5 °/s
torque limits $\mathbf{M}_{l,r}$	1	200 N m
	2	200 N m
	3	100 N m
	4	100 N m
	5	100 N m
	6	30 N m
	7	30 N m

Table A.1: Limits of the Kuka LWR IV+ [47].

Parameters	Value	Unit
d_1	0.4	m
d_2	0.39	m
d_3	0.078	m
d_4	0.157	m
d_5	0.285	m
m_7	1.560 194	kg
m_8	0.222 714	kg
m_9	0.186 075	kg
$c_{7,x}$	0.003 744 162	m
$c_{7,y}$	0.010 787 826	m
$c_{7,z}$	0.100 607 856	m
$c_{8,x}$	0.000 031 095	m
$c_{8,y}$	−0.047 241 586	m
$c_{8,z}$	0.000 865 948	m
$c_{9,x}$	0.000 00	m
$c_{9,y}$	−0.001 036 458	m
$c_{9,z}$	0.049 164 415	m
$I_{7,xx}$	$3013.860\,599 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{7,yy}$	$2032.125\,341 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{7,zz}$	$2118.801\,210 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{7,xy}$	$294.699\,322 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{7,xz}$	$-174.961\,59 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{7,yz}$	$-272.694\,32 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{8,xx}$	$353.874\,015 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{8,yy}$	$27.959\,845 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{8,zz}$	$354.071\,411 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{8,xy}$	$0.435\,201 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{8,xz}$	$0.230\,209 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{8,yz}$	$8.203\,600 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{9,xx}$	$1288.884\,845 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{9,yy}$	$1216.719\,951 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{9,zz}$	$80.024\,782 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{9,xy}$	$0.000\,000 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{9,xz}$	$0.000\,000 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
$I_{9,yz}$	$9.476\,641 \cdot 10^{-6}$	$\mathrm{kg\,m^2}$
g	9.81	$\mathrm{m/s^2}$

Table A.2: Parameters of the system. All parameters with the index 7 include the mass and inertia of link 7 of the robot and of the mounting enclosure of the spherical inverted pendulum. The remaining robot parameters are used from [59, 60].

B Technical Drawings

The technical drawings were created in Solid Edge on the basis of [41, 42, 54, 61–68]. The parts list is given in Table B.1. The technical drawings are shown in Figure B.1 to Figure B.10. All distances are given in mm.

Name	Vendor	Material	quantity
6001-2Z	SKF		3
608-2Z	SKF		1
Axis 8		stainless steel	1
Axis 9		stainless steel	1
BA 080 AB 01 A A 00	RLS		1
BA 150 AB 02 A A 00	RLS		1
Bearing enclosure 8		aluminium	1
Bearing enclosure 9		aluminium	1
BR10 SC B 14B 12 C D 00	RLS		1
BR10 SC B 14B 16 C D 00	RLS		1
Counterweight		stainless steel	1
DIN 912 M2×20			6
DIN 912 M3×10			3
DIN 912 M4×16			2
DIN 912 M4×20			1
DIN 912 M6×10			7
DIN 912 M6×20			9
DIN 913 M3×8			1
DIN 913 M5×5			1
DIN 471 8×0.8			1
DIN 471 12×1			1
DIN EN 50022 35×80		aluminium	1
Distance shell $\varnothing 3.0 \times 0.4 \times 8$		aluminium	6
Distance ring $\varnothing 15 \times 1.5 \times 5$		aluminium	1
EK1100	Beckhoff		1
EL5002	Beckhoff		1
End cover		aluminium	3
Flange		aluminium	1
Frame		aluminium	1
KL9505	Beckhoff		1
KMK 1	SKF		1
ISO 10642 M3×10			9
ISO 10642 M5×10			2
ISO 4032 M6			1
Rod		aluminium	1
Slip ring mount		aluminium	1
SRA-73540-6A	Moog Components Group		1

Table B.1: Parts list of the spherical inverted pendulum.

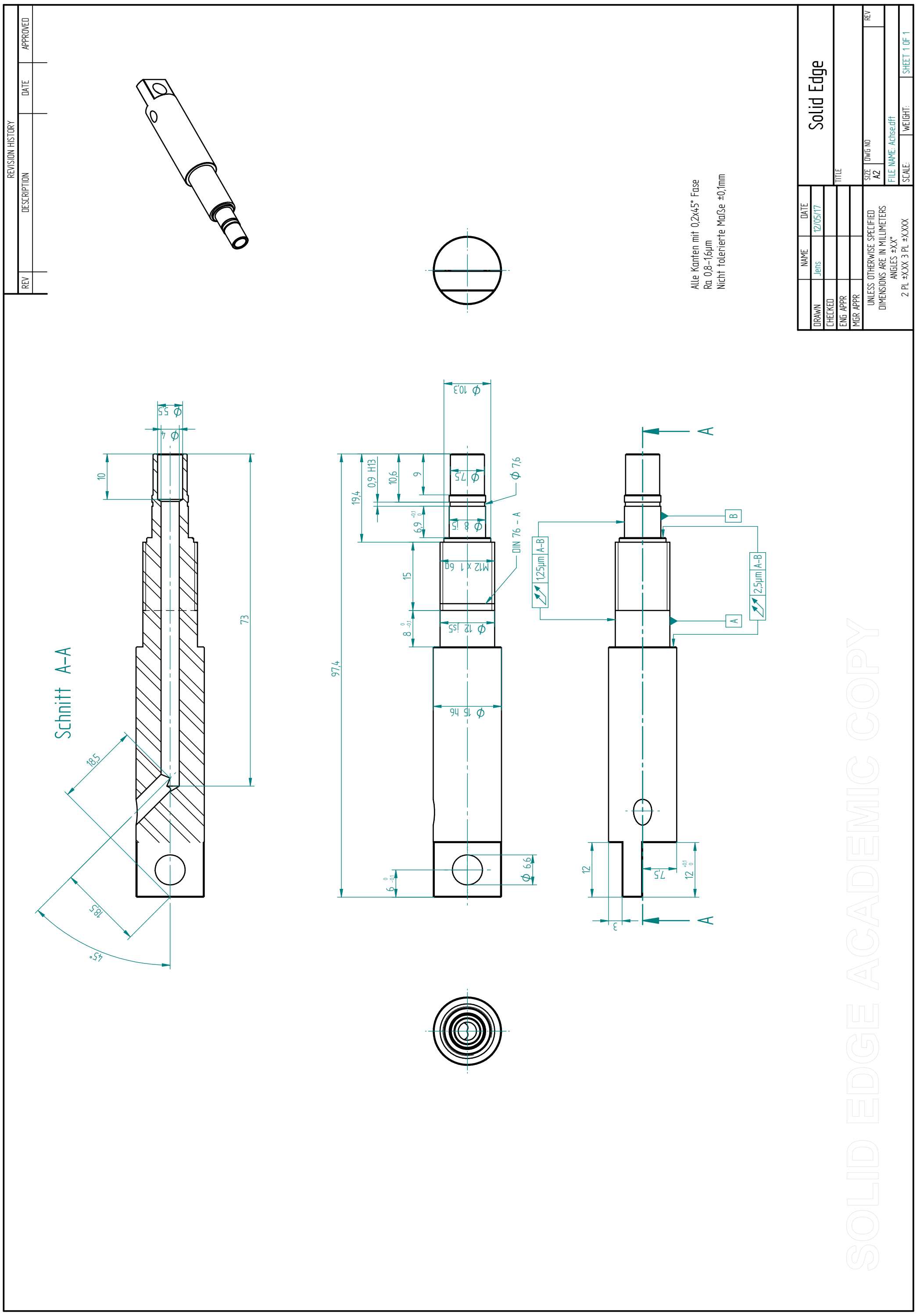

Figure B.1: Axis 8.

