TECHNISCHE UNIVERSITÄT WIEN

ACIN
AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

Robotic Rope Winding

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. M. Schwegel

submitted at the

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

by
Ali Haddadi
Matriculation number 01127376

Vienna, November 2018

Complex Dynamical Systems Group
A-1040 Wien, Gußhausstr. 27–29, Internet: http://www.acin.tuwien.ac.at

Acknowledgments

I would like to express my gratitude to Univ.-Prof. Dr. techn. Andreas Kugi, not only for the opportunity to conduct my thesis on this interesting topic, but also for his brilliant lectures, which sparked my interest in control engineering. I am deeply grateful to my supervisor Dipl.-Ing. Michael Schwegel for his great help with words and deeds throughout my work. I also want to thank Assoc. Prof. Dr.-Ing. Wolfgang Kemmetmüller for his constructive suggestions.

Special thanks go to Dipl.-Ing. Christian Hartl-Nesic, who never hesitated to help me with his theoretical and practical expertise. Furthermore, I would like to thank Dipl.-Ing. Bernhard Bischof for the interesting and helpful discussions. I thank the technicians at ACIN, Rainer Danzinger and Ing. Nikolaus Hofbauer, for their help with the construction of the experimental setup. I would like to thank my fellow students Jens Kretschmer and Thomas Weingartshofer. It was great to share experiences on working on the robot.

Last but certainly not least, I want to thank my family for the endless love and support. Thanks for always being there for me.

Vienna, November 2018

Abstract

The goal of this thesis is to use an industrial robot for winding a rope around objects, such as rods. In addition, the rope tension is to be controlled during this winding process.

For a model-based control approach, the dynamic model of the robot is derived. The desired path for winding around the rods is specified using a teach-in procedure. A path following controller based on transverse feedback linearization is employed to make the end-effector of the robot follow the desired path.

The end-effector consists of a rope spool and a motor, which is used to control the tension of the rope. The required end-effector orientation during the winding process is derived from the path, i. e. only a position teach-in is required. Using the Frenet-Serret frame and the parallel transport frame, end-effector orientations are derived that are adapted to the rope winding application.

The proposed method is examined in both simulations and experiments. The results validate the feasibility of the method for winding a rope using a robot without *a priori* information on the required end-effector orientation.

Kurzzusammenfassung

In dieser Arbeit wird ein Industrieroboter eingesetzt, um ein Seil um Objekte, wie z. B. Stangen, zu wickeln. Dabei soll zusätzlich die Seilspannung während des Wickelvorganges geregelt werden.

Für eine modellbasierte Regelung wird zunächst das dynamische Modell des Roboters hergeleitet. Der gewünschte Pfad zum Wickeln des Seils um die Stangen wird durch einen Teach-In-Vorgang vorgegeben. Unter Verwendung eines Pfadfolgereglers, der auf der transversalen Eingangs-Zustandslinearisierung basiert, wird erreicht, dass der Roboter-Endeffektor dem vorgegebenen Pfad folgt.

Der Endeffektor besteht aus einer Seilspule und einem Motor, der zur Regelung der Seilspannung verwendet wird. Eine geeignete Orientierung des Endeffektors während des Wickelvorganges wird aus dem vorgegebenen Pfad abgeleitet, d. h. es wird lediglich ein Teach-In der Position benötigt. Es werden der Frenet-Serret-Rahmen und der Paralleltransport-Rahmen eingesetzt und daraus Endeffektor-Orientierungen abgeleitet, die an die Aufgabe des Seilwickelns angepasst sind.

Die präsentierte Methode wird sowohl in Simulationen als auch im Rahmen von Experimenten untersucht. Die Ergebnisse zeigen die Anwendbarkeit dieser Methode zum Wickeln eines Seils durch einen Roboter, ohne dass *a priori* Informationen über die benötigte Endeffektor-Orientierung vorhanden sind.

Contents

1 Introduction **1**

2 Mathematical Modeling **2**
2.1 Forward Kinematics . . . 2
2.2 Inverse Kinematics . . . 5
2.3 Manipulator Jacobian Matrix . . . 10
2.4 Dynamic Model of Rigid-Body Robots . . . 11
2.5 Dynamic Model of Robots with Flexible Joints . . . 13

3 Path Following Control **16**
3.1 Literature Review . . . 16
3.2 Problem Statement . . . 17
3.2.1 Admissible Paths . . . 18
3.2.2 Control Objectives . . . 18
3.3 Coordinate Transformation . . . 19
3.3.1 Parallel Transport Frame . . . 19
3.3.2 Orthogonal Projection . . . 20
3.3.3 Tangential Subsystem . . . 21
3.3.4 Transversal Subsystem . . . 22
3.3.5 Rotational Subsystem . . . 23
3.3.6 Diffeomorphism . . . 23
3.4 Feedback Linearization . . . 24
3.5 Path Parametrization . . . 27

4 Robotic Winding of a Rope **29**
4.1 Literature Review . . . 29
4.2 Problem Statement . . . 30
4.3 Experimental Setup . . . 30
4.4 Path Recording . . . 31
4.5 Desired Orientation of the End-effector . . . 32
4.5.1 Frenet-Serret Frame . . . 33
4.5.2 Parallel Transport Frame . . . 35
4.5.3 Modification of the Spool Vector . . . 35
4.6 Motion Along the Path . . . 38
4.6.1 Setpoint Filter . . . 39
4.6.2 Optimal Path Parameter . . . 39
4.7 Simulation Results . . . 40
4.8 Experimental Results . . . 47

5 Conclusions **56**

A Parameters **58**
A.1 Robot Parameters 58
A.2 Motor Parameters 58
A.3 Parameters of the Path Following Controller 59

List of Figures

2.1 Robot *KUKA LWR IV+* and the coordinate frames 3
2.2 Arm angle ψ defined between the arm plane and the reference plane. . . . 6
2.3 Relevant coordinate frames for the derivation of the inverse kinematics. . 6
2.4 Model of a flexible joint. 14
2.5 Cascade control of a robot with flexible joints. 15

3.1 Orthonormal frame at the point $\boldsymbol{\sigma}_t(\vartheta^*)$ of the path γ_t. 21

4.1 Rope wound by the robot around three rods. 30
4.2 3-D view of the robot together with the end-effector. 31
4.3 3-D sketch of the end-effector. 32
4.4 Recorded path around three rods. 33
4.5 Projection of the spool vector $\mathbf{B}$ onto the $\mathbf{r}$-$\mathbf{z}$-plane. 37
4.6 Allowed range of the spool vector $\mathbf{B}$. 37
4.7 Reflection of the spool vector $\mathbf{B}$ through the $\boldsymbol{\kappa}$-$\mathbf{z}$-plane. 38
4.8 Rate limiter and setpoint filter for achieving a smooth trajectory. 39
4.9 (Simulation) Distance of the system output to the path. 40
4.10 (Simulation) Tangential coordinate η_1 and its time derivative η_2. 41
4.11 (Simulation) Time evolution of the optimal path parameter ϑ^*. 41
4.12 (Simulation) Transversal coordinates ξ_1 and ξ_3 and velocities ξ_2 and ξ_4. . 42
4.13 (Simulation) Use of the Frenet-Serret frame for winding a rope. 43
4.14 (Simulation) Use of the parallel transport frame for winding a rope. . . . 43
4.15 (Simulation) Orientation error $\mathbf{e}_r$. 44
4.16 (Simulation) Joint torques $\boldsymbol{\tau}$. 45
4.17 (Simulation) Joint angles $\mathbf{q}$. 46
4.18 Measured distance of the system output to the path. 47
4.19 Measured tangential coordinate η_1 and velocity η_2. 48
4.20 Measured transversal coordinates ξ_1 and ξ_3 and velocities ξ_2 and ξ_4. . . 49
4.21 Time evolution of the optimal path parameter ϑ^*. 49
4.22 Experimental result of using the Frenet-Serret frame for winding a rope. . 50
4.23 Experimental result of using the parallel transport frame for winding a rope. 50
4.24 Measured orientation error $\mathbf{e}_r$. 51
4.25 Torque of the spool motor τ_m and its desired value τ_m^d. 52
4.26 Estimated external torques $\boldsymbol{\tau}_{ext}$. 53
4.27 Joint torques $\boldsymbol{\tau}$. 54
4.28 Measured joint angles $\mathbf{q}$. 55

1 Introduction

Industrial robots have been employed for automating various tasks for decades. The goal of this thesis is to use a robot for winding a rope around objects, such as rods, in the workspace of the robot. Additionally, the tension of the rope has to be controlled during the winding process. The presented method could have applications in cable installation, flexible packaging of objects or large-scale 3D printing by substituting the rope with suitable materials.

In this work, a lightweight industrial robot, *KUKA LWR IV+* , which is designed for a safe human-robot interaction, is employed. In Chapter 2, the mathematical model of the robot is developed. First, the forward and inverse kinematics of the robot are derived. Secondly, the equations of motion are found assuming a rigid-body model for the robot. Lastly, this dynamic model is extended to take into account the joint flexibility of the robot.

The desired path for winding a rope around the rods is specified using a teach-in procedure. This path is traversed by the robot end-effector by applying the path following controller designed in Chapter 3. This control strategy uses transverse feedback linearization, which splits up the motion of the end-effector into tangential and transversal dynamics with respect to the path. Controlling the transversal dynamics yields asymptotic convergence of the robot end-effector to the specified path. The desired motion of the end-effector along the path is achieved by controlling the tangential dynamics. In order to control the orientation of the end-effector an input-output linearization is performed.

The robot end-effector consists of a rope spool and a motor, which is used to control the tension of the rope. A main aspect of this thesis, as described in Chapter 4, is to appropriately specify the orientation of the robot end-effector during the winding process. This is not a trivial task as it requires additional information on how the rods are oriented and when the orientation of the robot end-effector has to be changed. Analyzing the winding process shows that the desired path can be utilized to derive the desired orientation. For this purpose, two moving frames, the Frenet-Serret frame and the parallel transport frame, are employed. These frames need to be modified in order to derive suitable end-effector orientations for the rope winding process.

The simulation and experimental results presented in Chapter 4 show the feasibility of the proposed method for rope winding. Results of employing the Frenet-Serret frame and the parallel transport frame are compared to determine which frame is more suitable for the rope winding application.

2 Mathematical Modeling

In order to design controllers for robots it is almost indispensable to have a mathematical model describing the dynamics of the robot. Moreover, having a model makes it possible to perform dynamical simulations of the robot.

The model of a robot arm can consist of rigid or flexible links. This distinction also applies to the joints. The robot investigated in this work, *KUKA LWR IV+*, is treated as a system of rigid links that are connected by elastic joints. This joint flexibility mainly arises from the use of harmonic drive gears. Compared to rigid manipulators, robots with elastic joints require both an extended mathematical model and an adapted control approach.

This chapter begins with the description of the robot forward kinematics. Then an analytical inverse kinematics of the robot is derived. In Section 2.4, the dynamic model is determined assuming that the robot has rigid joints. In Section 2.5, this model is modified in order to take into account the joint flexibility. Finally, it will be analyzed under what circumstances the model of rigid robots can be used for controlling robots with flexible joints.

2.1 Forward Kinematics

Given a configuration of the robot joints, the forward kinematics determines the resulting Cartesian position and orientation of the end-effector.

For the calculation of the forward kinematics, in a first step coordinate frames are introduced, which are rigidly attached to each of the robot links. The position and orientation of the end-effector are then determined using homogeneous transformations between these frames.

Figure 2.1 shows the 7-axis robot in its zero position together with the coordinate frames. The assignment of the coordinate frames follows the Denavit-Hartenberg (DH) convention consisting of four rules, see [1]. Assuming that frame $i-1$ is already given, the successive frame i is defined as follows:

1. The $\mathbf{z}_i$-axis points along the axis of rotation.

2. If $\mathbf{z}_i$ and $\mathbf{z}_{i-1}$ have a unique common normal, the origin $\mathbf{0}_i$ is given by the intersection point of this common normal with the $\mathbf{z}_i$-axis. Otherwise, the origin can be arbitrarily placed along the $\mathbf{z}_i$-axis.

3. The $\mathbf{x}_i$-axis is normal to the plane spanned by $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$, while its positive direction is arbitrary, i. e. $\mathbf{x}_i = \pm(\mathbf{z}_{i-1} \times \mathbf{z}_i)$. In Figure 2.1, all $\mathbf{x}$-axes point towards the viewer.

4. The $\mathbf{y}_i$-axis is given by the cross product of $\mathbf{x}_i$ and $\mathbf{z}_i$ to ensure the right-handedness of the resulting coordinate system, i. e. $\mathbf{y}_i = \mathbf{z}_i \times \mathbf{x}_i$.

The origin and the $\mathbf{x}$-axis of the base frame $\mathbf{x}_0\mathbf{y}_0\mathbf{z}_0$ are free to be chosen. Similarly, the direction of the $\mathbf{z}$-axis of the end-effector frame is arbitrary since it is not followed by a further joint. Here, it is chosen to be the same as the $\mathbf{z}_6$-axis such that the 7th and 6th frame coincide.

The DH-parameters $(\vartheta_i, d_i, a_i, \alpha_i)$ of frame i are defined in the following order:

1. ϑ_i is the angle by which the previous frame $i-1$ is to be rotated about $\mathbf{z}_{i-1}$ such that $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ are parallel.

2. d_i is the distance along the $\mathbf{z}_{i-1}$-axis to align the $\mathbf{x}$-axes.

3. a_i is the distance along the $\mathbf{x}$-axis to make the origins of the frames coincide.

4. α_i is the angle by which the previous frame $i-1$ is to be rotated about the $\mathbf{x}$-axis such that $\mathbf{z}_{i-1}$ and $\mathbf{z}_i$ are parallel.

In the special case where $\mathbf{z}_i$ and $\mathbf{z}_{i-1}$ are parallel, there is no unique common normal between them. In this case the parameter d_i, which specifies the offset to the common normal, can be freely chosen.

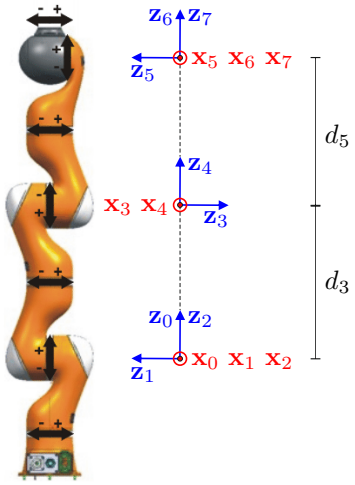


Figure 2.1: Robot *KUKA LWR IV+* and the coordinate frames according to the DH-convention.

Table 2.1 lists the DH-parameters of the robot. For revolute joints the parameter ϑ_i is the degree of freedom and is denoted by q_i. The only non-zero distances d_3 and d_5 are also depicted in Figure 2.1. The numeric values of these parameters can be found in Appendix A.

i	ϑ_i	d_i	a_i	α_i
1	q_1	0	0	$\pi/2$
2	q_2	0	0	$-\pi/2$
3	q_3	d_3	0	$-\pi/2$
4	q_4	0	0	$\pi/2$
5	q_5	d_5	0	$\pi/2$
6	q_6	0	0	$-\pi/2$
7	q_7	0	0	0

Table 2.1: Denavit-Hartenberg parameters of *KUKA LWR IV+* according to the frames shown in Figure 2.1.

Homogeneous Transformation

A vector $\mathbf{p}_i \in \mathbb{R}^3$ expressed in the coordinate frame i can be transformed into the frame $i-1$ using the homogeneous transformation matrix $\mathbf{T}_{i-1}^{i} \in \mathbb{R}^{4\times4}$ by

$$\begin{bmatrix} \mathbf{p}_{i-1} \\ 1 \end{bmatrix} = \mathbf{T}_{i-1}^{i} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} . \tag{2.1}$$

One of the advantages of using DH-parameters $(\vartheta_i, d_i, a_i, \alpha_i)$ is the fact that the homogeneous transformation matrix can be easily composed of four basic transformations

$$\begin{aligned} \mathbf{T}_{i-1}^{i} &= \mathrm{Rot}(\mathbf{z}, \vartheta_i)\mathrm{Trans}(\mathbf{z}, d_i)\mathrm{Trans}(\mathbf{x}, a_i)\mathrm{Rot}(\mathbf{x}, \alpha_i) \\ &= \begin{bmatrix} \cos(\vartheta_i) & -\sin(\vartheta_i) & 0 & 0 \\ \sin(\vartheta_i) & \cos(\vartheta_i) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_i) & -\sin(\alpha_i) & 0 \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos(\vartheta_i) & -\sin(\vartheta_i)\cos(\alpha_i) & \sin(\vartheta_i)\sin(\alpha_i) & a_i\cos(\vartheta_i) \\ \sin(\vartheta_i) & \cos(\vartheta_i)\cos(\alpha_i) & -\cos(\vartheta_i)\sin(\alpha_i) & a_i\sin(\vartheta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} . \end{aligned} \tag{2.2}$$

The robot pose describes the position and orientation of the end-effector expressed in the robot base coordinate system. It can be represented by a homogeneous transformation matrix, which is calculated by consecutive multiplication of the matrices $\mathbf{T}_{i-1}^{i}$, $i = 1, \ldots, 7$, i.e.

$$\mathbf{T}_0^7 = \mathbf{T}_6^7 \ldots \mathbf{T}_0^1 . \tag{2.3}$$

The resulting matrix has the form

$$\mathbf{T}_0^7(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_0^7(\mathbf{q}) & \mathbf{d}_0^7(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} , \tag{2.4}$$

where $\mathbf{R}_0^7(\mathbf{q}) \in SO(3)$ is a rotation matrix that performs a transformation on vectors expressed in the end-effector frame into the base frame. The vector $\mathbf{d}_0^7(\mathbf{q}) \in \mathbb{R}^3$ specifies the translation between these two frames. Using the rotation matrix $\mathbf{R}_0^7(\mathbf{q})$ and the position vector $\mathbf{d}_0^7(\mathbf{q})$ the orientation and the position of the end-effector are determined in the base coordinate system.

2.2 Inverse Kinematics

Let $\mathbf{y}$ denote the robot end-effector pose describing its position and orientation in the 3-D Cartesian space. The set of all possible poses $\mathbf{y}$ is the task space of the robot. The configuration space of the robot is given by all joint configurations $\mathbf{q}$ within the mechanical joint limits of the robot.

While the forward kinematics calculates the robot pose as a function of the joint configuration $\mathbf{q}$ by

$$\mathbf{y} = \mathbf{h}(\mathbf{q}) , \tag{2.5}$$

the inverse kinematics determines the joint configuration $\mathbf{q}$ as a function of the given robot pose, i. e.

$$\mathbf{q} = \mathbf{h}^{-1}(\mathbf{y}) . \tag{2.6}$$

Note that finding an analytical inverse kinematics in closed form $\mathbf{h}^{-1}(.)$ is not possible for all robot types.

The path following control strategy that will be discussed in Chapter 3 is designed in the task space, so that the inverse kinematics is not required. However, the robot end-effector is driven to the starting point of the path by means of a computed torque controller that is designed in the configuration space. Hence, the inverse kinematics is employed to obtain the required joint angles for the desired starting point.

The robot *KUKA LWR IV+* features a so-called Spherical-Roll-Spherical (S-R-S) structure. A joint with 3 intersecting axes at a single point is called a spherical joint. Comparable to the human arm, the first three joints $(1, 2, 3)$ of the robot build up a virtual spherical joint as the shoulder S, the 4th joint is interpreted as the elbow E, and the last three joints $(5, 6, 7)$ form a virtual spherical joint as the wrist W.

The configuration space of the robot has 7 degrees of freedom, whereas for specifying any position and orientation in the Cartesian space 6 degrees of freedom are sufficient. Hence, this robot is said to be a redundant manipulator. A redundant robot has a self-motion, i. e. it can vary its joint configuration without changing the end-effector pose. In other words, there is no unique joint configuration for a given pose. In order to parametrize this redundancy, an arm angle ψ is adopted, which is defined as the angle between the arm

plane and the reference plane, see Figure 2.2. The arm plane is spanned by the shoulder S, the elbow E and the wrist W, and the reference plane is the arm plane for the case $q_3 = 0$, i. e. when the joint axes 2 and 4 are parallel. In the following, quantities in the reference plane are denoted by the superscript o as in q_1^o, $\mathbf{R}_0^{3,o}$, etc.

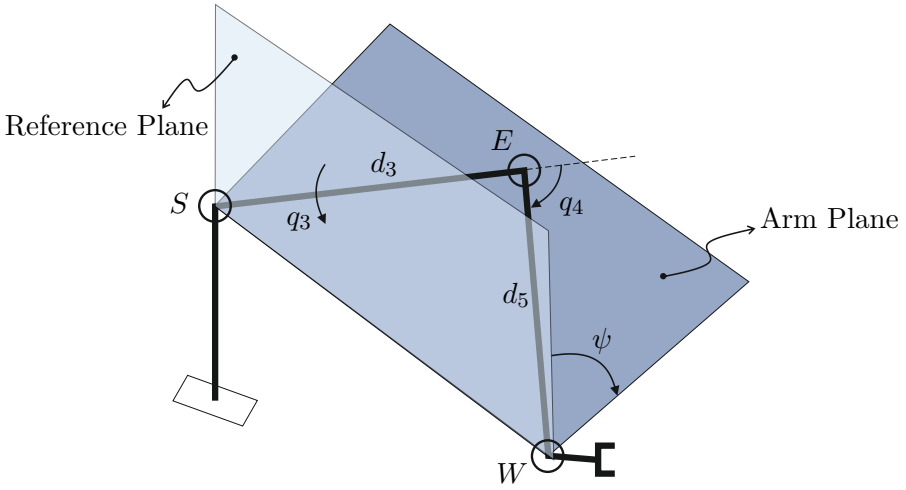


Figure 2.2: Arm angle ψ defined between the arm plane and the reference plane.

