Investigation and Implementation of Workspace restrictions for the KUKA LBR iiwa

Sebastain Schleisner Hjorth

Control and Automation

June 5, 2019

Master Thesis

STUDENT REPORT
AALBORG UNIVERSITY

AALBORG UNIVERSITY
STUDENT REPORT

Master Thesis - Control and Automation
Fredrik Bajers Vej 7
9220 Aalborg

Title:

Investigation and Implementation of Workspace restrictions for the KUKA LBR iiwa

Project period:
February 1st, 2019 - June 6th, 2019

Date of hand in:
June 6th, 2019

Author:
Sebastian Schleisner Hjorth

Supervisor:
Jan Dimon Bendtsen

Number of pages: 83

Synopsis:

This project concerns itself with the modeling, control and workspace restriction of a collaborative and redundant industrial manipulator. The manipulator used in this work is the KUKA LBR iiwa, for which the designed solution is verified in simulation as well as on the real manipulator The kinematics and dynamics of the KUKA LBR iiwa are modeled based on the screw theory approach. The KUKA LBR iiwa is controlled via an energy-aware impedance control. Furthermore, the project describes a method to which restricts the Cartesian workspace with the help of virtual walls and wrenches. In addition to these Cartesian boundaries, the project describes the implementation of the avoidance of joint boundaries with a feature called joint limit avoidance. The overall control strategy was for simulation purposes implemented in MATLAB and in the real world with the help of KUKAs own Fast-Research-Interface. The energy-aware impedance control was verified to work as intended. Furthermore, it was concluded that it is possible to restrict the Cartesian and Joint workspace of the KUKA LBR iiwa, with the help of potential fields, while it being in a compliant state.

Abstract

Dette projekt handler om modellering, kontrol og arbejdsområde begrænsning af en kollaborativ og redundanten industri robot. Robotten, som bliver brugt i dette projekt er den så kaldte KUKA LBR iiwa, som bliver styret via en energibesparende impedans kontroller. Kinematiken og dynamiken af KUKA LBR iiwaen er modelleret baseret på skrue-teori. Desuden beskriver projektet en metode, som begrænser det kartesiske arbejdsområde ved hjælp af virtuelle vægge og skuenøgler. Udover disse kartesiske grænser beskriver projektet implementeringen af undgåelse af KUKA LBR iiwaens led grænser med en funktion kaldet Joint Limit Avoidance. Den overordnede kontrolstrategi blev for simuleringsformål implementeret i MATLAB og i den virkelige verden ved hjælp af KUKAs eget Fast-Research-Interface. Den energibesparende impedans kontrol blev verificeret og fungere som ønsket. Endvidere blev det konkluderet, at det er muligt at begrænse det kaskesiske- og led-arbejdsområde i KUKA LBR iiwa ved hjælp af Potential Fields, mens det er i en kompatible tilstand.

Preface

This report was written in the spring of 2019, by Sebastian Schleisner Hjorth and documents the Master Thesis of the Control and Automation study at Aalborg University. It concerns the modeling of a robotic manipulator with help of screw theory, the implementation and validation of an Energy- and Power-based Impedance Controller and Cartesian workspace restriction on the KUKA LBR iiwa.
In order to understand the thesis a fundamental knowledge in Linear Algebra, Calculus, kinematic and dynamic modeling of serial manipulators is needed. Furthermore, a basic understanding of Lie groups in a robotics context is useful, although a short introduction to the most important theorems is given in this work.

The KUKA LBR iiwa was modeled, as well as the overall control strategy was tested in MATLAB®. The implementation of the control strategy on the real KUKA LBR iiwa was realized via a KUKA FRI Client written C++. The 3D model of the KUKA LBR iiwa which was used in some of the plots was taken from the Robotic System Toolbox from MATLAB® and the model-specific data (e.g. Inertia Tensor, link mass, center of mass, etc.) which was used to create an own model of the KUKA LBR iiwa was taken from an URDF file provided by `https://github.com/kuka-isir/iiwa_description`. With parts of section 2.1, 3.2 and Appendix C,**??** being taken from previous work done by the student in [1].
With the MATLAB code is available until 1st of July 2019 under the following link, `https://gitlab.com/Hjorth/lbr_iiwa_matlab_lib`. All source references throughout the report follow the IEEE referencing method, hence the references are stated by a number placed inside of square brackets (e.g.[1]). Important notation and abbreviations used in this work can be found after the preface.
A special thanks are given to the Professors Ole Madsen and Casper Schou, for giving me access to the KUKA LBR iiwa and their support during the project.

Sebastian Schleisner Hjorth

Nomenclature

List of Notation

$\xi, \widetilde{\xi}$	$\in \mathbb{R}^6$	body Twist and spatial Twist represented as a column vector
$\widehat{\xi}, \widehat{\widetilde{\xi}}$	$\in \mathbb{R}^{6\times 6}$	body Twist and spatial Twist in matrix form
ω	$\in \mathbb{R}^3$	Screw axis of the respective twist in form of a unit column vector
$\widehat{\omega}$	$\in \mathbb{R}^{3\times 3}$	Screw axis of the respective twist in form of a skew-symmetric matrix
$V, \widetilde{V}$	$\in \mathbb{R}^6$	body and spatial Cartesian Velocity in represented as a column vector
$\widehat{V}, \widehat{\widetilde{V}}$	$\in \mathbb{R}^{6\times 6}$	body and spatial Cartesian Velocity in matrix-form
$\mathbf{J}$		Jacobian matrix
W	$\in \mathbb{R}^{1\times 6}$	Wrench represented as a row vector
$f,\ m$	$\in \mathbb{R}^{1\times 3}$	Cartesian force and momentum represented as a row vector
E, T ,U		Total, kinetic and potential energy in $\frac{\mathrm{kg\cdot m^2}}{\mathrm{s^2}}$
P		Power in $\frac{\mathrm{kg\cdot m^2}}{\mathrm{s^3}}$
$q_i, \dot{q}_i$		Joint state and velocity of the i^{th} joint in rad,$\frac{\mathrm{rad}}{\mathrm{s}}$
τ_i		Joint torque of the i^{th} joint in Nm
p_j^i	$\in \mathbb{R}^3$	position vector between frame i and j
$\mathbf{R}_j^i$	$\in \mathbb{R}^{3\times 3}$	Rotation matrix between frame i and j
$\mathbf{H}_j^i$	$\in \mathbb{R}^{4\times 4}$	Homogeneous transformation matrix between frame i and j
$Ad_{\mathbf{H}_j^i}$	$\in \mathbb{R}^{6\times 6}$	Adjoint transformation between frame i and j
$\mathbf{M}$		Mass matrix
m_i		Mass of the i^{th} link in kg
$\boldsymbol{\mathcal{I}}$	$\in \mathbb{R}^{3\times 3}$	Inertia Tensor
$\mathbf{B}$		Damping matrix in Joint space
β		Damping scaling ratio
λ		Energy scaling ratio
v		Linear velocity in $\frac{\mathrm{m}}{\mathrm{s}}$
$\mathbb{Q}$		Configuration manifold
as()		Gives the skew-symmetric part of a square matrix
tr()		Tensor trace operator
$\mathbf{I}_n$	$\in \mathbb{R}^{n\times n}$	Identity matrix

List of Abbreviations

HRI Human-Robot-Interaction

COM Center of Mass

DOF Degree of Freedom

LBR Leicht-Bau-Roboter

iiwa intelligent industrial work assistant

FK Forward Kinematic

FRI Fast Research Interface

KRC KUKA Robot Controller

KLI KUKA Line interface

KONI KUKA Option Network Interface

Table of content

Introduction 1

When the first industrial manipulators were introduced to the world in the early 1960s [2]. They were only capable of completing simple pick and place or welding tasks. Since then a lot has changed, robotic solutions have revolutionized our way of living and are an essential part of our daily life. Most of them being serial manipulators in an industrial environment (e.g. production facilities) are also known under the phrase of *industrial manipulators*.

These industrial manipulators have come a long way since they were introduced to the world and applied in an industrial setting. Nowadays they are capable of completing a complex task with a velocity and an accuracy a human could never achieve. And the way how and where these manipulators are applied is already changing again.
In recent years the development of Collaborative serial manipulators, like the KUKA LBR iiwa (Figure 1.1), has been a hot topic in the industry. Previously industrial manipulators were restricted to operate in a static and well-defined environment behind a fence.

Figure 1.1: KUKA LBR iiwa [1]

These Collaborative robots or Co-bots are designed and controlled in such a way, that they are capable of sharing their workspace with the an autonomous quantity like a human. This change to the application environment brings new challenges and opportunities to the ways how a manipulator interacts with its environment.
The main challenges being the occurrence of unplanned interaction in the form of a collision or physical input by an autonomous quantity. In order to be able to cope with these kinds of interaction, so-called reactive control schemes can be implemented. These control schemes use the current state of the robot and the task description at each time step as input for the computation of the joint forces applied in the next time step.[3]

[1] `https://de.wikipedia.org/wiki/Datei:KUKA_LBR_iiwa.jpg`

Which compared to traditional robot control schemes (e.g. position and velocity control) is capable of dealing with a deviation between the manipulators current position and its planned trajectory, in case of an unplanned interaction introduced by an autonomous quantity, takes place. However, due to this freedom in its movement, the manipulator can collide with any objects in its reach. This is of course unwanted, as it could damage tools or products.
This raises the question, how can the manipulator be hindered from colliding with an object in its reach when being forced away from its pre-planned trajectory? One solution would be to limit the Cartesian workspace of a collaborative manipulator in such a way, that it can not collide with objects within its dexterous workspace.

The overall question the project tries to answer is:
How to restrict the Cartesian workspace of a redundant serial manipulator controlled by a reactive control scheme?

1.1 Concept/Use Case

The idea behind restricting the Cartesian workspace of a manipulator serves the purpose of being able to implement a manipulator in areas in which it can not utilize all of its dexterous workspace. One should imagine a production facility, in which the production line is already implemented and the company decides to support a worker in a specific task with the implementation of a Co-Bot like the KUKA LBR iiwa.
However as the production line already exists, one would have to redesign parts of the production line, in order to facilitate the manipulator. This would result in a substantial increase in cost. A visual representation of this problem can be seen in Figure 1.2, where for example there are machinery, floor, work-platform or simply the workspace of another robot in the manipulator's dexterous workspace.

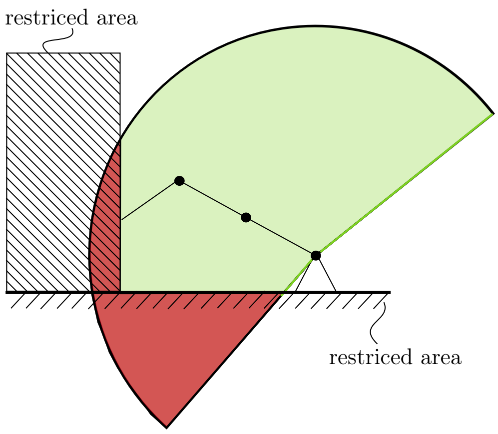

Figure 1.2: Illustrates an abstract visualization of a scenario in which there are fixed limitation/obstacle in the manipulator's dexterous workspace. The green area represents the part of the manipulator's dexterous workspace, in which it is allowed to move freely. With the red area representing the area of the robots dexterous workspace in which it is not allowed to move in.

The previously mentioned increase in cost might not be sustainable for the company. One might argue, that the above-stated problem might not occur, while the manipulator is following a preplanned trajectory. As the trajectory can be planned in such a way that the manipulator moves collision free within its workspace.
However, the application for the manipulator is to support the worker in its job, which means the worker has to interact with the manipulator. This raises the question, how should the Human-Robot-Interaction (HRI) be handled from a control point of view. The most intuitive approach for handling these interactions between the worker and the manipulator would be if the worker could freely interact with the manipulator without having to follow a predefined procedure (e.g. pressing a button). Such an approach can be realized by implementing a reactive control scheme for the control of the Co-bot.
However as the Worker interacts with the manipulator be it by pulling/pushing it away from its preplanned trajectory it might collide with one of the above-stated obstacles or reaches a joint limit, which would bring the robot to a standstill. In case one of the just described scenarios occur, valuable time is lost as the manipulator has to be reinitialized, the possible damage has to inspected and in the worst case components have to be replaced.
A more simple and cost-efficient solution for avoiding collisions with objects within the manipulator's dexterous workspace would be to restrict its the Cartesian workspace with help virtual walls. A abstract visualization of this concepts can be seen in Figure 1.3.

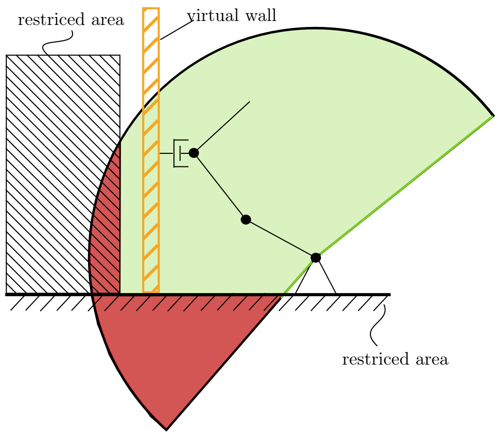

Figure 1.3: Illustrates an abstract of a scenario in which there are fixed limitation/obstacle in the manipulator's dexterous workspace. Where a predefined virtual wall is placed between the manipulator and the restricted area, with a Cartesian damper being placed between these to entities.

In which a predefined virtual wall is placed between the manipulator and the restricted area, between which a Cartesian damper is being placed. This damper forces manipulators link away from the constraint and thereby hindering the link from reaching the restricted area. When combining the concepts of Cartesian workspace restriction with the help of a virtual wall, a reactive control scheme and joint limit avoidance for the individual joints, it should be possible to avoid any of the above-stated scenarios while having an intuitive HRI.

1.2 Report Outline

The remain of this report is structured as follows:

- Chapter 2 elaborates on the main concepts needed for the mathematical modeling of a serial manipulator with screw theory. It contains a small summary of the general concepts within screw theory as well as the mathematical derivation of the kinematic and dynamic model.

- Chapter 3 focuses on the mathematical description of the in section 1.1 introduced concept, description of the chosen control scheme and on the concept of joint limit avoidance. This covers the design of simple virtual walls, their repelling force, a detailed description of the reactive control scheme used in this project and a detailed inside into the control strategy used for joint limit avoidance.

- Chapter 4 the reader is presented with the tests, which were conducted for the verification of the concept and its mathematical approach described in section 1.1 and chapter 3 respectively.

- Chapter 5 summaries the finding of the work and what conclusion can be drawn from them. As well as gives a short summary of some of the topics, which could be investigated in future projects.

Mathematical modelling of a Serial Manipulator 2

In this chapter, the reader is introduced to the main concepts for the mathematical modeling of the kinematic and dynamic behavior of the KUKA LBR iiwa. The modeling in this work is described with the help of screw theory, which as a concept was developed in [4] and used in the context of robotic application in [5, 6, 7]. The chapter is structured in the following way: *Kinematics*(2.1); *Dynamics*(2.2).

2.1 Kinematic model

This section discusses the theory behind the geometrical representation of the manipulator's motion without taking the applied forces into account.[8]
By first giving an overview of the basic concepts needed for describing the kinematic model of a serial manipulator with screw theory in subsection 2.1.1. Followed by the description of Forward Kinematic model in subsection 2.1.2 and Differential Kinematic model in subsection 2.1.3.

2.1.1 General Kinematic Concepts

It is well known in the fields of robotics, that in general any point p_j in a frame Ψ_j can be expressed relative to another frame Ψ_i with the help of a homogeneous transformation $\mathbf{H}_j^i$ as shown in Equation 2.1.

$$\begin{bmatrix} p^i \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{R}_j^i & p_j^i \\ \mathbf{0} & 1 \end{bmatrix}}_{\mathbf{H}_j^i} \begin{bmatrix} p^j \\ 1 \end{bmatrix} \tag{2.1}$$

The homogeneous transformation matrix $\mathbf{H}_j^i$ belongs to the Lie Group called *Special Euclidean Group* $SE(3)$ Equation 2.2

$$SE(3) = \left\{ \begin{pmatrix} \mathbf{R} & p \\ 0 & 1 \end{pmatrix} : \mathbf{R} \in SO(3), p \in \mathbb{R}^{3\times 1} \right\} \tag{2.2}$$

where p_i^j is the position and $\mathbf{R}_i^j$ the rotation of Frame Ψ_i with respect to frame Ψ_j. It is important to note that $\mathbf{R}_i^j$ is a member of the *Special Orthogonal Group* $SO(3)$ (2.3), which is also a Lie Group.[9]

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}^{-1} = \mathbf{R}^\top, \det(\mathbf{R}) = 1\} \tag{2.3}$$

The motion between these frames can be described with the *Charles theorem*, which states the following: *"Any rigid body motion can be accomplished by means of a rotation about a unique geometrical line in space, followed by a translation along the same line."* [10, 6, 5]. Where the mentioned line is the so-called *screw axis* ω and the resulting motion around the screw axis is referred to as the *screw motion*.

An infinitesimal screw motion is called a twist ξ, which describes the instantaneous velocity of this rigid body in terms of its linear and angular component.[6] In case the twist ξ describes a purely rotational displacement about its screw axis ω, it has the vector format and matrix-format as shown respectively in Equation 2.4 and Equation 2.5[5]. The vector can also be referred to as twist coordinates.

$$\xi = \begin{bmatrix} \nu \\ \omega \end{bmatrix} = \begin{bmatrix} -\omega \times u \\ \omega \end{bmatrix} \tag{2.4}$$

$$\widehat{\xi} = \begin{bmatrix} \widehat{\omega} & \nu \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \widehat{\omega} & -\omega \times u \\ 0 & 0 \end{bmatrix} \tag{2.5}$$

The mentioned linear and angular components of the instantaneous velocity ξ are respectively represent as the vectors $v \in \mathbb{R}^3$ and $\omega \in \mathbb{R}^3$. Where ω is a unit vector and ν is defined as the cross product between the screw axis ω and a point u on ω with respect to its reference frame, as seen in Figure 2.1.

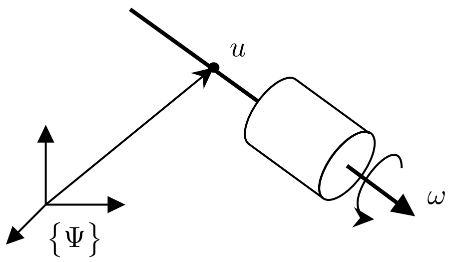

Figure 2.1: Shows an abstract visualization of the placement of a twist with in a rotational joint, in reference to a fixed frame Ψ.