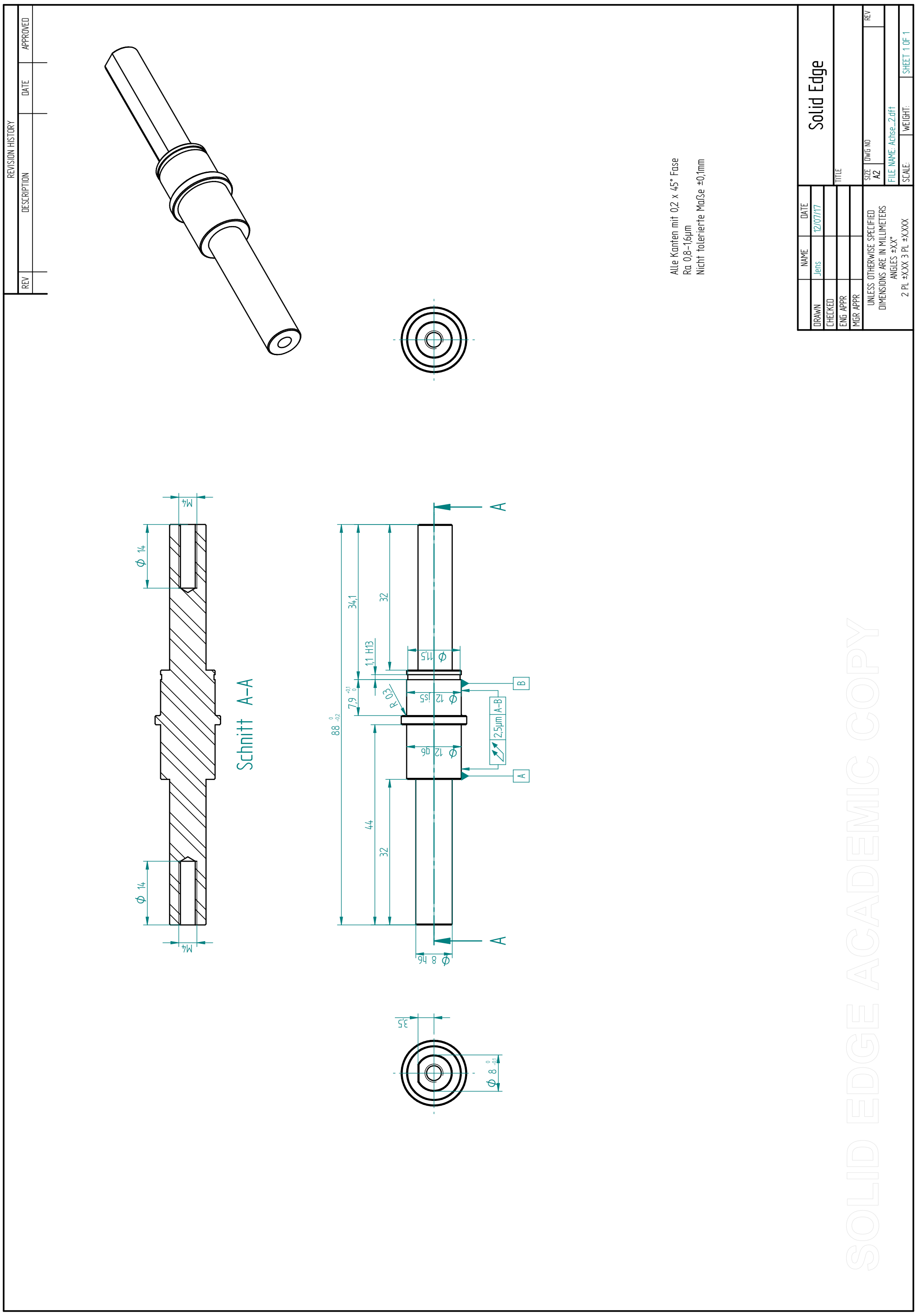

Figure B.2: Axis 9.