Based on the approach in [2], the joint angles are computed as a function of the arm angle ψ. This choice allows that the robot can be regarded as a virtual non-redundant manipulator for a fixed arm angle. Figure 2.3 shows the relevant coordinate frames for

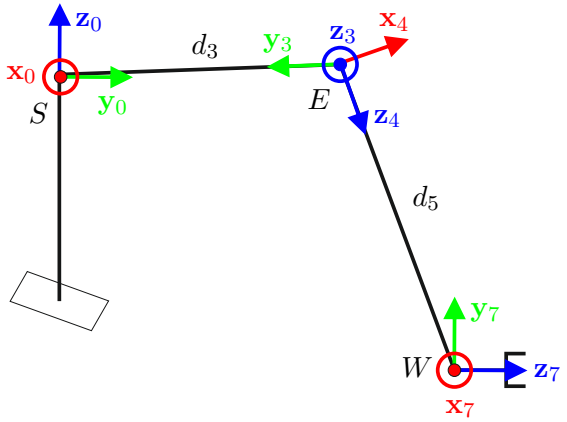


Figure 2.3: Relevant coordinate frames for the derivation of the inverse kinematics.

the calculation of the inverse kinematics. Let $\mathbf{l}_{se,3}$ and $\mathbf{l}_{ew,4}$ be constant vectors denoting the shoulder-elbow and the elbow-wrist distance in the 3-frame and 4-frame, respectively

$$\mathbf{l}_{se,3} = \begin{bmatrix} 0 & -d_3 & 0 \end{bmatrix}^{\mathrm{T}}, \tag{2.7a}$$

$$\mathbf{l}_{ew,4} = \begin{bmatrix} 0 & 0 & d_5 \end{bmatrix}^{\mathrm{T}}. \tag{2.7b}$$

The position of the end-effector in the base frame is given by

$$\mathbf{x}_0^7 = \mathbf{R}_0^3 \left(\mathbf{l}_{se,3} + \mathbf{R}_3^4 \, \mathbf{l}_{ew,4} \right) \ . \tag{2.8}$$

The end-effector orientation in the base frame can be determined by using the rotation matrix

$$\mathbf{R}_0^7 = \mathbf{R}_0^3 \, \mathbf{R}_3^4 \, \mathbf{R}_4^7 \ . \tag{2.9}$$

According to Figure 2.3 where the base coordinate system is placed in the shoulder, the vector $\mathbf{x}_{sw,0}$ which connects the shoulder to the wrist is equal to $\mathbf{x}_0^7$

$$\mathbf{x}_{sw,0} = \mathbf{x}_0^7 = \mathbf{R}_0^3 \left(\mathbf{l}_{se,3} + \mathbf{R}_3^4 \, \mathbf{l}_{ew,4} \right) \ . \tag{2.10}$$

This vector points along the axis around which the self-motion of the robot occurs. Let $\mathbf{e}_{sw,0}$ be the unit vector of this axis. The rotation of the wrist W by the arm angle ψ in the base frame can be calculated using the axis-angle representation in the form

$$\mathbf{R}_0^w(\psi) = \mathbf{I}_3 + \sin(\psi)\mathbf{S}(\mathbf{e}_{sw,0}) + \big(1 - \cos(\psi)\big) \, \mathbf{S}^2(\mathbf{e}_{sw,0}) \ . \tag{2.11}$$

Herein, $\mathbf{I}_3 \in \mathbb{R}^{3\times3}$ is the identity matrix and $\mathbf{S}(\mathbf{e}_{sw,0})$ denotes the skew-symmetric matrix of the vector $\mathbf{e}_{sw,0}$. The skew-symmetric matrix of a vector $\mathbf{a}$ has the form

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \ . \tag{2.12}$$

Using (2.11) the wrist orientation can be determined by the rotation matrix

$$\mathbf{R}_0^4(\psi) = \mathbf{R}_0^w(\psi) \, \mathbf{R}_0^{4,o} \ , \tag{2.13}$$

where $\mathbf{R}_0^{4,o}$ determines the orientation of the wrist in the reference plane. This equation can be written as

$$\underbrace{\mathbf{R}_0^3(\psi) \, \mathbf{R}_3^4(\psi)}_{\mathbf{R}_0^4(\psi)} = \mathbf{R}_0^w(\psi) \, \underbrace{\mathbf{R}_0^{3,o} \, \mathbf{R}_3^{4,o}}_{\mathbf{R}_0^{4,o}} \ . \tag{2.14}$$

Since for a given pose the elbow joint angle q_4 is constant and independent of ψ, i. e. $\mathbf{R}_3^4(\psi) = \mathbf{R}_3^{4,o}$, the rotation matrix $\mathbf{R}_0^3(\psi)$ is calculated to be

$$\mathbf{R}_0^3(\psi) = \mathbf{R}_0^w(\psi) \, \mathbf{R}_0^{3,o} \ . \tag{2.15}$$

By inserting this result in (2.9) the orientation of the end-effector in the base frame can be obtained using the rotation matrix

$$\mathbf{R}_0^7(\psi) = \mathbf{R}_0^w(\psi) \, \mathbf{R}_0^{3,o} \, \mathbf{R}_3^{4,o} \, \mathbf{R}_4^7(\psi) \ . \tag{2.16}$$

The elbow joint angle q_4, which is independent of ψ, is uniquely determined using the law of cosines according to Figure 2.2

$$\cos(q_4) = \frac{\left\|\mathbf{x}_{sw,0}\right\|^2 - d_3^2 - d_5^2}{2d_3\, d_5} \ . \tag{2.17}$$

Setting $q_3 = 0$ in (2.10) the shoulder joint angles q_1^o and q_2^o in the reference plane can be found by solving the equation

$$\mathbf{x}_{sw,0} = \underbrace{\mathbf{R}_0^{1,o}(\mathbf{q})\mathbf{R}_1^{2,o}(\mathbf{q})\mathbf{R}_2^3(\mathbf{q})\Big|_{q_3=0}}_{\mathbf{R}_0^{3,o}(\mathbf{q})} \left(\mathbf{l}_{se,3} + \mathbf{R}_3^4(\mathbf{q})\,\mathbf{l}_{ew,4}\right) \ . \tag{2.18}$$

Next, the shoulder joint angles q_1 and q_2 can be determined as a function of the arm angle ψ. Inserting (2.11) in (2.15) yields

$$\mathbf{R}_0^3(\psi) = \mathbf{A}_s \sin(\psi) + \mathbf{B}_s \cos(\psi) + \mathbf{C}_s \ , \tag{2.19}$$

with the matrices

$$\mathbf{A}_s = \mathbf{S}(\mathbf{e}_{sw,0})\,\mathbf{R}_0^{3,o} \ , \tag{2.20a}$$

$$\mathbf{B}_s = -\mathbf{S}^2(\mathbf{e}_{sw,0})\,\mathbf{R}_0^{3,o} \ , \tag{2.20b}$$

$$\mathbf{C}_s = \left(\mathbf{I}_3 + \mathbf{S}^2(\mathbf{e}_{sw,0})\right)\mathbf{R}_0^{3,o} = \left[\mathbf{e}_{sw,0}\,\mathbf{e}_{sw,0}^{\mathrm{T}}\right]\,\mathbf{R}_0^{3,o} \ . \tag{2.20c}$$

On the other hand, using the forward kinematics the rotation matrix $\mathbf{R}_0^3(\mathbf{q})$ is given by

$$\mathbf{R}_0^3(\mathbf{q}) = \begin{bmatrix} * & \cos(q_1)\sin(q_2) & * \\ * & \sin(q_1)\sin(q_2) & * \\ \sin(q_2)\cos(q_3) & -\cos(q_2) & -\sin(q_2)\sin(q_3) \end{bmatrix} \ , \tag{2.21}$$

where only the relevant elements are shown. The shoulder angle q_1 can be calculated by dividing the $(2,2)$-element of the matrix $\mathbf{R}_0^3(\mathbf{q})$ by its $(1,2)$-element. Since every entry of $\mathbf{R}_0^3(\mathbf{q})$ has to be equal to the corresponding entry of $\mathbf{R}_0^3(\psi)$, combining (2.19) and (2.21) yields

$$\tan(q_1) = \frac{\left(\mathbf{R}_0^3\right)_{22}}{\left(\mathbf{R}_0^3\right)_{12}} = \frac{a_{s22}\,\sin(\psi) + b_{s22}\,\cos(\psi) + c_{s22}}{a_{s12}\,\sin(\psi) + b_{s12}\,\cos(\psi) + c_{s12}} \ , \tag{2.22}$$

with a_{sij}, b_{sij} and c_{sij} as the (i,j)-th elements of the matrices $\mathbf{A}_s$, $\mathbf{B}_s$ and $\mathbf{C}_s$, respectively. Similarly, the shoulder joint angles q_2 and q_3 are calculated to be

$$\cos(q_2) = -\left(\mathbf{R}_0^3\right)_{32} = -a_{s32}\,\sin(\psi) - b_{s32}\,\cos(\psi) - c_{s32} \ , \tag{2.23a}$$

$$\tan(q_3) = \frac{-\left(\mathbf{R}_0^3\right)_{33}}{\left(\mathbf{R}_0^3\right)_{31}} = \frac{-a_{s33}\,\sin(\psi) - b_{s33}\,\cos(\psi) - c_{s33}}{a_{s31}\,\sin(\psi) + b_{s31}\,\cos(\psi) + c_{s31}} \ . \tag{2.23b}$$

In order to find the wrist angles q_5, q_6 and q_7, first the rotation matrix $\mathbf{R}_4^7(\psi)$ is obtained from (2.16) in the form

$$\mathbf{R}_4^7(\psi) = \left(\mathbf{R}_0^w(\psi)\,\mathbf{R}_0^{3,o}\,\mathbf{R}_3^{4,o}\right)^{\mathrm{T}}\,\mathbf{R}_0^7(\psi)\ . \tag{2.24}$$

Using (2.11) the matrix $\mathbf{R}_4^7(\psi)$ is given by

$$\mathbf{R}_4^7(\psi) = \mathbf{A}_w\sin(\psi) + \mathbf{B}_w\cos(\psi) + \mathbf{C}_w\ , \tag{2.25}$$

with the matrices

$$\mathbf{A}_w = \mathbf{R}_4^3\,\mathbf{A}_s^{\mathrm{T}}\,\mathbf{R}_0^7\ , \tag{2.26a}$$
$$\mathbf{B}_w = \mathbf{R}_4^3\,\mathbf{B}_s^{\mathrm{T}}\,\mathbf{R}_0^7\ , \tag{2.26b}$$
$$\mathbf{C}_w = \mathbf{R}_4^3\,\mathbf{C}_s^{\mathrm{T}}\,\mathbf{R}_0^7\ , \tag{2.26c}$$

with $\mathbf{A}_s$, $\mathbf{B}_s$ and $\mathbf{C}_s$ from (2.20). The rotation matrix $\mathbf{R}_4^7(\mathbf{q})$ resulting from the forward kinematics is given by

$$\mathbf{R}_4^7(\mathbf{q}) = \begin{bmatrix} * & * & -\cos(q_5)\sin(q_6) \\ * & * & -\sin(q_5)\sin(q_6) \\ \sin(q_6)\cos(q_7) & -\sin(q_6)\sin(q_7) & \cos(q_6) \end{bmatrix}\ . \tag{2.27}$$

Comparing this result to $\mathbf{R}_4^7(\psi)$ from (2.25), the wrist joint angles q_5, q_6 and q_7 are obtained as a function of the arm angle ψ

$$\tan(q_5) = \frac{-\left(\mathbf{R}_4^7\right)_{23}}{-\left(\mathbf{R}_4^7\right)_{13}} = \frac{-a_{w23}\,\sin(\psi) - b_{w23}\,\cos(\psi) - c_{w23}}{-a_{w13}\,\sin(\psi) - b_{w13}\,\cos(\psi) - c_{w13}}\ , \tag{2.28a}$$
$$\cos(q_6) = \left(\mathbf{R}_4^7\right)_{33} = a_{w33}\,\sin(\psi) + b_{w33}\,\cos(\psi) + c_{w33}\ , \tag{2.28b}$$
$$\tan(q_7) = \frac{-\left(\mathbf{R}_4^7\right)_{32}}{\left(\mathbf{R}_4^7\right)_{31}} = \frac{-a_{w32}\,\sin(\psi) - b_{w32}\,\cos(\psi) - c_{w32}}{a_{w31}\,\sin(\psi) + b_{w31}\,\cos(\psi) + c_{w31}}\ , \tag{2.28c}$$

with a_{wij}, b_{wij} and c_{wij} as the (i,j)-th elements of the matrices $\mathbf{A}_w$, $\mathbf{B}_w$ and $\mathbf{C}_w$, respectively.

Since the joints have mechanical limits (see Appendix A), not every solution $\mathbf{q}$ of the inverse kinematics is physically feasible. Hence, the results are modified in a further step to ensure that the resulting joint configuration $\mathbf{q}$ is within the joint limits. In this work, the arm angle ψ is utilized to avoid joint limit violations. First, for an arbitrary arm angle ψ the joint angles $\mathbf{q}$ are calculated using the equations (2.17), (2.22), (2.23) and (2.28). Assume that the resulting first joint angle q_1 is below its lower limit, i. e. $q_1 < q_1^{min}$. In order to obtain an alternative configuration, first a value of π is added to q_1 and the result is wrapped to the interval $[-\pi, \pi]$. The final result will be within the joint limits but it will change the given end-effector pose. To preserve the original pose, the sign of the joint

q_2 is changed and a value of π is added to q_3. Similarly, the limit violations of the joints q_3 and q_5 can be remedied. Altogether, for $i = 1, 3, 5$ this method can be written as

$$q_i := q_i + k(q_i) \ , \tag{2.29a}$$
$$q_{i+1} := -q_{i+1} \ , \tag{2.29b}$$
$$q_{i+2} := q_{i+2} + k(q_i) \ , \tag{2.29c}$$

where

$$k(q_i) = \begin{cases} \pi, & \text{if } q_i < q_i^{min} \\ -\pi, & \text{if } q_i > q_i^{max} \end{cases} \ , \tag{2.30}$$

see, e. g., [3]. To rectify the limit violation of q_7, the joint angles q_5 and q_6 are modified. In this case, (2.29) and (2.30) with $k(q_7)$ have to be used. The solutions for the joints q_2, q_4 and q_6 are given by the function arccos(.), which has a range of $[0, \pi]$. Hence, the resulting joint angles are above their lower limit. In case any of these joints exceed their upper limit, the calculation of the inverse kinematics is repeated for a different arm angle ψ.

2.3 Manipulator Jacobian Matrix

The manipulator Jacobian matrix $\mathbf{J}_0^i(\mathbf{q}) \in \mathbb{R}^{6\times 7}$ establishes the relation between the joint velocities $\dot{\mathbf{q}}(t) \in \mathbb{R}^7$ and the linear and angular velocity of each link in the form

$$\begin{bmatrix} \mathbf{v}_0^i \\ \boldsymbol{\omega}_0^i \end{bmatrix} = \mathbf{J}_0^i(\mathbf{q})\, \dot{\mathbf{q}} \ . \tag{2.31}$$

Herein, the linear velocity $\mathbf{v}_0^i(t) \in \mathbb{R}^3$ and the angular velocity $\boldsymbol{\omega}_0^i(t) \in \mathbb{R}^3$ of the center of mass of the ith link are expressed in the base coordinate system denoted by the subscript 0. The manipulator Jacobian matrix consists of two parts

$$\mathbf{J}_0^i(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{v,0}^i(\mathbf{q}) \\ \mathbf{J}_{\omega,0}^i(\mathbf{q}) \end{bmatrix} \ , \tag{2.32}$$

where $\mathbf{J}_{v,0}^i(\mathbf{q}) \in \mathbb{R}^{3\times 7}$ is used to calculate the linear velocity and $\mathbf{J}_{\omega,0}^i(\mathbf{q}) \in \mathbb{R}^{3\times 7}$ determines the angular velocity.

The Jacobian matrix $\mathbf{J}_{v,0}^i(\mathbf{q})$ is calculated as the partial derivative of the position vector $\mathbf{p}_0^i$ with respect to the joint angles $\mathbf{q}$, i. e.

$$\mathbf{J}_{v,0}^i(\mathbf{q}) = \frac{\partial \mathbf{p}_0^i}{\partial \mathbf{q}} \, , \qquad i = 1, \ldots, 7 \ . \tag{2.33}$$

Consider the position vector of the center of mass of the ith robot link expressed in the coordinate frame attached to the link

$$\mathbf{p}_i^i = \begin{bmatrix} c_{i,x} & c_{i,y} & c_{i,z} \end{bmatrix}^{\mathrm{T}} \ . \tag{2.34}$$

This vector is transformed into the base coordinate system using the homogeneous transformation matrix $\mathbf{T}_0^i(\mathbf{q})$ to obtain $\mathbf{p}_0^i$

$$\begin{bmatrix} \mathbf{p}_0^i \\ 1 \end{bmatrix} = \mathbf{T}_0^i(\mathbf{q}) \begin{bmatrix} \mathbf{p}_i^i \\ 1 \end{bmatrix} . \tag{2.35}$$

Using the Jacobian matrix resulting from (2.33) the linear velocity of this point in the base coordinate system is given by

$$\mathbf{v}_0^i = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p}_0^i = \mathbf{J}_{v,0}^i(\mathbf{q})\dot{\mathbf{q}} . \tag{2.36}$$

It can be shown, see, e. g., [4], that the components of the vector of angular velocity

$$\boldsymbol{\omega}_0^i = \begin{bmatrix} \omega_{x,0}^i & \omega_{y,0}^i & \omega_{z,0}^i \end{bmatrix}^{\mathrm{T}} , \qquad i = 1, \ldots, 7 , \tag{2.37}$$

can be extracted from the corresponding skew-symmetric matrix

$$\mathbf{S}(\boldsymbol{\omega}_0^i) = \begin{bmatrix} 0 & -\omega_{z,0}^i & \omega_{y,0}^i \\ \omega_{z,0}^i & 0 & -\omega_{x,0}^i \\ -\omega_{y,0}^i & \omega_{x,0}^i & 0 \end{bmatrix} , \qquad i = 1, \ldots, 7 . \tag{2.38}$$

On the other hand, this matrix can be calculated by

$$\mathbf{S}(\boldsymbol{\omega}_0^i) = \dot{\mathbf{R}}_0^i \left(\mathbf{R}_0^i\right)^{\mathrm{T}} , \tag{2.39}$$

where $\dot{\mathbf{R}}_0^i$ denotes the derivative of the rotation matrix $\mathbf{R}_0^i$ with respect to time.

The Jacobian matrix $\mathbf{J}_{\omega,0}^i(\mathbf{q})$ is given by

$$\mathbf{J}_{\omega,0}^i(\mathbf{q}) = \frac{\partial \boldsymbol{\omega}_0^i}{\partial \mathbf{q}} , \qquad i = 1, \ldots, 7 . \tag{2.40}$$

For a given robot state $(\mathbf{q}, \dot{\mathbf{q}})$ the angular velocity of the ith link can be obtained using the Jacobian matrix by

$$\boldsymbol{\omega}_0^i = \mathbf{J}_{\omega,0}^i(\mathbf{q})\dot{\mathbf{q}} , \qquad i = 1, \ldots, 7 . \tag{2.41}$$

2.4 Dynamic Model of Rigid-Body Robots

Using the Euler-Lagrange equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial}{\partial \dot{q}_i} L\right) - \frac{\partial}{\partial q_i} L = \tau_i , \qquad i = 1, \ldots, n \tag{2.42}$$

the dynamic model of a mechanical system consisting of n rigid bodies can be derived. Herein, L denotes the Lagrangian function of the system that is defined as

$$L = T - V , \tag{2.43}$$

with the kinetic energy T and the potential energy V. Furthermore, q_i, $\dot{q}_i$ and τ_i denote the generalized coordinates, velocities and forces of the ith body, respectively.

The equations of motion of a robot with n degrees of freedom have the form

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}\,(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \mathbf{g}\,(\mathbf{q}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext}\ , \tag{2.44}$$

where $\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}} \in \mathbb{R}^n$ denote the generalized coordinates, velocities and accelerations, respectively. The symmetric, positive definite mass matrix $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n\times n}$ includes the masses and moments of inertia of the robot links, the expression $\mathbf{C}\,(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}}$ contains the centrifugal and Coriolis terms and $\mathbf{g}\,(\mathbf{q}) \in \mathbb{R}^n$ is the vector of potential forces. The generalized forces $\boldsymbol{\tau} \in \mathbb{R}^n$ are motor torques that are applied to each link as control inputs and $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ are external forces that act as disturbances on the joints. Note that in this model, the friction in the joints is neglected.

The special structure of (2.44) is due to the following two conditions that robotic manipulators satisfy, see, e. g., [4]. First, the kinetic energy is a quadratic function of the vector of joint velocities $\dot{\mathbf{q}}$, i. e.

$$T = \frac{1}{2}\sum_{j=1}^{n}\sum_{i=1}^{n} m_{ij}(\mathbf{q})\dot{q}_i\dot{q}_j = \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{M}(\mathbf{q})\dot{\mathbf{q}}\ , \tag{2.45}$$

and second, the potential energy $V(\mathbf{q})$ depends only on the joint angles $\mathbf{q}$ and is independent of the joint velocities $\dot{\mathbf{q}}$.

In order to obtain the dynamic model of the robot, the individual components in (2.44) need to be determined. The mass matrix consists of a translational and a rotational part, which in turn are the sum of the contributions from each link, i.e.

$$\mathbf{M}(\mathbf{q}) = \mathbf{M}_v(\mathbf{q}) + \mathbf{M}_\omega(\mathbf{q}) = \sum_{i=1}^{n}\Big(\mathbf{M}_{v,i}(\mathbf{q}) + \mathbf{M}_{\omega,i}(\mathbf{q})\Big)\ . \tag{2.46}$$

The rotational parts of the mass matrix are given by

$$\mathbf{M}_{\omega,i}(\mathbf{q}) = \Big(\mathbf{J}^i_{\omega,0}(\mathbf{q})\Big)^{\mathrm{T}}\,\mathbf{I}^i_0\mathbf{J}^i_{\omega,0}(\mathbf{q})\ . \tag{2.47}$$

The expression

$$\mathbf{I}^i_0 = \mathbf{R}^i_0\mathbf{I}^i_i{\mathbf{R}^i_0}^{\mathrm{T}} \tag{2.48}$$

is a similarity transformation that transforms the inertia tensor $\mathbf{I}^i_i$ of the ith link from the frame attached to the link into the inertial frame. The translational parts of the mass matrix are calculated as

$$\mathbf{M}_{v,i}(\mathbf{q}) = m_i\,\Big(\mathbf{J}^i_{v,0}(\mathbf{q})\Big)^{\mathrm{T}}\,\mathbf{J}^i_{v,0}(\mathbf{q})\ , \tag{2.49}$$

where m_i denotes the mass of the ith link.

The (k,j)-th element of the matrix $\mathbf{C}\,(\mathbf{q},\dot{\mathbf{q}})$ is given by

$$c_{kj}(\mathbf{q}) = \sum_{i=1}^{n} c_{ijk}(\mathbf{q})\dot{q}_i = \sum_{i=1}^{n}\frac{1}{2}\left\{\frac{\partial}{\partial q_i}m_{kj}(\mathbf{q}) + \frac{\partial}{\partial q_j}m_{ki}(\mathbf{q}) - \frac{\partial}{\partial q_k}m_{ij}(\mathbf{q})\right\}\dot{q}_i\ , \tag{2.50}$$

where c_{ijk} are the so-called Christoffel symbols of the first kind and $m_{ij}(\mathbf{q})$ is the (i,j)-th element of the mass matrix $\mathbf{M}(\mathbf{q})$.

In order to obtain the vector of potential forces $\mathbf{g}(\mathbf{q})$, which does not depend on the joint velocity and acceleration, first the total potential energy V of the robot is calculated. Using the vector

$$\boldsymbol{\gamma}_0 = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^{\mathrm{T}} , \tag{2.51}$$

which gives the direction of the gravitational acceleration g in the inertial frame, the potential energy is calculated to be

$$V = \sum_{i=1}^{n} V_i = \sum_{i=1}^{n} -m_i \boldsymbol{\gamma}_0^{\mathrm{T}} \, \mathbf{p}_0^i , \tag{2.52}$$

with m_i and $\mathbf{p}_0^i$ as the mass and the position vector of the center of mass of the ith link, respectively. The vector of the potential forces $\mathbf{g}(\mathbf{q})$ is given by the partial derivative of the potential energy with respect to the joint angles $\mathbf{q}$, i. e.

$$\mathbf{g}(\mathbf{q}) = \frac{\partial V}{\partial \mathbf{q}} . \tag{2.53}$$

Finally, the relation between the generalized forces $\boldsymbol{\tau}$ and the motor torques $\boldsymbol{\tau}_m$ is described. Since the motor for actuating the ith link is rigidly attached at the end of the preceding link $(i-1)$, two torques act on the link i: The motor torque $\tau_{m,i}$ acts about the z-axis of the frame $(i-1)$. The other torque, $-\tau_{m,i+1}$, is the reaction torque of the motor attached at the end of the link and acts about the z-axis of the ith frame. Their effect on the generalized coordinates $\mathbf{q}$ is obtained using the Jacobian matrix $\mathbf{J}^i_{\omega,0}(\mathbf{q})$. Summing up the results for all the links yields

$$\boldsymbol{\tau} = \sum_{i=1}^{6} \left(\mathbf{J}^i_{\omega,0}(\mathbf{q}) \right)^{\mathrm{T}} \mathbf{R}_0^{i-1} \left(\begin{bmatrix} 0 \\ 0 \\ \tau_{m,i} \end{bmatrix} + \mathbf{R}^i_{i-1} \begin{bmatrix} 0 \\ 0 \\ -\tau_{m,i+1} \end{bmatrix} \right) + \left(\mathbf{J}^7_{\omega,0}(\mathbf{q}) \right)^{\mathrm{T}} \mathbf{R}_0^6 \begin{bmatrix} 0 \\ 0 \\ \tau_{m,7} \end{bmatrix} . \tag{2.54}$$

Note that in contrast to the other links, only one torque acts on the 7th link. This calculation results in

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_{m,1} & \tau_{m,2} & \tau_{m,3} & \tau_{m,4} & \tau_{m,5} & \tau_{m,6} & \tau_{m,7} \end{bmatrix}^{\mathrm{T}} = \boldsymbol{\tau}_m \tag{2.55}$$

2.5 Dynamic Model of Robots with Flexible Joints

Since *KUKA LWR IV+* has flexible joints, a rigid-body model does not completely describe its dynamics. Figure 2.4 illustrates the model of a flexible joint. Here, ϑ_i is the angle of the motor shaft and q_i is the joint angle. The joint torque is denoted by τ_i.

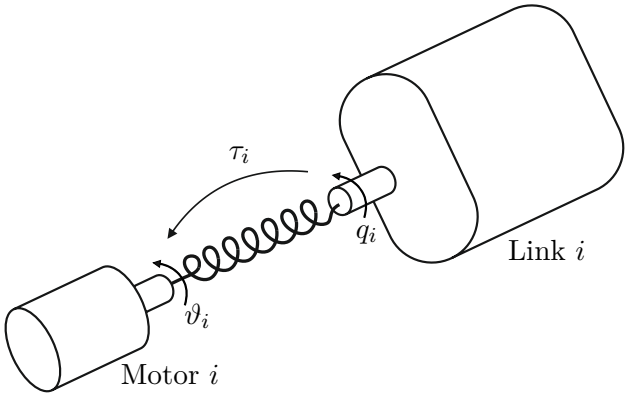


Figure 2.4: Model of a flexible joint.