In order to describe a relative motion of a rigid body, the expression in Equation 2.6 can be used. This expression describes the ridig bodies relative motion based on its initial and the twists exponential. A visualization of the mapping can bee seen in Figure 2.2

$$\mathbf{H}(q) = e^{\widehat{\xi}q}\mathbf{H}(0) \tag{2.6}$$

Where $e^{\widehat{\xi}q} \in SO(3)$ is the exponential of the twist ξ, which for a purely rotational joint has the structure seen in Equation 2.7 and is dependent on the angle of rotation $q \in \mathbb{R}$.

$$e^{\widehat{\xi}q} = \begin{bmatrix} e^{\widehat{\omega}q} & \left(\mathbf{I} - e^{\widehat{\omega}q}\right)\left(\omega \times \nu\right) \\ 0 & 1 \end{bmatrix} \tag{2.7}$$

In which $e^{\widehat{\omega}q}$, is the exponential map for a relative rotational motion from its initial to its final orientation around the screw-axis ω. As mentioned before $\omega \in \mathbb{R}^3$ is a unit vector specifying the direction of rotation and $q \in \mathbb{R}$ being the angle of rotation around this axis.[6]

$$e^{\widehat{\omega} q} = \mathbf{I} + q\widehat{\omega} + \frac{q^2}{2\,!}\widehat{\omega}^2 + \frac{q^3}{3\,!} + \ldots \tag{2.8}$$

The exponential map shown in Equation 2.8, is defined as an infinite series describing the integral action of $\omega \times q$ from time 0 to t. An alternative and computational more efficient formulation for this exponential map is shown in Equation 2.9, which in [6] and [5] is referred to as *Rodrigues formula.*

$$e^{\widehat{\omega} q} = \mathbf{I} + \widehat{\omega} \sin q + \widehat{\omega}^2 \big(1 - \cos q\big) \tag{2.9}$$

In other words $e^{\widehat{\omega} q}$, which describes the rotation around an axis ω by an angle q is equivalent to a rotation matrix $\mathbf{R}(\omega, q) = e^{\widehat{\omega} q}$ for $\mathbf{R} \in SO(3)$.[6]

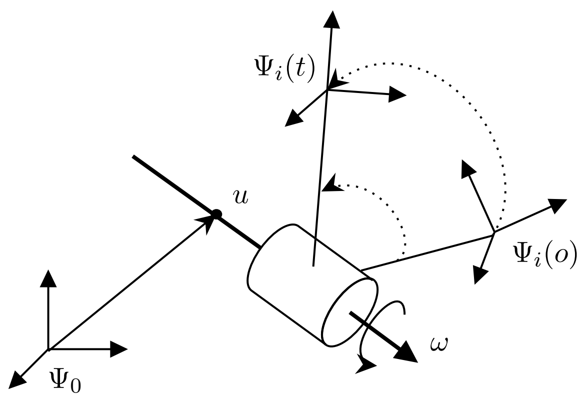

Figure 2.2: Shows an abstract visualization of relative motion of an rigid body from it initial configuration to its configuration at time, for a rotational joint, in reference to a fixed frame Ψ_0.

The above discussed concept concerned itself with the motion of rigid bodies in relation to a reference frame. Another important general concept for the screw-theory based modeling of an serial manipulator is: *How to express twists in another frame?*
This change of coordinates can be achieved by pre- and post-multiplying the twist $\widehat{\xi}$ by the respective homogeneous transformation matrix as seen in Equation 2.10.

$$\widehat{\xi}_i^{j,i} = \mathbf{H}_i^j \widehat{\xi}_i^i \mathbf{H}_j^i \tag{2.10}$$

However it can also be achieved by an *adjoint transformation* as shown in Equation 2.11.

$$\xi_i^j = Ad_{\mathbf{H}_i^j} \xi_i^i \tag{2.11}$$

Where $Ad_{\mathbf{H}_i^j} \in \mathbb{R}^{6\times 6}$ (Equation 2.12)is the so called adjoint transformation matrix for the homogeneous transformation $\mathbf{H}_i^j$ between frame Ψ_j and Ψ_i.

$$Ad_{\mathbf{H}_i^j} = \begin{bmatrix} \mathbf{R}_i^j & \widehat{p}_i^j \mathbf{R}_i^j \\ 0 & \mathbf{R}_i^j \end{bmatrix} \tag{2.12}$$

In other words $Ad_{\mathbf{H}_i^j}$ describes the relation between twists expressed in different frames.

2.1.2 Forward Kinematics

The **Forward Kinematics**, also known as the **Direct Kinematics** of a serial-manipulator, describes the geometric configuration of the *end-effector/tool center point* given the relative positions and orientations of each pair of adjoint links of the manipulator.[9]
The above-explained general concept in subsection 2.1.1 for describing the orientation and position of an arbitrary point p_j in another frame Ψ_i can be used to express the motion of a serial-manipulators link in relation to each other.
In case of a *n-link* open-chain manipulator as seen in Figure 2.3, where the pose of each link is represented by a frame Ψ_i, with the base-frame Ψ_0 being fixed.

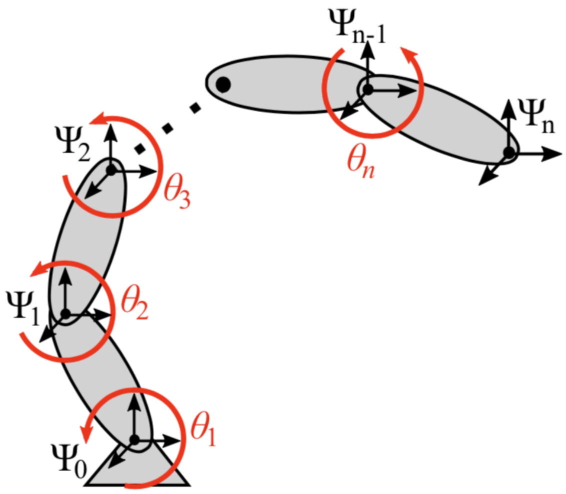

Figure 2.3: Illustrates an abstract representation of a serial kinematic chain. [11]

In order to describe the motion of each frame in reference to the base-frame Ψ_0 the chain rule is applied as seen in Equation 2.13.

$$\mathbf{H}_n^0(q_1, \ldots, q_n) = \mathbf{H}_1^0(q_1)\mathbf{H}_2^1(q_2)\ldots\mathbf{H}_n^{n-1}(q_n) \tag{2.13}$$

The translational and rotational motion of a rigid body described along the axis of a twist ξ is given by Equation 2.14, where $\mathbf{H}_i^{i-1}(0)$ is the reference configuration of the i^{th} joint and $e^{\widehat{\xi}_i q}$ (2.7) represents the motion along ξ.

$$\mathbf{H}_i^{i-1}(q) = e^{\widehat{\xi}_i^{i-1,i-1} q_i}\mathbf{H}_i^{i-1}(0) \tag{2.14}$$

By substituting each homogeneous transformation in Equation 2.13 with its equivalent part from Equation 2.14 the motion of each link is described by the twists of each joint.

$$\mathbf{H}_n^0(q_1, \ldots, q_n) = \underbrace{e^{\widehat{\xi}_1^{0,0} q_1}\mathbf{H}_1^0(0)}_{\mathbf{H}_1^0(q_1)}\underbrace{e^{\widehat{\xi}_2^{1,1} q_2}\mathbf{H}_2^1(0)}_{\mathbf{H}_2^1(q_2)}\ldots\underbrace{e^{\widehat{\xi}_n^{n-1,n-1} q_n}\mathbf{H}_n^{n-1}(0)}_{\mathbf{H}_n^{n-1}(q_n)} \tag{2.15}$$

Now in order to describe each links motion in reference to the Ψ_0-frame. The identity property of $\mathbf{H}$ is used to express the relation between the sequential links in the Ψ_0-frame as seen in Equation 2.16.

In Equation 2.16 the homogeneous transformations between each of its neighbouring link are mapped to the Ψ_0-frame by using the identity property of $\mathbf{H}$, resulting in Equation 2.17 the so called *Brockett's product of exponential formula*[6][9].

$$\mathbf{H}_n^0(q_1,\dots,q_n) = e^{\hat{\xi}_1^{0,0}q_1}\underbrace{\mathbf{H}_1^0(0)e^{\hat{\xi}_2^{1,1}q_2}\mathbf{H}_0^1(0)}_{e^{\hat{\xi}_2^{0,1}q_2}}\underbrace{\mathbf{H}_1^0(0)\,\mathbf{H}_2^1(0)e^{\hat{\xi}_3^{2,2}q_3}\mathbf{H}_3^2(0)\,\mathbf{H}_0^3(0)}_{e^{\hat{\xi}_3^{0,2}q_3}}\mathbf{H}_3^0(0)$$

$$\dots\underbrace{e^{\hat{\xi}_n^{n-1,n-1}q_n}\mathbf{H}_n^{n-1}(0)\,\mathbf{H}_0^n(0)}_{\hat{\xi}_n^{0,n-1}q_n}\mathbf{H}_n^0(0) \tag{2.16}$$

$$\mathbf{H}_n^0(q_1,...,q_n) = e^{\hat{\xi}_1^{0,0}q_1}e^{\hat{\xi}_2^{0,1}q_2}e^{\hat{\xi}_3^{0,2}q_3}\dots e^{\hat{\xi}_n^{0,n-1}q_n}\mathbf{H}_n^0(0) \tag{2.17}$$

Where $e^{\hat{\xi}_n^{0,n-1}q_n}$ is the transformation matrix between in Ψ_{n-1} and Ψ_n expressed in the Ψ_0. For the LBR iiwa the homogeneous transformation of the Ψ_{tcp}-frame expressed and seen in the 0-frame H_{tcp}^0, in its initial position can be seen in Equation 2.18.

$$\mathbf{H}_{tcp}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1266 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.18}$$

A visualization of the placement of the frames for the KUKA LBR iiwa can be seen in Figure 2.4. With the general Rotation-matrix $\mathbf{R}_{tcp}^0$ (**??**) and position-vector p_{tcp}^0 (**??**) of the LBR iiwa, as well as the spatial twist to calculate them can be found in **??**.

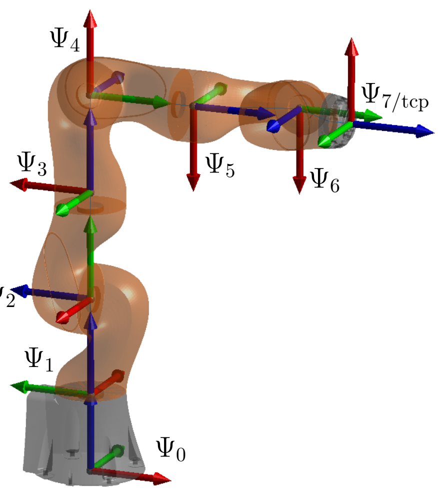

Figure 2.4: A visualiztions of the KUKA LBR iiwa with its body frames, where the blue arrow represents the frames z-axis, green its x-axis and red its y-axis.

2.1.3 Differential Kinematics

The subsection 2.1.2 presented the reader on how to calculate the end-effector frame's position and orientation for a given joint configuration of a serial-manipulator. This section discusses the relationship between the Cartesian velocities and joint velocities of a serial manipulator via the so-called **Jacobian**.

General

The Differential Kinematics of a manipulator can be derived with the help of one of the two following approaches, the **geometric Jacobian** and the most commonly known **analytic Jacobian**.
These two approaches are fundamentally different in their derivation. The geometric Jacobian is based on the manipulator's geometrical structure, which is represented with twists from the previously mentioned screw theory. Whereas, the analytic Jacobian is derived by taking the time-derivative of the joint coordinates $q \in \mathbb{R}^n$, which are found by differentiating the Forward Kinematics of the manipulator.[5]
In this work, the geometric approach is used as it is directly related to the approach used in subsection 2.1.2. The geometric Jacobian can be split into two different kinds, the **spatial Jacobian $\widetilde{\mathbf{J}}$** and **body Jacobian $\mathbf{J}$**. The body Jacobian describing the links velocity in reference to its body frame and the spatial Jacobian describing the links spatial velocity in reference to the base-frame.

Spatial Jacobian

In this section the relationship between a joint velocity $\dot{q}$ and the end-effector's spatial velocity given by the twist $\widetilde{\xi}$ is discussed.
The velocities $\dot{q}_i$ in joint space are expressed as the end-effector instantaneous spatial velocity $\widetilde{V}_n$ by mapping them with the spatial Jacobian $\widetilde{\mathbf{J}}(q) \in \mathbb{R}^{6\times n}$ as seen in Equation 2.19. In other words we look at the end-effectors Cartesian velocity in respect to the fixed inertial frame Ψ_0 and the joint velocities $\dot{q} \in \mathbb{R}^{n\times 1}$, with n being equal to the number of joints.

$$\widetilde{V}_n = \widetilde{\mathbf{J}}_n(q)\dot{q} \tag{2.19}$$

Where the i^{th} column of $\widetilde{\mathbf{J}}(q)$ Equation 2.20 is the i^{th} joint spatial twist $\widetilde{\xi}_i$ of the manipulators current configuration.

$$\widetilde{\mathbf{J}}_n(q) = \left[\widetilde{\xi}_1, \widetilde{\xi}_2, \ldots, \widetilde{\xi}_n\right] \qquad \text{Where} \qquad \widetilde{\xi}_i = \xi_i^{0,i-1} = Ad_{\mathbf{H}^0_{i-1}} \xi_i^{i-1,i-1} \tag{2.20}$$

One might see that the spatial twists described in Equation 2.20 are the same as used in Brockett's product of exponential in Equation 2.17, which means that is the i^{th} column of the spatial Jacobian represents the i^{th} joint twist, mapped to the current configuration of the serial-manipulator.[6]
Giving it the property of finding the initial joint twists by inspection only. A visual representation of the initial joint twists can be seen in Figure 2.5 and as mentioned in subsection 2.1.2 the spatial twists can be found in **??**.

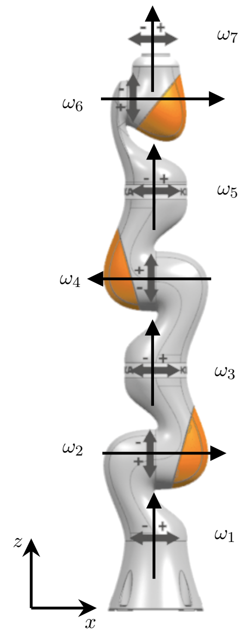

Figure 2.5: Illustrates the placement of the spatial screw-axis $\tilde{\omega}_i$ of the initial spatial joint twists $\widetilde{\xi}_i$.[12]

2.1.4 Body Jacobian

The body Jacobian $\mathbf{J}_n$ (Equation 2.21) is described by the body twist ξ, which describe the instantaneous Cartesian velocity of a point represented in the end-effector frame in its current configuration.[6]

$$V_n = \mathbf{J}_n(q)\dot{q} \tag{2.21}$$

As the body Jacobian $\mathbf{J}_n$ as well as the spatial Jacobian $\tilde{J}$ describe the instantaneous Cartesian velocity of the end-effector just from a different reference frame, it is possible to derived body Jacobian from the spatial Jacobian. This means that the spatial twist and body twist can be related to each other with an Adjoint transformation as seen in Equation 2.22.

$$\mathbf{J}_n(q) = Ad_{\mathbf{H}_0^n(q)}\widetilde{\mathbf{J}}_n(q) \quad \text{Where} \quad \xi_i = Ad_{\mathbf{H}_0^i(0)}\widetilde{\xi}_i \tag{2.22}$$

This relation is important and will be used later in this work for the derivation of the dynamic model described in section 2.2.

2.2 Dynamic model

This section concerns itself with the formulation of the dynamics of a serial-manipulator. In a general sense, dynamics concerns itself with how the motion of rigid mechanisms is generated/influenced by applying forces to the mechanism.[13]

The section will first give an overview of the basic concepts needed for describing the dynamic model of a serial manipulator with screw theory in subsection 2.2.1. Followed by the derivation of the general equitation of motion with the Newton-Euler approach in subsection 2.2.2, which is based on concepts introduced in [5, 7].

2.2.1 General Dynamic Concepts

Previously in subsection 2.1.1 the reader was introduced to the general concept of screw theory for describing the kinematic model of an open-chain manipulator. In this section, the reader will be introduced to the general concept for describing the dynamic model an open-chain manipulator with screw theory.

In screw theory, the forces acting on a rigid body can be represented by a single force along the screw-axis in combination with a momentum [6]. This force is called a *wrench* $W = [f, m]$,which is dual to the twists.[14]
The dual only states that the theorems applied to twist can also be applied to wrenches [6]. This dual is described with the so-called Poinsot's theorem, which shows that every wrench is equivalent to a force $f \in \mathbb{R}^{1\times 3}$ and momentum $m \in \mathbb{R}^{1\times 3}$ applied along the same screw axis as the twist.[6, 4]
Due to this theorem, it is possible to map a Wrench into joints torques, which can be described with the help of the transposed Jacobian as seen in Equation 2.26.
The relation between wrenches and torques can be derived by looking at the work $\mathbb{W}$ (Equation 2.23) generated due to the displacement of the end-effector through an applied wrench over the time interval $t \in [t_1, t_2]$.

$$\mathbb{W} = \int_{t_1}^{t_2} WV \; dt \tag{2.23}$$

Where V is the Cartesian velocity of the body and the work performed due to the wrench is assumed to be frictionless. However the work $\mathbb{W}$ can also be expressed in joint space in terms of the joint velocities $\dot{q}$ and joint torques $\tau \in \mathbb{R}^{1\times n}$ as seen in Equation 2.24.

$$\int_{t_1}^{t_2} \tau\dot{q} \; dt = \mathbb{W} = \int_{t_1}^{t_2} WV \; dt \tag{2.24}$$

As the relationship between the joint-space and Cartesian-space holds for any time interval, they must be equal can therefore (2.24) can be rewritten like in (2.25).[6]

$$\begin{aligned}
\tau\dot{q} &= WV \\
&\Downarrow \\
\tau\dot{q} &= W\big(\mathbf{J}(q)\dot{q}\big) \\
&\Downarrow \\
\big(\tau\dot{q}\big)^\top &= \big(W\big(\mathbf{J}(q)\dot{q}\big)\big)^\top \\
&\Downarrow \\
\dot{q}^\top\tau^\top &= \big(\mathbf{J}(q)\dot{q}\big)^\top W^\top \\
&\Downarrow \\
\dot{q}^\top\tau^\top &= \dot{q}^\top\mathbf{J}(q)^\top W^\top
\end{aligned} \tag{2.25}$$

From Equation 2.25 one can derived Equation 2.26, as $\dot{q}$ only specifies the magnitude and direction and therefore has no influence on the location of the applied torque or wrench.[6][5]

$$\tau^\top = \mathbf{J}(q)^\top W^\top \tag{2.26}$$

Note that wrenches can like the twists introduced in subsection 2.1.1, be expressed in reference to different frames. The ones being in reference to the fixed base-frame are denotated as $\widetilde{W}$ and the ones in reference to the respective body-frame as W. Where $\widetilde{W}$ is known as *spatial wrench* and W as *body wrench*.