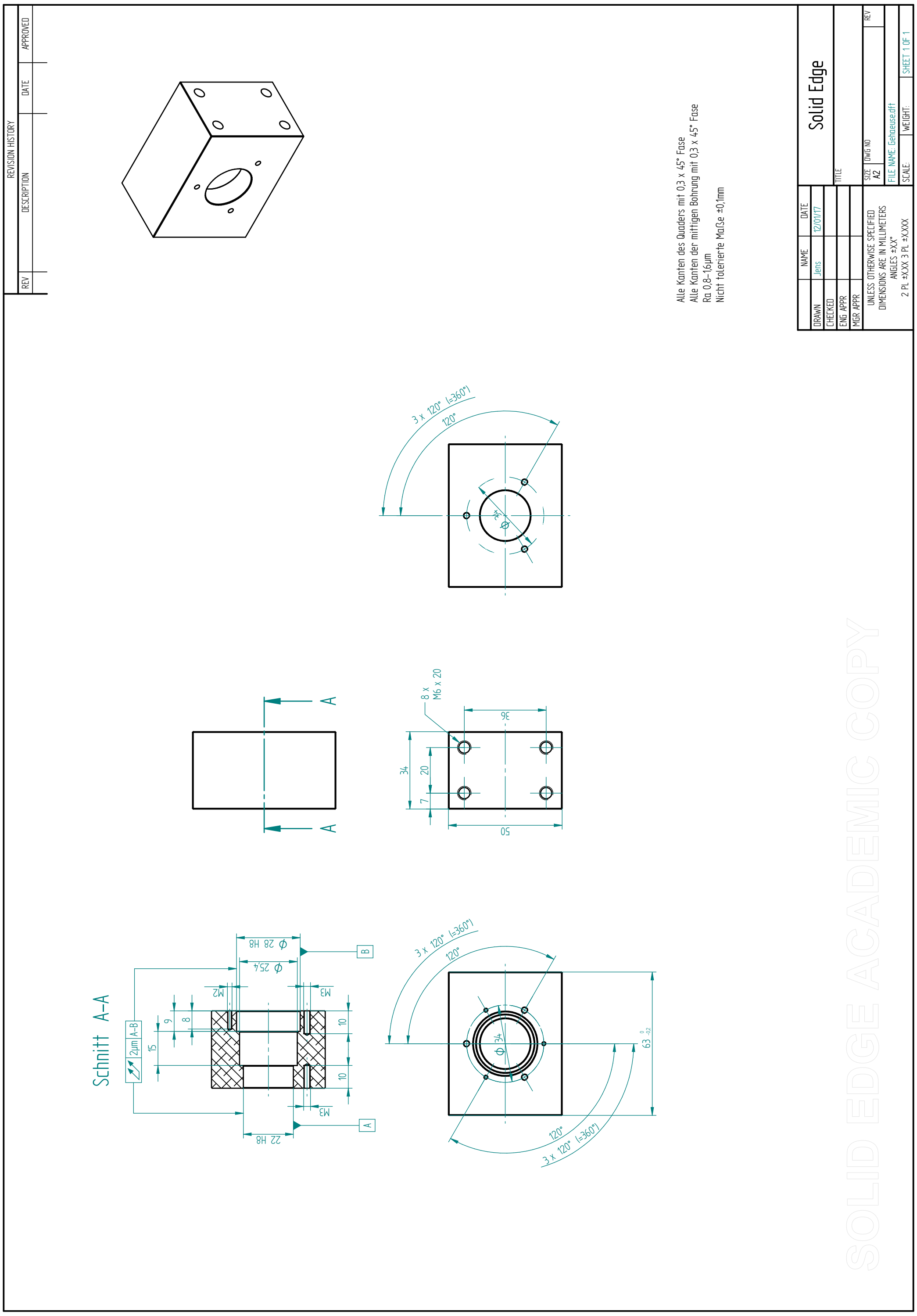

Figure B.3: Bearing enclosure 8.

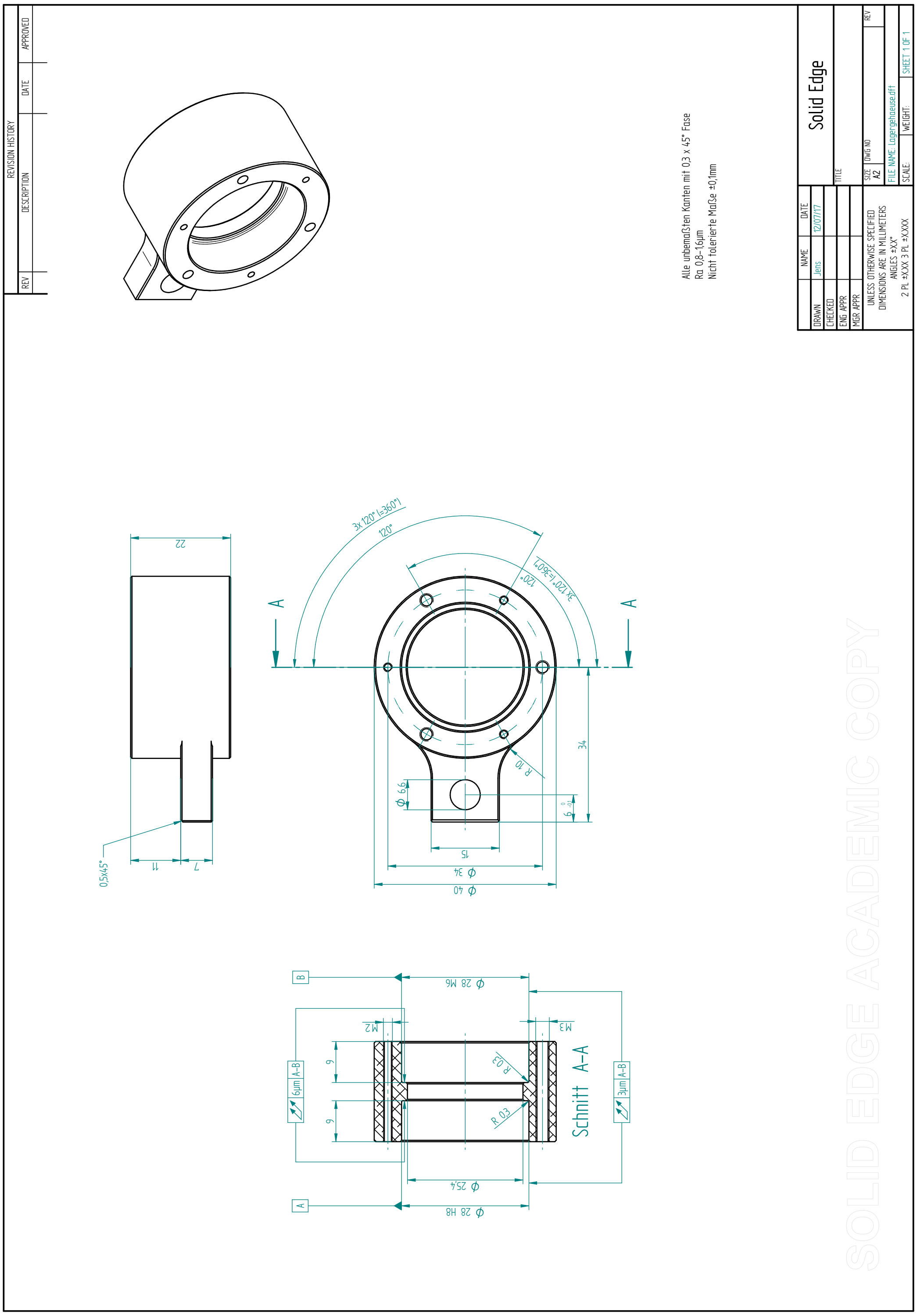

Figure B.4: Bearing enclosure 9.