Taking into account the coupling between the motors and the links, the dynamic model is given by

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \underbrace{\mathbf{K}(\boldsymbol{\vartheta} - \mathbf{q})}_{\boldsymbol{\tau}} + \boldsymbol{\tau}_{ext} \ , \tag{2.56a}$$

$$\mathbf{B}\ddot{\boldsymbol{\vartheta}} + \mathbf{K}(\boldsymbol{\vartheta} - \mathbf{q}) = \boldsymbol{\tau}_m \ , \tag{2.56b}$$

see [5]. In this model, $\mathbf{K} \in \mathbb{R}^{n \times n}$ is a positive definite, diagonal matrix representing the stiffness of the transmission gears, and the positive definite, diagonal matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$ describes the inertia of the motor dynamics. Furthermore, $\boldsymbol{\vartheta} \in \mathbb{R}^n$ comprises the angles of the motor shafts, and $\boldsymbol{\tau}_m \in \mathbb{R}^n$ are the motor torques.

The dynamics of the motor torques (2.56b) are considerably faster than the link dynamics (2.56a). Using the singular perturbation theory and by introducing the small scalar parameter ε the model is split up into two separate subsystems, see [5]. The dynamics of the motor torques form the fast part of the system, whereas the dynamics of the links are the slower subsystem. This partitioning allows for designing a cascade control. Figure 2.5 shows the schematic of this cascade control. Using the parameter ε the matrix $\mathbf{K}$, which contains high stiffness values, can be formally replaced by

$$\mathbf{K} = \frac{\mathbf{K}_\varepsilon}{\varepsilon^2} \ , \tag{2.57}$$

with a positive definite diagonal matrix $\mathbf{K}_\varepsilon$.

The joint torques $\boldsymbol{\tau}$ can be controlled in an inner loop using the control law

$$\boldsymbol{\tau}_m = \boldsymbol{\tau}_d - \mathbf{K}_\tau(\boldsymbol{\tau} - \boldsymbol{\tau}_d) - \varepsilon \mathbf{D}_\tau \dot{\boldsymbol{\tau}} \ , \tag{2.58}$$

with positive definite matrices $\mathbf{K}_\tau, \mathbf{D}_\tau \in \mathbb{R}^{n \times n}$ as controller gains. Furthermore, $\boldsymbol{\tau}_d$ is the desired reference torque, which is the resulting control input of the outer loop motion controller that is designed for a rigid-body robot model.

By combining (2.56)-(2.58), using the relation

$$\ddot{\boldsymbol{\vartheta}} = \mathbf{K}^{-1}\ddot{\boldsymbol{\tau}} + \ddot{\mathbf{q}} \ , \tag{2.59}$$

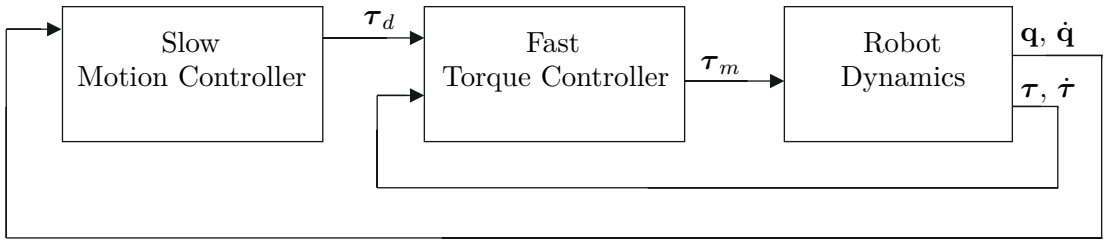


Figure 2.5: Cascade control of a robot with flexible joints.

and setting ε to zero, the closed-loop system for the link dynamics, from the input $\boldsymbol{\tau}_d$ to the link state $(\mathbf{q}, \dot{\mathbf{q}})$, is given by

$$\left[\mathbf{M}(\mathbf{q}) + (\mathbf{I}_n + \mathbf{K}_\tau)^{-1}\mathbf{B}\right]\ddot{\mathbf{q}} + \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \mathbf{g}\left(\mathbf{q}\right) = \boldsymbol{\tau}_d \ , \tag{2.60}$$

where $\mathbf{I}_n$ is the identity matrix. Since the matrix $\mathbf{B}$ and the controller gain $\mathbf{K}_\tau$ are constant, the closed-loop system can be rewritten as

$$(\mathbf{M}(\mathbf{q}) + \tilde{\mathbf{B}})\ddot{\mathbf{q}} + \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \mathbf{g}\left(\mathbf{q}\right) = \boldsymbol{\tau}_d \ , \tag{2.61}$$

where $\tilde{\mathbf{B}}$ is a constant diagonal matrix. Comparing this equation to (2.44) with $\boldsymbol{\tau}_{ext} = \mathbf{0}$ reveals that this approach is equivalent to simply modifying the robot mass matrix $\mathbf{M}(\mathbf{q})$ to $(\mathbf{M}(\mathbf{q}) + \tilde{\mathbf{B}})$.

Since in the case of *KUKA LWR IV+* the robot controller unit, *KRC*[1], is equipped with a joint stiffness compensation, the robot can be modeled and controlled as a rigid-body robot (2.61) by the end-user. This approach was experimentally verified as follows. A computed torque controller based on the extended model (2.61) was used to follow a desired trajectory in the joint space. Choosing the identity matrix for $\tilde{\mathbf{B}}$ resulted in very small control errors in all except for the second joint. Analysis of the elements of the mass matrix $\mathbf{M}(\mathbf{q})$ for different configurations showed that its second diagonal element is comparatively larger than the diagonal elements for the other links. To compensate for this effect, the second diagonal element of $\tilde{\mathbf{B}}$ was set to zero, i. e.

$$\tilde{\mathbf{B}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} , \tag{2.62}$$

which improved the control performance considerably.

[1] *KUKA* Robot Control

3 Path Following Control

When a geometrical curve is to be traversed by the robot end-effector, one can generally employ two types of control strategies. In trajectory tracking control, a trajectory of the end-effector is stabilized. This trajectory specifies the time parametrization of the geometrical curve together with the required velocity. In this case, each point on the curve is passed through at a defined instant of time.

In contrast to that, in path following control the curve has no *a priori* time parametrization and the primary objective is the convergence of the end-effector to the path. One of the advantages of this approach is that the speed of traversal of the path can be specified independently of the path itself.

In this chapter, first a short review of path following control approaches is given. In Section 3.2, the problem is formulated, and the control objectives and admissible paths are discussed. By introducing an orthonormal frame in Section 3.3 a coordinate transformation is performed, which enables linearizing the system dynamics with respect to the path in Section 3.4. Finally, the parametrization of the path is specified.

3.1 Literature Review

A common approach for path following control is the transverse feedback linearization that splits up the motion into dynamics tangential to the path and dynamics orthogonal to the path. This method was introduced by Banaszuk and Hauser in [6], where the transverse dynamics of a single-input nonlinear control-affine system are linearized with respect to a periodic orbit. By stabilizing the transverse dynamics the system output converges asymptotically to the path. In [7], Nielsen and Maggiore determined necessary and sufficient conditions under which a multi-input nonlinear control-affine system is locally transversally feedback linearizable with respect to a given invariant submanifold. Using transverse feedback linearization, path following controllers were designed for a magnetically levitated positioning system in [8], for a quadrotor in [9] and for a laboratory crane in [10]. Hladio *et al.* [11] showed that the tangential dynamics of a mechanical system are also linearizable if the system has enough control inputs such that the tangential dynamics are reachable. In this case, a desired motion on the path can be achieved. The presented method in [11] requires not only a parametrization of the path but also its implicit representation.

Using transverse feedback linearization a path following control method for parametrized paths was proposed in [12]. By constructing an orthonormal frame with respect to the path the tangential and transversal coordinates can be defined. As orthonormal frame the Frenet-Serret frame is used, which requires the path curvature to be non-zero. In [13], Bischof *et al.* use the parallel transport frame from [14] instead to overcome this limitation. In this way, it is possible to design path following controllers for paths including

segments with vanishing curvature. Furthermore, the use of the parallel transport frame also simplifies the path following control law compared to [12].

In order to systematically satisfy the input and state constraints, Faulwasser *et al.* [15] proposed a model predictive control approach to path following problems. In this method the time evolution of the path parameter is considered as an extra constraint to the optimal control problem that is to be solved.

In this work, the approach in [13] is used to design a path following controller for the robot *KUKA LWR IV+*. The transverse feedback linearization is applied to control the position of the end-effector, whereas for the orientation an input-output linearization is used.

3.2 Problem Statement

A path following controller is designed to drive the output of a dynamical system to a desired path in the output space. Furthermore, the motion along the path has to fulfill specified requirements. In this chapter, the dynamic model

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \underbrace{\mathbf{C}\left(\mathbf{q},\dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \mathbf{g}\left(\mathbf{q}\right)}_{\mathbf{n}(\mathbf{q},\dot{\mathbf{q}})} = \boldsymbol{\tau} + \boldsymbol{\tau}_{ext} \ , \tag{3.1}$$

of the rigid-body robot, as derived in Section 2.4, is considered. The system output $\mathbf{y} \in \mathbb{R}^6$ is chosen to be the end-effector pose

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_t \\ \mathbf{y}_r \end{bmatrix} = \begin{bmatrix} \mathbf{h}_t\left(\mathbf{q}\right) \\ \mathbf{h}_r\left(\mathbf{q}\right) \end{bmatrix} = \mathbf{h}\left(\mathbf{q}\right) \ , \tag{3.2}$$

where $\mathbf{y}_t \in \mathbb{R}^3$ is the position of the end-effector in Cartesian coordinates and $\mathbf{y}_r \in \mathbb{R}^3$ is its orientation. The function $\mathbf{h}\left(\mathbf{q}\right)$ is the forward kinematics that maps the joint angles $\mathbf{q}$ to the pose $\mathbf{y}$. The path following controller is used to control the position $\mathbf{y}_t$, and an input-output linearization is applied for the orientation $\mathbf{y}_r$. For the representation of the orientation several possibilities exist. Although Euler angles or roll-pitch-yaw angles are geometrically easy to interpret, they suffer from the problem of representation singularities. Unit quaternions, however, do not have this issue and provide simpler mathematical calculations. A quaternion consists of a scalar μ and a vector part $\boldsymbol{\epsilon}$ in the form

$$Q = \{\mu \quad \boldsymbol{\epsilon}\} = \{\mu \quad \epsilon_1 \quad \epsilon_2 \quad \epsilon_3\} \ . \tag{3.3}$$

To represent the orientation of the end-effector $\mathbf{y}_r$ the vector part of the corresponding unit quaternion is used. Using the normalization condition

$$\|Q\|^2 = \mu^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1 \tag{3.4}$$

the scalar part of the unit quaternion can be readily obtained, if required.

3.2.1 Admissible Paths

For the approach taken in this work, parametrized paths with a path parameter $\vartheta \in \mathcal{T} \subseteq \mathbb{R}$ are used. The path is defined to be

$$\gamma_t = \left\{ \bar{\mathbf{y}}_t \in \mathbb{R}^3 : \bar{\mathbf{y}}_t = \boldsymbol{\sigma}_t(\bar{\vartheta}), \bar{\vartheta} \in \mathcal{T} \right\} , \tag{3.5}$$

where the function $\boldsymbol{\sigma}_t(\vartheta) : \mathcal{T} \to \mathbb{R}^3$ is the parametrization of the path.

The path γ_t has to be a regular curve, meaning that its tangent vector must not be zero, see [16]. Hence, for the path parametrization $\boldsymbol{\sigma}_t(\vartheta)$ one has

$$\boldsymbol{\sigma}_t'(\vartheta) = \left(\frac{\partial \boldsymbol{\sigma}_t}{\partial \vartheta} \right) (\bar{\vartheta}) \neq \mathbf{0} , \qquad \forall \bar{\vartheta} \in \mathcal{T} . \tag{3.6}$$

Moreover, in order for the feedback law (Section 3.4) to be continuous, the path is required to be threefold continuously differentiable, i. e. $\gamma_t \in \mathcal{C}^3$.

3.2.2 Control Objectives

The objectives of the path following control can be formulated as follows, see [8]:

($\mathcal{O}_1$) Asymptotic Convergence to γ_t: the system output $\mathbf{y}_t$ converges asymptotically to the path, i. e.

$$\left\| \mathbf{y}_t \left(t \right) \right\|_{\gamma_t} \to 0 \quad \text{for} \quad t \to \infty , \tag{3.7}$$

where

$$\left\| \mathbf{y}_t(t) \right\|_{\gamma_t} = \inf_{\bar{\mathbf{y}}_t \in \gamma_t} \left\| \mathbf{y}_t(t) - \bar{\mathbf{y}}_t \right\|_2 \tag{3.8}$$

gives the shortest distance of the system output to the path γ_t.

($\mathcal{O}_2$) Invariance Property: If the system state $[\bar{\mathbf{q}}^{\mathrm{T}}(t_0) \quad \dot{\bar{\mathbf{q}}}^{\mathrm{T}}(t_0)]^{\mathrm{T}}$ at time t_0 is an element of the controlled invariant subset Γ^* of

$$\Gamma = \left\{ \begin{bmatrix} \bar{\mathbf{q}} \\ \dot{\bar{\mathbf{q}}} \end{bmatrix} \in \mathbb{R}^{14} : \mathbf{h}_t(\bar{\mathbf{q}}) \in \gamma_t \right\} , \tag{3.9}$$

then one has

$$\left\| \mathbf{y}_t \left(t \right) \right\|_{\gamma_t} = 0 , \qquad \forall t \geq t_0 , \tag{3.10}$$

i. e. the system remains on the path for all $t \geq t_0$.

($\mathcal{O}_3$) Tangential Motion: The motion on the path γ_t meets application-specific requirements, e. g. the direction and the speed of traversing the path can be specified.

3.3 Coordinate Transformation

The motion of the end-effector can be partitioned into tangential and transversal dynamics with respect to the path γ_t. The tangential subsystem describes the motion along the path, whereas the transversal subsystem describes the motion orthogonal to the path. For this purpose, an orthonormal frame must be constructed that is well defined for each point of the path. One way to assign a frame to a curve is the use of Frenet-Serret frame [16] as utilized by Gill *et al.* in [12] in their approach to path following control. The Frenet-Serret formulas require the curve with parametrization $\boldsymbol{\sigma}_t(\vartheta)$ to have second-order regularity, i. e. its first two derivatives with respect to ϑ are linearly independent. Since this condition implies that the curvature of $\boldsymbol{\sigma}_t(\vartheta)$ must not be zero for any $\vartheta \in \mathcal{T}$, the path following control will be limited to paths that do not include straight line segments. Furthermore, using the Frenet-Serret frame the basis vectors orthogonal to the path have a sudden change of sign at inflection points of the curve. This causes the frame to have discontinuities along the curve. To cope with these limitations, the parallel transport frame is used as in [13].

3.3.1 Parallel Transport Frame

For a $\mathcal{C}^2$ curve the parallel transport frame requires the first derivative to be non-zero. The first basis vector is given by the unit tangential vector to the path γ_t

$$\mathbf{e}_{\parallel}(\vartheta) = \frac{\boldsymbol{\sigma}_t'(\vartheta)}{\left\|\boldsymbol{\sigma}_t'(\vartheta)\right\|_2} \,, \tag{3.11}$$

with $\boldsymbol{\sigma}_t'(\vartheta)$ as the tangential vector to the path at ϑ.

To obtain the parallel transport frame, the remaining basis vectors $\mathbf{e}_{\perp}$ and $\mathbf{e}_{\pitchfork}$ are constructed in such a way that their first derivatives are parallel to the first basis vector $\mathbf{e}_{\parallel}$, see [14], i. e.

$$\mathbf{e}_i'(\vartheta) = \kappa_i(\vartheta)\mathbf{e}_{\parallel}(\vartheta)\,, \qquad i \in \{\perp, \pitchfork\} \,, \tag{3.12}$$

with a scalar $\kappa_i(\vartheta)$. Together with the orthonormality conditions, a system of differential-algebraic equations is given for each normal vector $\mathbf{e}_i(\vartheta)$, $i \in \{\perp, \pitchfork\}$

$$\mathbf{e}_i'(\vartheta) = \kappa_i(\vartheta)\mathbf{e}_{\parallel}(\vartheta)\,, \qquad \mathbf{e}_i\left(\vartheta_0\right) = \mathbf{e}_{i,0} \tag{3.13a}$$

$$0 = 1 - \mathbf{e}_i^{\mathrm{T}}(\vartheta)\mathbf{e}_i(\vartheta) \tag{3.13b}$$

$$0 = \mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta)\mathbf{e}_i(\vartheta) \,, \tag{3.13c}$$

where $\mathbf{e}_{i,0}$ is the initial condition that is chosen from the null space of $\mathbf{e}_{\parallel,0}$ and also fulfills (3.13b) and (3.13c). Note that this system of equations is overdetermined since there are 5 scalar equations with the 4 unknowns κ_i and the elements of $\mathbf{e}_i$. To solve this system of equations, first, (3.13c) is differentiated with respect to ϑ, which results in

$$0 = \left(\mathbf{e}_{\parallel}'(\vartheta)\right)^{\mathrm{T}} \mathbf{e}_i(\vartheta) + \mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta)\mathbf{e}_i'(\vartheta) \,. \tag{3.14}$$

Multiplying (3.13a) by $\mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta)$ yields

$$\mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta)\mathbf{e}_i'(\vartheta) = \kappa_i(\vartheta)\underbrace{\mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta)\mathbf{e}_{\parallel}(\vartheta)}_{=1} \ . \tag{3.15}$$

By combining (3.14) and (3.15), $\kappa_i(\vartheta)$ is given by

$$\kappa_i(\vartheta) = -\left(\mathbf{e}_{\parallel}'(\vartheta)\right)^{\mathrm{T}}\mathbf{e}_i(\vartheta) \ , \tag{3.16}$$

which is then plugged into (3.13a) to obtain the ordinary differential equations for $i \in \{\perp, \pitchfork\}$

$$\mathbf{e}_i'(\vartheta) = -\left(\mathbf{e}_{\parallel}'(\vartheta)\right)^{\mathrm{T}}\mathbf{e}_i(\vartheta)\mathbf{e}_{\parallel}(\vartheta) \, , \qquad \mathbf{e}_i\left(\vartheta_0\right) = \mathbf{e}_{i,0} \ . \tag{3.17}$$

For real-time implementation on a digital control system, the result must be in discrete time form. As shown in [13], the overdetermined system of equations (3.13) can be discretized in time and then solved using a least-squares approach. The optimal solution for the first normal vector $\mathbf{e}_{\perp}$ is given by

$$\mathbf{e}_{\perp,k}^{*} = \frac{\mathbf{e}_{\perp,k-1}^{*} - \mathbf{e}_{\parallel,k}^{\mathrm{T}}\mathbf{e}_{\perp,k-1}^{*}\mathbf{e}_{\parallel,k}}{\sqrt{1 - \left(\mathbf{e}_{\parallel,k}^{\mathrm{T}}\mathbf{e}_{\perp,k-1}^{*}\right)^2}} \, , \qquad \mathbf{e}_{\perp,1} = \mathbf{e}_{\perp}\left(\vartheta_0\right) \ . \tag{3.18}$$

for the time steps $k = 2, 3, \ldots$. While a similar equation also holds for the second normal vector $\mathbf{e}_{\pitchfork}$, it is simpler to calculate $\mathbf{e}_{\pitchfork}$ using the cross product

$$\mathbf{e}_{\pitchfork}(\vartheta) = \mathbf{e}_{\parallel}(\vartheta) \times \mathbf{e}_{\perp}(\vartheta) \ , \tag{3.19}$$

which is a consequence of the orthogonality condition of the basis vectors.

3.3.2 Orthogonal Projection

In general, the system output is not located on the path. By means of the optimization problem

$$\vartheta^{*} = P_{\mathcal{T}}\left(\mathbf{y}_t\right) = \underset{\vartheta \in \mathcal{T}}{\arg\min} \left\| \mathbf{y}_t - \boldsymbol{\sigma}_t\left(\vartheta\right) \right\|_2^2 \tag{3.20}$$

an orthogonal projection $P_{\mathcal{T}}\left(\mathbf{y}_t\right)$ is defined, which determines the closest point $\mathbf{y}_t^{*} = \boldsymbol{\sigma}_t(\vartheta^{*})$ on the path γ_t to the system output $\mathbf{y}_t$, see Figure 3.1. The first-order necessary condition and the second-order sufficient condition for optimality of this optimization problem are given by

$$\left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^{*})\right)^{\mathrm{T}}\boldsymbol{\sigma}_t'(\vartheta^{*}) = 0 \ , \tag{3.21}$$

and

$$\left\|\boldsymbol{\sigma}_t'(\vartheta^{*})\right\|_2^2 - \left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^{*})\right)^{\mathrm{T}}\boldsymbol{\sigma}_t''(\vartheta^{*}) > 0 \ , \tag{3.22}$$

respectively. By introducing

$$\alpha(\mathbf{y}_t) = \frac{\left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\right)^{\mathrm{T}} \boldsymbol{\sigma}_t''(\vartheta^*)}{\left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2^2} , \tag{3.23}$$

and using (3.22) a feasible neighborhood of the path can be defined as

$$\mathcal{Y}_t = \{\bar{\mathbf{y}}_t \in \mathbb{R}^3 : \alpha\left(\bar{\mathbf{y}}_t\right) < 1\} . \tag{3.24}$$

If the system output is outside this feasible neighborhood, no unique path parameter ϑ^* can be obtained.

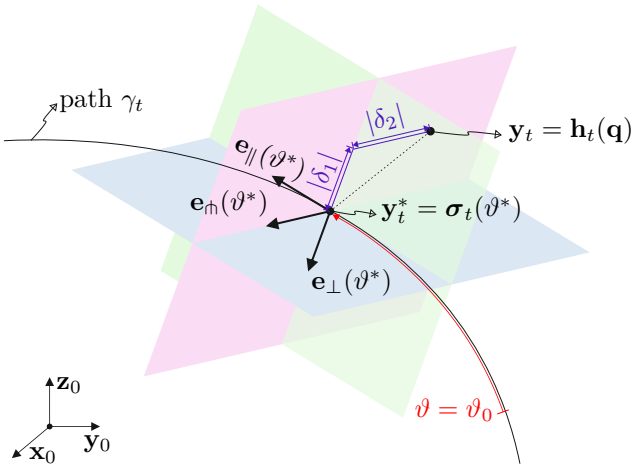


Figure 3.1: Orthonormal frame at the point $\boldsymbol{\sigma}_t(\vartheta^*)$ of the path γ_t.

In the following, the parallel transport frame is used together with the optimal path parameter ϑ^* to perform a coordinate transformation.

3.3.3 Tangential Subsystem

As proposed in [13], for the first tangential coordinate the arc length

$$\eta_1 = g\left(\mathbf{y}_t\right) = \int_{\vartheta_0}^{\vartheta^*} \left\|\boldsymbol{\sigma}_t'\left(\tau\right)\right\|_2 \mathrm{d}\tau \tag{3.25}$$

is chosen, which is the traversed length along the curve from the starting point $\boldsymbol{\sigma}_t(\vartheta_0)$ to the current point $\boldsymbol{\sigma}_t(\vartheta^*)$, with ϑ^* according to (3.20). The second tangential coordinate is the time derivative of the arc length

$$\eta_2 = \dot{\eta}_1 = \left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2 \dot{\vartheta}^* , \tag{3.26}$$

with the time derivative of the optimal path parameter $\dot{\vartheta}^*$, which is calculated as follows. Differentiating the first order optimality condition (3.21) with respect to time yields

$$\left(\dot{\mathbf{y}}_t - \boldsymbol{\sigma}_t'(\vartheta^*)\dot{\vartheta}^*\right)^{\mathrm{T}} \boldsymbol{\sigma}_t'(\vartheta^*) + \left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\right)^{\mathrm{T}} \boldsymbol{\sigma}_t''(\vartheta^*)\dot{\vartheta}^* = 0 . \tag{3.27}$$

Using (3.23) one has

$$\dot{\mathbf{y}}_t^{\mathrm{T}}\boldsymbol{\sigma}_t'(\vartheta^*) - \dot{\vartheta}^* \underbrace{\left(\boldsymbol{\sigma}_t'(\vartheta^*)\right)^{\mathrm{T}}\boldsymbol{\sigma}_t'(\vartheta^*)}_{\left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2^2} + \alpha(\mathbf{y}_t)\left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2^2 \dot{\vartheta}^* = 0 \ . \tag{3.28}$$

This equation can be solved for $\dot{\vartheta}^*$ to obtain

$$\dot{\vartheta}^* = \frac{\dot{\mathbf{y}}_t^{\mathrm{T}}\boldsymbol{\sigma}_t'(\vartheta^*)}{(1-\alpha(\mathbf{y}_t))\left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2^2} = \frac{\dot{\mathbf{y}}_t^{\mathrm{T}}\mathbf{e}_{\parallel}(\vartheta^*)}{(1-\alpha(\mathbf{y}_t))\left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2} \ , \tag{3.29}$$

where (3.11) is used. Finally, by introducing

$$\beta(\mathbf{y}_t) = \frac{1}{1-\alpha(\mathbf{y}_t)} \ , \tag{3.30}$$

the time derivative of the optimal path parameter $\dot{\vartheta}^*$ can be rewritten as

$$\dot{\vartheta}^* = \frac{\beta(\mathbf{y}_t)\mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta^*)}{\left\|\boldsymbol{\sigma}_t'(\vartheta^*)\right\|_2}\dot{\mathbf{y}}_t \ . \tag{3.31}$$

Hence, the second tangential coordinate η_2 from (3.26) is given by

$$\eta_2 = \underbrace{\beta(\mathbf{y}_t)\mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta^*)}_{(\nabla g)^{\mathrm{T}}}\underbrace{\dot{\mathbf{y}}_t}_{\nabla\mathbf{h}_t\dot{\mathbf{q}}} \ , \tag{3.32}$$

with the gradient $(\nabla g)^{\mathrm{T}} = \partial g/\partial \mathbf{y}_t$ and the Jacobian matrix $\nabla\mathbf{h}_t = \partial\mathbf{h}_t/\partial\mathbf{q}$.