Therefore the just described relation between the body wrench W and the torques τ also holds for the relation between the spatial wrench $\widetilde{W}$ and τ. Furthermore, W and $\widetilde{W}$ are related to each other by an adjoint transformation as seen in Equation 2.27, which describes a change of coordinates.

$$W^\top = Ad_{\mathbf{H}}^\top \widetilde{W}^\top \tag{2.27}$$

This relation means that as $\widetilde{W}$ and W describe the same force acting on the same rigid body the Work $\mathbb{W}$ generated by the pair $\widetilde{W}$ and $\widetilde{V}$, is equal to the work generated by W and V. As they only differ in their point of reference. In other words the $\tau^\top$ computed in Equation 2.26 is equal to the one resulting from $\widetilde{\mathbf{J}}(q)^\top \widetilde{W}^\top$.

2.2.2 General Equation of Motion

The previous section introduced the reader to general concepts within dynamic modeling, in the following section these concepts will be used to describe the derivation of the general equation of motion of the KUKA LBR iiwa.
The general equation of motion for a n-link serial manipulator can be represented in joint space, as a non-linear second-order differential equation in the canonical form seen in Equation 2.28.[14]

$$\mathbf{M}(q)\ddot{q} + \bar{\mathbf{C}}(q,\dot{q})\dot{q} + \bar{\mathbf{G}}(q) = \tau^\top \tag{2.28}$$

With q,$\dot{q}$ and $\ddot{q} \in \mathbb{R}^n$ being the generalized joint position, velocity and acceleration respectively, $\mathbf{M}(q) \in \mathbb{R}^{n\times n}$ is the mass matrix in from of a symmetric positive-definite matrix, $\bar{\mathbf{C}}(q,\dot{q})\dot{q} \in \mathbb{R}^n$ the centrifugal and Coriolis torques, $\bar{\mathbf{G}}(q)$ the gravitational torques and $\tau^\top$ the equivalent torques in the joint space.[5, 15, 8]

The general equation of motion in Equation 2.28 is typically derived by either the energy-based Lagrangian formalism or the force-balance-based Newton-Euler formalism.
As mentioned above the Euler-Lagrangian formulation derives the general equation of motion from the kinetic and potential energy of the manipulator, as seen in Equation 2.29.[5, 15, 8]

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}(q,\dot{q})}{\partial \dot{q}} - \frac{\partial \mathcal{L}(q,\dot{q})}{\partial q} = 0 \quad \text{Where} \quad \mathcal{L}(q,\dot{q}) = T(q,\dot{q}) - U(q) \tag{2.29}$$

With $T(q,\dot{q})$ and $U(q)$ being the total kinetic and potential energy of the manipulator and $q \in \mathbb{R}^n$ the general coordinates in form of the manipulators joint variables. However, as mentioned in [5] the derivation of Equation 2.28 quickly becomes computational demanding as the complexity of manipulator increases. For this reason, it was decided to use the Newton-Euler approach described in [5].

Opposed to the Lagrangian formalism, the Newton-Euler approach is a recursive algorithm. And is used for calculation of the inverse dynamics for a serial manipulator. This algorithm is split up into the following two stages:

- ***Stage 1*** **Forward iteration:** Computes the Cartesian velocity and acceleration of each link, starting from the base-frame Ψ_0 and iterating to the end-frame Ψ_{tcp}.
- ***Stage 2*** **Backwards iteration:** Computes the Wrenches applied to each link, starting from the end-frame Ψ_{tcp} and iterating to the base-frame Ψ_0.

The remaining of this section will be split into the two separate parts for the above listed *Stages 1* and *2*.

Forward iteration

At first a body/link-specific reference frame Ψ_i $(i = 1 \dots n)$ is attached at the CoM of each link, where the base-frame and end-frame are denoted as Ψ_0 and Ψ_{n+1}. In which the Inertia of the i^{th} link M_{c_i} is defined as shown in Equation 2.30, with m_i being the mass and $\boldsymbol{\mathcal{I}}_i$ Inertia tensor of the link.

$$\mathbf{M}_{c_i} = \begin{bmatrix} m_i\mathbf{I}_3 & 0 \\ 0 & \boldsymbol{\mathcal{I}}_i \end{bmatrix} \tag{2.30}$$

In the next step, one has to express the twist of the i^{th} joint in the body-specific reference frame Ψ_i, which denoted as ξ_i. This can achieved by using the relation described in Equation 2.22, where it is stated how ξ_i is directly related to the spatial expression of the same joint $\widetilde{\xi_i}$. The body twist ξ_i is then used to calculate the homogeneous transformation $\mathbf{H}^i_{i-1}(q)$ in Equation 2.31.

$$\mathbf{H}^i_{i-1}(q) = \left(\mathbf{H}^{i-1}_i(q)\right)^{-1} = \left(\mathbf{H}^{i-1}_i(0)e^{\widehat{\xi_i}q_i}\right)^{-1} \tag{2.31}$$

With the help this expression it is now possible to compute the instantaneous Cartesian velocity V_i. As seen in Equation 2.32, V_i is defined as the sum of the instantaneous Cartesian velocity of the previous link denoted as V_{i-1} and instantaneous Cartesian velocity introduced by the joint-rate $\dot{q}_i$ around the twist ξ_i.

$$V_i = Ad_{\mathbf{H}^i_{i-1}(q)}V_{i-1} + \xi_i\dot{q}_i \tag{2.32}$$

In the last step of the forward iteration is the calculation of the instantaneous Cartesian acceleration $\dot{V}_i$ (Equation 2.33).[5] The exact derivation of can be found in Appendix A. In Equation 2.33, $\dot{V}_i$ is defined as the sum of the joint-acceleration $\ddot{q}$ around the twist ξ_i, the instantaneous Cartesian acceleration of the previous link denoted as V_{i-1} mapped into the Ψ_i frame and

$$\begin{aligned} \dot{V}_i &= \xi_i\ddot{q}_i + Ad_{\mathbf{H}^i_{i-1}(q)}\dot{V}_{i-1} + [V_i, \xi_i\dot{q}_i] \\ &= \xi_i\ddot{q}_i + Ad_{\mathbf{H}^i_{i-1}(q)}\dot{V}_{i-1} + ad_{V_i}\xi_i\dot{q}_i \end{aligned} \tag{2.33}$$

With the operator $[\cdot,\cdot]$ found in Equation 2.34 being the so called **Lie Bracket** operator, which can be seen as a generalization of the cross-product on $\mathbb{R}^3$ to a twists in $\mathbb{R}^6$.

$$[V_i,V_j] = V_iV_j - V_jV_i = ad_{V_i}V_j \quad \text{Where} \quad ad_{V_i} = \begin{bmatrix} \widehat{\omega}_i & \widehat{\nu}_i \\ 0 & \widehat{\omega}_i \end{bmatrix} \tag{2.34}$$

Where ad_{V_i} describes the twist V_i in form of 6×6 matrix. A derivation of this generalization can be found in [6] and [5].

Backwards iteration

After the twists V and accelerations $\dot{V}$ for each link are computed, the algorithm calculate recursively the wrenches applied at each link and the resulting joint torques by iterating backward from the end-effector to the base.
The first step of each iteration is to calculate the *spatial momentum* $\boldsymbol{\mathcal{X}}_i$ of the link, is defined as seen in Equation 2.35.

$$\boldsymbol{\mathcal{X}}_i = \mathbf{M}_{c_i} V_i \tag{2.35}$$

Followed by the computation of the wrench acting on the i^{th}-link given the twists V_i and accelerations $\dot{V}_i$ in Equation 2.36, which describes the dynamic equations for a single rigid body.

$$\begin{aligned} W_i^\top &= \mathbf{M}_{c_i} \dot{V}_i - ad_{V_i}^\top (\boldsymbol{\mathcal{X}}_i) \\ &= \mathbf{M}_{c_i} \dot{V}_i - ad_{V_i}^\top (\mathbf{M}_{c_i} V_i) \end{aligned} \tag{2.36}$$

The next step computes the total wrench W_i acting on the i^{th} link. To achieve this, one has to include the wrench, which is applied by the previous link $i+1$ on the *ith*-link (Equation 2.37).

$$\mathbf{M}_{c_i} \dot{V}_i - ad_{V_i}^\top (\mathbf{M}_{c_i} V_i) = W_i^\top - Ad_{\mathbf{H}_i^{i+1}}^\top W_{i+1}^\top \tag{2.37}$$

This equation then solve for Wrench W_i as seen in Equation 2.38, in order to describe the total wrench acting on the i^{th}-link. Which after the reformulation is defined by the sum of the wrench being applied to the link through the previous joint $i+1$ and the wrench resulting from the twist V_i and acceleration $\dot{V}_i$, those wrenches being expressed in Ψ_i.

$$W_i^\top = \mathbf{M}_{c_i} \dot{V}_i - ad_{V_i}^\top (\mathbf{M}_{c_i} V_i) + Ad_{\mathbf{H}_i^{i+1}}^\top W_{i+1}^\top \tag{2.38}$$

The result of Equation 2.38 is then multiplied by the body twist of the respective link ξ_i, as in Equation 2.39, resulting the joint torques acting on the i^{th}-joint.

$$\tau_i = \xi_i W_i^\top \tag{2.39}$$

A visualization of the velocities, acceleration and Wrenches applied to a link by it's neighbouring links is given in Figure 2.6.

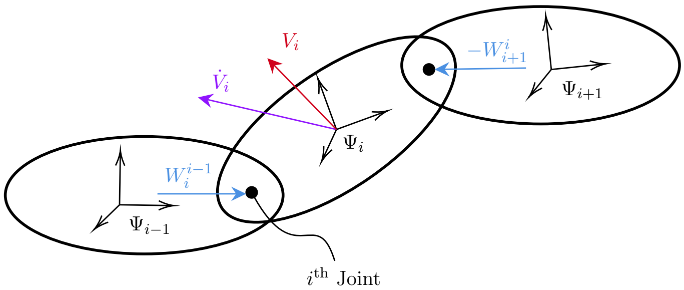

Figure 2.6: Shows an abstract visualization of the working wrenches and resulting twists on the links.

From this it is possible to derive the canonical form as described in Equation 2.28, which the interested reader can find in Appendix B

This concludes the derivation of the Mathematical modeling of an open-chain manipulator and we can focus on how to control such a system.

Controller 3

As mentioned in chapter 1, the aim of this project was to investigate the possible use of a reactive control scheme for the **KUKA LBR iiwa** in combination with the Cartesian workspace restriction in form of virtual walls.
In the previous chapters, the reader got introduced to the general concept of virtual walls (subsection 3.1.1) and the approaches used to mathematically model the KUKA LBR iiwa (chapter 2).

The reader will furthermore get an detailed insight into the mathematical description of the previously introduced concept (subsection 3.1.1) in section 3.1, as well as the chosen control strategy in section 3.2, the concept of joint limit avoidance in section 3.5 and the computation of the final control input in section 3.6.

3.1 Cartesian Constraint Control

As mentioned previously this section will discuss the description of the previously introduced concept of Cartesian workspace restriction with the help of virtual walls. This section will provide the reader with a more detailed overview of the in section 1.1 described concept. Followed by the mathematical description of the virtual walls (subsection 3.1.2), the placement of the Cartesian damper (subsection 3.1.3) and the definition of the repelling force generated to keep the manipulator its boundaries in subsection 3.1.4.

3.1.1 Overview

In this part of the section, the reader is introduced to the general mathematical interpretation of the above-described concept. In pursuance of keeping the manipulator within the set boundaries, it is of great importance to know:

1. How are the Constraints/Virtual Walls defined?
   - In general virtual walls can be described by any smooth manifold, however, in this work, it was decided to model the virtual walls as 2D-planes. The mathematical description can be found in subsection 3.1.2.
2. What distance do the restricted links of the manipulator have to the Constraints at time t?
   - This distance can be found by projecting the point position of the manipulator's links onto the constraint.
3. How should the manipulator approach the constraints?
   - The repelling force from the constraints should be introduced gradually to the affected links, in order to minimize abrupt behavior.

From these questions, it is possible to define a more precise interpretation of the in **??** previously defined concept. As mentioned above the virtual walls are defined as 2D-planes within 3D-workspace of the manipulator. With the knowledge of the position of this plane and the position of each link relative to the manipulator's world frame; it is possible to find position and distance between these two entities. From there it is possible to define a function, which gradually increases the gain of the repelling force of the constraint. As mentioned previously this is done in pursuance of minimizing the non-linear behavior of the repelled link, when encountering a constraint. However it the movements of the manipulator should not be affected by the constraint as long as it is outside a certain perimeter. A visual representation of this interpretation can be seen in Figure 3.1.

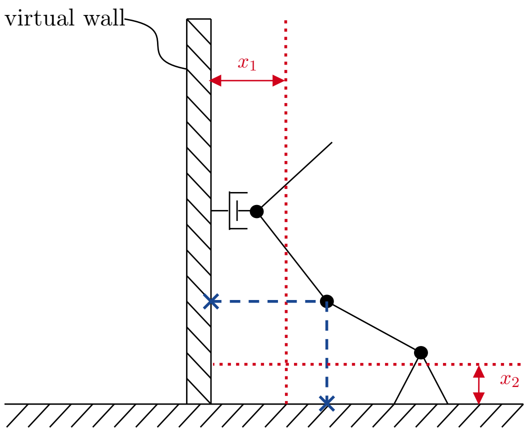

Figure 3.1: Visualizes an abstract visualization of the Concept with its main variables and features. Where x_j are the activation distances of the respective constraints the lines attached to the second are the projection of link onto the constraints.

Where the crosshatched areas are representing virtual constraints, the blue dashed lines and crosses visualize projection-line and the projected points of the manipulators link onto the constraints. With the red dotted line being the distance to the constraint at which the repelling force of the constraint (represented as damper) is activated/deactivated. In the remaining of this section, the mathematical description of the above-described concept is being given.

3.1.2 Constraints/Virtual Walls

The following section gives a generalized mathematical description of the creation of Cartesian constraints in the form of virtual walls. In general virtual walls can be described by any smooth manifold $C \in \mathbb{R}^3$. As mentioned earlier, in this work the virtual walls will be described in the form of 2D-planes in the Cartesian workspace. In order to create such a plane one has to chose three independent points $P_1, P_2, P_3 \in \mathbb{R}^3$, which span a vector space $C_j \in \mathbb{R}^3$. From these 3 independent points one derives the normal vector $\vec{n}$ as seen in Equation 3.1, which then in combination of a chosen origin P_o is used to described any point on this plan (Equation 3.2). A sensible choice for the origin would be any of these three points P_1, P_2, P_3 as they are known to be $\in C_j$.

$$\vec{n}_j = \overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3} = \begin{bmatrix} n_{j_x} \\ n_{j_y} \\ n_{j_z} \end{bmatrix} \tag{3.1}$$

This vector space can be described with Equation 3.2

$$n_{j_x}x + n_{j_y}y + n_{j_z}z + \vec{n}_j \cdot P_o = 0 \tag{3.2}$$

With being able to describe with Equation 3.2 any point on the plane/constraint. The next step is to find the point of representing links point position on this plane, which is done by projecting this point position onto the plane.

3.1.3 Projection

The projection of any arbitrary point p_i in space, onto the plane C_j can be described by Equation 3.3. Which describes the projection p_{i,C_j} on the plane by the sum of the point p_i and the product of the normalized normal vector of the plane $\vec{n}_j$ and the Scalar t.

$$p_{i,C_j} = p_i + \vec{n}_j t = \begin{bmatrix} p_{i_x} \\ p_{i_y} \\ p_{i_z} \end{bmatrix} + \begin{bmatrix} n_{j_x} \\ n_{j_y} \\ n_{j_z} \end{bmatrix} t \tag{3.3}$$

Where t can be seen as a form of distance measure between the point p_i and the plane C_j. Which can be found as seen in Equation 3.4, by substituting x, y, z in Equation 3.2 by the xyz-coordinates of p_{i,C_j} as defined in Equation 3.3.

$$\begin{aligned}
0 =& n_{j_x}\left(p_{i_x} + n_{j_x}t\right) + n_{j_y}\left(p_{i_y} + n_{j_y}t\right) + n_{j_z}\left(p_{i_z} + n_{j_z}t\right) + \vec{n_j} \cdot P_o \\
0 =& n_{j_x}p_{i_x} + n_{j_x}n_{j_x}t + n_{j_y}p_{i_y} + n_{j_y}n_{j_y}t + n_{j_z}p_{i_z} + n_{j_z}n_{j_z}t + \vec{n_j} \cdot P_o \\
0 =& t\left(n_{j_x}n_{j_x} + n_{j_y}n_{j_y} + n_{j_z}n_{j_z}\right) + n_{j_x}p_{i_x} + n_{j_y}p_{i_y} + n_{j_z}p_{i_z} + \vec{n_j} \cdot P_o \\
t =& -\frac{n_{j_x}p_{i_x} + n_{j_y}p_{i_y} + n_{j_z}p_{i_z} + \vec{n_j} \cdot P_o}{n_{j_x}^2 + n_{j_y}^2 + n_{j_z}^2}
\end{aligned} \tag{3.4}$$

The resulting value of t is then simply inserted into Equation 3.3, which results in the projected point p_{i,C_j} of point p_i on the plane C_j, a visualization of this can be seen in Figure 3.2.