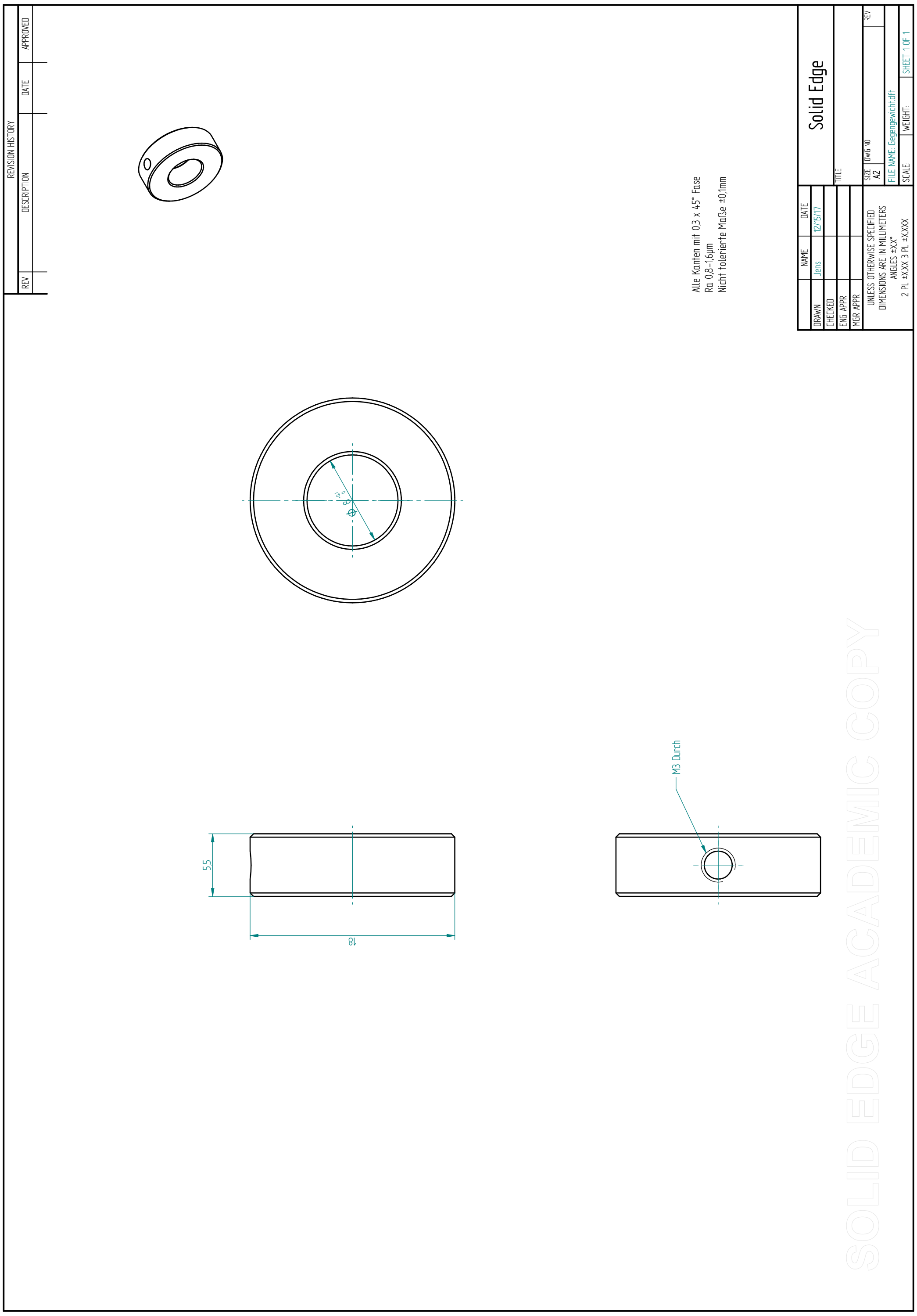

Figure B.5: Counterweight.

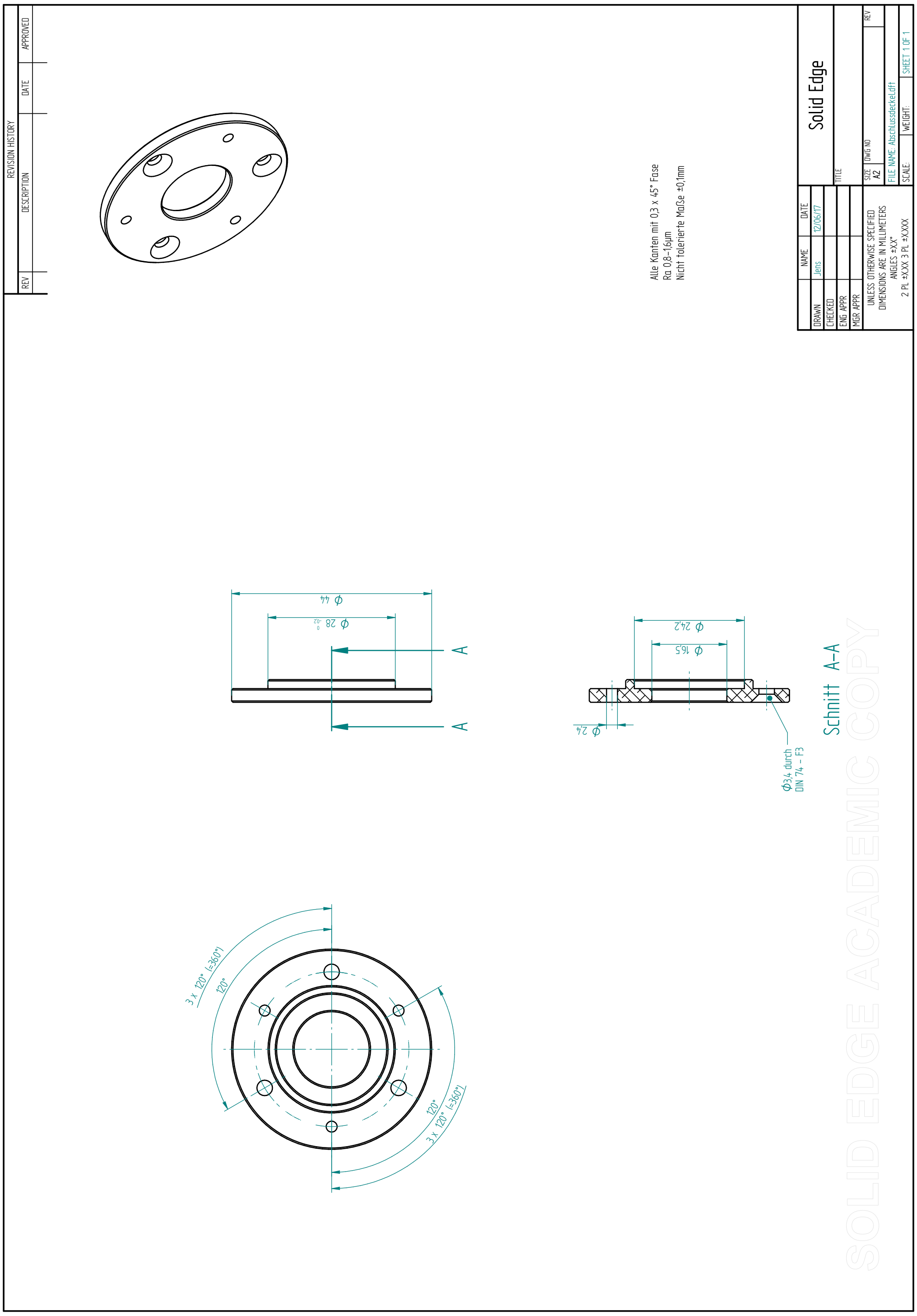

Figure B.6: End cover.