3.3.4 Transversal Subsystem

According to (3.21), the difference $(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*))$ determines the component of the system output $\mathbf{y}_t$ that is locally orthogonal to the path γ_t. As proposed in [13], by projecting this term onto the basis vectors $\mathbf{e}_{\perp}$ and $\mathbf{e}_{\pitchfork}$ of the local coordinate frame, the transversal coordinates

$$\xi_1 = \delta_1(\mathbf{y}_t) = \mathbf{e}_{\perp}^{\mathrm{T}}(\vartheta^*)\left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\right) \tag{3.33}$$

and

$$\xi_3 = \delta_2(\mathbf{y}_t) = \mathbf{e}_{\pitchfork}^{\mathrm{T}}(\vartheta^*)\left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\right) \tag{3.34}$$

are defined, see Figure 3.1. It is obvious that if the output is on the path, the transversal coordinates ξ_1 and ξ_3 are zero.

Differentiation of ξ_1 and ξ_3 with respect to time yields

$$\xi_2 = \dot{\xi}_1 = \underbrace{\left(\mathbf{e}_{\perp}'(\vartheta^*)\right)^{\mathrm{T}}\dot{\vartheta}^*\left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\right)}_{\overset{(3.13a)}{=}0} - \underbrace{\mathbf{e}_{\perp}^{\mathrm{T}}(\vartheta^*)\boldsymbol{\sigma}_t'(\vartheta^*)\dot{\vartheta}^*}_{\overset{(3.13c)}{=}0} + \underbrace{\mathbf{e}_{\perp}^{\mathrm{T}}(\vartheta^*)}_{(\nabla\delta_1)^{\mathrm{T}}}\underbrace{\dot{\mathbf{y}}_t}_{\nabla\mathbf{h}_t\dot{\mathbf{q}}} \ , \tag{3.35}$$

and

$$\xi_4 = \dot{\xi}_3 = \underbrace{\left(\mathbf{e}'_{\text{⫛}}(\vartheta^*)\right)^{\mathrm{T}} \dot{\vartheta}^* \left(\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\right)}_{\overset{(3.13a)}{=}0} - \underbrace{\mathbf{e}_{\text{⫛}}^{\mathrm{T}}(\vartheta^*)\boldsymbol{\sigma}'_t(\vartheta^*)\dot{\vartheta}^*}_{\overset{(3.13c)}{=}0} + \underbrace{\mathbf{e}_{\text{⫛}}^{\mathrm{T}}(\vartheta^*)}_{\left(\nabla\delta_2\right)^{\mathrm{T}}} \underbrace{\dot{\mathbf{y}}_t}_{\nabla\mathbf{h}_t\dot{\mathbf{q}}} \ . \tag{3.36}$$

with the gradients $(\nabla\delta_1)^{\mathrm{T}} = \partial\delta_1/\partial\mathbf{y}_t$ and $(\nabla\delta_2)^{\mathrm{T}} = \partial\delta_2/\partial\mathbf{y}_t$ and the Jacobian matrix $\nabla\mathbf{h}_t$ of the output $\mathbf{y}_t$. The first and second term in (3.35) and (3.36) vanish as a consequence of the use of the parallel transport frame with (3.13a) and (3.13c) and due to (3.21).

3.3.5 Rotational Subsystem

The rotational part of the system output $\mathbf{y}_r$ and its time derivative are chosen for the first and second rotational coordinates $\boldsymbol{\zeta}_1$ and $\boldsymbol{\zeta}_2$

$$\boldsymbol{\zeta}_1 = \mathbf{y}_r = \mathbf{h}_r\left(\mathbf{q}\right) \ , \tag{3.37}$$

$$\boldsymbol{\zeta}_2 = \dot{\boldsymbol{\zeta}}_1 = \dot{\mathbf{y}}_r = \nabla\mathbf{h}_r\dot{\mathbf{q}} \ , \tag{3.38}$$

with the Jacobian matrix $\nabla\mathbf{h}_r$ of the output $\mathbf{y}_r$.

3.3.6 Diffeomorphism

Equations (3.25) and (3.32)-(3.38) define a $\mathcal{C}^1$-diffeomorphism $\boldsymbol{\Phi}$, which establishes a mapping between the generalized coordinates $\mathbf{q}$ and velocities $\dot{\mathbf{q}}$ and the virtual output $\hat{\mathbf{y}}^{\mathrm{T}} = \hat{\mathbf{h}}^{\mathrm{T}}\left(\mathbf{q}\right) = [\eta_1 \quad \xi_1 \quad \xi_3 \quad \boldsymbol{\zeta}_1^{\mathrm{T}}]$ in the form

$$\begin{bmatrix}\hat{\mathbf{y}}\\ \dot{\hat{\mathbf{y}}}\end{bmatrix} = \begin{bmatrix}\eta_1\\ \xi_1\\ \xi_3\\ \boldsymbol{\zeta}_1\\ \eta_2\\ \xi_2\\ \xi_4\\ \boldsymbol{\zeta}_2\end{bmatrix} = \begin{bmatrix} g\circ\mathbf{h}_t\left(\mathbf{q}\right)\\ \delta_1\circ\mathbf{h}_t\left(\mathbf{q}\right)\\ \delta_2\circ\mathbf{h}_t\left(\mathbf{q}\right)\\ \mathbf{h}_r\left(\mathbf{q}\right)\\ \left(\nabla g\right)^{\mathrm{T}}\nabla\mathbf{h}_t\,\dot{\mathbf{q}}\\ \left(\nabla\delta_1\right)^{\mathrm{T}}\nabla\mathbf{h}_t\,\dot{\mathbf{q}}\\ \left(\nabla\delta_2\right)^{\mathrm{T}}\nabla\mathbf{h}_t\,\dot{\mathbf{q}}\\ \nabla\mathbf{h}_r\,\dot{\mathbf{q}}\end{bmatrix} = \boldsymbol{\Phi}\left(\mathbf{q},\dot{\mathbf{q}}\right) \ . \tag{3.39}$$

For a proof that the mapping (3.39) is a diffeomorphism, see [13]. Using this coordinate transformation the pose of the end-effector and its time derivative are transformed to the corresponding path-related coordinates. This allows for a separation of the dynamics into tangential, transversal and rotational subsystems.

The Jacobian of $\boldsymbol{\Phi}$ is calculated to be

$$\nabla\boldsymbol{\Phi} = \begin{bmatrix}\hat{\mathbf{J}}\left(\mathbf{q}\right) & \mathbf{0}\\ * & \hat{\mathbf{J}}\left(\mathbf{q}\right)\end{bmatrix} \ , \tag{3.40}$$

Herein, the matrix $\hat{\mathbf{J}}(\mathbf{q})$ is the transformation of the Jacobian matrix $\mathbf{J}(\mathbf{q})$ of the system output $\mathbf{y}$ with respect to the path, i. e.

$$\hat{\mathbf{J}}(\mathbf{q}) = \mathbf{L}(\mathbf{q})\,\mathbf{J}(\mathbf{q}) \ , \tag{3.41}$$

with the matrices

$$\mathbf{L}(\mathbf{q}) = \begin{bmatrix} \mathbf{E}(\mathbf{q}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} , \qquad \mathbf{E}(\mathbf{q}) = \begin{bmatrix} \beta(\mathbf{y}_t)\mathbf{e}_{\parallel}^{\mathrm{T}} \\ \mathbf{e}_{\perp}^{\mathrm{T}} \\ \mathbf{e}_{\pitchfork}^{\mathrm{T}} \end{bmatrix} , \tag{3.42}$$

where $\mathbf{I}_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix. Analyzing equations (3.32), (3.35), (3.36) and (3.38) shows that the matrix $\hat{\mathbf{J}}(\mathbf{q})$ relates the time derivative of the virtual output $\dot{\hat{\mathbf{y}}}$ to the generalized velocities $\dot{\mathbf{q}}$ in the form

$$\dot{\hat{\mathbf{y}}} = \begin{bmatrix} \dot{\eta}_1 \\ \dot{\xi}_1 \\ \dot{\xi}_3 \\ \dot{\boldsymbol{\zeta}}_1 \end{bmatrix} = \hat{\mathbf{J}}(\mathbf{q})\,\dot{\mathbf{q}} \ . \tag{3.43}$$

This equation is the counterpart of

$$\dot{\mathbf{y}} = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{3.44}$$

for the original output $\mathbf{y}$.

3.4 Feedback Linearization

Using the equations (3.32), (3.35), (3.36) and (3.38) the second-order time derivative of the tangential, transversal and rotational coordinates is calculated to be

$$\ddot{\eta}_1 = \left((\nabla\beta)^{\mathrm{T}}\,\dot{\mathbf{y}}_t \mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta^*) + \beta(\mathbf{y}_t)\left(\mathbf{e}_{\parallel}'(\vartheta^*)\right)^{\mathrm{T}}\dot{\vartheta}^* \right)\dot{\mathbf{y}}_t + \beta(\mathbf{y}_t)\mathbf{e}_{\parallel}^{\mathrm{T}}(\vartheta^*)\ddot{\mathbf{y}}_t \tag{3.45a}$$

$$\ddot{\xi}_1 = \left(\mathbf{e}_{\perp}'(\vartheta^*)\right)^{\mathrm{T}}\dot{\vartheta}^*\dot{\mathbf{y}}_t + \mathbf{e}_{\perp}^{\mathrm{T}}(\vartheta^*)\ddot{\mathbf{y}}_t \tag{3.45b}$$

$$\ddot{\xi}_3 = \left(\mathbf{e}_{\pitchfork}'(\vartheta^*)\right)^{\mathrm{T}}\dot{\vartheta}^*\dot{\mathbf{y}}_t + \mathbf{e}_{\pitchfork}^{\mathrm{T}}(\vartheta^*)\ddot{\mathbf{y}}_t \tag{3.45c}$$

$$\ddot{\boldsymbol{\zeta}}_1 = \ddot{\mathbf{y}}_r \ , \tag{3.45d}$$

with $(\nabla\beta)^{\mathrm{T}} = \partial\beta(\mathbf{y}_t)/\partial\mathbf{y}_t$. This is equivalent to differentiating (3.43) with respect to time to obtain

$$\ddot{\hat{\mathbf{y}}} = \begin{bmatrix} \ddot{\eta}_1 \\ \ddot{\xi}_1 \\ \ddot{\xi}_3 \\ \ddot{\zeta}_1 \end{bmatrix} = \dot{\hat{\mathbf{J}}}(\mathbf{q},\dot{\mathbf{q}})\,\dot{\mathbf{q}} + \hat{\mathbf{J}}(\mathbf{q})\,\ddot{\mathbf{q}} \ . \tag{3.46}$$

Using (3.1) the generalized accelerations $\ddot{\mathbf{q}}$ are given by

$$\ddot{\mathbf{q}} = \mathbf{M}^{-1}(\mathbf{q})(\boldsymbol{\tau} + \boldsymbol{\tau}_{ext} - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}})) \ , \tag{3.47}$$

so that (3.46) can be rewritten as

$$\ddot{\hat{\mathbf{y}}} = \dot{\hat{\mathbf{J}}}\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \hat{\mathbf{J}}\left(\mathbf{q}\right)\left(\mathbf{M}^{-1}(\mathbf{q})(\boldsymbol{\tau} + \boldsymbol{\tau}_{ext} - \mathbf{n}(\mathbf{q}, \dot{\mathbf{q}}))\right) \ . \tag{3.48}$$

It can be shown that the use of the feedback transformation

$$\boldsymbol{\tau} = \mathbf{n}\left(\mathbf{q}, \dot{\mathbf{q}}\right) - \boldsymbol{\tau}_{ext} + \mathbf{M}(\mathbf{q})\hat{\mathbf{J}}^{\dagger}\left(\mathbf{q}\right)\left(\mathbf{v} - \dot{\hat{\mathbf{J}}}\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}}\right) \tag{3.49}$$

yields a linear input-output relation from the virtual control input $\mathbf{v}$ to the virtual output $\hat{\mathbf{y}}$, which is expressed as 6 integrator chains of length two

$$\ddot{\hat{\mathbf{y}}} = \mathbf{v} \ , \tag{3.50}$$

where the virtual control input is

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_t \\ \mathbf{v}_r \end{bmatrix} = \begin{bmatrix} v_{\parallel} \\ v_{\perp} \\ v_{\pitchfork} \\ \mathbf{v}_r \end{bmatrix} \ . \tag{3.51}$$

Note that in (3.49) the generalized inverse of the matrix $\hat{\mathbf{J}}\left(\mathbf{q}\right)$,

$$\hat{\mathbf{J}}^{\dagger}\left(\mathbf{q}\right) = \hat{\mathbf{J}}^{\mathrm{T}}\left(\mathbf{q}\right)\left(\hat{\mathbf{J}}\left(\mathbf{q}\right)\hat{\mathbf{J}}^{\mathrm{T}}\left(\mathbf{q}\right)\right)^{-1} \ , \tag{3.52}$$

is used since the original system (3.1) has $n = 7$ control inputs, whereas only 6 virtual inputs exist. In order to stabilize the self-motion of the robot, see, e. g., [17], a term $\boldsymbol{\tau}_N$

$$\boldsymbol{\tau}_N = \left(\mathbf{I}_7 - \hat{\mathbf{J}}^{\dagger}\left(\mathbf{q}\right)\hat{\mathbf{J}}\left(\mathbf{q}\right)\right)\left(-\mathbf{K}_1\dot{\mathbf{q}} - \mathbf{K}_0\mathbf{q}\right) \tag{3.53}$$

is added to the control law (3.49), i. e.

$$\boldsymbol{\tau} = \mathbf{n}\left(\mathbf{q}, \dot{\mathbf{q}}\right) - \boldsymbol{\tau}_{ext} + \mathbf{M}(\mathbf{q})\hat{\mathbf{J}}^{\dagger}\left(\mathbf{q}\right)\left(\mathbf{v} - \dot{\hat{\mathbf{J}}}\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}}\right) + \boldsymbol{\tau}_N \ , \tag{3.54}$$

with the identity matrix $\mathbf{I}_7$ and positive definite diagonal matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ as PD controller gains.

With the help of the virtual control inputs the fulfillment of the objectives of the path following control can be discussed. Due to the linear relation between the inputs $v_{\perp}$ and $v_{\pitchfork}$ and the transversal states $(\boldsymbol{\xi}, \dot{\boldsymbol{\xi}})$ according to (3.50) and because of the reachability of this linear system, a linear state controller can be used to asymptotically stabilize the transversal subsystem. Thus, the asymptotic convergence to $\boldsymbol{\sigma}_t(\vartheta)$ can be guaranteed and objective $(\mathcal{O}_1)$ is fulfilled.

By setting the transversal coordinates ξ_1 and ξ_3 and their time derivatives to zero, i. e.

$$\boldsymbol{\Phi}_\xi(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \xi_1 \\ \xi_3 \\ \xi_2 \\ \xi_4 \end{bmatrix} = \begin{bmatrix} \delta_1 \circ \mathbf{h}_t(\mathbf{q}) \\ \delta_2 \circ \mathbf{h}_t(\mathbf{q}) \\ (\nabla\delta_1)^{\mathrm{T}} \nabla\mathbf{h}_t \, \dot{\mathbf{q}} \\ (\nabla\delta_2)^{\mathrm{T}} \nabla\mathbf{h}_t \, \dot{\mathbf{q}} \end{bmatrix} = \mathbf{0} \ , \tag{3.55}$$

the controlled invariant subset $\Gamma^* = \left\{ [\bar{\mathbf{q}}^{\mathrm{T}} \quad \dot{\bar{\mathbf{q}}}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{14} \, : \, \boldsymbol{\Phi}_\xi(\bar{\mathbf{q}}, \dot{\bar{\mathbf{q}}}) = \mathbf{0} \right\}$ is found. Elements of this subset have the property that their corresponding system output is on the path, and its velocity is tangential to the path. If at t_0 the system state is an element of Γ^*, then by choosing $v_\perp = 0$ and $v_{\pitchfork} = 0$ it can be achieved that

$$\xi_i = 0 \, , \qquad i = 1, 2, 3, 4 \tag{3.56a}$$

$$\|\mathbf{y}_t\|_{\gamma_t} = 0 \, , \qquad \forall t > t_0 \ , \tag{3.56b}$$

i. e. the system output remains on the path and objective $(\mathcal{O}_2)$ is satisfied.

Since the virtual input $v_\parallel$ is in the tangential direction, it allows to control the motion along the path and objective $(\mathcal{O}_3)$ is fulfilled.

Control Law

The control law for the tangential and transversal coordinates is designed as a PD state controller combined with a feedforward part as follows

$$\mathbf{v}_t = \begin{bmatrix} \ddot{\eta}_1^d - a_{\eta,1}\dot{e}_\eta - a_{\eta,0}e_\eta \\ \ddot{\xi}_1^d - a_{\xi,1}\dot{e}_{\xi_1} - a_{\xi,0}e_{\xi_1} \\ \ddot{\xi}_3^d - a_{\xi,1}\dot{e}_{\xi_3} - a_{\xi,0}e_{\xi_3} \end{bmatrix} \ , \tag{3.57}$$

where the desired values are denoted by the superscript d, and the errors between the actual and desired values are given by

$$e_\eta = \eta_1 - \eta_1^d \tag{3.58a}$$

$$e_{\xi_i} = \xi_i - \xi_i^d \, , \qquad i = 1, 3 \ . \tag{3.58b}$$

The controller gains, $a_{i,j} > 0$ with $i \in \{\eta, \xi\}$ and $j = 0, 1$, are obtained using pole placement so that the characteristic polynomial of the closed-loop system is guaranteed to be a Hurwitz polynomial and thus, the error dynamics are exponentially stable.

Although the use of an integral action in the controller has the advantage of making the steady-state control error zero for constant disturbances, it can cause an undesirable limit cycle in combination with stick-slip friction.

For the rotational coordinates a PD controller is employed in the form

$$\mathbf{v}_r = -a_{r,0}\mathbf{e}_r - a_{r,1}\dot{\mathbf{y}}_r \ , \tag{3.59}$$

with the positive controller gains $a_{r,0}$ and $a_{r,1}$. Note that for the D-action not the time derivative of the error $\dot{\mathbf{e}}_r$ but the angular velocity

$$\dot{\mathbf{y}}_r = \mathbf{J}^7_{\omega,0}(\mathbf{q})\dot{\mathbf{q}} \tag{3.60}$$

is used. For the orientation error $\mathbf{e}_r$, quaternions are used as described in the following. Using rotation matrices the error between the desired end-effector orientation $\mathbf{R}_d$ and its current orientation $\mathbf{R}(\mathbf{q})$ can be defined as $\mathbf{R}(\mathbf{q})\mathbf{R}_d^{\mathrm{T}}$. With the help of the corresponding quaternions Q_d and $Q(\mathbf{q})$, the error can be expressed as

$$\Delta Q = Q(\mathbf{q}) \otimes Q_d^{-1} \ . \tag{3.61}$$

By means of the calculation rules of quaternions $Q = \{\mu \quad \boldsymbol{\epsilon}\}$,

$$Q^{-1} = \{\mu \quad -\boldsymbol{\epsilon}\} \tag{3.62a}$$

$$Q_1 \otimes Q_2 = \{\mu_1\mu_2 - \boldsymbol{\epsilon}_1^{\mathrm{T}}\boldsymbol{\epsilon}_2, \quad \mu_1\boldsymbol{\epsilon}_2 + \mu_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2\} \ , \tag{3.62b}$$

it can be shown, see, e. g., [18], that $\Delta Q = \{1, \mathbf{0}\}$, if and only if the desired and the actual orientation are aligned, i. e. $\mathbf{R}_d$ and $\mathbf{R}$ are equal. Therefore, the orientation error of the end-effector can be regarded as the vector part of the error in (3.61), i. e.

$$\mathbf{e}_r = \Delta\boldsymbol{\epsilon} = -\mu(\mathbf{q})\boldsymbol{\epsilon}_d + \mu_d\boldsymbol{\epsilon}(\mathbf{q}) - \boldsymbol{\epsilon}(\mathbf{q}) \times \boldsymbol{\epsilon}_d \ . \tag{3.63}$$

3.5 Path Parametrization

As discussed in Section 3.2.1 the path is required to be threefold continuously differentiable in order to guarantee continuous control inputs $\boldsymbol{\tau}$. Hence, quartic splines, which are of class $\mathcal{C}^3$, can be used as a desired path. Quartic splines are piecewise polynomial functions of order four, which are used to interpolate between given points. At the interpolation points the adjacent curve segments have to satisfy continuity conditions for their function values and the value of their first three derivatives. For details on spline interpolation see, e. g., [19] and [20]. By constructing the quartic spline, the path is given as a segmented parametrized curve

$$\boldsymbol{\sigma}(\vartheta) = \begin{cases} \boldsymbol{\sigma}_0(\vartheta)\,, & \vartheta_0 \le \vartheta < \vartheta_1 \\ \boldsymbol{\sigma}_1(\vartheta)\,, & \vartheta_1 \le \vartheta < \vartheta_2 \\ \vdots & \\ \boldsymbol{\sigma}_{l-1}(\vartheta)\,, & \vartheta_{l-1} \le \vartheta < \vartheta_l \end{cases} , \tag{3.64}$$

consisting of l path segments

$$\boldsymbol{\sigma}_s(\vartheta) = \begin{bmatrix} \sigma_{s,x}(\vartheta) \\ \sigma_{s,y}(\vartheta) \\ \sigma_{s,z}(\vartheta) \end{bmatrix} = \sum_{j=0}^{4} \begin{bmatrix} a_{j,s}(\vartheta - \vartheta_s)^j \\ b_{j,s}(\vartheta - \vartheta_s)^j \\ c_{j,s}(\vartheta - \vartheta_s)^j \end{bmatrix} , \qquad s = 0, \ldots, l-1 \ , \tag{3.65}$$

with the polynomial coefficients $a_{j,s}$, $b_{j,s}$, $c_{j,s}$ for $j = 0, \ldots, 4$.

In a further step the path parameter ϑ is normalized by dividing by its maximum value ϑ_{max}, i. e.

$$\vartheta_n = \frac{\vartheta}{\vartheta_{max}} = \frac{\vartheta}{\vartheta_l} \ , \tag{3.66}$$

such that $\vartheta_n \in [0, 1]$. Note that additionally, the polynomial coefficients of the spline must be modified. The coefficients of the normalized curve $\boldsymbol{\sigma}_n(\vartheta_n)$ are calculated to be

$$a_{j,s,n} = \vartheta_{max}^j a_{j,s} \tag{3.67a}$$

$$b_{j,s,n} = \vartheta_{max}^j b_{j,s} \tag{3.67b}$$

$$c_{j,s,n} = \vartheta_{max}^j c_{j,s} \ , \tag{3.67c}$$

where $j = 0, \ldots, 4$.