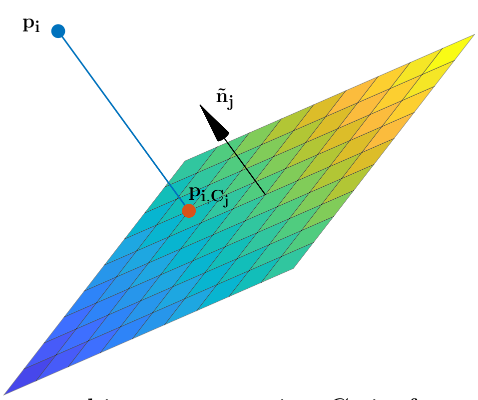

Figure 3.2: Visualizes an arbitrary constraint C_j in form of a plane, with its normal vector $\vec{n}_j$ and the projection p_{i,C_j} onto the plane of position of an arbitrary point p_i.

In order to put the above described mathematical derivation into context to an manipulator, the following changes in notation are necessary. Therefore for the remainder of this work, p_i will be referred to as p_i^0, which describes the position of the i^{th} link in $\mathbb{R}^3$ in reference to the manipulators base-frame. In addition p_{i,C_j} will be referred to p_{i,C_j}^0 and describes the projection of the i^{th} link onto the j^{th} constraint C in reference to the manipulators base-frame.

3.1.4 Approach function

In this work, the enforcement of the above-defined constraints is done by the help of a modified version of an artificial repulsive potential field. Firstly introduced in [16] by O.Khatib for real-time obstacle avoidance of industrial manipulators. The general concept behind the *artificial potential field approach* can be summarized by the following statement made by Khatib in [16]: *"The manipulator moves in a field of forces. The position to be reached is an attractive pole for the end-effector, and obstacles are repulsive surfaces for the manipulator parts."*. It has the beneficial properties of being real-time applicable as well as describing the approach towards a constraint by a non-negative and smooth function. These are important properties as it makes it possible to gradually decrease the manipulator's motion and thereby being able to meet the manipulator's stability conditions.[16]
As the work does not focuses on the path planning itself, but rather on the limitation of the manipulator's Cartesian workspace, while being in a compliant state; the focus within the topic of artificial potential field approach will be limited to repulsive potential fields for the remaining of the work. The repulsive potential field is defined by the potential function as described by Equation 3.5, which in this application serves as a transition function between the free and restricted motion. This function is defined as a non-negative smooth surface for any given joint configuration q. The resulting potential $U_{C_j,i}(q)$ (Equation 3.5) increase towards infinity as the i^{th} constraint link of the manipulator approach the constraint C_j.

$$U_{C_j,i}(q) = \begin{cases} \frac{\kappa_j}{\gamma}\left(\frac{1}{d_{i,C_j}(q)} - \frac{1}{x_j}\right)^{\gamma} & \text{if } d_{i,C_j}(q)) \leq x_j \\ 0 & \text{if otherwise} \end{cases} \tag{3.5}$$

Where $\kappa_j > 0$ is a scaling factor for the potential generated by $\gamma > 0$, $d_{i,C_j}(q)$ is the shortest euclidean distance between the points p_i^0 and $p_{i,C_j}^0 \in C_j$ and where x_j is the distance at which the constraint activates. The repelling force introduced to keep the link way from the restricted area can be defined as a Wrench $W_{C_j}^{i,i}$ acting on the respective link can be derived as seen in Equation 3.6. Where its derivation is based on the concepts described in [17].

$$W_{C_j}^{i,i\top} = \begin{bmatrix} f_{C_j}^{i,i\top} \\ m_{C_j}^{i,i\top} \end{bmatrix} \tag{3.6}$$

where $f_{C_j}^{i,i} \in \mathbb{R}^3$ (3.8)and $m_{C_j}^{i,i} \in \mathbb{R}^3$ (3.9) represent the force and momentum applied by the constraint C_j onto the i^{th} body [13]. The potential $U_{C_j,i}$ is then used to calculate the gain $\mathbf{Q}_{t_i}$ which represents the co-stiffness for a translational repelling force and is used for the computation of the repelling force $W_{C_j}^{i,i}$ described in Equation 3.8 and Equation 3.9.

$$\mathbf{Q}_{t_i}(q) = \mathbf{I}_3 U_{C_j,i}(q) \tag{3.7}$$

As it is not necessary to generate a repelling force for the rotational part, $\mathbf{Q}_{r_i}$ and $\mathbf{Q}_{c_i}$ are set equal to 0. This results in that $W_{C_j}^{i,i}$ solely represents a repelling force due to a translational displacement.

$$\widehat{f}_{C_j}^{i,i} = -as\big(\mathbf{Q}_{t_i}(q)\mathbf{R}_{C_j}^{i}\widehat{p}_i^{C_j}\mathbf{R}_i^{C_j}\big) - \mathbf{R}_{C_j}^{i} as\big(\mathbf{Q}_{t_i}(q)\widehat{p}_i^{C_j}\big)\mathbf{R}_i^{C_j} \tag{3.8}$$

$$\widehat{m}_{C_j}^{i,i} = -as\big(\mathbf{Q}_{t_i}(q)\mathbf{R}_{C_j}^{i}\widehat{p}_i^{C_j}\widehat{p}_i^{C_j}\mathbf{R}_i^{C_j}\big) \tag{3.9}$$

In order to express the wrench $W_{C_j}^{i,i}$ in the inertial reference 0-frame with the coordinate transformation $Ad_{\mathbf{H}_0^i}^{\top}$ (3.11), seen in Equation 3.10

$$W_{C_j}^{0,tcp\top} = Ad_{\mathbf{H}_0^{tcp}}^{\top} W_{C_j}^{tcp,tcp\top} \tag{3.10}$$

$$Ad_{\mathbf{H}_0^i}^{\top} = \begin{bmatrix} \mathbf{R}_i^0 & 0 \\ \widehat{p}_i^0\mathbf{R}_i^0 & \mathbf{R}_i^0 \end{bmatrix} \tag{3.11}$$

Now where the wrench is expressed in the inertial reference frame, the importance behind this change of reference will be seen later in subsection 3.3.1.

This concludes the mathematical description of the constraints, in the remaining of this chapter the reader will be introduced to the implemented reactive control scheme.

3.2 Reactive control

As previously mentioned a traditional robot control schemes like position and velocity controllers are divided into two subsequent stages: motion planning and motion execution. Where during the motion planning, the desired position and orientation for the end-effector frame is described by a function of time and reference coordinates, which then is transformed into joint angles. These "pre-calculated" joint angles/velocities form a joint trajectory, which is first executed when the planning is finished.[15]
However reactive control schemes merge the planning and execution phase [18]. Reactive control schemes use the current state of the robot and the task description at each time step as input for the computation of the joint forces applied in the next time step. This control scheme provides greater flexibility for the robot during the execution of the task compared to traditional control schemes and will, therefore, be investigated closer. One of these reactive control schemes is the so-called impedance controller.[3]

In other words, the benefit of an Impedance controller over a position/velocity controller is that a dynamic relationship between the different state variables is made by controlling the impedance of the manipulator, instead of controlling just a single state variable.[3]
In general, one can describe an impedance control controlled manipulator by a mass-spring-damper system with adjustable parameters [14]. The interaction with such a system can be described as an energy exchange between the manipulator and its environment [13]. This energy exchange is described as the mapping from the flow (i.e motion) to effort (i.e force).[13, 19]

In other words, the manipulator's controller reacts to the deviation between the end-effectors motion and desired motion by generating forces countering the deviation of motion.[19] A visualization of general setup of the implemented control strategy can be seen in Figure 3.3.

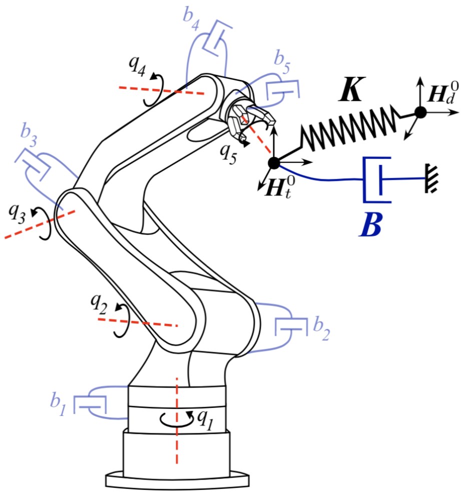

Figure 3.3: Visualizes virtual springs and dampers of the control strategy. Where $\mathbf{K}$ represents the Cartesian spring between the tool center point of the robot and the desired transformation. The damping is represented in the Cartesian damping B and the damping for each individual joint $b_1, \ldots, b_5$.[20]

In this work, an Energy/Power-aware Impedance controller which was first introduced in [9] is used. This control strategy includes methods like Energy shaping and Damping injection. These methods minimize/counter-act autonomously the non-linear behavior of a normal Impedance controller, by observing the energy introduced to the manipulator and power the manipulator is able to exchange with its environment.
The remainder of this section will elaborate on the Energy/Power-aware Impedance controller mentioned above.

3.3 Motion Springs

This section describes derivation of τ_{motion}, which as mentioned above and visualized in Figure 3.3 is generated through a spring. The section will elaborate on the modeling of the mentioned spring in subsection 3.3.1 and the concept of energy scaling in subsection 3.3.2.

3.3.1 Mathematical description of a Spring

This section concerns itself with the mathematical description of the motion generating spring in an impedance controller. The 6 dimensional spring illustrated in Figure 3.3, which generates the motion of the end-effector from its current transformation $\mathbf{H}^0_{tcp}$ towards it goal transformation $\mathbf{H}^0_d$ is modeled as the Wrench $W_K^{tcp,tcp}$ as seen in Equation 3.12.[13]

The reason behind modeling the spring as a wrench can be referred back to the discovery of Poinsot, which states that any forces acting on a rigid body, can be expressed as a single force and momentum acting along a line.[4]

$$W_K^{tcp,tcp^\top} = \begin{bmatrix} f_K^{tcp,tcp^\top} \\ m_K^{tcp,tcp^\top} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{K}_t & \mathbf{K}_c \\ \mathbf{K}_c^\top & \mathbf{K}_r \end{bmatrix}}_{\mathbf{K}} \Delta\chi; \mathbf{K}_{r,t,c} \in \mathbb{R}^{3x3} \tag{3.12}$$

Where $\Delta\chi^\top = \begin{bmatrix} \Delta\theta_K^{tcp,tcp^\top} & \Delta q_K^{tcp,tcp^\top} \end{bmatrix}$ expresses an infinitesimal twist displacement, which is in relation to an relative configuration $\mathbf{H}_j^i$ numerically expressed as $\Delta\xi = \dot{\mathbf{H}}_d^{tcp}\mathbf{H}_{tcp}^d$ [13]. The elements $\mathbf{K}_t$,$\mathbf{K}_r$ represent the stiffness for translation, rotation of the spring respectively. As proven in [21], $\mathbf{K}_c$ is the maximal decoupling between the translational and rotational terms, in other words it defines how much the translational and rotational springs influence each other. It to note that entries of K can be freely chosen.

In pursuance of describing $W_K^{tcp,tcp}$ purely in terms of energy, which is essential for being able to use the concept of energy scaling which will be introduced in later in subsection 3.3.2, the force $f_K^{tcp,tcp}$ (3.14) and momentum $m_K^{tcp,tcp}$ (3.13) can be formulated. A more detailed derivation of $f_K^{tcp,tcp}$ and $m_K^{tcp,tcp}$ can be found in [13].

$$\widehat{m}_K^{tcp,tcp} = -2as\big(\mathbf{G}_r\mathbf{R}_{tcp}^d\big) - as\big(\mathbf{G}_t\mathbf{R}_d^{tcp}\widehat{p}_{tcp}^d\widehat{p}_{tcp}^d\mathbf{R}_{tcp}^d\big) - 2as\big(\mathbf{G}_c\widehat{p}_{tcp}^d\mathbf{R}_{tcp}^d\big) \tag{3.13}$$

$$\widehat{f}_K^{tcp,tcp} = -\mathbf{R}_d^{tcp} as\big(\mathbf{G}_t\widehat{p}_{tcp}^d\big)\mathbf{R}_{tcp}^d - as\big(\mathbf{G}_t\mathbf{R}_d^{tcp}\widehat{p}_{tcp}^d\mathbf{R}_{tcp}^d\big) - 2as\big(\mathbf{G}_c\mathbf{R}_{tcp}^d\big) \tag{3.14}$$

Where the $\mathbf{G}_{r,t,c}$ are the co-stiffnesses for translation, rotation and the coupling springs and $as()$ the operator returning the anti-symmetric part of a square matrix. The co-stiffnesses are introduced for the convention between $\Delta\chi$ and the Rotation matrices $\mathbf{R}$.

$$\mathbf{G}_{r,t,c} = \frac{1}{2}\,\mathrm{tr}\,\big(\mathbf{K}_{r,t,c}\big)\mathbf{I} - \mathbf{K}_{r,t,c} \tag{3.15}$$

In order to express the elastic wrench $W_K^{tcp,tcp}$ in the inertial reference 0-frame with the coordinate transformation $Ad_{\mathbf{H}_0^{tcp}}^\top$ (3.17), seen in Equation 3.16

$$W_K^{0,tcp^\top} = Ad_{\mathbf{H}_0^{tcp}}^\top W_K^{tcp,tcp^\top} \tag{3.16}$$

$$Ad_{\mathbf{H}_0^{tcp}}^\top = \begin{bmatrix} \mathbf{R}_{tcp}^0 & 0 \\ \widehat{p}_{tcp}^0\mathbf{R}_{tcp}^0 & \mathbf{R}_{tcp}^0 \end{bmatrix} \tag{3.17}$$

Now where the elastic wrench is expressed in the inertial reference frame (3.16) and due to the dual nature of force and motion mentioned above in subsection 2.2.1. [4, 6], the driving joint torques τ_{motion} can be computed by Equation 3.18.

Where the joint torques τ_{motion} result from the combination of the elastic wrench $W_K^{0,tcp\top}$ and the wrench generated by the constraint $W_{C_j}^{0,i}$ (3.6). This is only possible because they are described both described in the same refernce frame.

$$\tau_{\text{motion}}^\top = \widetilde{\mathbf{J}}^\top(q)\Big(W_K^{0,tcp\top} - \sum_{i=1}^{n}\big(\sum_{j=1}^{z} W_{C_j}^{0,i\top}\big)\Big) \tag{3.18}$$

Where n is equal to the number of joints and z equal to the number of constraints.

3.3.2 Energy shaping

Energy shaping is a form of passivity-based control: The controller is a passive Port-Hamiltonian System, so it can inject only a finite amount of energy [22]. By shaping the energy of the system in order to assign a strict minimum in the desired configuration.[23]

The energy based safety metric demands a limit on the total energy of the system (3.19) and this can be achieved by regulating the amount of potential energy in the spatial springs of the control system [24].

$$E_{\text{total}} = T_{\text{total}} + U_{\text{total}} \tag{3.19}$$

The kinetic energy in the system is described in Equation 3.20 and total potential energy stored in the springs are defined by Equation 3.21.

$$T_{\text{total}}(q,\dot{q}) = \frac{1}{2}\dot{q}^\top \mathbf{M}(q)\dot{q} \tag{3.20}$$

$$U_{\text{total}}(\mathbf{R}_{tcp}^d, p_{tcp}^d) = U_r(\mathbf{R}_{tcp}^d) + U_t(\mathbf{R}_{tcp}^d, p_{tcp}^d) + U_c(\mathbf{R}_{tcp}^d, p_{tcp}^d) \tag{3.21}$$

Where U_r, U_t and U_c (see Equation 3.22) are the potential energy stored in the rotational, transnational and Coupling spring respectively.

$$\begin{aligned} U_r(\mathbf{R}_{tcp}^d) &= -\operatorname{tr}\left(\mathbf{G}_r \mathbf{R}_{tcp}^d\right) \\ U_t(\mathbf{R}_{tcp}^d, p_{tcp}^d) &= -\frac{1}{4}\operatorname{tr}\left(\widehat{p}_{tcp}^d \mathbf{G}_t \widehat{p}_{tcp}^d\right) - \frac{1}{4}\operatorname{tr}\left(\widehat{p}_{tcp}^d \mathbf{R}_{tcp}^d \mathbf{G}_r \mathbf{R}_d^{tcp} \widehat{p}_{tcp}^d\right) \\ U_c(\mathbf{R}_{tcp}^d, p_{tcp}^d) &= \operatorname{tr}(\mathbf{G}_c \mathbf{R}_d^{tcp} \widehat{p}_{tcp}^d) \end{aligned} \tag{3.22}$$

In which the tr() operator is used to sum the potential energy along each axis together. Based on the total energy stored of the system with the initial stiffness's E_{total} and a chosen maximum energy $E_{\max}$ which the system is allowed to store. A scaling parameter is computed λ as in Equation 3.23.

$$\lambda = \begin{cases} 1 & \text{if } E_{\text{total}} \leqslant E_{\max} \\ \frac{E_{\max} - T_{\text{total}}}{U_{\text{cur}}} & \text{otherwise} \end{cases} \tag{3.23}$$

Where the second line in Equation 3.23 can be seen as a linear factor between the maximal allowed potential energy stored in the springs and the current potential energy stored in the springs (3.24).

$$\frac{E_{\max} - T_{\text{total}}}{U_{\text{cur}}} = \frac{U_{\max}}{U_{\text{cur}}} \tag{3.24}$$

As the potential energy stored in the spatial springs (3.22) are proportional to the Co-stiffness $\mathbf{G}_{r,t,c}$ [25], it is possible by proportionally scaling $\mathbf{G}_{r,t,c}$ by λ as seen in Equation 3.25) to scale the potential energy in the system.

$$\mathbf{G}_{r,t,c} \leftarrow \lambda \mathbf{G}_{r,t,c} \tag{3.25}$$

By limiting the potential energy stored in the spatial springs, the energy injected by the controller is limited and thereby guaranteeing that only a finite amount of energy is introduced to the system.