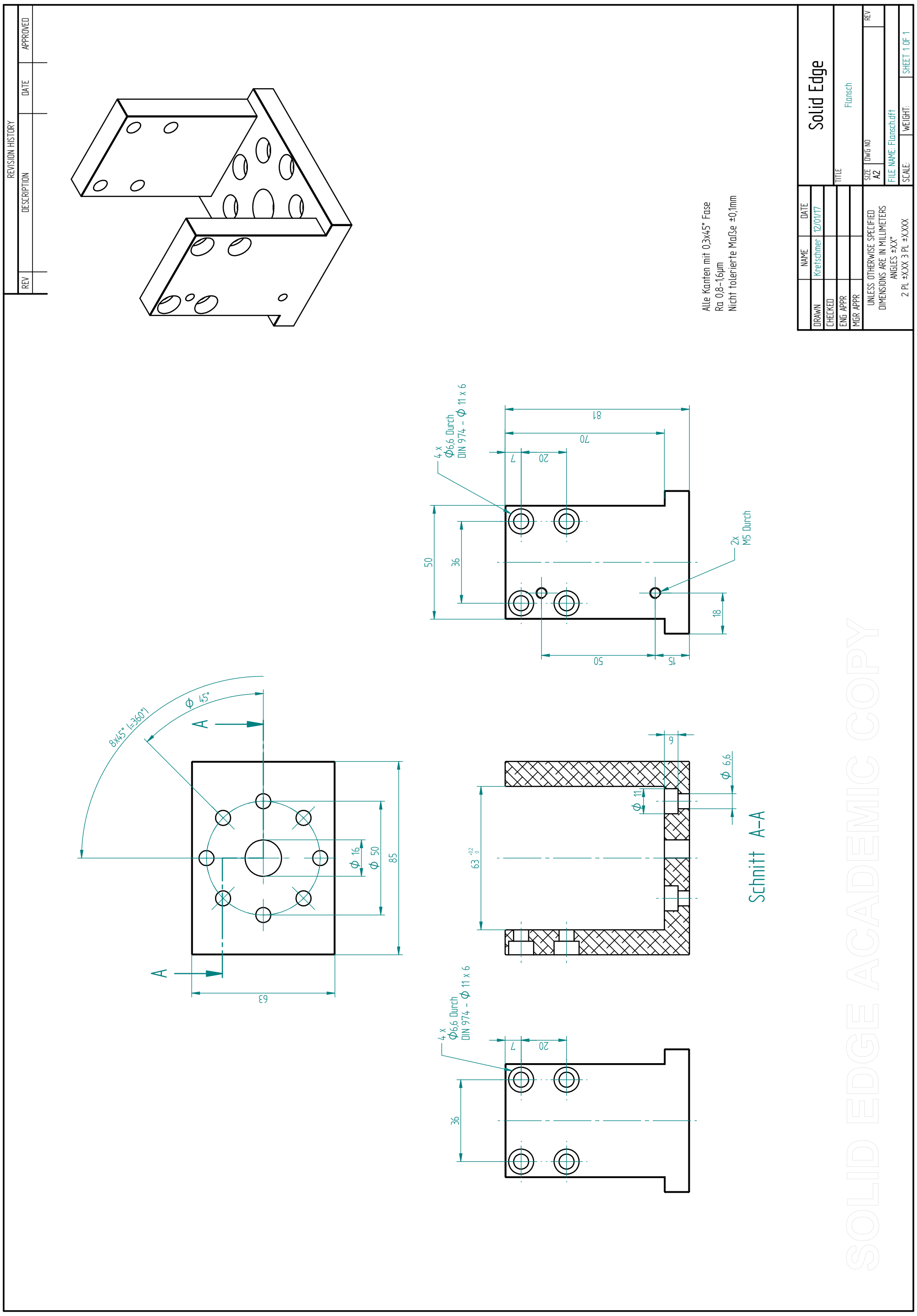

Figure B.7: Flange.

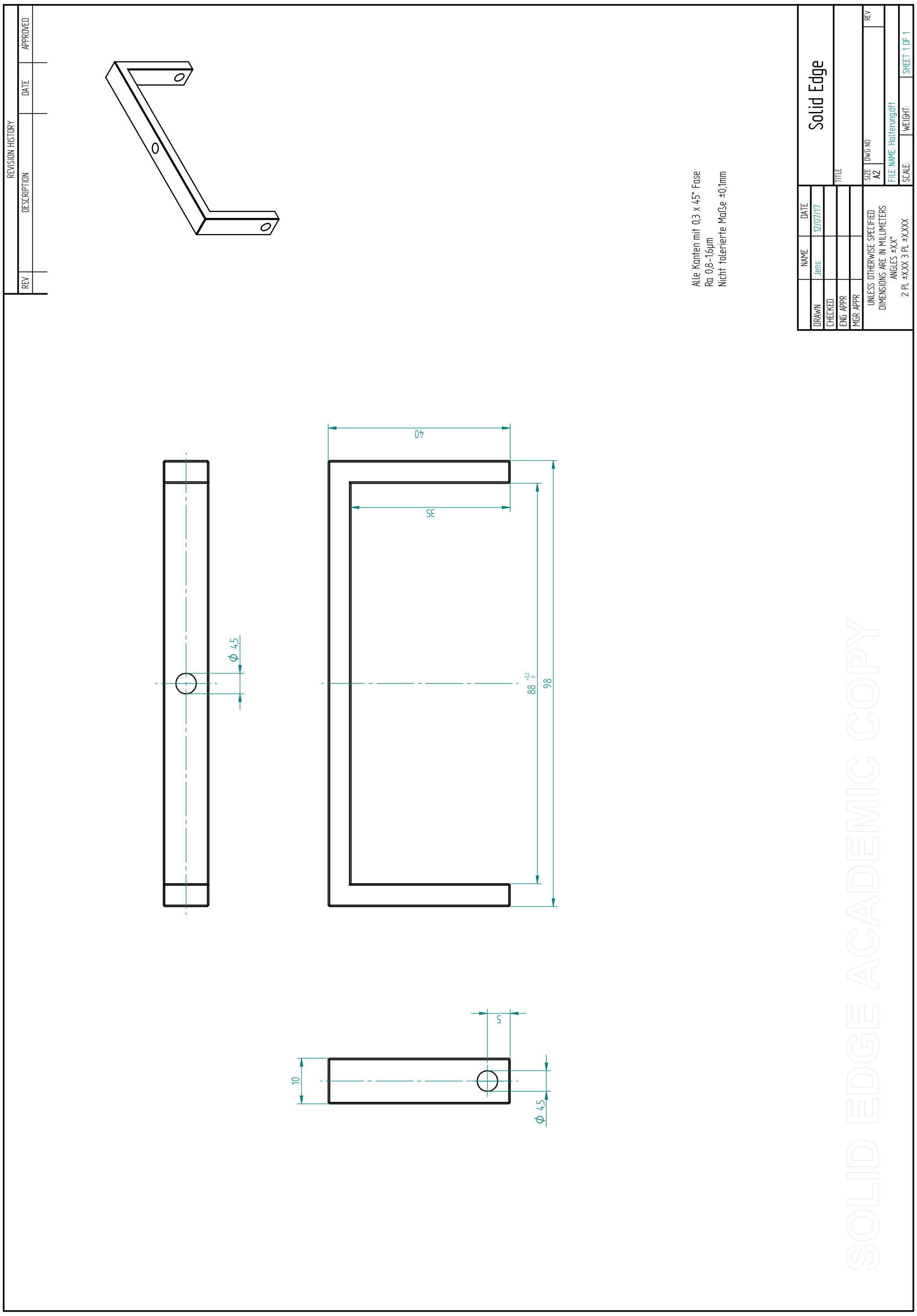

Figure B.8: Frame.