4 Robotic Winding of a Rope

The procedure of winding a rope around objects in the workspace of a robot can be divided into 3 subtasks. First, the end-effector is to be controlled to follow a defined path. This is achieved using the path following control described in the previous chapter. Secondly, the orientation of the end-effector has to be specified depending on the orientation of the objects to be wound around. Thirdly, during the winding process the tension of the rope is required to be controlled.

In this chapter, after a literature review the problem is described in more detail in Section 4.2. Then, in Section 4.3 the experimental setup is explained. In Section 4.4, it is shown how the desired path is designed. The desired orientation of the end-effector is derived in Section 4.5. In Section 4.6, the motion along the path is discussed. The chapter concludes by presenting simulation and experimental results.

4.1 Literature Review

The use of robots for handling linear flexible objects, such as ropes, has been investigated in various works. In [21] a method is proposed for placing a rope in a target shape on a tabletop using a dual-arm robot and making a clove hitch in a further step. Vinh *et al.* analyzed the motion required to tie a knot by a human and extracted key points which are used for a point-to-point motion control of a robot arm to tie a knot, see [22]. In [23] and [24] the dynamic knotting of a flexible rope is investigated by developing a rope deformation model. Starting from the multi-link approximation of the rope, its equations of motion can be replaced by an algebraic equation under the constraint of the high-speed motion of a robot arm. The model is used for motion planning for tying a knot.

Using industrial robots Hultman *et al.* presented robotized cable winding of the stators of electric machines, see [25]. In a robot cell two industrial robots cooperate to wind cables in the stator slots in a predefined pattern.

In [26] an architectural structure is built using two synchronized industrial robots. Carbon fiber is spanned between defined control points to complete the desired structure over a steel frame that is held and moved by the robots. The fiber is delivered by a spool that is placed on the floor. In contrast to that, in [27] the fiber spool is part of the end-effector of a single robot. Using an adhesive the fiber is placed on the inner surface of a membrane that surrounds the robot. Here, the contact force between the newly applied fiber and the placed fiber is controlled to avoid pulling off the placed fibers and also deforming the membrane. For this force control the robot is allowed to deviate from the desired path by online modification of the original path.

In the present work, a rope is wound around arbitrarily oriented rods that are placed within the workspace of a robot. During the winding process the tension of the rope is controlled to be constant.

4.2 Problem Statement

Given some rods in the workspace of the robot, a path has to be defined for the winding process. This path is traversed by the robot end-effector using the path following control algorithm discussed in Chapter 3. Having a rope spool mounted on the robot end-effector and fixing the rope end, the rope is wound around the rods when the path is traversed.

Another objective is to ensure a constant rope tension during the winding. This is achieved by a motor that drives the rope spool. By controlling the motor torque the rope tension is kept constant. One of the requirements for choosing the motor is the capability of having forces close to the maximum force that the robot can withstand. On the other hand, the motor itself has to be as small and light as possible. A high force can be achieved using a small motor if a gear is used. However, this adds additional friction that is difficult to control.

One of the challenges during the winding process is to specify the orientation of the end-effector. That is, for arbitrarily oriented rods the end-effector orientation must be appropriately defined. This is covered in detail in Section 4.5.

Figure 4.1 shows the desired result of rope winding by the robot.

(a) side view (b) top view

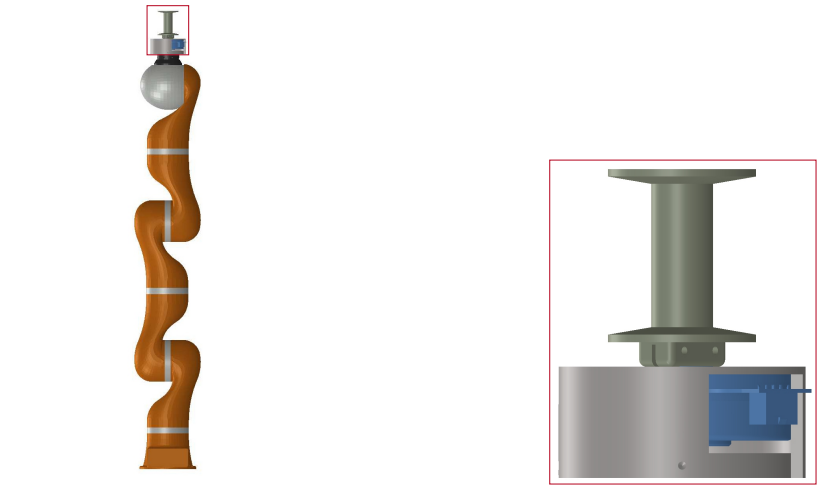
Figure 4.1: Rope wound by the robot around three rods.

4.3 Experimental Setup

As a trade-off between the aforementioned requirements for the motor a direct drive electronically commutated motor is employed. The motor *EC 90 Flat* from *Maxon Motor* has a high torque density and can develop torques up to 560 mNm. Because of its external rotor, it cannot be directly mounted on the robot. For that purpose a suitable motor housing had to be designed and constructed.

Figure 4.2 shows the robot together with the end-effector. The components of the

end-effector are illustrated in Figure 4.3 in detail. The lower part of the motor housing is mounted on the mounting flange of the robot and is laterally fixed with the upper part of the case. The upper part encloses the motor and is fixed with its upper face. The spool is directly attached to the motor shaft.



(a) Robot with end-effector. (b) End-effector.

Figure 4.2: 3-D view of the robot together with the end-effector.

Due to the construction of the end-effector it was possible to use the motor torque controller provided by the manufacturer for controlling the rope tension. Given the desired rope tension force F_r^d, the required motor torque τ_m^d is calculated to be

$$\tau_m^d = r_a F_r^d \ , \tag{4.1}$$

where r_a is the average radius of the rope spool. Here, the following simplifying assumptions were made:

- The rope is rigid, and thus the deformation and dynamics of the rope are not modeled.
- It is assumed that the rope layers do not change the distance between the rope contact point and the motor shaft axis considerably. Hence, the average radius can be used for calculating the motor torque from the desired rope tension force.
- The stretched rope is orthogonal to the spool axis.

4.4 Path Recording

A teach-in procedure is performed to record the desired path. By compensating the gravitational forces and friction of the robot it is possible to easily move the robot joints to achieve the desired poses. The Cartesian position values of the end-effector along the path are stored. This results in a large number of data points. Since in a next step a

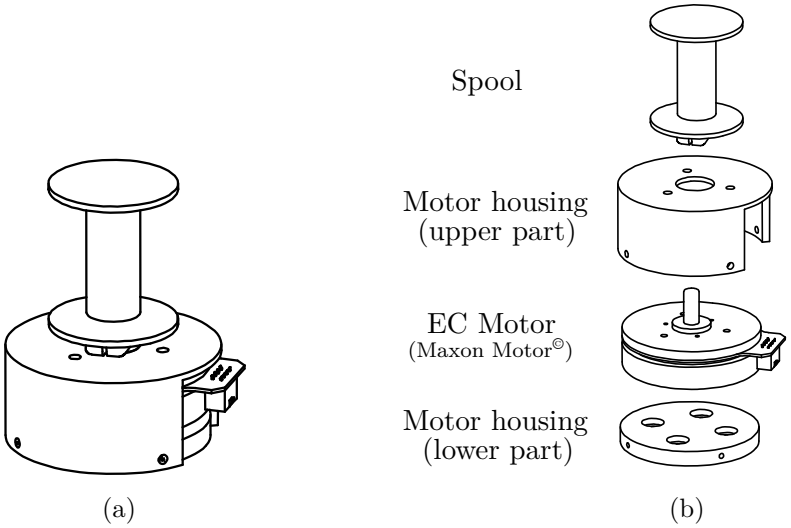


Figure 4.3: 3-D sketch of the end-effector: (a) Assembled end-effector. (b) Parts of the end-effector.

quartic spline interpolation is to be performed between the points, only a limited number of points that roughly describe the desired path are selected.

As no collision monitoring is employed, the path has to be constructed in such a way that the spool does not collide with the stretched rope. Hence, the user has to perform the teach-in of the path with the mounted end-effector and rope so that possible collisions can be observed and avoided.

In the post-processing of the recorded path, the fact that the spool orientation is directly derived from the path must be especially taken into consideration. To this end, parts of the path containing small back and forth motions are manually modified since they would cause undesirable changes in the spool orientation. For the path shown in Figure 4.4 this is done by flattening the part of the path around the left vertical rod, i. e. the z-coordinates are set to an equal value. The torsion in the end part of the path is not flattened since this would cause a collision of the spool with the stretched rope.

4.5 Desired Orientation of the End-effector

When winding a rope around a rod, the orientation of the spool must be appropriately controlled using the control law derived in the previous chapter. In the case where all the rods have the same orientation, a single spool orientation that is antiparallel to the rod orientation can be kept constant during the whole winding process. However, this is not applicable to the situation where each rod to be wound around has a different orientation. For a vertically mounted rod the spool must also be vertically oriented, and for winding around a subsequent horizontal rod, the spool orientation must be changed accordingly. This would require additional external information on how the rods are oriented and when the spool has to change its orientation. However, this work is aimed at deriving the

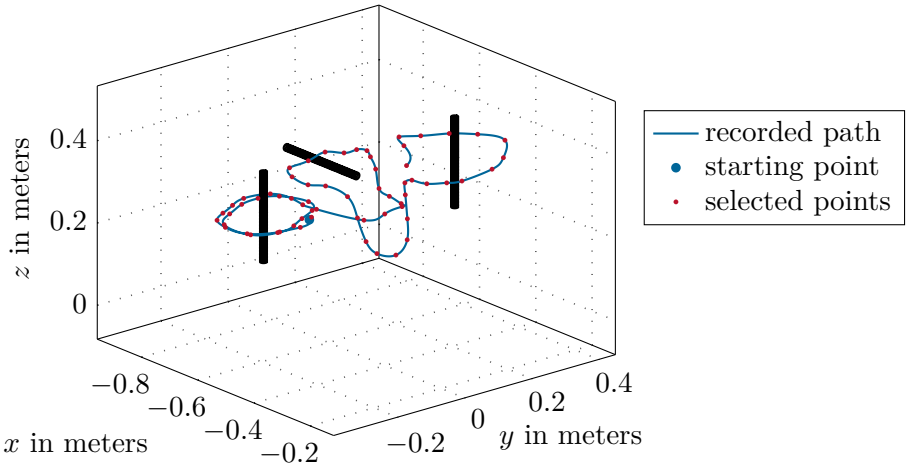


Figure 4.4: Recorded path around three rods and selected points from which a quartic spline is constructed. The robot base is in the origin.

required spool orientation from the given path.

One way of representing the end-effector orientation is through the use of rotation matrices. The columns of a rotation matrix can be interpreted as the basis vectors of the end-effector frame expressed in the base frame. Hence the problem of specifying the desired end-effector orientation is equivalent to finding a well-defined coordinate frame at each point of the path being traversed. This gives rise to using the concept of moving frames from differential geometry, which provide a coordinate system at each point of a curve. This coordinate system can be used to derive an appropriate end-effector orientation during the winding process. In this work, two common moving frames, the Frenet-Serret frame and the parallel transport frame, are employed and the results are compared.

4.5.1 Frenet-Serret Frame

For a three times differentiable parametrized curve $\boldsymbol{\sigma}(\vartheta) \in \mathbb{R}^3$ the basis vectors of the Frenet-Serret frame are given by

$$\mathbf{T}(\vartheta) = \frac{\boldsymbol{\sigma}'(\vartheta)}{\|\boldsymbol{\sigma}'(\vartheta)\|_2} \tag{4.2a}$$

$$\mathbf{N}(\vartheta) = \frac{\mathbf{T}'(\vartheta)}{\|\mathbf{T}'(\vartheta)\|_2} \tag{4.2b}$$

$$\mathbf{B}(\vartheta) = \mathbf{T}(\vartheta) \times \mathbf{N}(\vartheta) \ , \tag{4.2c}$$

where $\mathbf{T}$ is the unit tangent vector to the path, $\mathbf{N}$ is the unit normal vector and $\mathbf{B}$ is the unit binormal vector. The normal vector $\mathbf{N}$ points in the direction of the center of curvature of the path. The osculating plane is spanned by the vectors $\mathbf{T}$ and $\mathbf{N}$. When the Frenet-Serret frame is used for the end-effector orientation, the osculating plane defines the plane, in which the end-effector moves. Furthermore, the binormal vector $\mathbf{B}$ specifies the direction of the spool axis, and thus it is referred to as the spool vector in the following.

In order to express the normal vector $\mathbf{N}$ in terms of the derivatives of the curve $\boldsymbol{\sigma}(\vartheta)$, first, using (4.2a) the vector $\mathbf{T}'$ is written as

$$\begin{aligned}\mathbf{T}'(\vartheta) &= \frac{\mathrm{d}}{\mathrm{d}\vartheta}\left(\frac{\boldsymbol{\sigma}'(\vartheta)}{\|\boldsymbol{\sigma}'(\vartheta)\|_2}\right) = \frac{\boldsymbol{\sigma}''\left\|\boldsymbol{\sigma}'\right\|_2 - \boldsymbol{\sigma}'\frac{(\boldsymbol{\sigma}'')^{\mathrm{T}}\boldsymbol{\sigma}'+(\boldsymbol{\sigma}')^{\mathrm{T}}\boldsymbol{\sigma}''}{2\|\boldsymbol{\sigma}'\|_2}}{\|\boldsymbol{\sigma}'\|_2^2} \\ &= \frac{\boldsymbol{\sigma}''\left\|\boldsymbol{\sigma}'\right\|_2^2 - \boldsymbol{\sigma}'\left((\boldsymbol{\sigma}')^{\mathrm{T}}\boldsymbol{\sigma}''\right)}{\|\boldsymbol{\sigma}'\|_2^3} \,.\end{aligned} \tag{4.3}$$

For simplicity the dependency of the curve $\boldsymbol{\sigma}(\vartheta)$ on the path parameter ϑ is omitted here. Using the identity

$$\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = \mathbf{b}(\mathbf{a}\cdot\mathbf{c}) - \mathbf{c}(\mathbf{a}\cdot\mathbf{b}) \,, \tag{4.4}$$

with $\mathbf{a} = \mathbf{c} = \boldsymbol{\sigma}'$ and $\mathbf{b} = \boldsymbol{\sigma}''$, (4.3) can be written as

$$\mathbf{T}'(\vartheta) = \frac{\boldsymbol{\sigma}'\times(\boldsymbol{\sigma}''\times\boldsymbol{\sigma}')}{\|\boldsymbol{\sigma}'\|_2^3} \,. \tag{4.5}$$

By inserting this result in (4.2b) the unit normal vector $\mathbf{N}(\vartheta)$ is given by

$$\mathbf{N}(\vartheta) = \frac{\boldsymbol{\sigma}'\times(\boldsymbol{\sigma}''\times\boldsymbol{\sigma}')}{\|\boldsymbol{\sigma}'\times(\boldsymbol{\sigma}''\times\boldsymbol{\sigma}')\|} = \frac{\boldsymbol{\sigma}'\times(\boldsymbol{\sigma}''\times\boldsymbol{\sigma}')}{\|\boldsymbol{\sigma}'\|\,\|\boldsymbol{\sigma}''\times\boldsymbol{\sigma}'\|} \,, \tag{4.6}$$

where the last equation follows from the orthogonality of $\boldsymbol{\sigma}'$ to $(\boldsymbol{\sigma}''\times\boldsymbol{\sigma}')$.

In the following, some of the properties of the Frenet-Serret frame that result in discontinuities in the spool orientation are discussed, and it is shown how these issues can be solved.

Using the Frenet-Serret formulas the frame is undefined at path segments with straight lines, i. e. where the curvature $\boldsymbol{\sigma}''(\vartheta)$ vanishes. In this case the frame of the previous point is adopted for the current point.

When passing through inflection points of the path the sign of the normal vector $\mathbf{N}$ changes abruptly, which also results in an abrupt change in the spool vector $\mathbf{B}$. This discontinuity is detected by comparing the magnitude of the difference between the current and previous normal vector against a threshold. By reversing the sign of the normal vector a continuous spool orientation along the path is achieved.

When the osculating plane changes, a jump discontinuity occurs and the frame becomes discontinuous. Let Q_a and Q_b denote the corresponding quaternions of the two consecutive frames before and after the jump. To restore continuity, starting from Q_a the time derivative of the quaternion is integrated to reach the desired final value Q_b. To this end, the error ΔQ between the consecutive quaternions Q_a and Q_b is determined by

$$\Delta Q = Q_a \otimes Q_b^{-1} \,. \tag{4.7}$$

The vector part of (4.7) is given by

$$\Delta\boldsymbol{\epsilon} = -\mu_a\boldsymbol{\epsilon}_b + \mu_b\boldsymbol{\epsilon}_a - \boldsymbol{\epsilon}_a\times\boldsymbol{\epsilon}_b \,. \tag{4.8}$$

This difference is then multiplied by a scalar parameter k_q to obtain

$$\boldsymbol{\omega} = k_q \Delta\boldsymbol{\epsilon} \ , \tag{4.9}$$

which can be interpreted as the scaled required angular velocity of the end-effector for the transition from Q_a to Q_b. Using the corresponding quaternion Q_ω of $\boldsymbol{\omega}$ the time derivative of the quaternion is given by

$$\dot{Q} = \frac{1}{2} Q_\omega \otimes Q_a = \frac{1}{2} \{0 \quad \boldsymbol{\omega}\} \otimes Q_a \ , \tag{4.10}$$

see, e. g., [18]. The discrete time integration with a sampling time T_s is performed as follows

$$Q_k = Q_{k-1} + T_s \dot{Q} \ . \tag{4.11}$$

This result is then normalized to obtain a unit quaternion. It can be converted to the rotation matrix $\mathbf{R}_{0,k}^7$, from which the spool vector $\mathbf{B}_k$ at the k-th time step is extracted.

4.5.2 Parallel Transport Frame

Consider an initial frame $(\mathbf{T}, \mathbf{N}, \mathbf{B})$ on a curve, with the tangent vector $\mathbf{T}$ and the normal vectors $\mathbf{N}$ and $\mathbf{B}$ to the curve. When moving along the curve, the parallel transport frame turns only as much as necessary for $\mathbf{N}$ and $\mathbf{B}$ to remain normal to the curve. Compared to the Frenet-Serret frame, the change of the parallel transport frame only depends on the curvature of the curve and not on its torsion, which results in a smaller change. The discontinuities discussed for the Frenet-Serret frame do not occur when using the parallel transport frame. A more extensive comparison between these two frames together with examples can be found in [28].

Recall that the coordinate system $(\mathbf{e}_\parallel, \mathbf{e}_\perp, \mathbf{e}_{\pitchfork})$ required for the path following control in Chapter 3 was calculated by the parallel transport frame. Hence, the same basis vectors from (3.11), (3.18) and (3.19) can also be used for determining the desired orientation of the end-effector, i. e.

$$\mathbf{T} = \mathbf{e}_\parallel \tag{4.12a}$$

$$\mathbf{N} = \mathbf{e}_\perp \tag{4.12b}$$

$$\mathbf{B} = \mathbf{e}_{\pitchfork} \ , \tag{4.12c}$$

with $\mathbf{B}$ representing the spool vector.

4.5.3 Modification of the Spool Vector

As mentioned earlier in this chapter, the spool orientation depends on the direction change of the traversed path. In this section, the obtained spool vector $\mathbf{B}$ is modified in a few steps so that it is adapted to the winding application. It is assumed that the free end of the vertical rods points upwards, and the horizontal rods point towards the robot, see, e. g., Figure 4.1.

- Consider Figure 4.5 with a cylinder that is coaxial with the vertical axis of the robot base. The robot end-effector lies on the cylinder surface. Given the position $\mathbf{p}$ of the end-effector, the unit radial vector to this point is given by

$$\mathbf{r}^{\mathrm{T}}(\mathbf{p}) = \frac{\begin{bmatrix} p_x & p_y & 0 \end{bmatrix}}{\left\| \begin{bmatrix} p_x & p_y & 0 \end{bmatrix} \right\|} \ . \tag{4.13}$$

  Using the vector

$$\mathbf{z}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{4.14}$$

  which is parallel to the vertical axis of the cylinder, the vector tangential to the cylinder at $\mathbf{p}$ is defined by

$$\boldsymbol{\kappa}(\mathbf{p}) = \mathbf{z} \times \mathbf{r}(\mathbf{p}) \ . \tag{4.15}$$

  For winding around a horizontal rod pointing towards the robot, the tangential component of the spool vector is suppressed to avoid collisions with the rod. To this end, the spool vector is projected onto the plane spanned by the vectors $\mathbf{r}$ and $\mathbf{z}$, which is in the null space of the tangent vector $\boldsymbol{\kappa}$. The projection of the spool vector $\mathbf{B}$ onto this plane is given by

$$\mathbf{B}_p = \left(\mathbf{I}_3 - \frac{\boldsymbol{\kappa}\boldsymbol{\kappa}^{\mathrm{T}}}{\boldsymbol{\kappa}^{\mathrm{T}}\boldsymbol{\kappa}} \right) \mathbf{B} \ , \tag{4.16a}$$

$$\tilde{\mathbf{B}}_p = \frac{\mathbf{B}_p}{\left\| \mathbf{B}_p \right\|} \ . \tag{4.16b}$$

  Note that after each modification step a normalization of the vector is necessary to obtain a unit vector.

- In the **r**-**z**-plane the spool vector is modified such that it is restricted to the range spanned by the negative vertical vector $-z$ and the radial vector r, see Figure 4.6(a). This step is required since for a rod pointing upwards, winding is only possible with the spool pointing downwards. Hence, if the z-component of $\mathbf{B}$ is positive, its sign is reversed.

- A problem during winding around a horizontal rod is that the robot might get into wrist singularities, i. e. the joint axes 5 and 7 become parallel. To avoid such a situation, an inclination angle α for the spool vector is introduced, which specifies the minimum required angle between the radial vector $\mathbf{r}$ and the spool vector as shown in Figure 4.6(b). Let $\mathbf{B}_\alpha$ denote the limit for the spool vector. The z-component of $\mathbf{B}$ has to be modified to be

$$B_z := \min\left(B_z \, , B_{\alpha,z}\right) = \min\left(B_z \, , -\arctan(\alpha)\right) \ , \tag{4.17a}$$

$$\tilde{\mathbf{B}} = \frac{\mathbf{B}}{\|\mathbf{B}\|} \ . \tag{4.17b}$$

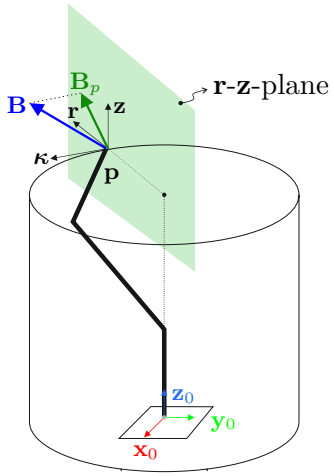


Figure 4.5: Projection of the spool vector $\mathbf{B}$ onto the $\mathbf{r}$-$\mathbf{z}$-plane to obtain $\mathbf{B}_p$.

(a) (b)

Figure 4.6: (a) Allowed range of the spool vector $\mathbf{B}$. (b) Allowed range with an inclination angle α to avoid the wrist singularity.

- The spool vector might have components that point towards the robot. This is an undesirable orientation considering the fact that the rods are usually placed away from the robot and point towards it. In this case the spool vector and the radial vector form an obtuse angle. To correct that, the spool vector is reflected through the $\boldsymbol{\kappa}$-$\mathbf{z}$-plane as shown in Figure 4.7. Since the unit radial vector $\mathbf{r}$ is the normal vector of this plane, the reflected spool vector $\mathbf{B}_r$ is given by

$$\mathbf{B}_r = \mathbf{B} - 2\mathbf{r}\mathbf{r}^{\mathrm{T}}\mathbf{B} \,, \tag{4.18a}$$

$$\tilde{\mathbf{B}}_r = \frac{\mathbf{B}_r}{\|\mathbf{B}_r\|} \,. \tag{4.18b}$$

In the following, the desired modified spool vector is denoted by $\mathbf{B}_d$.