3.4 Design of Damping

Generally it is not a good choice to choose $\mathbf{B} \in \mathbb{R}^{n \times n}$ as a constant and diagonal matrix, as dynamics of the system is state depended. This is why it was chosen to make the damping $\mathbf{B}(q) \in \mathbb{R}^{n \times n}$ dependent on the change of the state of the Manipulator.
The resulting damping torques are for any of the following approaches calculated by Equation 3.26.

$$\tau_{\text{Damping}}^{\top} = \mathbf{B}(q)\dot{q} \tag{3.26}$$

3.4.1 Joint space damping

As mentioned in the beginning of this chapter, choosing the values of $\mathbf{B}$ to be a constant is in general not the ideal choice. For this reason, $\mathbf{B}$ is made depended on the state of the system, the most obvious way to do this is by multiplying it by the diagonal elements of the $\mathbf{M}(q)$, $\mathbf{M}_{diag}(q)$ as seen in Equation 3.27.

$$\mathbf{B}(q) = \mathbf{M}_{diag}(q)\mathbf{B} \tag{3.27}$$

Where

$$\mathbf{M}_{diag}(q) = \begin{bmatrix} \mathbf{M}_{1,1}(q) & 0 & \dots & 0 \\ 0 & \mathbf{M}_{2,2}(q) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{M}_{n,n}(q) \end{bmatrix} \tag{3.28}$$

This has the advantage, that $\mathbf{B}(q)$ is not only depended on the velocity of the individual joints but also mass-inertia of each link. In this section the derived damping matrix $\mathbf{B}(q)$ will be from now referred to as $\mathbf{B}_{\text{init}}(q)$.

3.4.2 Damping injection

In the previous section it was discussed how to limit the energy introduced to the robot by the controller, but not the energy which the robot can exchange with its environment. In order to limit this energy exchange the so called *damping injection* method is used.
The power in (3.29) consist out of an gravitational part and motion part, which are defined as seen in Equation 3.30 and Equation 3.31.

$$P_{\text{ctrl}} = P_{\text{motion}} + P_{\text{Gravity}} \tag{3.29}$$

Where gravitational power represents the power the controller transferred to the robot for the compensation of the gravitational forces.

$$P_{\text{Gravity}} = \bar{\mathbf{G}}(q) \cdot \dot{q} \tag{3.30}$$

The energy the robot would exchange with an obstacle at the occurrence of an uncontrolled contact/collision is defined in Equation 3.31.

$$P_{\text{motion}} = \left(\widetilde{\mathbf{J}}(q)^\top W_K^{0,tcp\top} - \mathbf{B}(q)_{\text{init}} \dot{q} \right)^\top \dot{q} \tag{3.31}$$

In order to limit this exchanged energy the intial damping term of the controller (3.27) is scaled up by a scaling parameter β (3.33) as seen in Equation 3.32. This the scaling takes place whenever P_{motion} exceeds the user defined maximal power threshold P_{max}. As only the power, which is exchange in the direction of the desired configuration H_d^0 is of interest, the following statment must be fulfilled $P_{\text{motion}} > 0$ inorder increase the damping term. This ensures that only the motions towoards the desired configuration are punished if it exceeds the power based satfy metric. The reason for only increasing the damping in this direction is twofold: Firstly the spring which pulls end-effector towards the goal already has an damping effect for motions away from the desired configuration. Secondly in case of an interaction, where the manipulator is forced away, for example when pushed away from an hazardous interaction, its movement should not be punished.

$$\mathbf{B}(q) = \beta \mathbf{B}_{\text{init}}(q) \tag{3.32}$$

Where

$$\beta = \begin{cases} 1 & \text{if } P_{\text{motion}} \leqslant P_{\text{max}} \\ \frac{\left(\widetilde{\mathbf{J}}(q)^\top W_K^{0,tcp\top} \right)^\top \dot{q} - P_{\text{max}}}{\dot{q}^\top \mathbf{B}_{\text{init}}(q) \dot{q}} & \text{otherwise} \end{cases} \tag{3.33}$$

With the new calculated damping term described in Equation 3.32 the following equation for the damping torques τ_{Damping} (3.34) can be derived.

$$\tau_{\text{Damping}}^\top = \beta \mathbf{B}_{\text{init}}(q) \dot{q} \tag{3.34}$$

This concludes the mathematical description of the reactive control scheme used in this project, in the following section will the reader be introduced to the mathematical description of concept of *joint limit avoidance.*

3.5 Joint limit avoidance

As the controller in its compliant state does not follow any pre-planned trajectory, the risk for the manipulator reaching a joint limit increases drastically. This has the consequence that the KUKA LBR iiwa will go into a backlog, in which the manipulator stops is movement and must be reinitialized.

For this reason, it was decided to implement in addition to the Cartesian constraint, so-called Joint constraint which serves the purpose of hindering the manipulator to reach its joint limit. In order to keep the manipulators joint within its set constraint, the method of *joint limit avoidance* which was firstly introduced in [16] is implemented.

The concept of *joint limit avoidance* is similar to the way how the manipulator's links are kept within its respective Cartesian constraints (section 3.1), by generating a torque $\tau_q^\top \in \mathbb{R}^n$ that forces the joint to stay within the defined constraint. For the remainder of this section, the mathematical description of this control scheme will be described for a single joint. The function, which generates a torque into the opposite direction of the active constraints depends on the difference between the i^{th} joints current joint position q_i and its lower/upper joint limits $\underline{q}_{i,\text{limit}}/\bar{q}_{i,\text{limit}}$ as seen inEquation 3.35.

$$\begin{aligned} \tau_{\underline{q}_i}(q) &= \begin{cases} \frac{\Omega}{\underline{q}_i^2}\left(\frac{1}{\underline{q}_i} - \frac{1}{\underline{q}_{i,J}}\right) & \text{if } \underline{q}_i \leq \underline{q}_{i,J} \\ 0 & \text{otherwise} \end{cases} \\ \tau_{\bar{q}_i}(q) &= \begin{cases} -\frac{\Omega}{\bar{q}_i^2}\left(\frac{1}{\bar{q}_i} - \frac{1}{\bar{q}_{i,J}}\right) & \text{if } \bar{q}_i \leq \bar{q}_{i,J} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3.35}$$

As seen in Equation 3.35, the repelling torques $\tau_{\underline{q}_i}$ and $\tau_{\bar{q}_i}$ are not only depended on the distance between q_i and its minimal/maximal bounds, but also $\underline{q}_{i,J}/\bar{q}_{i,J}$ a distance at which the constraints turns active and a scaling factor $\Omega > 0$. Where $\underline{q}_i$ and $\bar{q}_i$ are calculated as seen in Equation 3.36.

$$\begin{aligned} \underline{q}_i &= q_i - \underline{q}_{i,\text{limit}} \\ \bar{q}_i &= \bar{q}_{i,\text{limit}} - q_i \end{aligned} \tag{3.36}$$

Equation 3.35 can be summarized into Equation 3.37.

$$\tau_{q_i}(q) = \begin{cases} \frac{\Omega}{\underline{q}_i^2}\left(\frac{1}{\underline{q}_i} - \frac{1}{\underline{q}_{i,J}}\right) & \text{if } \underline{q}_i \leq \underline{q}_{i,J} \\ -\frac{\Omega}{\bar{q}_i^2}\left(\frac{1}{\bar{q}_i} - \frac{1}{\bar{q}_{i,J}}\right) & \text{if } \bar{q}_i \leq \bar{q}_{i,J} \\ 0 & \text{otherwise} \end{cases} \tag{3.37}$$

A conceptual visualization of this feature is illustrated in Figure 3.4.

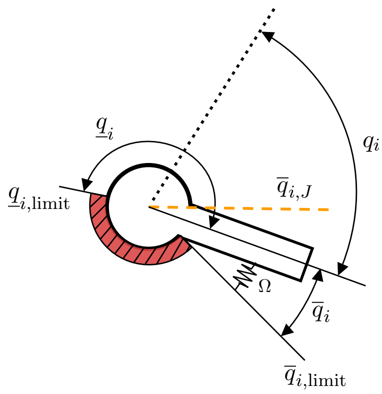

Figure 3.4: Visualizes the joint limit avoidance concept in an active state at the upper bound, based on the description in this section. Where the area colored in red is the area in which the joint of the manipulator can physically not reach, the dotted black line marks the midpoint of the joint and the dashed line in orange is the distance to the upper limit at which the constraint is active.

This concludes the mathematical description of the implemented joint limit avoidance for the individual joints.

3.6 Control Torques

This section briefly summarizes the main highlights of this chapter and describes the resulting torques τ, which are used to control the manipulator.

In section 3.1 the reader was introduced to the mathematical description of simple Cartesian Constraints in form of 2D plane in $\mathbb{R}^3$ as well as to the description the derivation of wrench $W_{C_j}^{0,i}$ which repelling the affected link from the violated constraint.
The reader was introduced in section 3.2 to the concept of reactive control schemes and their advantages over classic control schemes. In this section also a energy-aware impedance control strategy was described from which the torques τ_{motion} and τ_{Damping} were derived.
And lastly, an inside into the concept and mathematical description of a joint limit avoidance strategy and the generated torque $\tau_q^\top$ was given in section 3.5.
From the above-mentioned torques, one can formulate the torques τ, which are used to control the manipulator as shown in Equation 3.38.

$$\tau^\top = \underbrace{\tau_{\text{motion}}^\top - \tau_{\text{Damping}}^\top}_{\tau_{\text{ctrl}}^\top} + \tau_q^\top + \underbrace{\tau_{\text{Coriolis}}^\top}_{\bar{\mathbf{C}}(q,\dot{q})\dot{q}} + \underbrace{\tau_{\text{Gravity}}^\top}_{\bar{\mathbf{G}}(q)} \tag{3.38}$$

With $\tau_{\text{Coriolis}}^\top$ and $\tau_{\text{Gravity}}^\top$ being the torques representing the compensation Coriolis and gravitational force acting on the manipulator, which is extracted from the mathematical model described in section 2.2 and their derivation can be found in Appendix B.

This concludes the mathematical description of the implemented control strategy used in this work, in the following chapter the conducted experiments and their results will be described.

Validation 4

In chapter 1 the reader was introduced to one of the main challenges collaborative robots face when interacting with an autonomous quantity, namely the collision with objection within its dexterous workspace. Furthermore, the reader was introduced in section 1.1 to a scenario and a proposed concept which should mitigate this problem. This concept aims to keep the manipulator within its free Cartesian workspace by setting up Cartesian constraints and improving its maneuverability within this constraint workspace by implementing a feature called joint limit avoidance.
This chapter describes the validation of the implemented control strategies capabilities for fulfilling the above-mentioned concept. For this a set of requirements were defined, which are listed in Table 4.1.

Requirements:

1	The KUKA LBR iiwa should stay within the set safety metrics when interacted with: **1.1** With the end-effecotor being displaced by at least 0.2m in terms of position **1.2** With the end-effecotor being displaced by at least 40°
2	The KUKA LBR iiwa should stay within the set Cartesian constraints: **2.1** When encountering two differnt constraints at different time instances. **2.2** When encountering two differnt constraints at the same time instances. **2.3** variable force
3	The KUKA LBR iiwa should not exceed the jointspecific limits: **3.1** For both the upper and lower limit of a joint. **3.2** When approaching multiple joint limits at the once

Table 4.1: Lists the requirements set for the test.

Based on these requirements a set of tests were conducted, which tests the compliance of the chosen reactive control scheme, the capability to stay within the restricted Cartesian Workspace and ability to avoid joint limits the manipulator might violate during the interaction.
The remainder of this chapter will describe the experiments conducted in this work and will be structured in the following way.

- **Test 1:** Compliance under interaction (section 4.1)
- **Test 2:** Cartesian Constraints (section 4.2)
- **Test 3:** Joint constraint (section 4.3)

The above stated tests will be conducted with the in Table 4.2 defined control parameter and the general hardware setup together with the description of the implementation can be found in Appendix C.

Control variables	$\mathbf{K}_{t_{x,y,z}}$	$\mathbf{K}_{r_{x,y,z}}$	$\mathbf{K}_{c_{x,y,z}}$	$E_{\max}$ in J	$P_{\max}$ in W	b
	2000	100	0	2	0.5	5

Table 4.2: Control variables used during the different experiments.

4.1 Test 1: Compliance under interaction

This section describes the test, which was conducted to verify and evaluate the LBR iiwas behaviour when encountering an unplanned interaction (e.g. collision, user input). The main focus of this test is the reactive control scheme, which was implemented in this work and described in section 3.2. The test was conducted by having the LBR iiwas end-effector holding a Cartesian configuration $\mathbf{H}_d^0$, from which it is forced away by an upwards motion at a random point in time. As soon as the end-effector has reached a displacement of at least 0.2m along the z-axis the end-effector is released. At which point the end-effector should move back to $\mathbf{H}_d^0$. A visual representation of the test can be seen in Figure 4.1.

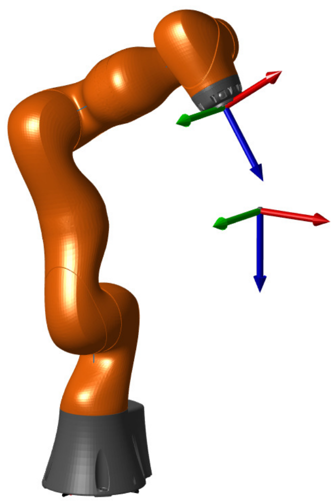

Figure 4.1: Visualizes the concept of "Compliance under interaction" test.

4.1.1 Position Displacement

This section examines the displacement between the reference position p_d^0 and current position p_{tcp}^0 of the end-effector. When inspecting Figure 4.2 it can be seen that the interaction begins at time $t = 1.3$s and reaches its maximum displacement of 0.33m along the z-axis and 0.24m along the y-axis at $t = 2.3$s and $t = 2.5$s respectively. As the end-effector is release at $t = 2.3$s and no external force is longer applied, it starts to move back towards reference configuration $\mathbf{H}_d^0$.

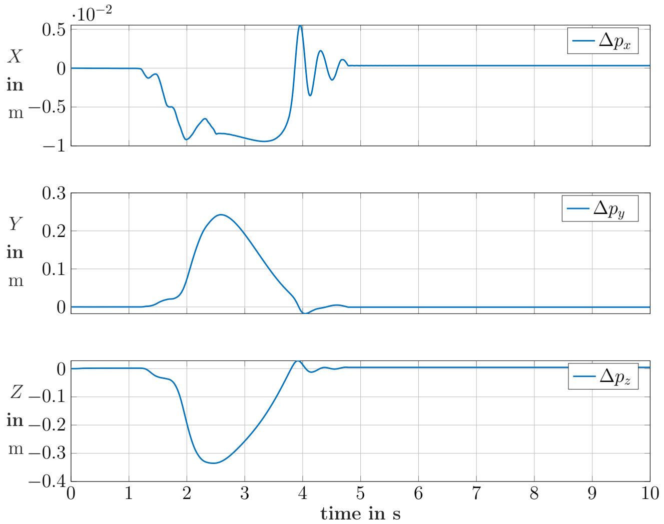

Figure 4.2: Shows the displacement between p_d^0 and p_{tcp}^0 along each axis.

4.1.2 Angular Displacement

In this section the angular displacement $\Delta\theta$ between the reference frame and current end-effector frame is examined. As seen in Figure 4.3 the angular displacement shows a similar behavior as already observed in subsection 4.1.1. The angular displacement $\Delta\theta$ increases as the end-effector is forced away from its reference frame and reaches its maximum deviation of 59° at time $t = 2.5$s. A visualization of this configuration can be seen in Figure 4.1. After the end-effector has reached its reference position p_d^0 it can be seen that there is a minor orientational offset of 1.2°. This can be related back to the small value which was chosen for the rotational spring stiffness $\mathbf{K}_o$. As the force, which is generated by the displacement $\Delta\theta$ and stiffness $\mathbf{K}_o$ is not big enough to force the end-effector to its reference orientation $\mathbf{R}_d^0$.

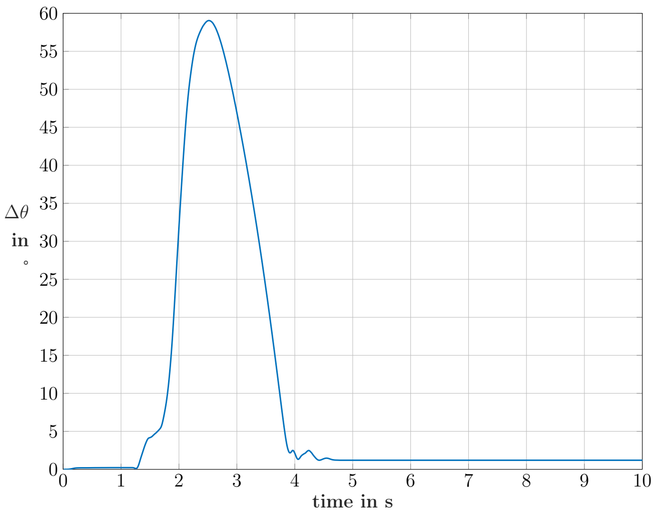

Figure 4.3: Shows the angular displacement between the reference and end-effectors orientation.

4.1.3 Energy Scaling

This section examines the energy introduced to the system and the effect which the reactive control schemes energy scaling feature (subsection 3.3.2) has on this energy. The reason for looking at the energy introduced to the system rather on the force is due to the fact that it is not possible to measure the force of a virtual spring, hence one looks at the potential energy injected to the system.
When looking at Figure 4.4 one can observe how the energy introduced to the system increases as energy stored in the virtual springs $E_{\text{total}_{\text{init}}}$ increase as the displacement between the reference frame and current end-effector frame increases. If one would release the end-effector with such a large amount of energy stored in the springs, the system would most likely get into an unstable and hazardous state. The energy scaling feature of the reactive control scheme mitigates this, by limiting the energy which can be introduced to the system to a set level E_{max}. This is done by downscaling the spring stiffnesses by the scaling factor λ as soon as $E_{\text{total}_{\text{init}}}$ exceeds E_{max}, which results in a new energy $E_{\text{total}_{\text{scaled}}}$.

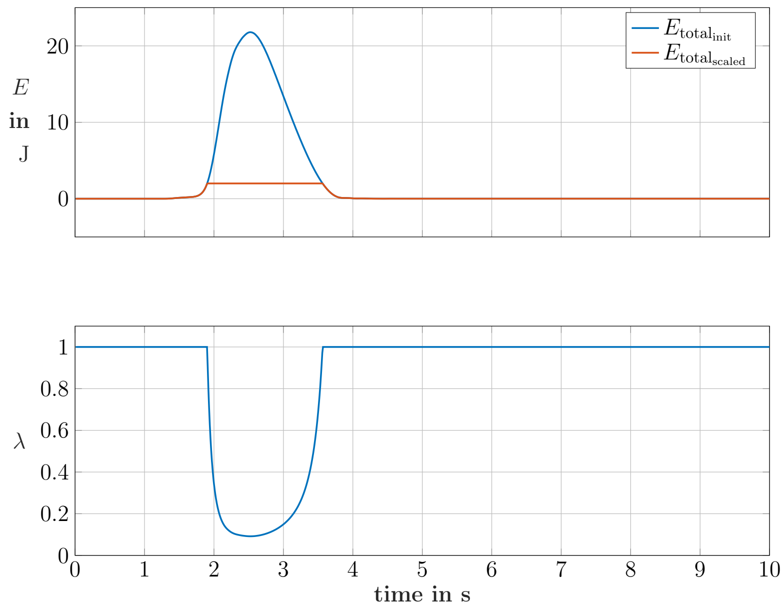

Figure 4.4: Shows the Energy $E_{\text{total}_{\text{init}}}$, $E_{\text{total}_{\text{scaled}}}$ and the energy scaling parameter λ.