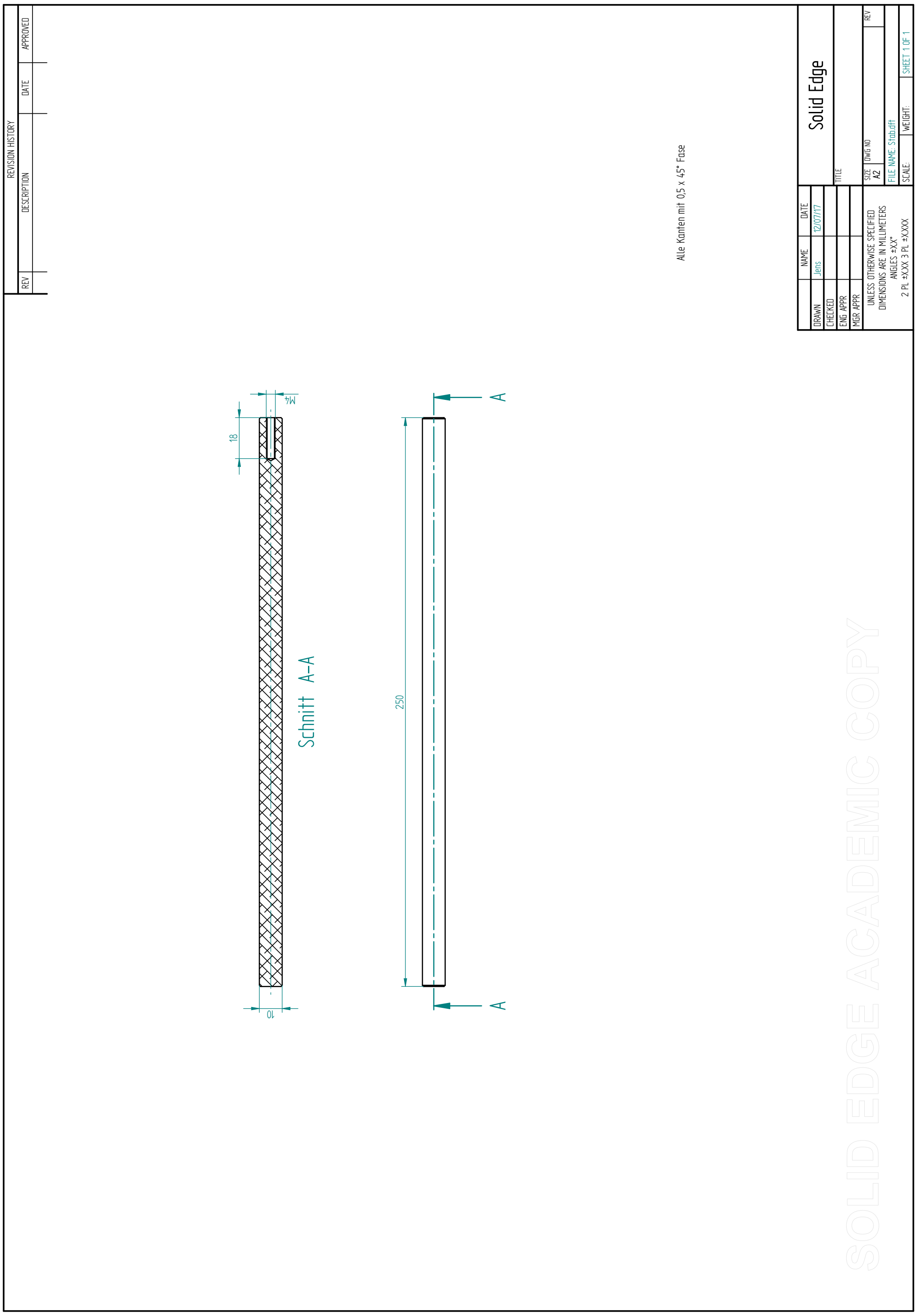

Figure B.9: Rod.

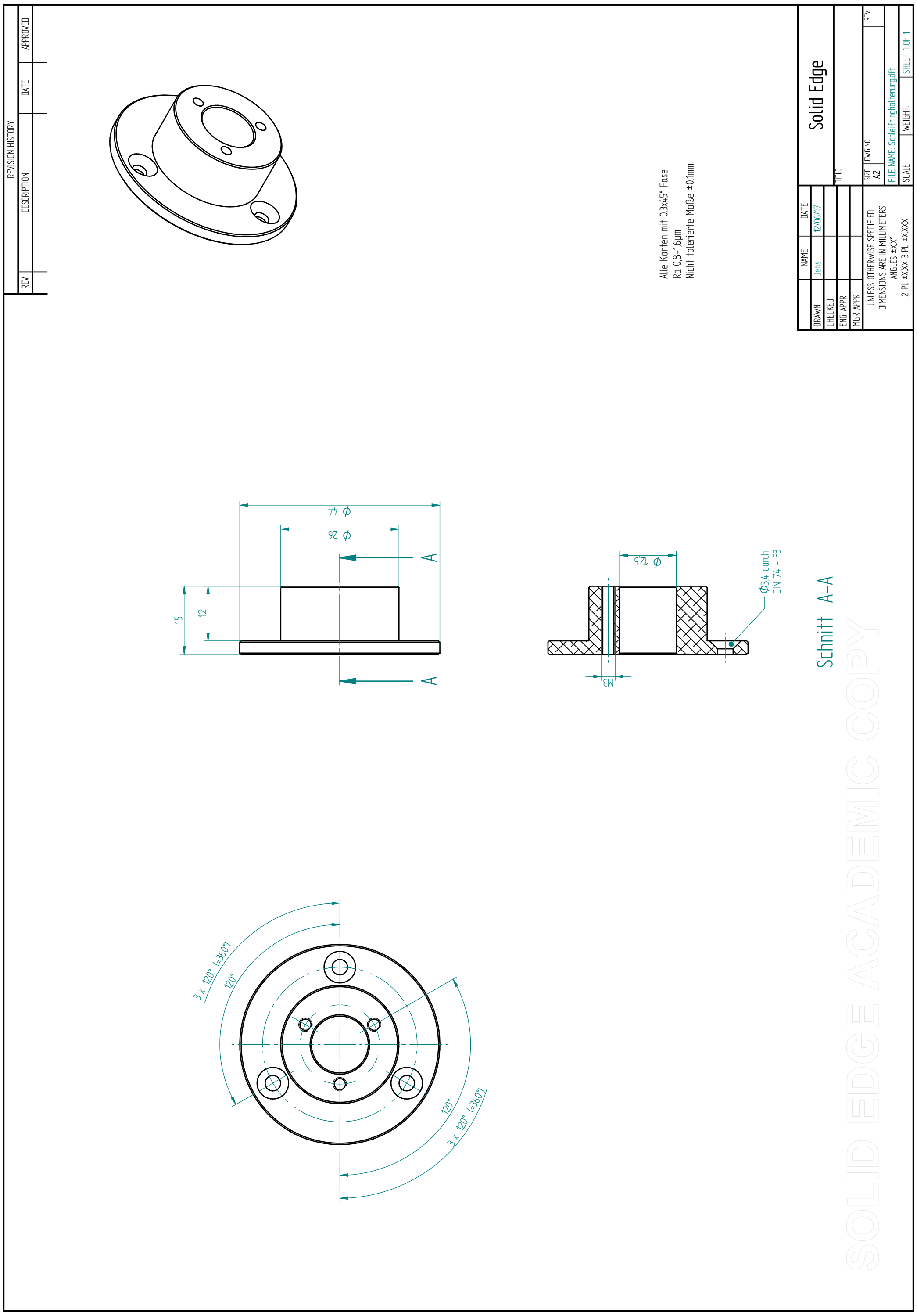

Figure B.10: Slip ring mount.