Orientation Error

From the previous calculations it can be seen that the modified spool vector $\mathbf{B}_d$ has no components in the direction of the tangential vector $\boldsymbol{\kappa}(\mathbf{p})$ to the cylinder, i. e. they are

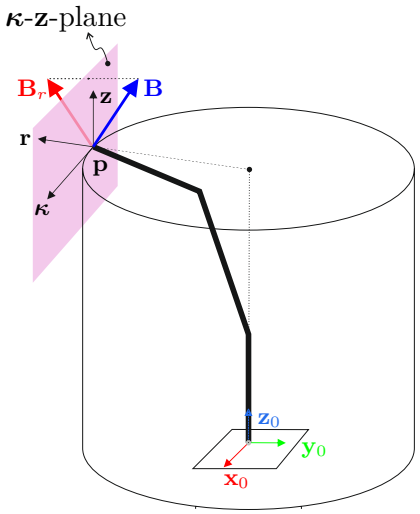


Figure 4.7: Reflection of the spool vector $\mathbf{B}$ through the $\boldsymbol{\kappa}$-$\mathbf{z}$-plane to obtain $\mathbf{B}_r$.

orthogonal to each other. The desired orientation of the end-effector can be expressed by the rotation matrix

$$\mathbf{R}_d = \begin{bmatrix} \boldsymbol{\kappa}(\mathbf{p}) & \mathbf{B}_d \times \boldsymbol{\kappa}(\mathbf{p}) & \mathbf{B}_d \end{bmatrix} . \tag{4.19}$$

To achieve this desired orientation during the winding around a rod, continuous changes of the joint angle q_7 may be required. This might cause a limit violation of the 7th joint. However, due to the symmetry of the end-effector, rotations about the 7th joint axis are not necessary and the only relevant vector for the orientation is the spool vector $\mathbf{B}_d$. Hence, for calculating the orientation error, not the whole rotation matrices but only their third basis vectors are used, which specify the spool vector. In [29] it is shown how a vector can be aligned with another vector using unit quaternions. The required unit quaternion for aligning the current unit spool vector $\mathbf{B}(\mathbf{q})$ with the desired modified unit spool vector $\mathbf{B}_d$ is given by

$$Q = \{\mu \quad \boldsymbol{\epsilon}\} = \left\{ \sqrt{\frac{1+c}{2}} \quad \sqrt{\frac{1}{2(1+c)}} \left(\mathbf{B}(\mathbf{q}) \times \mathbf{B}_d\right) \right\} , \tag{4.20}$$

with $c = \mathbf{B}(\mathbf{q}) \cdot \mathbf{B}_d$, see [29]. The vector part $\boldsymbol{\epsilon}$ of this quaternion is used as the orientation error $\mathbf{e}_r$ in the control law (3.59).

4.6 Motion Along the Path

Since the primary objective of path following control is the asymptotic convergence of the output to the path, the desired values for the transversal coordinates ξ_1^d and ξ_3^d and their time derivatives are set to zero. To obtain a smooth trajectory for the desired tangential coordinate η_1^d a setpoint filter is used.

Furthermore, the approach for finding the optimal path parameter is discussed in this section.

4.6.1 Setpoint Filter

In order to smooth a given signal $u(t)$ a linear filter in the form

$$G(s) = \frac{\hat{y}(s)}{\hat{u}(s)} = \frac{1}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{4.21}$$

is used. Herein, s is the Laplace variable and $\hat{u}(s)$ and $\hat{y}(s)$ are the Laplace transform of the filter input $u(t)$ and output $y(t)$, respectively. This third order filter also allows for calculating the first two continuous derivatives of the input using a state space representation. Applying a step input from u_{min} to u_{max} to this filter, a sufficiently smooth trajectory $y(t)$ is obtained. Additionally, it is desirable to limit the velocity $\dot{u}(t)$ to a maximum value $\dot{u}_{max}$. This can be achieved by using a rate limiter at the input of the filter. The resulting signals are shown in Figure 4.8.

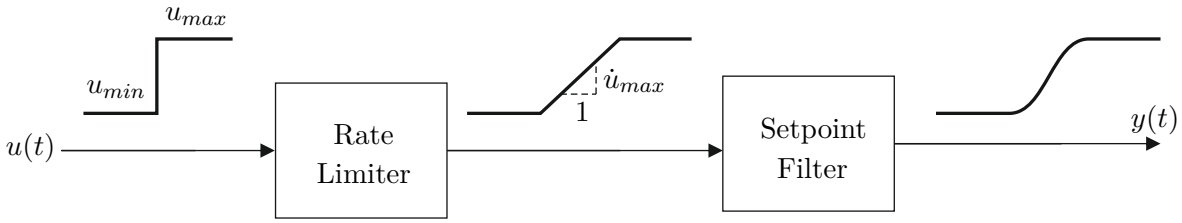


Figure 4.8: Rate limiter and setpoint filter for achieving a sufficiently smooth trajectory.

4.6.2 Optimal Path Parameter

For the numerical implementation of the path following control algorithm a time discretization in the form $t = kT_s$, $k = 1, 2, \ldots$ with the sampling time T_s is performed. Using the Newton method, the optimization problem (3.20) with the cost function

$$f(\vartheta) = \left\| \mathbf{y}_{t,k} - \boldsymbol{\sigma}_t(\vartheta) \right\|_2^2 \tag{4.22}$$

is iteratively solved in each time step $k = 1, 2, \ldots$. The iteration is given by

$$\vartheta_{k,i} = \vartheta_{k,i-1} - \frac{f'(\vartheta_{k,i-1})}{f''(\vartheta_{k,i-1})}, \qquad i = 1, 2, \ldots \tag{4.23}$$

with the initial condition $\vartheta_{k,0} = \vartheta^*_{k-1}$. The termination condition is given by

$$\left| \vartheta_{k,i} - \vartheta_{k,i-1} \right| < \varepsilon \tag{4.24}$$

with $\varepsilon > 0$. The derivatives of the cost function with respect to $\vartheta_{k,i-1}$ are calculated to be

$$f'(\vartheta_{k,i-1}) = -2 \left(\mathbf{y}_t - \boldsymbol{\sigma}_t \left(\vartheta_{k,i-1} \right) \right)^{\mathrm{T}} \boldsymbol{\sigma}'_t \left(\vartheta_{k,i-1} \right) \tag{4.25}$$

$$f''(\vartheta_{k,i-1}) = 2 \left\| \boldsymbol{\sigma}'_t \left(\vartheta_{k,i-1} \right) \right\|_2^2 - 2 \left(\mathbf{y}_t - \boldsymbol{\sigma}_t \left(\vartheta_{k,i-1} \right) \right)^{\mathrm{T}} \boldsymbol{\sigma}''_t \left(\vartheta_{k,i-1} \right) . \tag{4.26}$$

This algorithm should be initialized at the global optimum ϑ_0^* for the time step $k = 0$. For this purpose, a sufficient number of evenly spread points on the path γ_t are chosen. The point with the shortest distance to $\mathbf{y}_t(0)$ is used as the starting point for the local minimum search (4.23) to obtain ϑ_0^*.

Using the optimal solution $\vartheta_k^* = \vartheta_{k,i}$, the arc length η_1 from (3.25) is numerically calculated to be

$$\eta_{1,k} = \eta_{1,k-1} + \left(\vartheta_k^* - \vartheta_{k-1}^*\right) \left\| \boldsymbol{\sigma}_t'(\vartheta_k^*) \right\|_2 \ , \tag{4.27}$$

see [13].

4.7 Simulation Results

Using MATLAB/SIMULINK, simulations of the path following control algorithm applied to the robot model (3.1) were performed. The control parameters of the path following control are tuned such that small errors can be achieved while making sure that the joint torque and velocity limits of the robot are not violated. For a list of control parameters, see Appendix A.3.

In the following, the simulation results of traversing the path from Figure 4.4 are presented. The results of employing both the Frenet-Serret (FS) and parallel transport (PT) frame for the desired orientation are compared.

The distance of the system output to the path is given by

$$d = \left\| \mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*) \right\| \ . \tag{4.28}$$

As can be seen in Figure 4.9, starting with a distance $d_0 = 10\,\text{mm}$ the system output converges to the path when at $t = 2\,\text{s}$ the path following controller is turned on. At $t = 5\,\text{s}$ the traversal of the path is started. The increase of distance d in the interval $[50\,\text{s}, 60\,\text{s}]$ can be traced back to the sharp torsion in the end part of the path.

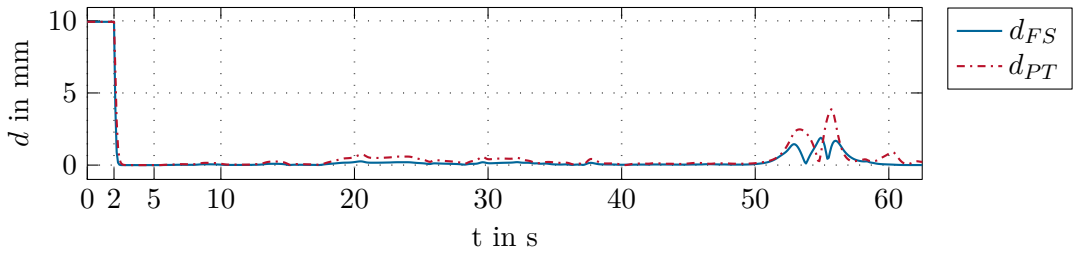


Figure 4.9: (Simulation) Distance of the system output to the path, $d = \left\| \mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*) \right\|$.

The tangential coordinate η_1 and its time derivative η_2 are shown in Figure 4.10 together with their desired values η_1^d and η_2^d, which are obtained using the setpoint filter from Section 4.6.1. Using a rate limiter the maximum tangential velocity η_2^d is limited to be $0.07\,\text{m/s}$. The desired tangential acceleration is also depicted in this figure, which is used as feedforward part for controlling the tangential motion according to (3.57).

Figure 4.11 shows the time evolution of the optimal path parameter ϑ^*. The initial optimum value is found at $\vartheta_0^* = 1.4 \times 10^{-4}$ and because of the normalization of the path parameter the value at the end of the path equals 1.

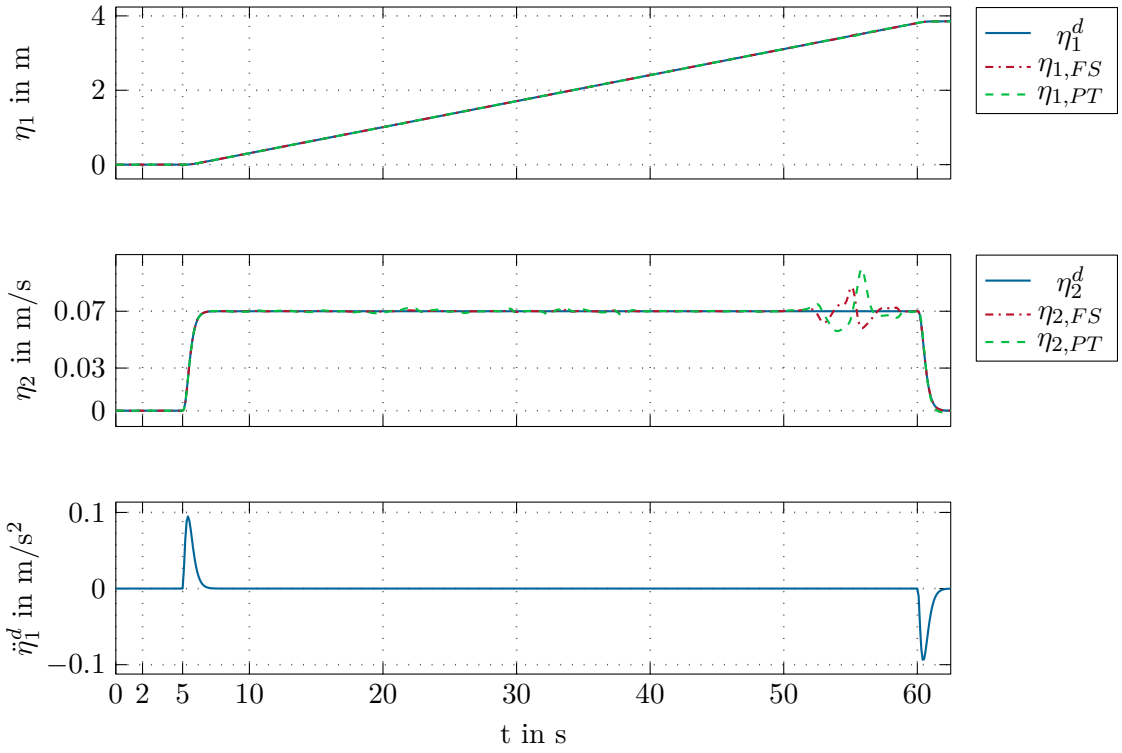


Figure 4.10: (Simulation) Tangential coordinate η_1 and its time derivative η_2 together with their desired values η_1^d and η_2^d and the desired tangential acceleration $\ddot{\eta}_1^d$.

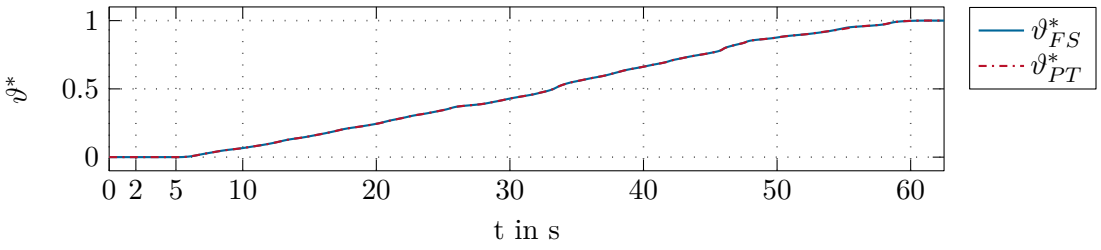


Figure 4.11: (Simulation) Time evolution of the optimal path parameter ϑ^*.

The transversal coordinates ξ_1 and ξ_3 and their time derivatives ξ_2 and ξ_4 are shown in Figure 4.12. This is in accordance with Figure 4.9 since the distance to the path can also be expressed as

$$d = \sqrt{\xi_1^2 + \xi_3^2} \, . \tag{4.29}$$

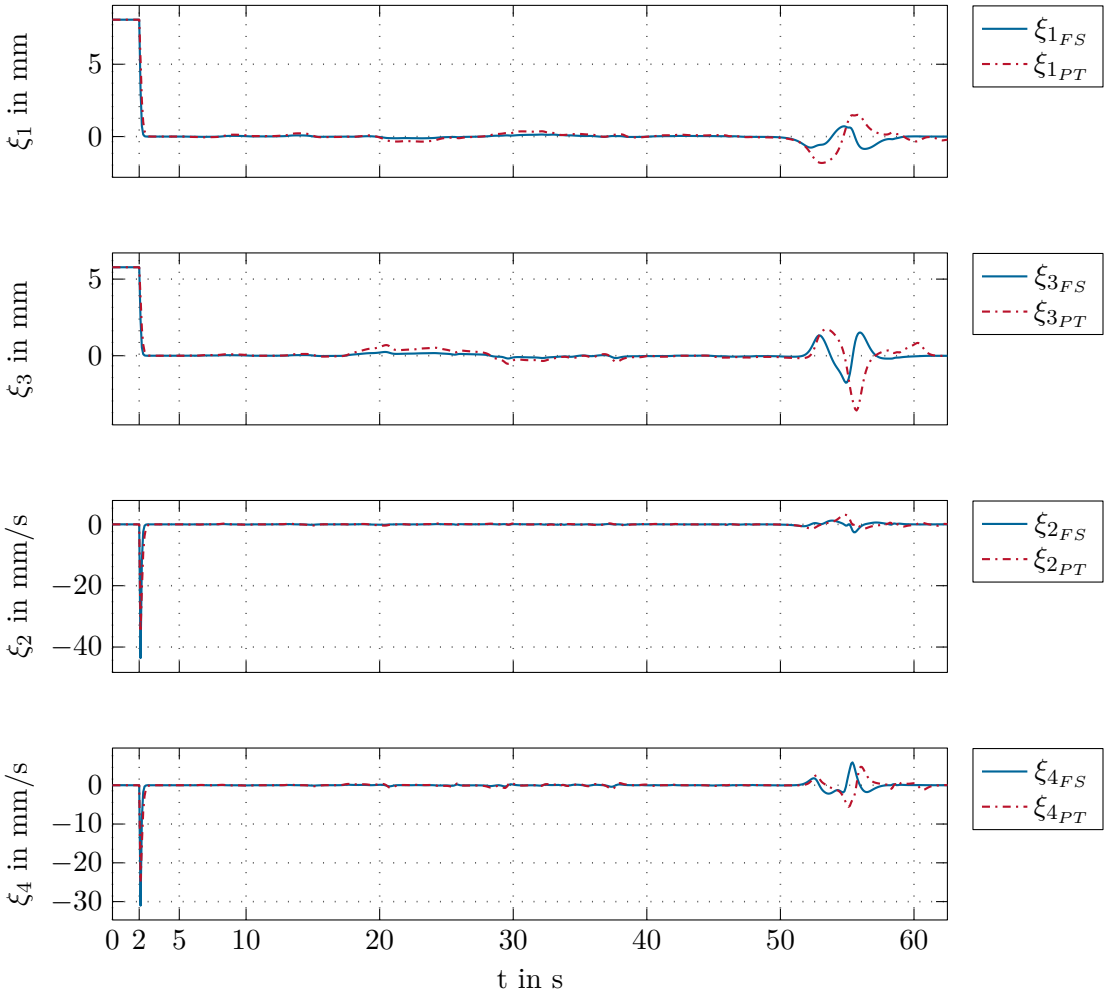


Figure 4.12: (Simulation) Transversal coordinates ξ_1 and ξ_3 and velocities ξ_2 and ξ_4.

The spool vector $\mathbf{B}$ and its desired direction $\mathbf{B}_d$ are illustrated in Figure 4.13 for the FS frame and in Figure 4.14 for the PT frame. The vector components of the orientation error $\mathbf{e}_r$ are shown in Figure 4.15. During the winding around the first vertical rod until $t = 30\,\mathrm{s}$ the orientation error is almost zero. In the subsequent part the orientation error becomes larger due to the changes in the curvature and torsion of the path. Setting the controller parameters to larger values can improve the result at the cost of higher joint velocities and torques close to the limits of the robot. An important observation is that even when the orientation error increases, it is immediately driven to zero after each increase.

Figure 4.16 shows the required joint torques $\boldsymbol{\tau}$. The resulting trajectories of the joint angles $\mathbf{q}$ can be seen in Figure 4.17.

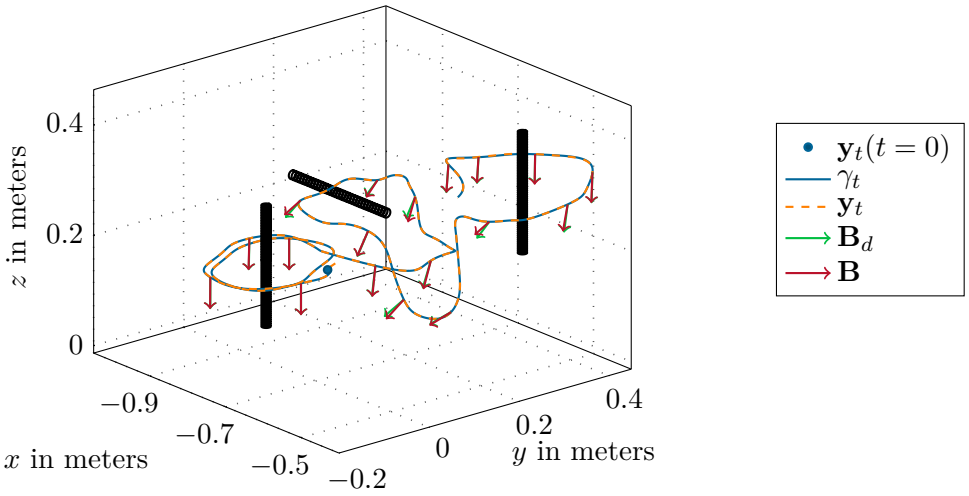


Figure 4.13: (Simulation) Use of the Frenet-Serret frame for winding a rope around three rods along the path γ_t. The spool vector $\mathbf{B}$ and its desired direction $\mathbf{B}_d$ are shown at exemplary points.

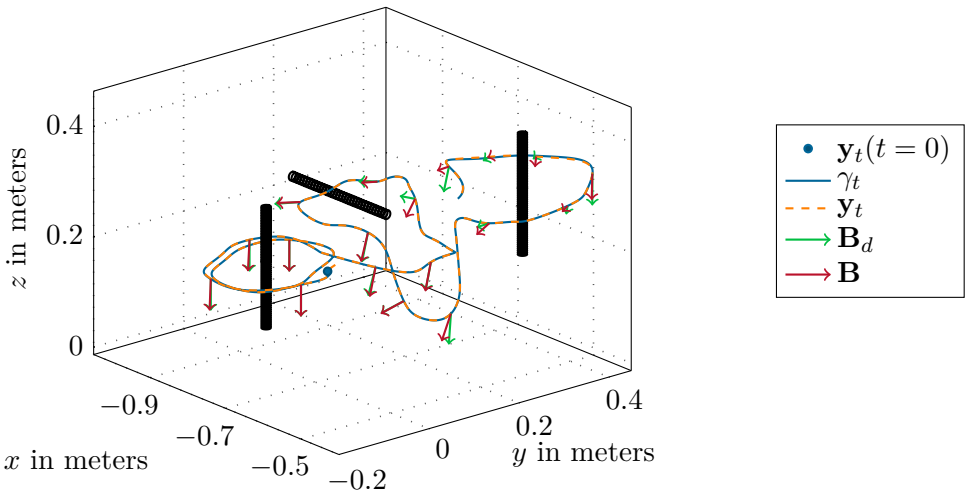


Figure 4.14: (Simulation) Use of the parallel transport frame for winding a rope around three rods along the path γ_t. The spool vector $\mathbf{B}$ and its desired direction $\mathbf{B}_d$ are shown at exemplary points.

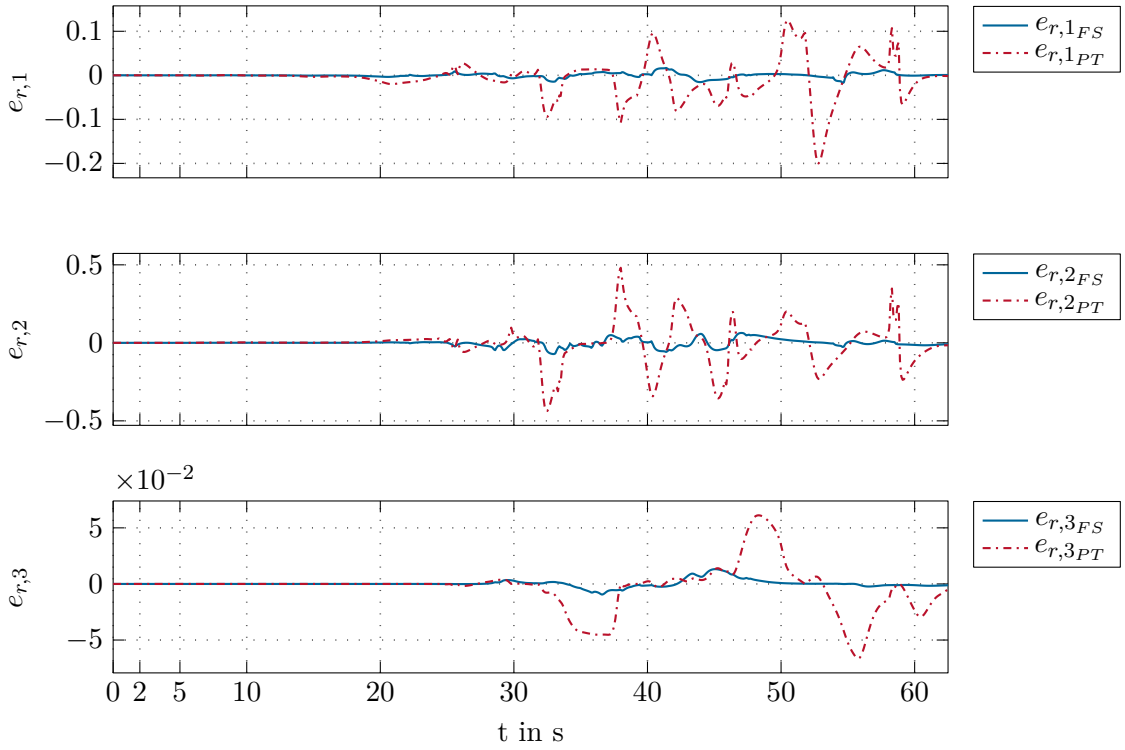


Figure 4.15: (Simulation) Orientation error $\mathbf{e}_r$ as the vector part of the quaternion error.