4.1.4 Power

Another important quantity in the field of compliant control strategy is the power P_{motion}, which system can exchange with its environment. In this section, this power is examined as well as the effect the damping injection feature of the implemented control scheme (subsection 3.4.2) has on this power is examined.
As seen in the first plot of Figure 4.5, the power increases in negative direction, as the system is injected with an external energy due to the external force applied during the interaction.

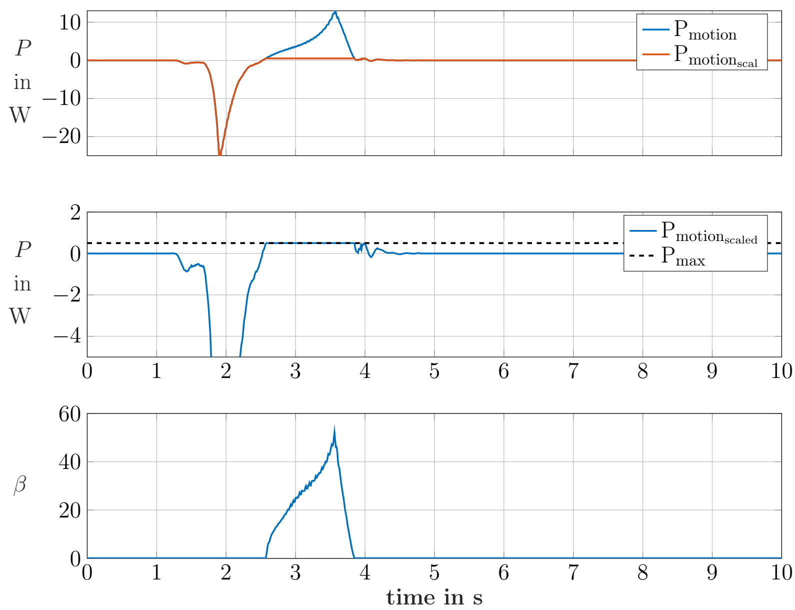

Figure 4.5: Shows the Power P_{motion}, $P_{\text{motion}_{\text{scaled}}}$ and the damping injection parameter β.

It can also be seen that as soon as the external force is no longer applied that the direction of the power exchange is reversed. In other words the system exerts the energy into the environment. As this exchange can be harmful and dangerous for objects and humans in the manipulators environment, it is important to limit this power to a "safe" threshold P_{max}.
In this control scheme this is done with previous mentioned damping injection feature, where P_{motion} is limited by increasing the joint damping by the scalar β. The influence β has on the P_{motion} can be seen Figure 4.5, as P_{motion} exceeds P_{max}, β increases which results in $P_{\text{motion}_{\text{scaled}}}$. This new power does not exceed the set Threshold P_{max} as one can observe in the second plot of Figure 4.5.

4.1.5 Test evaluation

As mentioned earlier this test was conducted for the purpose of confirming the behavior of the implemented reactive control scheme and its capability to handle unplanned interaction. The test has shown, that even though the displacement of end-effector exceeded the set requirements of 0.2m and 40°, the reactive control scheme was capable of keeping the KUKA LBR iiwa within the set safety metric. Thereby fulfilling the *Requirement 1* as listed in Table 4.1.

4.2 Test 2: Cartesian Constraints

The following sections describe the test of the LBR iiwas behavior when encountering a Cartesian constraint in simulation and in the real world. The test was conducted by having the LBR iiwa following a preplanned trajectory $\mathbf{H}_d^0$, which is designed in such a way that it violates the Cartesian constraints C_1 and C_2. These two constraints/virtual walls are placed in front and above the LBR iiwa and are defined as 2D planes in the Cartesian

Workspace of the manipulator as described in section 3.1. Where the virtual wall C_1 is placed in front and C_2 is placed above the LBR.

This test setup serves the purpose of simulating the scenario described in section 1.1, in which the human co-worker pushes/pulls the robot towards a constraint. This test was conducted in simulation as well as on the real-world LBR iiwa. A visualization of this test setup can be seen in Figure 4.6. Where the function describing the trajectory and the test specific Parameters can be found in section D.1 and Table 4.3 in respectively.

Parameters	C_1 vertices	C_2 vertices	$d_{0,c_{1/2}}$ in m	γ	$\kappa_{1/2}$
	$\begin{bmatrix}0.5\\1\\0\end{bmatrix}, \begin{bmatrix}0.5\\1\\0.9\end{bmatrix}, \begin{bmatrix}0.5\\-1\\0\end{bmatrix}, \begin{bmatrix}0.5\\-1\\0.9\end{bmatrix}$	$\begin{bmatrix}0.5\\1\\0.9\end{bmatrix}, \begin{bmatrix}0.5\\1\\0.9\end{bmatrix}, \begin{bmatrix}-1\\-1\\0.9\end{bmatrix}, \begin{bmatrix}-1\\1\\0.9\end{bmatrix}$	0.05	5	5

Table 4.3: Lists the "Cartesian Constraints" test specific parameters.

The vertices for the virtual walls C_1 and C_2 were implemented in such a way that the pre-planned trajectory violates both constraints. This was done so that the behaviour of LBR iiwa could be investigated for the following scenariors: When the manipulator encounters different constraints at different point in time. When the manipulator encounters multiple constraints at the same time. The activation distance $d_{0,c_{1/2}}$ of the Constraints C_1 and C_2 was chosen to be 0.05m in order to have an sufficient buffer such that the constraint is not being violated. The exponential of the transition function γ described in section 3.1 was set equal to 2, as it was stated in [16, 15] to be a reasonable choice. The scaling factor for the potential generated $\kappa_{1/2}$ was chosen to be is an equal to 5 as early testing has shown that it is sufficient value to keep the manipulator within the Cartesian constraints. It is to note that the parameters listed in Table 4.3 can be freely chosen by the user.

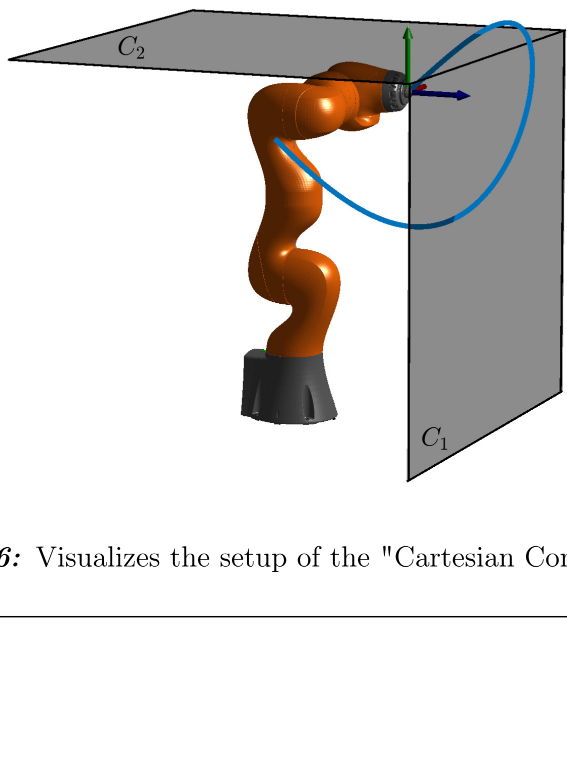

Figure 4.6: Visualizes the setup of the "Cartesian Constraints" test.

4.2.1 Reference tracking

As mentioned previously the test described in this section, was done by having the manipulators end-effector follow an pre-planned trajectory, which violates a set of virtual walls. This section discusses the reference tracking of the LBR iiwa. The x, y, z-components of trajectory p_d^0 and the end-effector p_{tcp}^0 position for the simulation and real-world test over time t is shown in Figure 4.7/4.8.

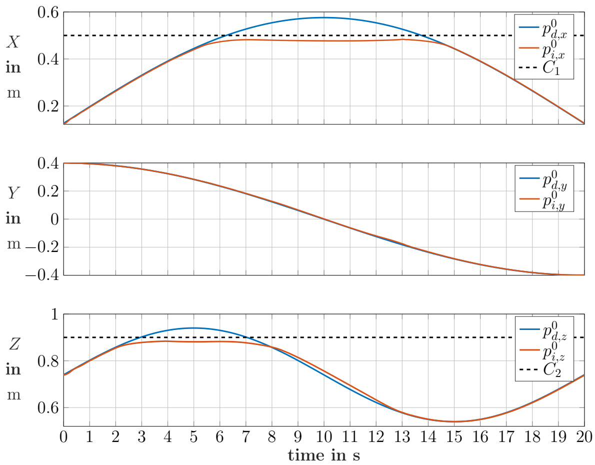

Figure 4.7: Shows the trajectory and end-effector postion for the simulation test.

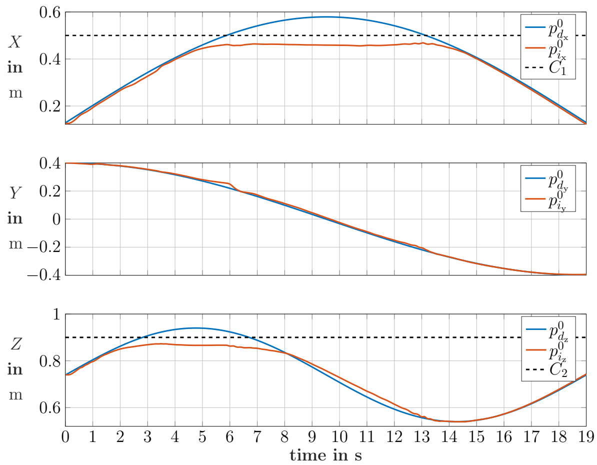

Figure 4.8: Shows the trajectory and end-effector postion for the real-world test.

By inspecting Figure 4.7/4.8, it can be seen that the virtual wall C_1 only restrict the movements of the end-effector along the x-axis of the base frame. Whereas the constraint C_2 only the movements along the base frames z-axis and the movement along the base frames y-axis has no restriction. When looking at Figure 4.7/4.8 it can be seen that the end-effector encounters the virtual wall C_2 by coming within the constraints activation distance d_{tcp,c_2} at time $t = 2$s. At this point a repelling wrench $W_{C_2}^{0,tcp}$ is gernerated, which forces the end-effector to stay within the set Cartesian boundaries. While the end-effector follows the trajectory along the constraint C_2, it encounters the constraint C_1 at time $t = 5.5$s for the simulation and $t = 5$s for the real-world test. At time $t = 8$s the trajectory no longer violate the virtual wall C_2 and end-effector rejoins the trajectory. However the end-effector is only able to track the trajectory with an offset, until the trajectory no longer violates the constraint C_1 at $t = 15$s and $t = 14$s for the simulation and real world test respectively. This behaviour results from the implemented energy scaling described in subsection 3.3.2 and will be elaborate on in subsection 4.2.4.

4.2.2 Position Displacement

This section examines the difference Δp between the desired and the current position of the end-effector in relation to the origin frame. As seen in Figure 4.9/4.10 the resulting displacement between p_d^0 and p_{tcp}^0 is like in subsection 4.2.1 visualized separately for the x, y, z-components. As already mentioned in subsection 4.2.1 one can clearly see, that as soon as the end-effector encounters either one of the virtual walls C_1 and C_2, the displacement along the restricted axes increases. When looking more closely on the z-axis of both test one is able to observe more clearly how the displacement along this axis Δp first matches the reference as the trajectory no longer violates the constraint C_1. As mentioned previously in subsection 4.2.1 this has to do with the nature of the energy scaling, which will be discussed in more detail in subsection 4.2.4.

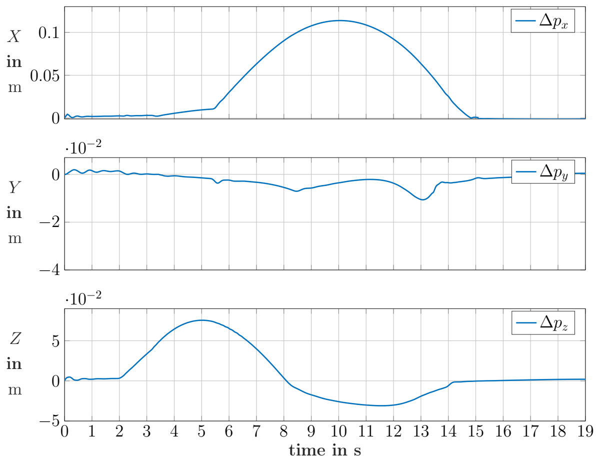

Figure 4.9: Shows the displacement in position Δp between the end-effectors desired position $p_d^0(t)$ and its real position $p_{tcp}^0(t)$ for the simulation test.

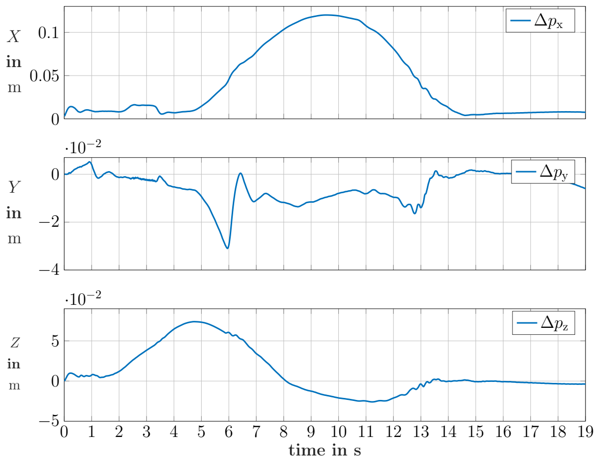

Figure 4.10: Shows the displacement in position Δp between the end-effectors desired position $p_d^0(t)$ and its real position $p_{tcp}^0(t)$ for the real-world test.

When shifting the focus on the y-axis one might observe a small discrepancy in the behavior between the simulation and real-world test in the time interval $t \in [4, 6.5]$, which can be related back to the difference between the modeled and the real LBR iiwa. The displacement Δp_z increases as the end-effector is forced away at time $t = 1$s the error increases along each axis and decreases again as soon as the trajectory no longer violates this constraint.

4.2.3 Angular displacement

In Figure 4.11/4.12 the angular displacement between the reference orientation $\mathbf{R}_d^0(t)$ and $\mathbf{R}_{tcp}^0(t)$ is illustrated. In this test the orientations reference $\mathbf{R}_d^0(t) \quad \forall t$ was chosen to be static. When comparing both figures, it can be seen that the LBR iiwa behavior in the real world is more sensitive to the disturbance in its motion during the transition phase when encountering or leaving a constraint, than in the simulation environment. However, outside these transition phases, its behavior is close to identical.
It can also be observed that the angular displacement $\Delta\theta$ is at its maximum at the point in time, in which also the overall displacement is at its maximum. This is due to the nature of the energy scaling, which will be briefly elaborated on in subsection 4.2.4 and the response seen in Figure 4.13/4.14. With this stated it has to be said that this response would not occur in the scenario described in section 1.1, as the reference frame would not be on the opposite side of a constraint and the manipulator would already be in a compliant state before encountering the first constraint.

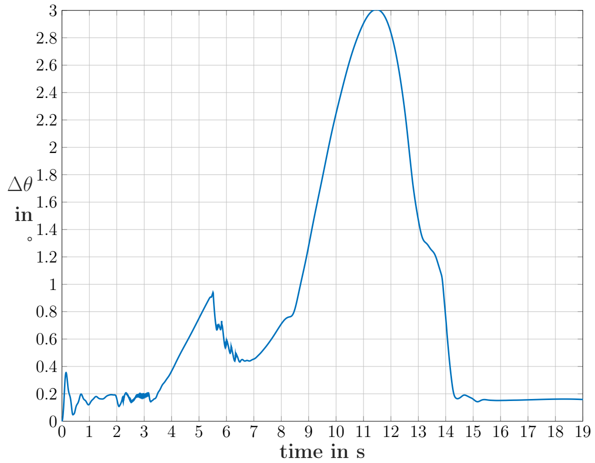

Figure 4.11: Visualizes the angular displacement between the desired and current orientation of the end-effector over time for the simulation test.

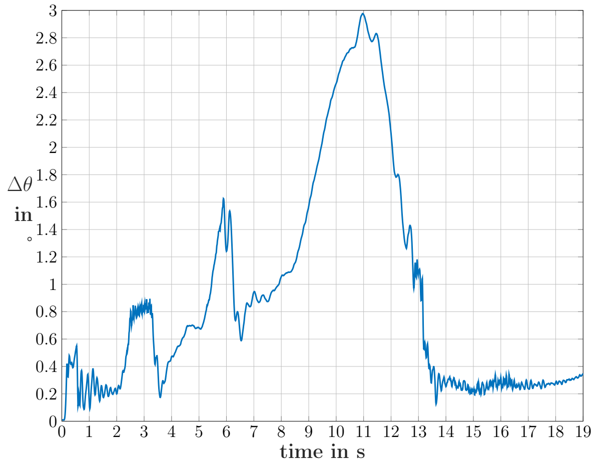

Figure 4.12: Visualizes the angular displacement between the desired and current orientation of the end-effector over time for the real-world test.

4.2.4 Energy

This section evaluates the effect which the energy scaling described in subsection 3.3.2 has on energy of the system. As already described in section 4.1, the total energy introduced to the system increases as the end-effector is forced away from its reference configuration $\mathbf{H}_d^0$. This behaviour can be seen in Figure 4.13/4.14, where $E_{\text{total}_{\text{init}}}$ increases as the end-effector encounters the constraints C_2/C_1 and the displacement between $\mathbf{H}_d^0$ and $\mathbf{H}_{tcp}^0$ increases. However asthe displacement increases also the scaling parameter λ increases inorder to keep $E_{\text{total}_{\text{scaled}}}$ within the maxmial threshold of 2J. As earlier mentioned, this is achieved by scaling the co-stiffenesses by λ, however as the co-stiffnesses decrease also the precision and responsiveness of the manipulator decreases.This results in the angular displacement $\Delta\theta$ and position displacement Δp_{y}(Figure 4.9/4.10).