Bibliography

[1] N. J. Mathew, K. K. Rao, and N. Sivakumaran, "Swing Up and Stabilization Control of a Rotary Inverted Pendulum," *Proceedings of the 10th IFAC International Symposium on Dynamics and Control of Process Systems*, vol. 46, no. 32, pp. 654–659, Dec. 2013.

[2] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River: Prentice Hall, 2010.

[3] K. H. Lundberg and T. W. Barton, "History of Inverted-Pendulum Systems," *Proceedings of the 8th IFAC Symposium on Advances in Control Education*, vol. 42, no. 24, pp. 131–135, 2010.

[4] J. Ahn and S. Jung, "Swing-up fuzzy control of an inverted pendulum system for control education with an experimental kit," *Proceedings of the 2008 International Conference on Control, Automation and Systems*, pp. 1709–1714, Oct. 2008.

[5] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4489–4496, Oct. 2010.

[6] K. J. Åström and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, no. 2, pp. 287–295, Feb. 2000.

[7] K. Yoshida, "Swing-up control of an inverted pendulum by energy-based methods," *Proceedings of the American Control Conference*, vol. 6, pp. 4045–4047, Jun. 1999.

[8] K. Furuta, T. Okutani, and H. Sone, "Computer control of a double inverted pendulum," *Computers & Electrical Engineering*, vol. 5, no. 1, pp. 67–84, Mar. 1978.

[9] S. A. Bortoff, "Robust Swing-Up Control for a Rotational Double Pendulum," *Proceedings of the 13th World Congress of IFAC*, vol. 29, no. 1, pp. 2810–2815, Jun. 1996.

[10] M.-C. Tsai and B.-H. Shen, "Synchronization control of parallel dual inverted pendulums driven by linear servomotors," *IET Control Theory Application*, vol. 1, no. 1, Jan. 2007.

[11] X. Albouy and L. Praly, "On The Use Of Dynamic Invariants And Forwarding For Swinging Up A Spherical Inverted Pendulum," *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 5, pp. 1667–1672, Dec. 2000.

[12] R. Yang, Y.-Y. Kuen, and Z. Li, "Stabilization of a 2-DOF spherical pendulum on X-Y table," *Proceedings of the 2000 IEEE International Conference on Control Applications*, pp. 724–729, Sep. 2000.

[13] J. Shen, A. K. Sanyal, N. A. Chaturvedi, D. Bernstein, and H. McClamroch, "Dynamics and control of a 3D pendulum," *Proceedings of the 43rd IEEE Conference on Decision and Control*, vol. 1, pp. 323–328, Dec. 2004.

[14] K. Furuta and T. O. and N. Ono, "Attitude control of a triple inverted pendulum," *International Journal of Control*, vol. 39, no. 6, pp. 1351–1365, Jun. 1984.

[15] H. Li, M. Zhihong, and W. Jiayin, "Variable universe adaptive fuzzy control on the quadruple inverted pendulum," *Science in China Series E: Technological Sciences*, vol. 45, no. 2, pp. 213–224, Apr. 2002.

[16] C. C. Chung and J. Hauser, "Nonlinear control of a swinging pendulum," *Automatica*, vol. 31, no. 6, pp. 851–862, Jun. 1995.

[17] K. J. Åström, J. Aracil, and F. Gordillo, "A family of smooth controllers for swinging up a pendulum," *Automatica*, vol. 44, no. 7, pp. 1841–1848, Jul. 2008.

[18] O. Boubaker, "The Inverted Pendulum Benchmark in Nonlinear Control Theory: A Survey," *International Journal of Advanced Robotic Systems*, vol. 10, no. 5, p. 233, Jan. 2013.

[19] M. F. Hamza, H. J. Yap, I. A. Choudhury, A. I. Isa, A. Y. Zimit, and T. Kumbasar, "Current development on using Rotary Inverted Pendulum as a benchmark for testing linear and nonlinear control algorithms," *Mechanical Systems and Signal Processing*, vol. 116, pp. 347–369, Feb. 2019.

[20] J. Yi, N. Yubazaki, and K. Hirota, "Upswing and stabilization control of inverted pendulum system based on the SIRMs dynamically connected fuzzy inference model," *Fuzzy Sets and Systems*, vol. 122, no. 1, pp. 139–152, Aug. 2001.

[21] Y. Xu, M. Iwase, and K. Furuta, "Time Optimal Swing-Up Control of Single Pendulum," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 3, pp. 518–527, Sep. 2001.

[22] K. Y. Chou and Y. P. Chen, "Energy based swing-up controller design using phase plane method for rotary inverted pendulum," *Proceedings of the 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 975–979, Dec. 2014.

[23] M. F. Hamza, H. J. Yap, and I. A. Choudhury, "Genetic Algorithm and Particle Swarm Optimization Based Cascade Interval Type 2 Fuzzy PD Controller for Rotary Inverted Pendulum System," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–15, 2015.

[24] S. Tanaka, X. Xin, and T. Yamasaki, "Experimental verification of energy-based swing-up control for a rotational pendulum," *Proceedings of the SICE Annual Conference*, vol. 51, pp. 534–539, Aug. 2012.

[25] Y.-F. Chen and A.-C. Huang, "Adaptive control of rotary inverted pendulum system with time-varying uncertainties," *Nonlinear Dynamics*, vol. 76, no. 1, pp. 95–102, Nov. 2013.

[26] M. A. Fairus, Z. Mohamed, M. N. Ahmad, and W. S. Loi, "LMI-based Multiobjective Integral Sliding Mode Control for Rotary Inverted Pendulum System Under Load Variations," *Jurnal Teknologi*, vol. 73, no. 6, pp. 125–137, Apr. 2015.

[27] Q. V. Dang, B. Allouche, L. Vermeiren, A. Dequidt, and M. Dambrine, "Design and implementation of a robust fuzzy controller for a rotary inverted pendulum using the takagi-sugeno descriptor representation," *Proceedings of the IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, vol. 4, pp. 176–181, Dec. 2014.

[28] V. Nath and R. Mitra, "Swing-up and control of Rotary Inverted Pendulum using poleplacement with integrator," *Proceedings of the Recent Advances in Engineering and Computational Sciences (RAECS)*, vol. 1, pp. 1–5, Mar. 2014.

[29] I. Hassanzadeh and S. Mobayen, "Controller Design for Rotary Inverted Pendulum System Using Evolutionary Algorithms," *Mathematical Problems in Engineering*, vol. 2011, pp. 1–17, 2011.

[30] B. Sprenger, L. Kucera, and S. Mourad, "Balancing of an Inverted Pendulum with a SCARA Robot," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, vol. 3, no. 2, pp. 91–97, Jun. 1998.