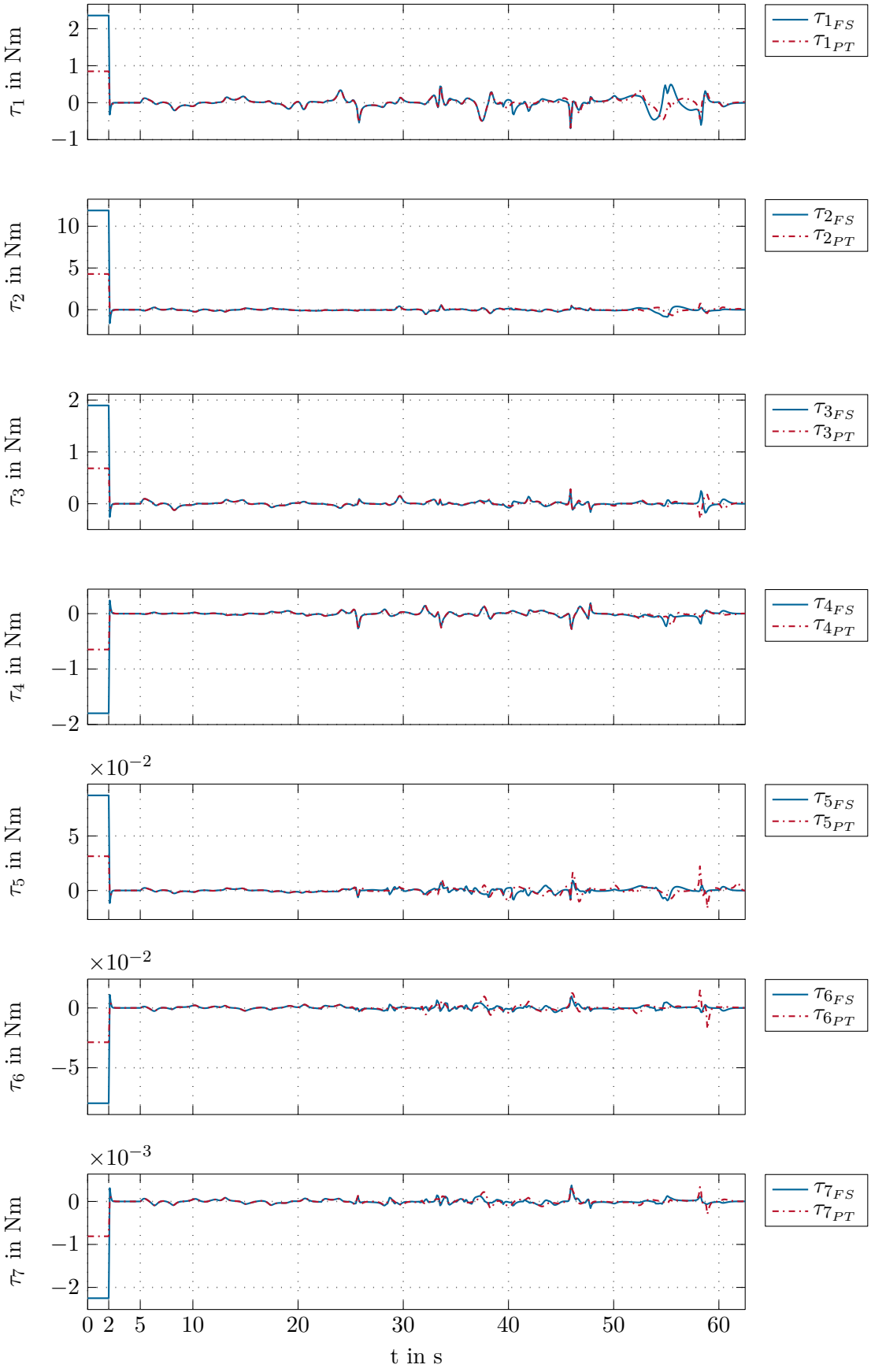


Figure 4.16: (Simulation) Joint torques $\boldsymbol{\tau}$.

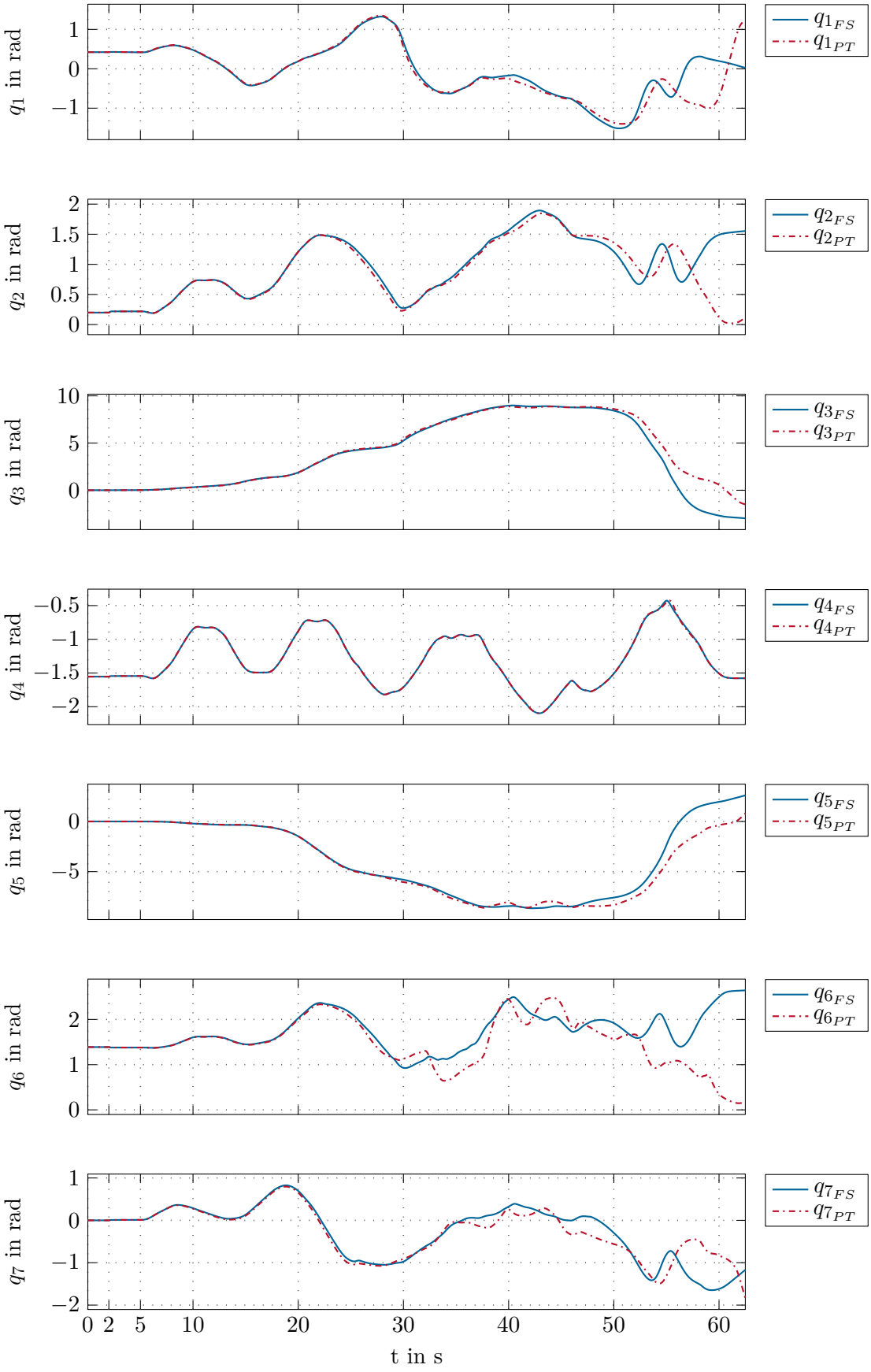


Figure 4.17: (Simulation) Joint angles **q**.

4.8 Experimental Results

The method presented in this work is implemented in MATLAB/SIMULINK, from which a C++ code is generated. This code is then used on the real-time system *TwinCAT* from *Beckhoff Automation*. The communication between *TwinCAT* and the *KUKA* controller unit takes place via *EtherCAT*. The sampling time used for the time discretization is $T_s = 1\,\text{ms}$.

First, using a computed torque (CT) controller in the form

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\mathbf{v} + \mathbf{C}\left(\mathbf{q}, \dot{\mathbf{q}}\right)\dot{\mathbf{q}} + \mathbf{g}\left(\mathbf{q}\right) \tag{4.30a}$$

$$\mathbf{v} = \ddot{\mathbf{q}}^d - \mathbf{K}_0(\mathbf{q} - \mathbf{q}^d) - \mathbf{K}_1(\dot{\mathbf{q}} - \dot{\mathbf{q}}^d) \tag{4.30b}$$

the end-effector is driven to the configuration corresponding to the starting point of the path, which is obtained using the inverse kinematics described in Section 2.2. The controller gains $\mathbf{K}_0$ and $\mathbf{K}_1$ are positive definite diagonal matrices. The vectors $\mathbf{q}^d$, $\dot{\mathbf{q}}^d$, $\ddot{\mathbf{q}}^d$ denote the desired trajectory of the joint angles, velocities and accelerations, respectively, which are obtained by using the setpoint filter from Section 4.6.1 for each joint.

Once the stationary target point is reached, the controller is switched to the path following control. In the following, the experimental results for the path from Figure 4.4 are presented.

Figure 4.18 shows the measured distance d of the end-effector to the path. The difference in the initial distance to the path for the FS and PT case is due to the fact that the CT controller does not always reach the exact same end pose, which is mainly caused by the unmodeled joint friction. At $t = 2.2\,\text{s}$, the path following controller is turned on. It can be seen that the initial distance to the path gets smaller, cf. control objective ($\mathcal{O}_1$). At $t = 2.8\,\text{s}$, the spool motor is turned on, which acts as a disturbance to the controller. This explains the increase of the distance at this point. The traversal of the path starts at $t = 5.1\,\text{s}$.

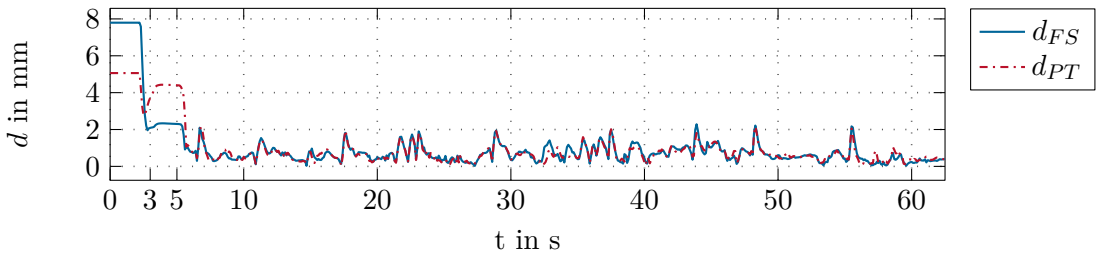


Figure 4.18: Measured distance of the system output to the path, $d = \|\mathbf{y}_t - \boldsymbol{\sigma}_t(\vartheta^*)\|$.

The tangential and transversal coordinates and their time derivatives are illustrated in Figure 4.19 and Figure 4.20. The results show the decoupling of the position and orientation control, as the quantities related to the position match in both cases FS and PT.

The differences between the experimental and simulation results can be explained by the measurement noise and the robot joint friction which are not taken into account in the simulation.

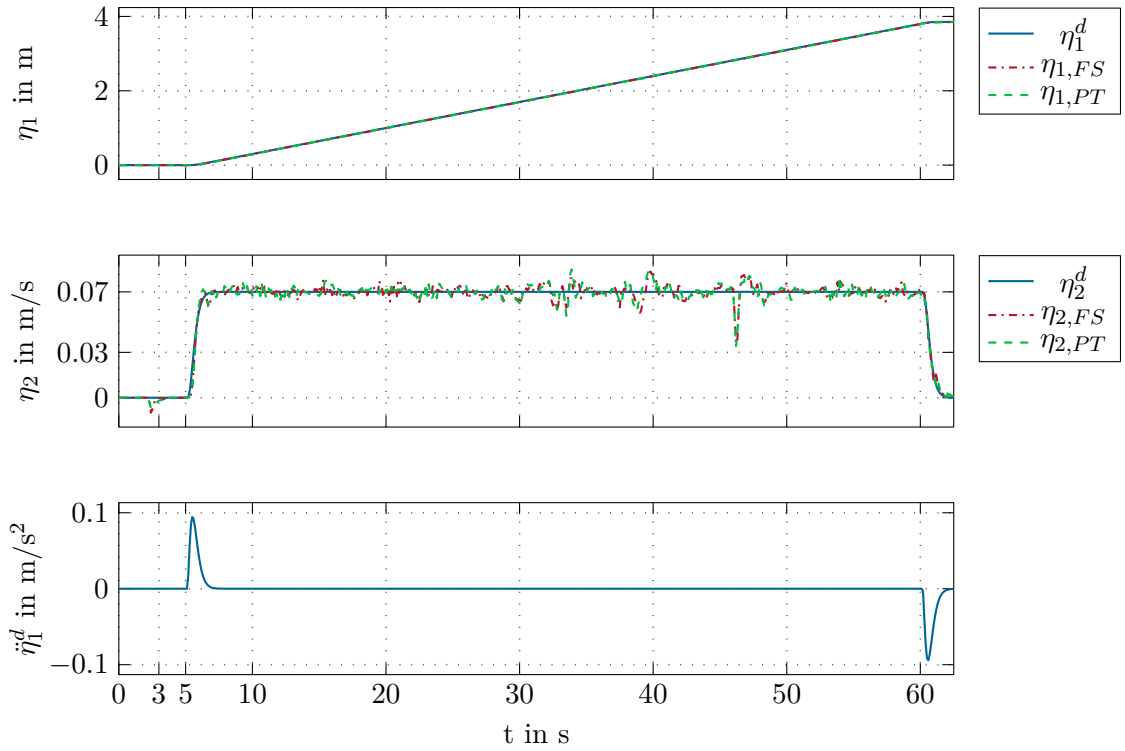


Figure 4.19: Measured tangential coordinate η_1 and velocity η_2, their desired values η_1^d and η_2^d and the desired tangential acceleration $\ddot{\eta}_1^d$.

The time evolution of the optimal path parameter is shown in Figure 4.21.

The desired and measured spool vector during the traversal of the path are shown in Figure 4.22 for the FS frame and in Figure 4.23 for the PT frame.

Figure 4.24 shows the vector components of the orientation error. It can be seen that at the beginning, i. e. during the winding around the left vertical rod, both frames yield the same result. Starting from the transition to the horizontal rod, the FS and PT frame behave differently.

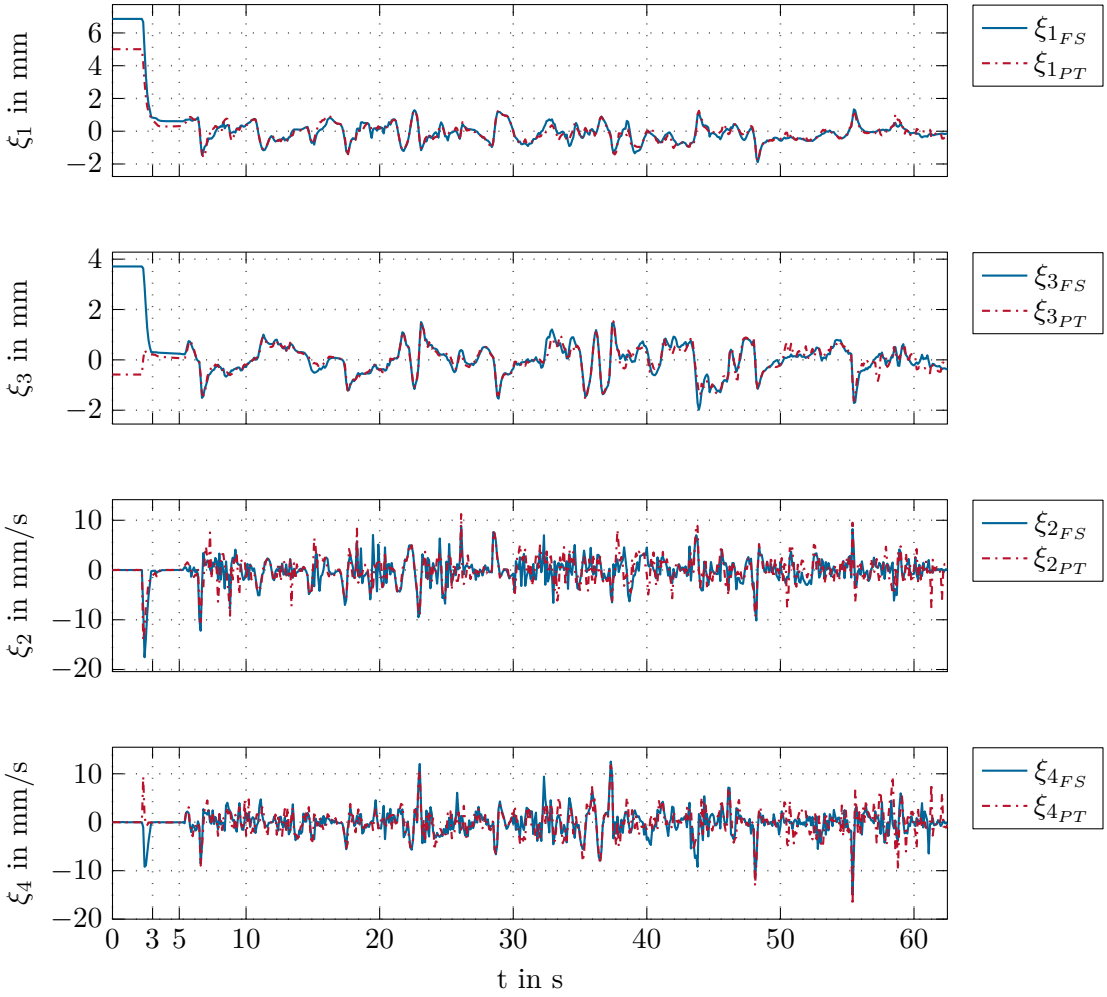


Figure 4.20: Measured transversal coordinates ξ_1 and ξ_3 and velocities ξ_2 and ξ_4.

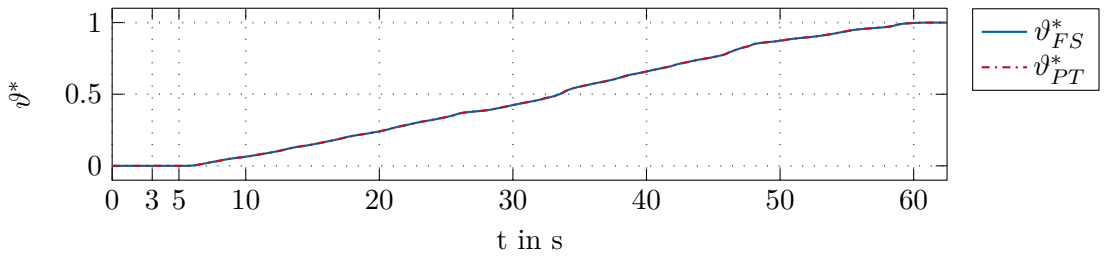


Figure 4.21: Time evolution of the optimal path parameter ϑ^*.

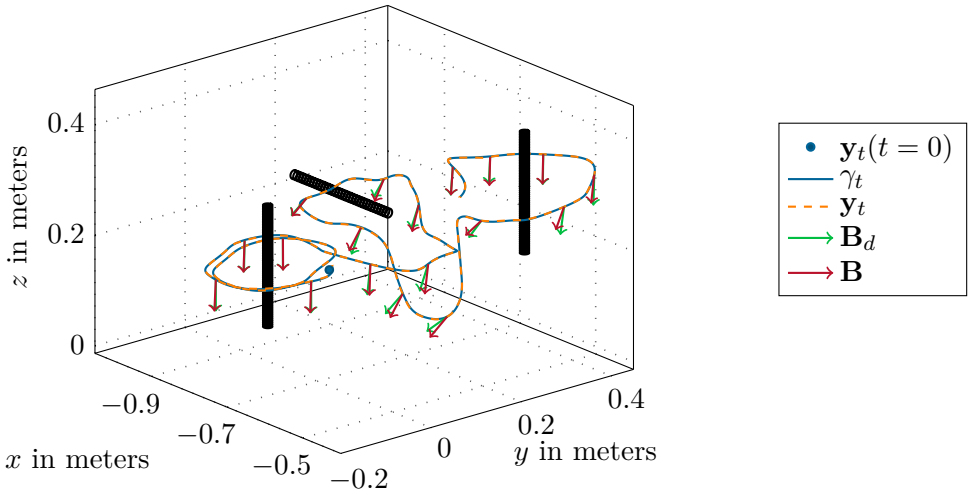


Figure 4.22: Experimental result of using the Frenet-Serret frame for winding a rope around three rods along the path γ_t. The spool vector $\mathbf{B}$ and its desired direction $\mathbf{B}_d$ are shown at exemplary points.

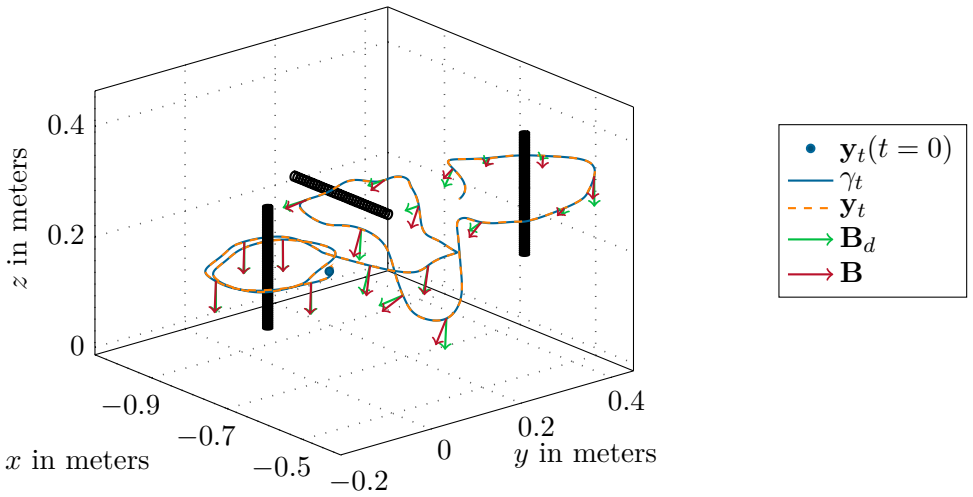


Figure 4.23: Experimental result of using the parallel transport frame for winding a rope around three rods along the path γ_t. The spool vector $\mathbf{B}$ and its desired direction $\mathbf{B}_d$ are shown at exemplary points.

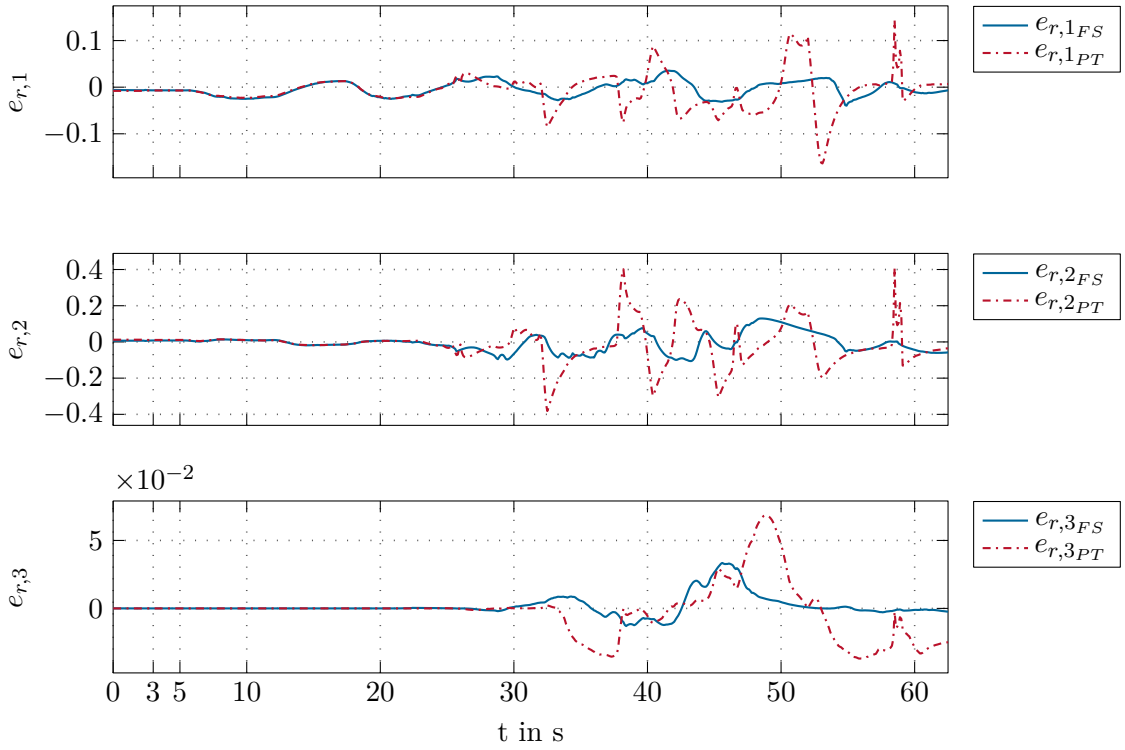


Figure 4.24: Measured orientation error $\mathbf{e}_r$ as the vector part of the quaternion error.