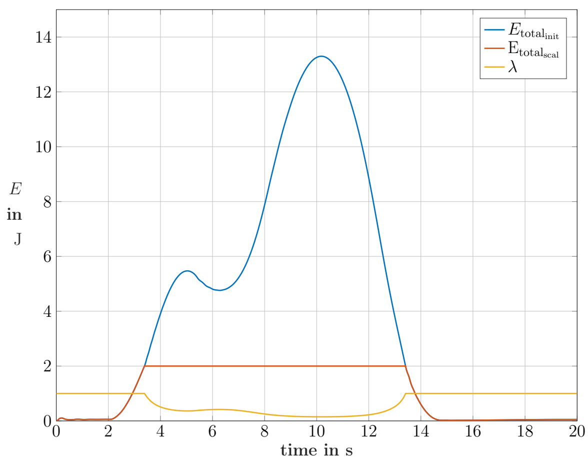

Figure 4.13: Visualizes the total energy E_{total} of the system over time for the simulation.

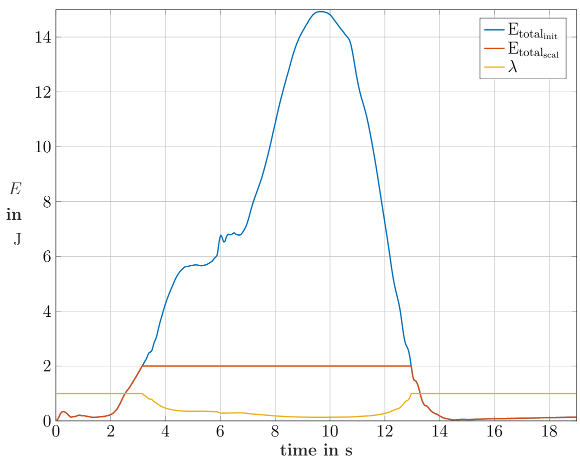

Figure 4.14: Visualizes the total energy E_{total} of the system over time for the real-world test.

4.2.5 Power

This section evaluates the effect the damping injection has on the system. As already described in section 4.1 the damping injection serves the purpose of limiting the energy the manipulator can exchange with its environment in case of an unplanned interaction. When comparing Figure 4.15 and Figure 4.16 with each other one can see that they behave similar to each other, where the $P_{\text{motion}_{\text{init}}}$ exceeds the set power limit $P_{\max}$ in two time instances. The first violation takes place at the beginning of the movement and the second as the end-effector rejoins the trajectory, for both instances, it is assumed that they take place due to the discrepancies between the real LBR iiwa and the created model for the simulation.

However, besides these two minor discrepancies in the simulation, both the simulation and real-world test do not exceed the set power threshold $P_{\max}$, which means the damping injection does not influence the motion of the LBR iiwa. When considering how the test was conducted the results are logical, as the end-effector keeps following the trajectory along the constraint and as it cannot move towards the desired configuration $\mathbf{H}_d^0$ and exchanges close to no energy with the environment.

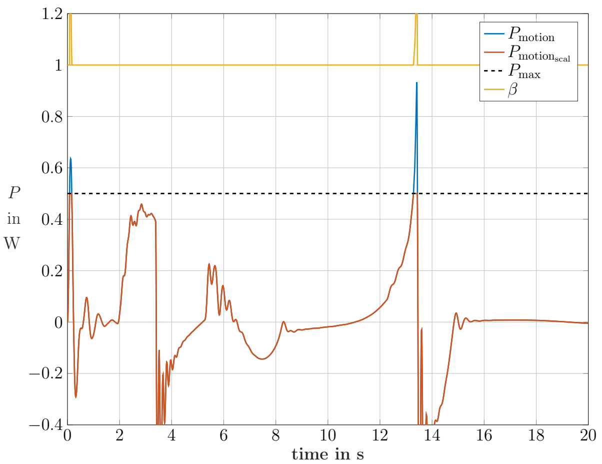

Figure 4.15: Visualizes the P_{motion} over time for the simulation test.

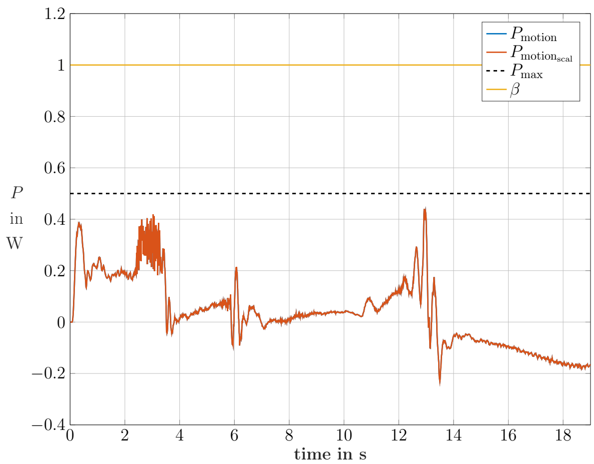

Figure 4.16: Visualizes the P_{motion} over time for the real-world test.

4.2.5.1 Constraint gains

The following section examines the behaviour of the in subsection C.2.2 described approach function for the Cartesian constraint. As stated earlier in subsection 4.2.1, the LBR iiwa behavior when encountering a virtual wall, resulted from the repelling wrench $W_{C_{1/2}}^{0,tcp}$ which is generated when the end-effector comes within a predefined threshold $d_{tcp,c_{1/2}}$. The approach function which generates this repelling wrench $W_{C_{1/2}}^{0,tcp}$ depends on the distance between manipulators end-effector and the Constraint. This function describes a potential which increases as the end-effector gets closer to the Constraint. In Figure 4.17 and 4.18 one can observe this behavior as the gain of the repelling wrench increases as the end-effector approaches the Constraints C_1 and C_2, as well as how they decrease as the end-effector distances itself from the constraints.

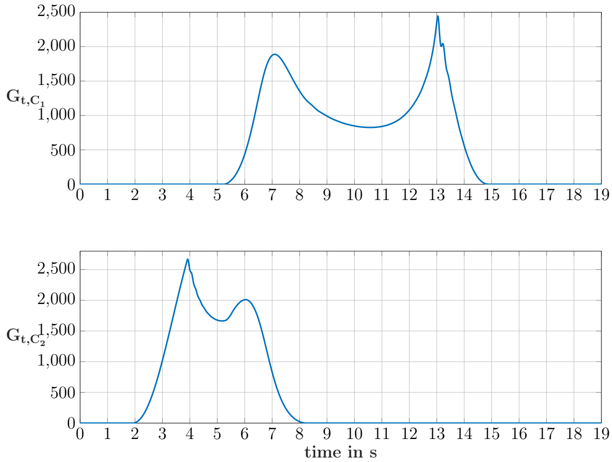

Figure 4.17: Shows the gains of the repelling force of the constraints C_1 and C_2 for the simulation.

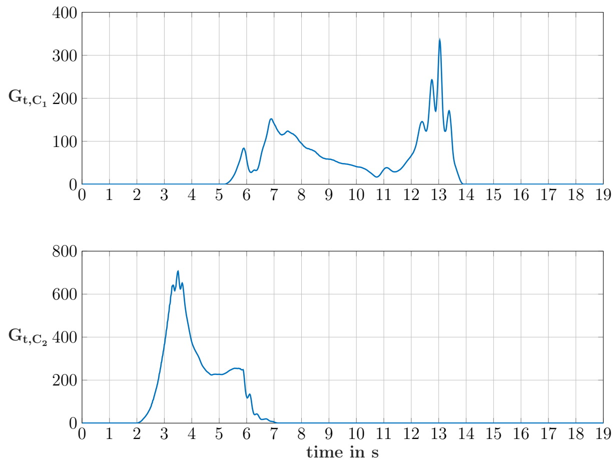

Figure 4.18: Shows the gains of the repelling force of the constraints C_1 and C_2 for the real-world test.

When examing Figure 4.17/4.18 the following observation can be made. Firstly, even though the gains of the simulation and the real-world follow the similar behavior as the gains in the simulation are significantly higher than the ones in the real world. it is assumed that this is due to the differences between the derived model and the real LBR iiwa, as it can be seen that the individual joint torques of the simulation are higher as well as the end-effector being able to get closer to the constraints than the real LBR iiwa.
Secondly the gains generating the repelling force increase whenever the end-effector either encounters or leaves a constraint. This behavior correlates with the displacement between the end-effectors current and the desired configuration, which result in the earlier discussed energy scaling. As one might recall, the stiffnesses of the elastic wrench $W_K^{0,tcp}$ are scaled down as the energy introduced by the springs to the system exceeds a threshold. This energy is proportional to the displacement, hence are the stiffnesses of the springs closer to their initial value the smaller the displacement and thereby the elastic wrench $W_K^{0,tcp}$ bigger. This results in the end-effector being able to get closer to the constraint and thereby increasing the repelling wrench $W_{C_{1/2}}^{0,tcp}$ significantly.

4.2.6 Joint space behaviour

This section examines the Joint space behavior of the LBR iiwa in terms of joint positions and torques for the simulation and real-world test in subsubsection 4.2.6.1 and 4.2.6.2.

4.2.6.1 Joint postions

In Figure 4.19 one can see the joint position for each individual joint for both the simulation and real-world test. This figure in a more general sense has no significant value for evaluating the performance of the implemented reactive scheme. As this scheme does not control the individual joints separately in order to follow a cartesian reference but generates joint torques based on the cartesian displacement between $\mathbf{H}_d^0$ and $\mathbf{H}_{tcp}^0$. Which means that the nullspace of the manipulator can be freely moved within its constraint without effecting the end-effectors configuration. However, this figure can be used to evaluate the performance of the implemented joint limit avoidance and can be taken as a quantitive measure for the evaluation of the LBR iiwas model which was created for the simulation.

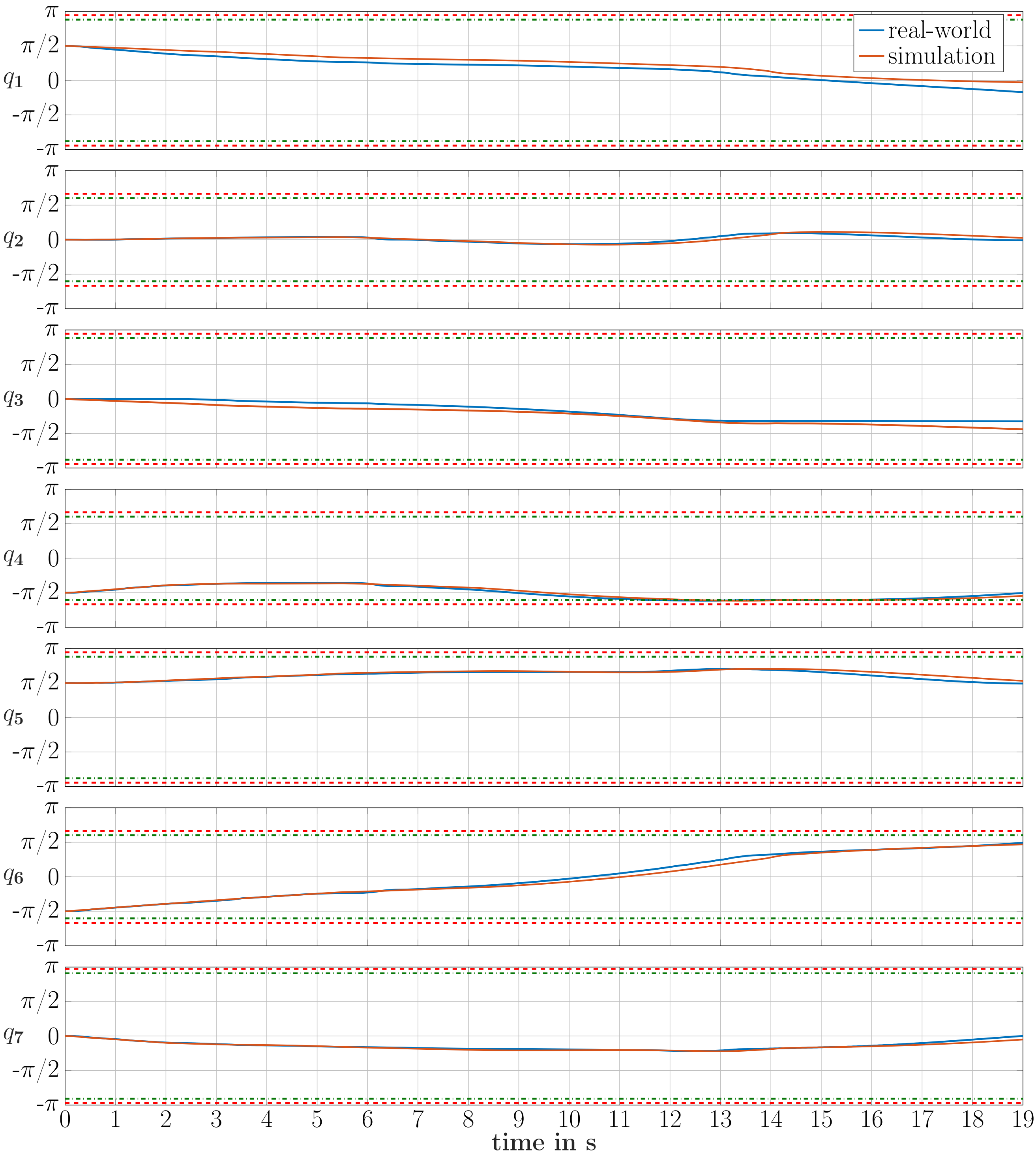

Figure 4.19: Illustrates the joint postion q_i for the simulation and real-world, with the upper/lower joint limits $\bar{q}_{i,\text{limit}}/\underline{q}_{i,\text{limit}}$ marked with a red dashed line and the upper/lower joint limit thresholds $\bar{q}_{i,J}/\underline{q}_{i,J}$ marked with a green dashed line.

When looking at Figure 4.19 on can see how closely the joint positions of the simulation and real-world test follow each other, this can be taken as indication that the model created for the simulation is in coherence with the real LBR iiwa. It can also be seen that all joint positions q_i stay with in its joint limits, with q_4 reaching within the joints lower joint limit threshold $\underline{q}_{4,J}$ at $t = 11$s. However as the joint limit avoidance takes effect at this point and generate a torque τ_{q_4} in the opposite direction and thereby forcing q_4 to stay within its boundaries. A more elaborate explanation and evaluation of this feature will begiven in section 4.3.

4.2.6.2 Joint torques

In Figure 4.20 and Figure 4.21 the motion generating joint torques from the simulation and real-world are visualized. When comparing these two figures it can clearly be seen that the torques from the real manipulator have more frequent and higher fluctuation as the ones in the simulation. This has mainly to do with the discrepancy between the real world LBR iiwa and the model created for the simulation.

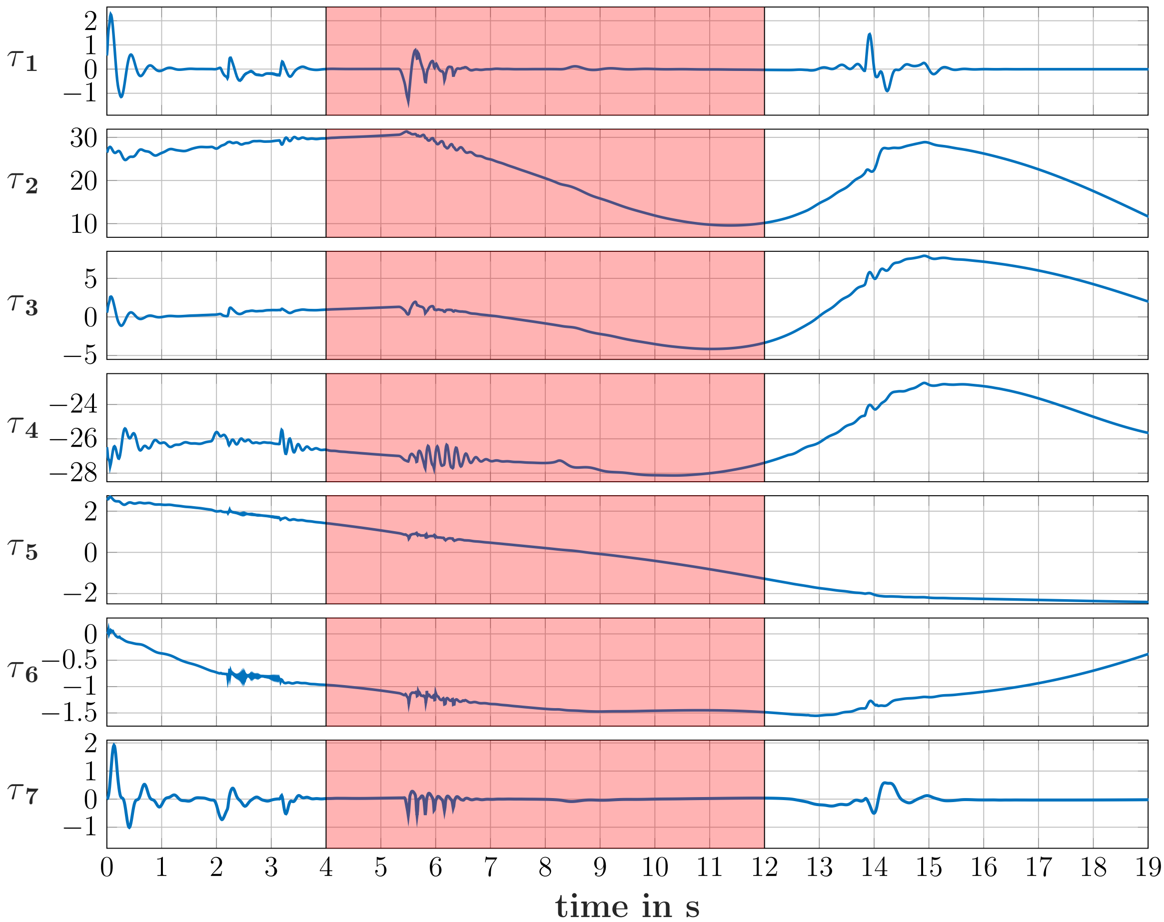

Figure 4.20: Shows the joint specifc torques during the simulation of this test.