[31] C. Y. Chung, J. W. Lee, S. M. Lee, and B. H. Lee, "Balancing of an inverted pendulum with a redundant direct-drive robot," *Proceedings of the International Conference on Robotics and Automation (ICRA)*, vol. 4, pp. 3952–3957, Apr. 2000.

[32] C. Y. Chung, "Balancing of an Inverted Pendulum With a Kinematically Redundant Robot," *International Journal of General Systems*, vol. 31, no. 1, pp. 1–15, Feb. 2002.

[33] M. Resch and H. Ollmann, *Stabilisierung eines inversen Raumpendels mit Hilfe eines 5 DOF-roboters*, Projektseminar, Johannes Kepler Universität Linz, Jun. 2004.

[34] M.-G. Yoon, "Dynamics and Stabilization of a Spherical Inverted Pendulum on a Wheeled Cart," *International Journal of Control, Automation and Systems*, vol. 8, no. 6, pp. 1271–1279, Dec. 2010.

[35] I. Soto and R. Campa, "Modelling and Control of a Spherical Inverted Pendulum on a Five-Bar Mechanism," *International Journal of Advanced Robotic Systems*, vol. 12, no. 7, pp. 95–110, Jul. 2015.

[36] G. Liu, D. Nešić, and I. Mareels, "Modelling And Stabilisation Of A Spherical Inverted Pendulum," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 1130–1135, Jul. 2005.

[37] T. D. Viet, P. T. Doan, H. Giang, H. K. Kim, and S. B. Kim, "Control of a 2-DOF omnidirectional mobile inverted pendulum," *Journal of Mechanical Science and Technology*, vol. 26, no. 9, pp. 2921–2928, Sep. 2012.

[38] H. Ludvigsen, A. Shiriaev, and O. Egeland, "Stabilization of Stable Manifold of Upright Position of the Spherical Pendulum," *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 22, no. 1, pp. 3–14, Jan. 2001.

[39] A. S. Shiriaev, H. Ludvigsen, and O. Egeland, "Swinging up the spherical pendulum via stabilization of its first integrals," *Automatica*, vol. 40, no. 1, pp. 73–85, Jan. 2004.

[40] J. Kretschmer, *Modellierung und Regelung eines Knickarmroboters mit sphärischem Pendel*, Automation and Control Institute, TU Wien, Bachelor thesis, Jan. 2014.

[41] *Wälzlager*, PUB BU/P1 10000/2 DE, SKF, Jan. 2014 (accessed 2018-10-22). [Online]. Available: `http://www.skf.com/binary/78-121486/Walzlager---10000_2-DE.pdf`.

[42] D. Muhs, H. Wittel, D. Jannasch, and J. Voßiek, *Roloff/Matek Maschinenelemente*, 17th ed. Wiesbaden: Vieweg, 2005.

[43] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. London: Springer-Verlag, Nov. 2008.

[44] B. C. Chachuat, *Nonlinear and Dynamic Optimization: From Theory to Practice*, 2007 (accessed 2018-11-24). [Online]. Available: `http://infoscience.epfl.ch/record/111939`.

[45] M. Papageorgiou, M. Leibold, and M. Buss, *Optimierung: Statische, dynamische, stochastische Verfahren für die Anwendung*, 4th ed. Berlin Heidelberg: Springer, 2015.

[46] *MATLAB 2017b Documentation*, The MathWorks, Inc., 2017.

[47] *Leichtbauroboter 4+ Betriebsanleitung*, BA LBR 4+ V2 de, KUKA Roboter GmbH, Jul. 2010.

[48] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Boston, MA, USA: Addison–Wesley, 1998.

[49] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," *Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, no. 12, pp. 868–871, Dec. 1977.

[50] M. Z. Huang and H. Varma, "Optimal Rate Allocation in Kinematically Redundant Manipulators - The Dual Projection Method," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 702–707, Apr. 1991.

[51] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Berlin Heidelberg: Springer-Verlag, 2008.

[52] *KR C2 lr Betriebsanleitung*, BA KR C2 lr V4 de, KUKA Roboter GmbH, Jan. 2010.

[53] *KUKA.FastResearchInterface 1.0*, KUKA.FRI 1.0 V2 de, KUKA Roboter GmbH, Sep. 2011.

[54] *Orbis true absolute rotary encoder*, BRD01_01, Issue 1, RLS, Oct. 2017.

[55] *EL500x - SSI-Geber-Interface*, Version 2.1, Beckhoff Automation GmbH & Co. KG, May 2017.

[56] *EK110x, EK15xx - EtherCAT-Buskoppler*, Version 3.2, Beckhoff Automation GmbH & Co. KG, May 2017.

[57] *EL95xx - EtherCAT Netzteilklemmen*, Version 3.6, Beckhoff Automation GmbH & Co. KG, Oct. 2017.

[58] M. Eom and D. Chwa, “Robust Swing-Up and Balancing Control Using a Nonlinear Disturbance Observer for the Pendubot System With Dynamic Friction,” *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 331–343, Apr. 2015.

[59] C. Gaz, F. Flacco, and A. D. Luca, “Identifying the Dynamic Model Used by the KUKA LWR: A Reverse Engineering Approach,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1386–1392, May 2014.

[60] ——, “Extracting Feasible Robot Parameters from Dynamic Coefficients using Nonlinear Optimization Methods,” *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2075–2081, May 2016.

[61] *SRA-73540: Compact slip ring capsule*, Moog Components Group, Jun. 2016.

[62] “ISO965-3: Metrisches ISO-Gewinde allgemeiner Anwendung - Toleranzen - Teil 3: Grenzabmaße für Konstruktionsgewinde,” DIN Deutsches Institut für Normung e.V., Standard, Nov. 1999.

[63] “DIN EN 20273: Durchgangslöcher für Schrauben,” DIN Deutsches Institut für Normung e.V., Standard, Apr. 1992.

[64] “DIN 471: Sicherungsringe (Halteringe) für Wellen - Regelausführung und schwere Ausführung,” DIN Deutsches Institut für Normung e.V., Standard, Feb. 2011.

[65] “DIN 974-1: Senkdurchmesser - Konstruktionsmaße - Teil 1: Schrauben mit Zylinderkopf,” DIN Deutsches Institut für Normung e.V., Standard, Feb. 2008.

[66] “DIN 974-2: Senkdurchmesser für Sechskantschrauben und Sechskantmuttern - Konstruktionsmaße,” DIN Deutsches Institut für Normung e.V., Standard, May 1991.

[67] “DIN 74: Senkungen für Senkschrauben ausgenommen Senkschrauben mit Köpfen nach DIN EN 27721,” DIN Deutsches Institut für Normung e.V., Standard, Apr. 2003.

[68] “DIN EN ISO 4026: Gewindestifte mit Innensechskant mit Kegelstumpf (ISO 4026:2003),” DIN Deutsches Institut für Normung e.V., Standard, May 2004.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit gemäß dem Code of Conduct – Regeln zur Sicherung guter wissenschaftlicher Praxis (in der aktuellen Fassung des jeweiligen Mitteilungsblattes der TU Wien), insbesondere ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel, angefertigt wurde. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In– noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, im Jänner 2019

Jens Kretschmer