The desired rope tension force during winding is set to $F_r^d = 6\,\mathrm{N}$. The desired and actual torque of the spool motor are shown in Figure 4.25. This motor torque in the end-effector acts as an external disturbance torque on the robot joints. These external torques are estimated by the *KUKA* controller unit. Figure 4.26 shows the estimated external torques $\boldsymbol{\tau}_{ext}$, which are compensated in the control law (3.49). The differences between the FS and PT case after $t = 54\,\mathrm{s}$ is a result of the fact that using the PT frame the spool orientation was not completely vertical when it reached the last vertical rod, and thus the stretched rope slightly missed the rod and could not be wound.

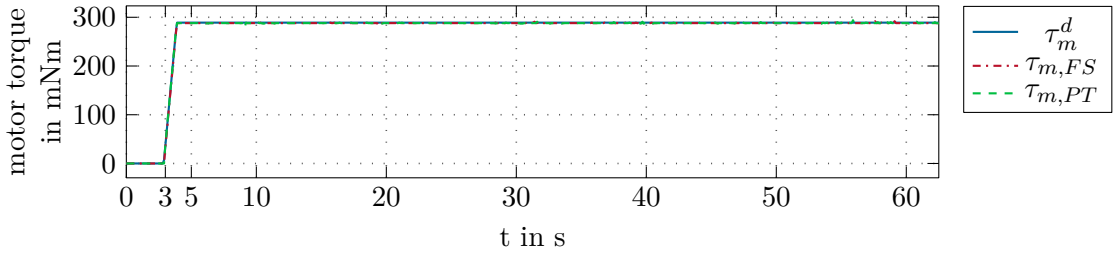


Figure 4.25: Torque of the spool motor τ_m and its desired value τ_m^d.

The required joint torques $\boldsymbol{\tau}$ for controlling the position and orientation of the end-effector along the path are shown in Figure 4.27.

Figure 4.28 shows the trajectories of the joint angles $\mathbf{q}$ during the winding process. As can be seen, the main differences between the FS and PT case are in the last three joints, the wrist joints, which determine the orientation of the end-effector.

Comparing the results, it can be stated that both the FS and PT frame can be used for determining the desired spool orientation. However, since the change of the PT frame depends only on the curvature and not the torsion of the path, it must be ensured that the path exhibits the required curvature before reaching the rod. Otherwise, it is possible that the spool fails to wind around the rod.

In contrast to the PT frame, the FS frame takes into account both the curvature and the torsion of the path. Moreover, during winding around a rod the osculating plane of the FS frame clearly shows the relation between the motion of the end-effector and the orientation of the rod.

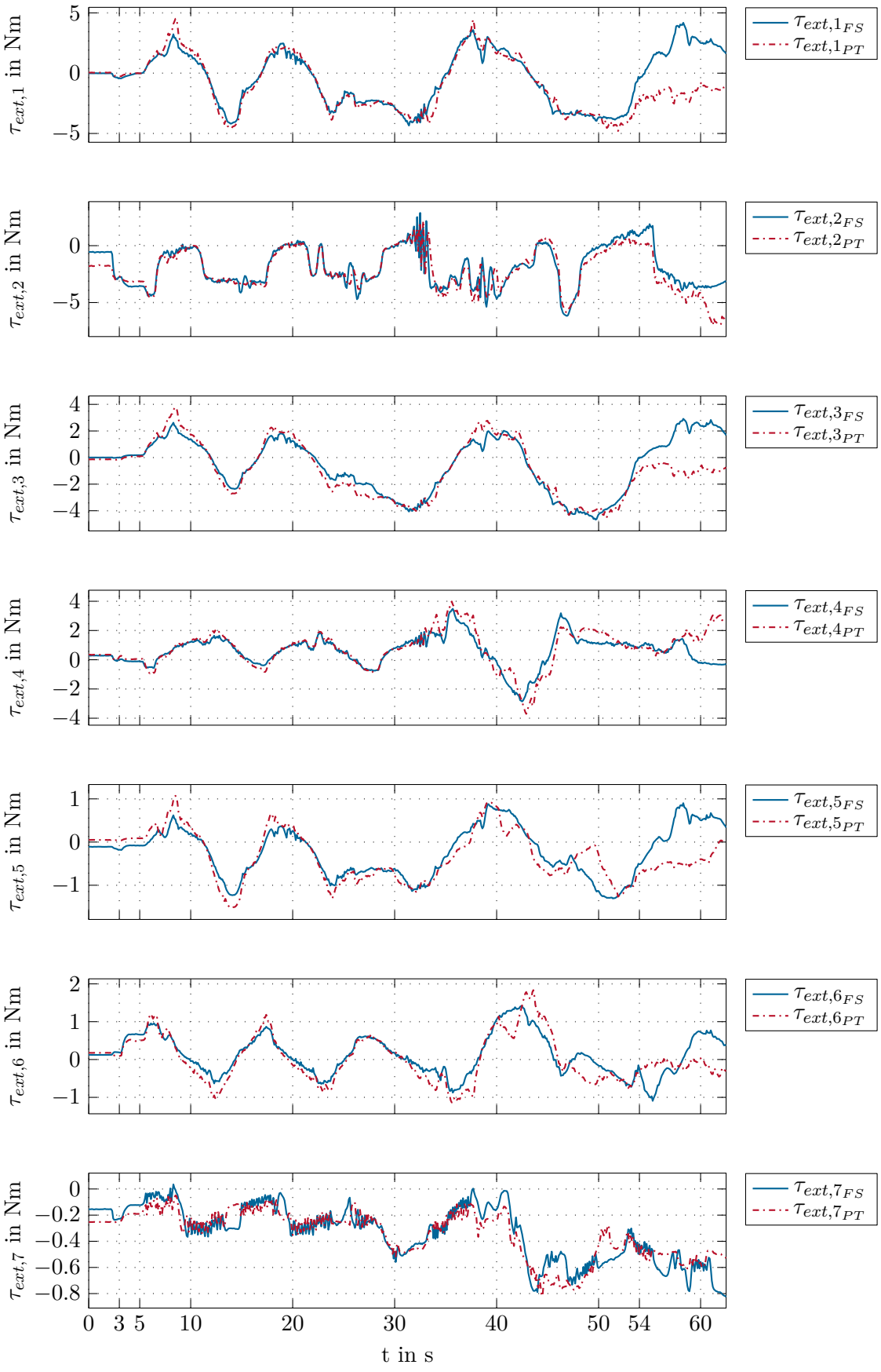


Figure 4.26: Estimated external torques $\boldsymbol{\tau}_{ext}$.

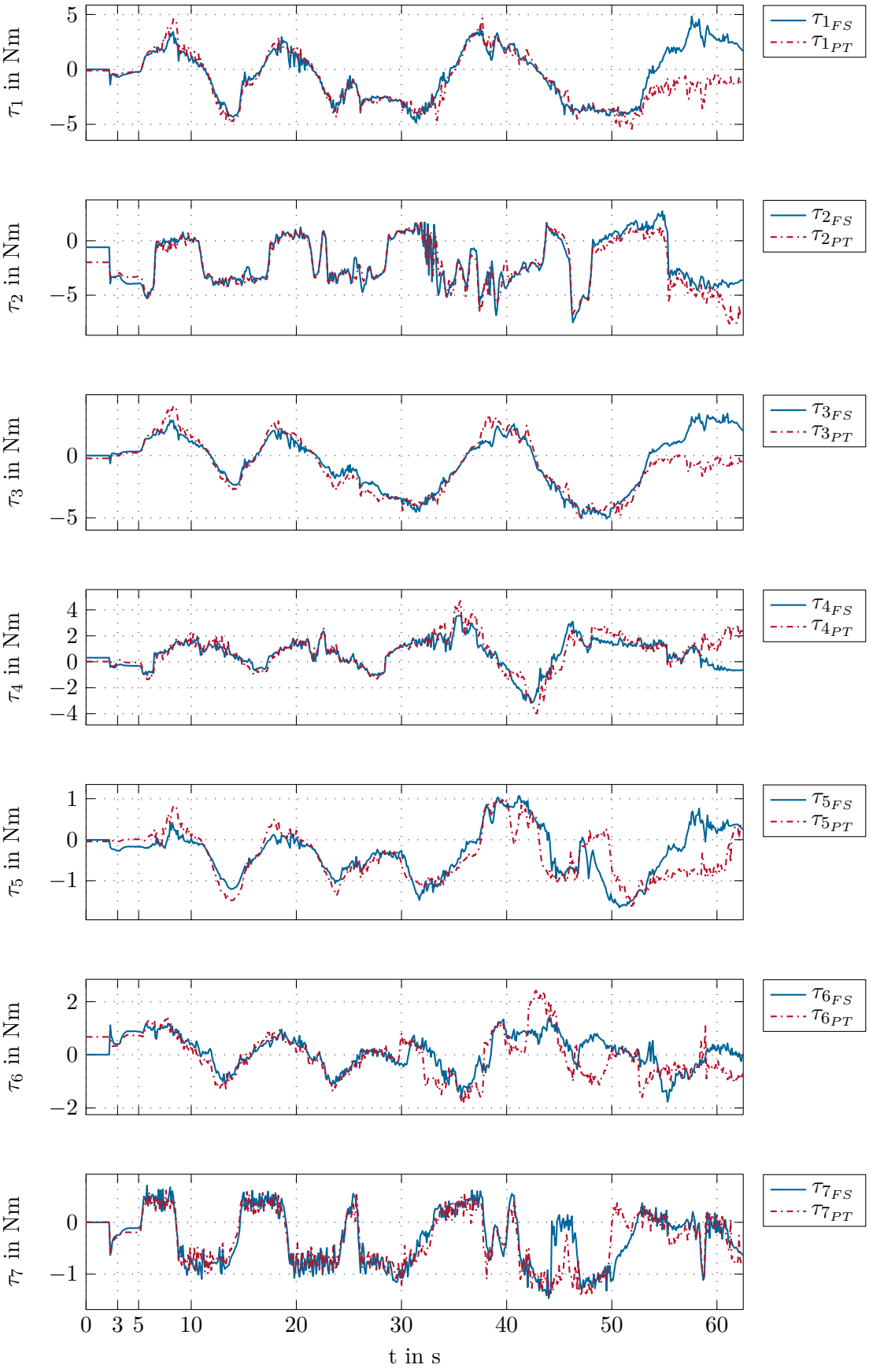


Figure 4.27: Joint torques $\boldsymbol{\tau}$.

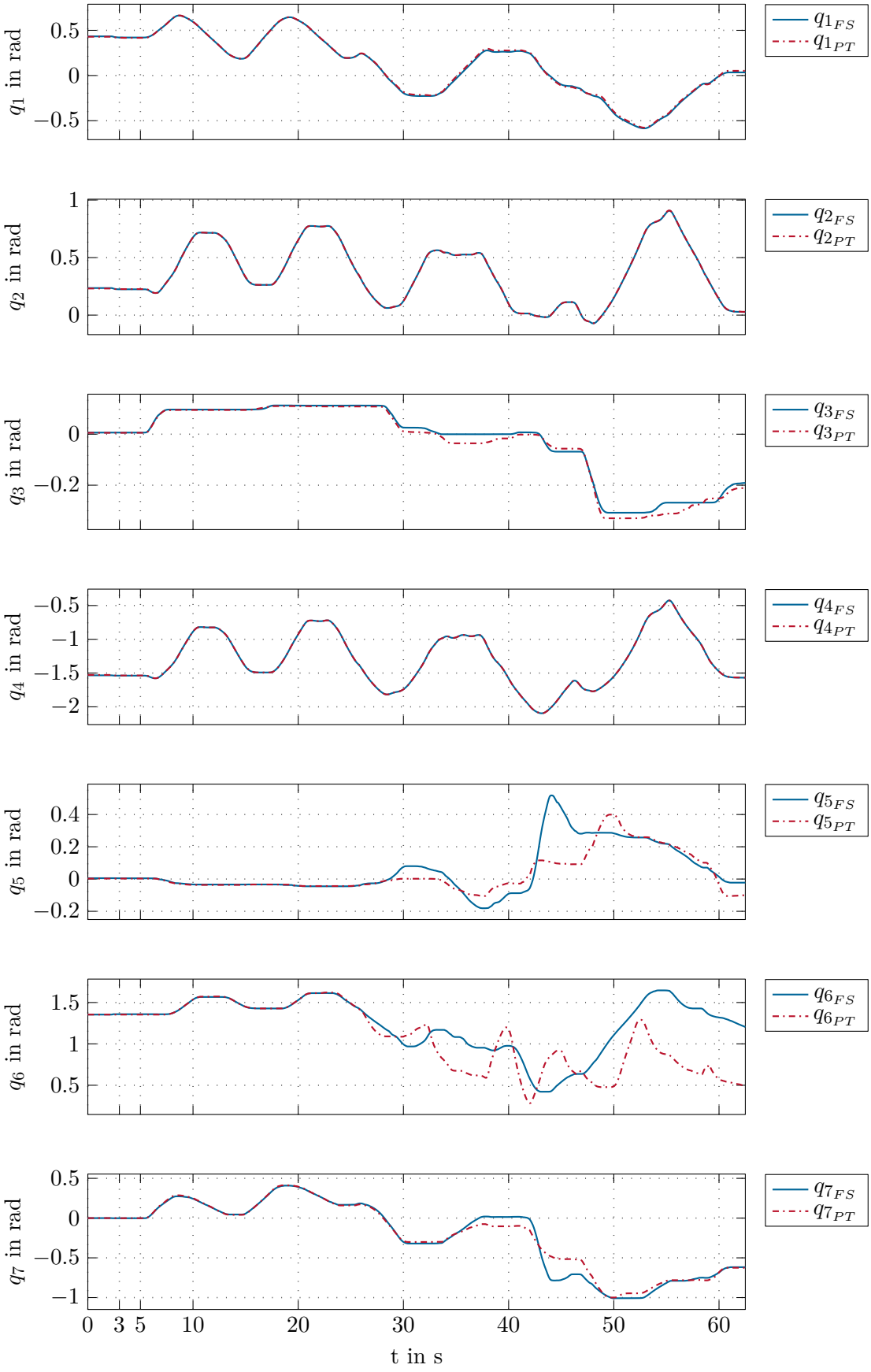


Figure 4.28: Measured joint angles $\mathbf{q}$.

5 Conclusions

This thesis presented a method for robotic winding of a rope around arbitrarily oriented rods in the robot workspace. During the winding process the rope tension is controlled to be constant.

The mathematical model of the robot *KUKA LWR IV+* was derived in Chapter 2. After calculation of the forward and inverse kinematics the dynamic model of the robot was determined in two steps. First, the equations of motion were derived assuming a rigid-body model for the robot. This model was extended in a further step to take into account the joint flexibility.

In a teach-in procedure, the desired path for winding around the rods was acquired. Using the path following control method described in Chapter 3 this path can be traversed by the robot end-effector. The method uses transverse feedback linearization to split up the motion of the end-effector into dynamics tangential and transversal to the path. The asymptotic convergence of the end-effector to the path is guaranteed by controlling the transversal dynamics, and the desired motion along the path is achieved by controlling the tangential dynamics. Furthermore, an input-output linearization was used to control the orientation of the end-effector.

The end-effector consists of a rope spool and a motor that is used for controlling the rope tension. Chapter 4 described the specification of the end-effector orientation during the winding process, which is a main aspect of this work. The Frenet-Serret frame and the parallel transport frame, two moving frames from differential geometry, were used to derive the desired orientation from the path. The simulation and experimental results show the applicability of the presented method for the robotic winding of a rope. Comparison of the results of employing the Frenet-Serret frame and the parallel transport frame reveal that the Frenet-Serret frame is more suitable for deriving the required end-effector orientation since it considers not only the curvature of the path but also its torsion. Furthermore, the relation between the motion of the end-effector and the orientation of the rods can be explained using the osculating plane of the Frenet-Serret frame.

In this work, the tangential vector of the path was used to derive the moving frame for the spool orientation. As an extension, this vector can be replaced by the direction of the stretched rope which can be acquired using a force sensor in the end-effector. In doing so, the orientation control can be used in the path recording stage where the path is still unknown.

Since the orientation of the end-effector is not stored during the teach-in procedure, the path can also be specified without using the robot. Combined with vision systems, the path can be adapted during the winding. This is useful to avoid collisions of the end-effector with the stretched rope or to cope with the position change of the rods.

The presented method could be used for flexible packaging. The operator specifies the desired path for wrapping an object that has an arbitrary shape. The only information

needed for determining the required orientation of the robot end-effector is the path. The tension of the material used for wrapping is specified as an independent parameter.

A Parameters

A.1 Robot Parameters

The non-zero DH-parameters of the robot are listed in Table A.1.

Parameter	Value	Unit
d_3	400	mm
d_5	390	mm

Table A.1: Non-zero DH-parameters of *KUKA LWR IV+*.

Table A.2 shows the joint limits of *KUKA LWR IV+*.

i	1	2	3	4	5	6	7
q_i^{max}	170°	120°	170°	120°	170°	120°	170°
q_i^{min}	−170°	−120°	−170°	−120°	−170°	−120°	−170

Table A.2: Upper and lower bounds of the joints of *KUKA LWR IV+*.

The robot parameters used in this work are the results of parameter identifications performed by Gaz *et al.* in [30] and [31].

A.2 Motor Parameters

The relevant parameters of the motor *EC 90 Flat* from *Maxon Motor* are listed in Table A.3.

Parameter	Value	Unit
Nominal voltage	36	V
Nominal current (max. continuous current)	4.76	A
Nominal torque (max. continuous torque)	560	mNm
Torque constant	109	mNm/A

Table A.3: Relevant parameters of the motor *EC 90 Flat*.

The motor current controller of the manufacturer is realized as a PI controller with feedforward part. The controller parameters listed in Table A.4 were determined in an auto-tuning process.

Parameter	Value	Unit
P-Gain	9.271	V/A
I-Gain	0.041	V/As
Feedforward	0.537	V/A

Table A.4: Control parameters of the motor *EC 90 Flat*.

A.3 Parameters of the Path Following Controller

The parameters of the path following controller are obtained using pole placement. The characteristic polynomial of the error dynamics of a linear second-order system is given by

$$p(s) = s^2 + a_1 s + a_0 \ . \tag{A.1}$$

By placing both poles of this system at λ the controller parameters are given by $a_0 = \lambda^2$ and $a_1 = -2\lambda$.

The poles λ_η, λ_ξ and λ_r chosen for the tangential, transversal and rotational subsystem, respectively, are listed in Table A.5. Herein, the poles are separately shown for the case of employing the Frenet-Serret (FS) frame and parallel transport (PT) frame for deriving the end-effector orientation.

	Simulation		Experiment	
Pole	FS	PT	FS	PT
λ_η	-10	-5	-5	-5
λ_ξ	-20	-12	-12	-12
λ_r	-10	-4	-4	-4

Table A.5: Poles for calculating the controller parameters.

Bibliography

[1] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control: the Computer Control of Robot Manipulators*, ser. MIT Press series in artificial intelligence. Cambridge, MA: MIT Press, 1981.

[2] M. Shimizu, H. Kakuya, W. K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical Inverse Kinematic Computation for 7-DOF Redundant Manipulators With Joint Limits and Its Application to Redundancy Resolution," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1131–1142, Oct. 2008.

[3] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Upper Saddle River, NJ: Pearson Education, 2005.

[4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. New York: John Wiley & Sons, 2005.

[5] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, ser. Springer Tracts in Advanced Robotics (STAR). Berlin Heidelberg: Springer-Verlag, 2008, vol. 49.

[6] A. Banaszuk and J. Hauser, "Feedback Linearization of Transverse Dynamics for Periodic Orbits," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, vol. 2, Dec. 1994, pp. 1639–1644.

[7] C. Nielsen and M. Maggiore, "On Local Transverse Feedback Linearization," *SIAM Journal on Control and Optimization*, vol. 47, no. 5, pp. 2227–2250, 2008.

[8] C. Nielsen, C. Fulford, and M. Maggiore, "Path Following Using Transverse Feedback Linearization: Application to a Maglev Positioning System," in *Proceedings of the American Control Conference*, Jun. 2009, pp. 3045–3050.

[9] A. Akhtar, S. L. Waslander, and C. Nielsen, "Path Following for a Quadrotor Using Dynamic Extension and Transverse Feedback Linearization," in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, Dec. 2012, pp. 3551–3556.

[10] M. Böck and A. Kugi, "Manifold Stabilization and Path-Following Control for Flat Systems with Application to a Laboratory Tower Crane," in *Proceedings of the 53rd IEEE Conference on Decision and Control*, Dec. 2014, pp. 4529–4535.

[11] A. Hladio, C. Nielsen, and D. Wang, "Path Following for a Class of Mechanical Systems," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2380–2390, Nov. 2013.

[12] R. J. Gill, D. Kulić, and C. Nielsen, "Spline Path Following for Redundant Mechanical Systems," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1378–1392, Dec. 2015.

[13] B. Bischof, T. Glück, and A. Kugi, "Combined Path Following and Compliance Control for Fully Actuated Rigid Body Systems in 3-D Space," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1750–1760, Sep. 2017.

[14] R. L. Bishop, "There is More than One Way to Frame a Curve," vol. 82, no. 3, pp. 246–251, Mar. 1975.

[15] T. Faulwasser, B. Kern, and R. Findeisen, "Model Predictive Path-Following for Constrained Nonlinear Systems," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec. 2009, pp. 8642–8647.

[16] V. A. Toponogov, *Differential Geometry of Curves and Surfaces - A Concise Guide*. Boston: Birkhäuser, 2006.

[17] A. D. Luca, G. Oriolo, and B. Siciliano, "Robot Redundancy Resolution at the Acceleration Level," *Laboratory Robotics and Automation*, vol. 4, pp. 97–106, 1992.

[18] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*. London: Springer-Verlag, 2009.

[19] C. de Boor, *A Practical Guide to Splines*. New York: Springer-Verlag, 2001.

[20] E. V. Shikin and A. I. Plis, *Handbook on Splines for the User*. Boca Raton: CRC Press, 1995.

[21] M. Takizawa, S. Kudoh, and T. Suehiro, "Method for Placing a Rope in a Target Shape and its Application to a Clove Hitch," in *Proceedings of the 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2015, pp. 646–651.

[22] T. V. Vinh, T. Tomizawa, S. Kudoh, and T. Suehiro, "A New Strategy for Making a Knot with a General-Purpose Arm," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2012, pp. 2217–2222.

[23] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Motion Planning for Dynamic Knotting of a Flexible Rope with a High-Speed Robot Arm," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2010, pp. 49–54.

[24] Y. Yamakawa, A. Namiki, and M. Ishikawa, "Simple Model and Deformation Control of a Flexible Rope using Constant, High-Speed Motion of a Robot Arm," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2012, pp. 2249–2254.

[25] E. Hultman and M. Leijon, "Robotized Stator Cable Winding," *Robotics and Computer-Integrated Manufacturing*, vol. 53, pp. 197–214, 2018.

[26] M. Prado, M. Dörstelmann, T. Schwinn, A. Menges, and J. Knippers, "Core-Less Filament Winding," in *Robotic Fabrication in Architecture, Art and Design 2014*, W. McGee and M. P. de Leon, Eds. Cham: Springer International Publishing, 2014, pp. 275–289.

[27] G. Schieber, V. Koslowski, J. Knippers, M. Dörstelmann, M. Prado, L. Vasey, and A. Menges, "Integrated Design and Fabrication Strategies for Fibrous Structures," in *Modelling Behaviour: Design Modelling Symposium 2015*, M. R. Thomsen, M. Tamke, C. G. and B. Faircloth, and F. Scheurer, Eds. Cham: Springer International Publishing, 2015, pp. 237–245.

[28] A. J. Hanson and H. Ma, "Parallel Transport Approach to Curve Framing," Department of Computer Science, Indiana University, Bloomington, IN, USA, Tech. Rep., 1995.

[29] A. J. Hanson, *Visualizing Quaternions*. San Francisco, CA, USA: Morgan Kaufmann Publishers, 2006.

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

[31] C. Gaz, F. Flacco, and A. D. Luca, "Extracting Feasible Robot Parameters from Dynamic Coefficients Using Nonlinear Optimization Methods," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 2075–2081.

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, November 2018

Ali Haddadi