However, a comparison between them can be drawn as they show similar behavior, especially at the time instances at which the manipulator encounters/leaves a constraint. It is to note, that only the interaction after the first encounter of the constraint C_2 up until the point at which the manipulator is not in contact with either one of the constraints is relevant for the evaluation of this test and is marked in red.
The reason for this is that the control strategy for the encounter of cartesian constraints (section 3.1), was designed for the purpose of keeping the LBR iiwa within the set boundaires while being already in the compliant mode. Therefore only the area marked in red is representative of the behavior along the constraints. When focusing on the area marked in red one can see minor oscillations as in the time interval $t \in [5, 7]$, which is the same interval in which the LBR iiwa encounters both constraints at the same time and is pulled into the area in which both constraints join. As the LBR iiwa is already in a compliant state, with the scaled down elastic force pulling the end-effector into the constraints and the repelling force of the constraint simply overwhelms this elastic force.
This behavior could be minimized by generating a smaller repelling force. However, as seen in Figure 4.19 this behavior has close to no effect on the joint positions q_i and as mentioned earlier the real use case does not see the end-effector being pulled towards a constraint but pushed against constraint by an external force.

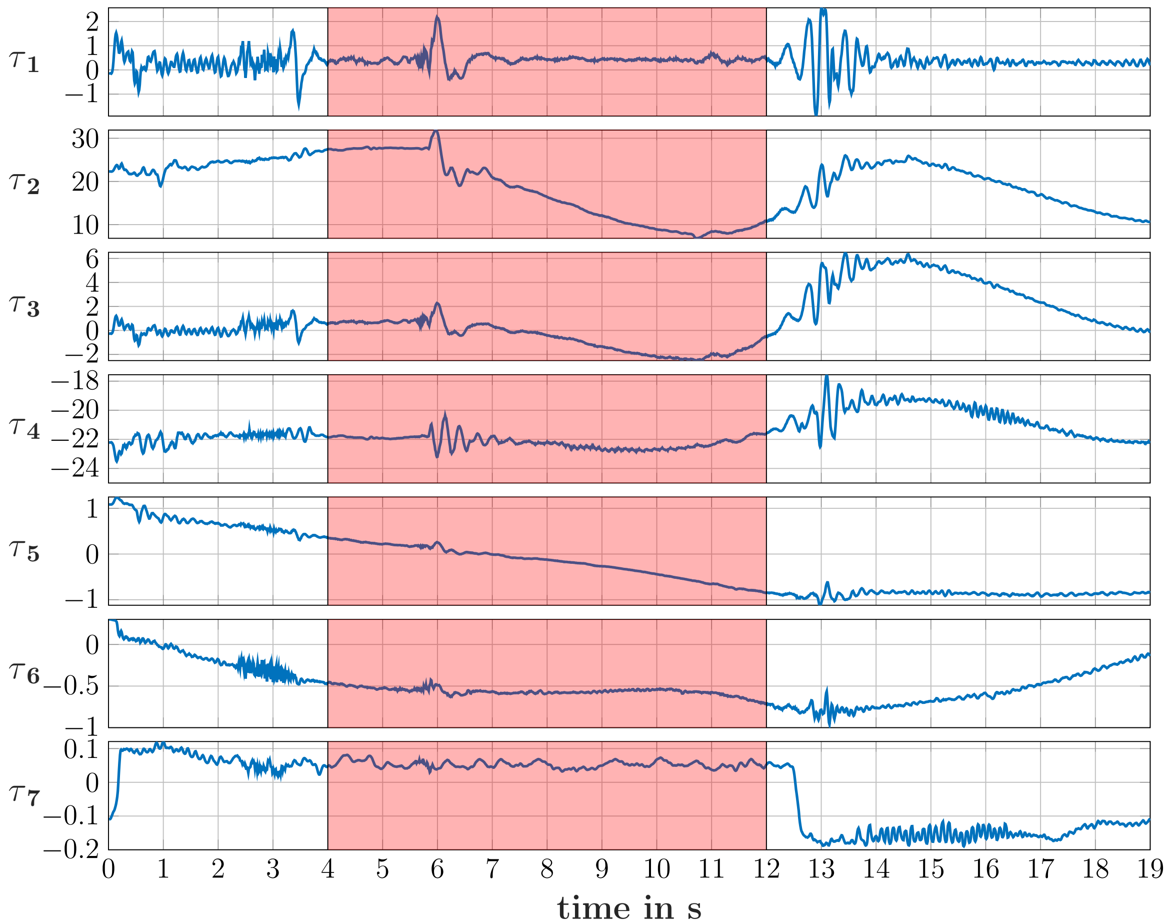

Figure 4.21: Shows the joint specifc torques during the real-world test.

4.2.7 Test evaluation

As mentioned earlier this test was conducted for the purpose of examining the behavior of the implemented Cartesian constraint control strategy when encountering a virtual wall when the LBR iiwa is in a compliant state. It was shown that the reactive control strategy is capable of handling the additional force generated by the virtual wall while the LBR iiwa is in a complaint. The test also showed that the implemented Cartesian constraint control strategy was capable of keeping the LBR iiwa within different Constraints at different time instances as well as then being subject to multiple constraints at the same time. It was also shown how the repelling force is gradually increased as the constraint link approaches the constraint instead of applying an abrupt force. Thereby fulfilling the *Requirement 2* as listed in Table 4.1.

4.3 Test 3: Joint Limit avoidance

The following section describes the test of the robot's behavior when encountering a joint constraint. As the purpose of this feature is to keep the manipulator reaching one of its joint limits while being in a compliant state, it was decided to conduct the test of this feature in the following way: Firstly LBR iiwa is forced into its compliant state, followed by maneuvering it towards the upper and lower joint limit of joint 2 and finally maneuvering it towards joint limits of joint 2,4 and 6 at the same time. The test was conducted by putting the LBR iiwa into a compliant state and maneuvering different links towards their individual joint limits. A visualization of the LBR in a none constraint versus a constratin configuration can be seen in Figure 4.22 and test specific parameter can be found in Table 4.4

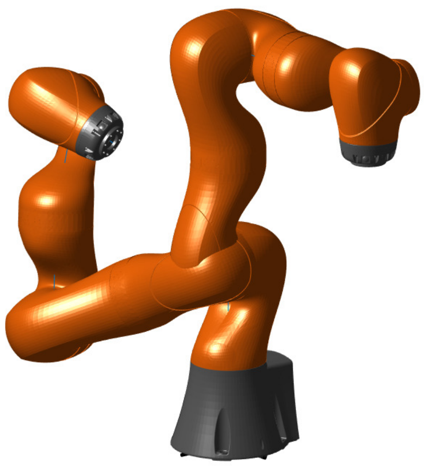

Figure 4.22: Visualises the intial and joint limit configration for joint 2, 3 and 4 of the LBR iiwa.

Parameters	$\bar{q}_{i,J}$ in rad	$\underline{q}_{i,J}$ in rad	Ω
	0.2	0.2	0.025

Table 4.4: Lists the "Joint limit avoidance" test specific parameters.

The upper and lower threshold or activation distance ($\bar{q}_{i,J}$, $\underline{q}_{i,J}$) of each joint were for ease of implementation set for all joint equaly.

4.3.1 Joint limits

As mentioned earlier in this section the two different tests were conducted for the validation of the implemented joint limit avoidance. WIth the first one examining the behavior of this feature at both the upper and lower limit of the 2nd joint. When looking at the Figure 4.23 one can see how the 2nd joint is first maneuvered towards the upper joint limit $\bar{q}_{2,\text{limit}}$ marked in red followed by a movement towards the lower joint limit $\underline{q}_{2,\text{limit}}$marked in grey. In both cases one can see how a torque gradually increasing τ_{q2} in the opposite direction of the joint limit is generated, as q_2 is forced towards the constraint and violates the respective thresholds $\bar{q}_{2,J}$, $\underline{q}_{2,J}$.

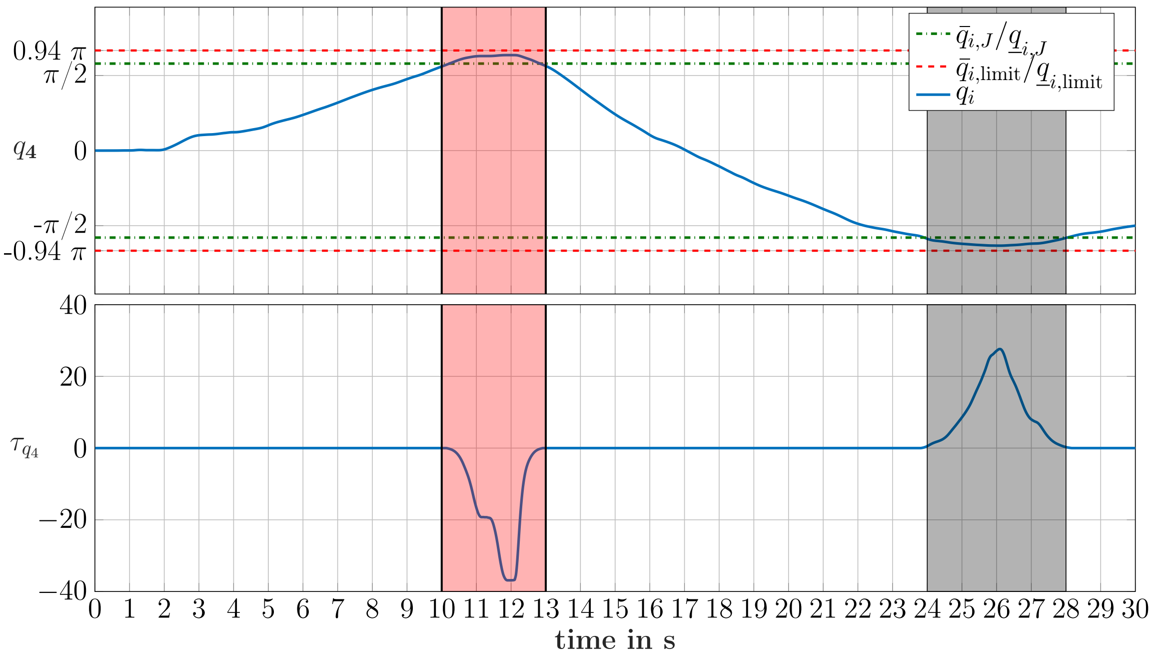

Figure 4.23: Shows the joint position of joint 2 and the torques generated by implemented joint limit avoidance feature.

In addition to the above-described test, it was also tested what happens when the manipulator is maneuvered towards the joint limits of joint 2, 4 and 6 at the same time. The result of this test is visualized in Figure 4.24, where it can be seen that it was not possible to force at least one of the joints into its limit, even though a significant force was applied.

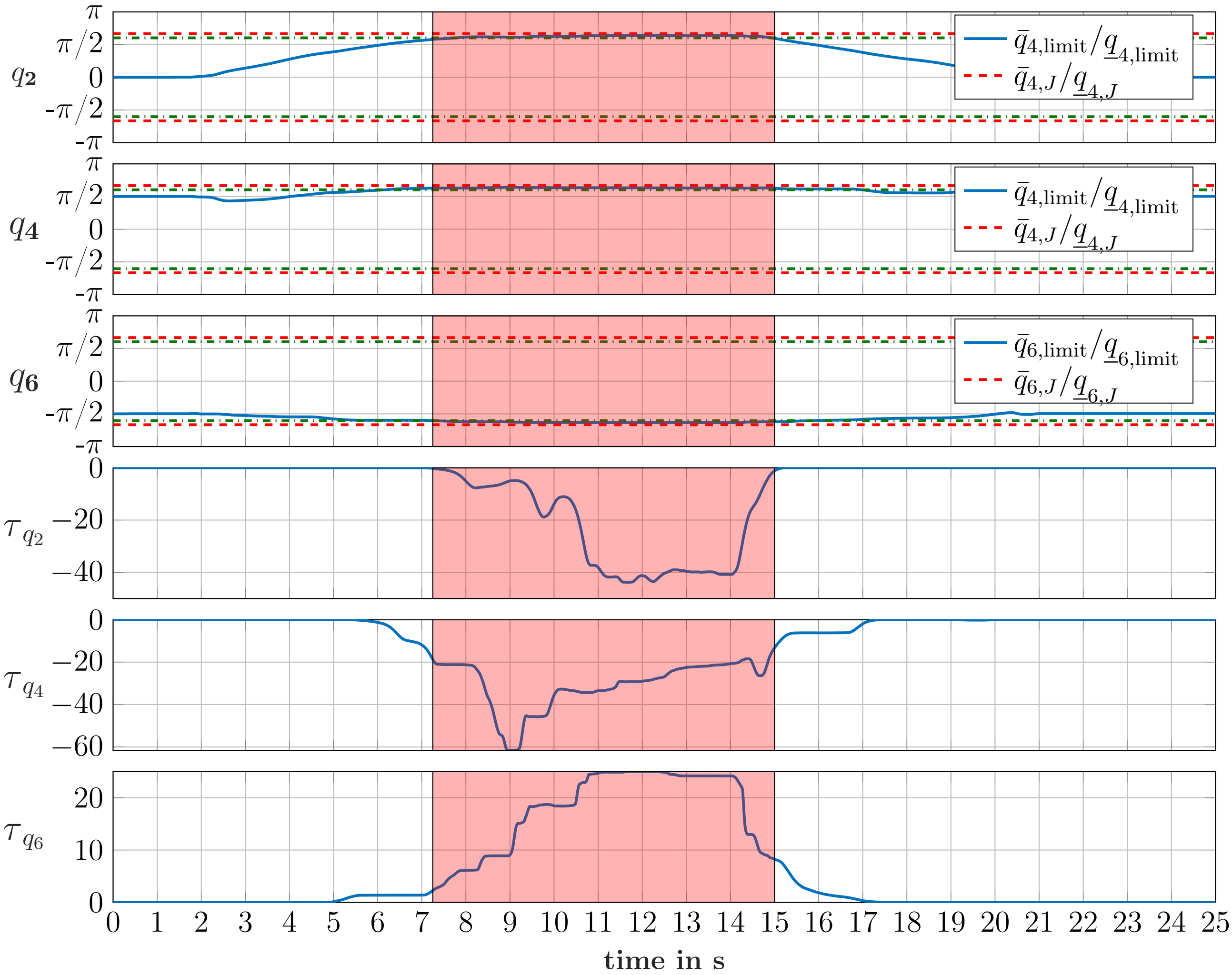

Figure 4.24: Shows the joint position of joint 2,4,6 and the torques gernerated by implemented joint limit avoidance feature.

4.4 Test evaluation

As mentioned earlier this test was conducted for the purpose of examining the behavior of the implemented joint limit avoidance strategy when approaching one or multiple joint limits. It was shown that the implemented joint limit avoidance strategy is capable of keeping the joints of the LBR iiwa within its limit, even when a significant force is applied. The test also showed how the torques which keep the manipulators joint within its limits gradually increase as it approaches a joint limit. Thereby fulfilling the *Requirement 3* as listed in Table 4.1.

Summary 5

5.1 Conclusion

The aim of this thesis was to investigate and implement Cartesian workspace restrictions in form of virtual walls in combination with an energy-aware Impedance controller and joint limit avoidance on the KUKA LBR iiwa.

The work does not only describe the implementation of the overall control strategy, but also a detailed description of the mathematical modeling of the KUKA LBR iiwas kinematics and dynamics using screw theory.
Where kinematics model of the KUKA LBR iiwa covers the derivation of the forward, as well as the differential kinematics. The dynamics of the KUKA LBR iiwa were modeled with the Newton-Euler approach. This model was then used to test the overall control strategy in simulation, before implementing it on the real manipulator.

The overall control strategy was verified by investigating the control strategies subparts behavior based on different test. The investigation of the reactive control scheme capability to handle unplanned interaction it was concluded that it is capable of handling such interactions. As it is able to autonomously adjust the stiffnesses of the spring and the joint damping, in order to stay within the set safety metrics.
The evaluation of the behavior and performance of the implemented the Cartesian constraint control concluded, that it is possible to keep the KUKA LBR iiwa stable and within the constraint even when encountering multiple constraints at the same time.
The investigation into the behavior of the KUKA LBR iiwa when encountering one or multiple joint limits and its capability of avoiding the violation of those joint limits, resulted in the KUKA LBR iiwa being able to staying within the the upper or lower joint limits, even when it is forced into multiple limits at once while being in a compliant state.

The overall conclusion which can be drawn from this is; that the overall implemented control strategy is capable of handling unplanned interaction, staying within the Cartesian constraints as well as avoiding the violation of the LBR iiwa's joint limits.

5.2 Future Work

This chapter gives recommendation for topics, which could be looked into in future works. As the controller implemented in this work has shown, it is possible to control the KUKA LBR iiwa with energy and springs, as well as the limiting its Cartesian workspace with virtual walls when the manipulator is in a compliant state.
This opens up the possibility of looking into:

- Virtual wall design - In this work, the virtual walls used to restrict the Cartesian workspace of the LBR iiwa were defined as a simple 2D plane. However this is not a must, one could also define a virtual wall in form of any smooth manifold $\mathcal{M} \in \mathbb{R}^3$ within the Cartesian workspace of the LBR iiwa.
- Workspace optimization - In this work, the virtual walls were defined as simple 2D planes and placed at locations which were specific for the test. A future research topic could be how to use the previously mentioned virtual wall design and maximize the restricted Cartesian workspace of the LBR iiwa. Thereby limiting the maneuverability of the LBR iiwa in its compliant state by a minimum.

References

[1] Sebastian Hjorth. Implementation of modern control algorithms for the kuka lbr iiwa. 2019.

[2] UNIMATE the first industrial robot. https://www.robotics.org/joseph-engelberger/unimate.com. Accessed: 2018-12-02.

[3] Neville Hogan. Impedance control: An approach to manipulation. part i, ii, iii. volume 107, pages 304 – 313, 07 1984.

[4] Robert Stawell Ball. *A treatise on the theory of screws*. University Press, 1900.

[5] Frank C. Park Kevin M. Lynch. *Modern Robotics : Mechanics, Planning, and Control*. Cambridge: Cambridge University Press, 2017.

[6] S. Shankar Sastry Richard M. Murray, Zexiang Li. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[7] F. C. Park, J. E. Bobrow, and S. R. Ploen. A lie group formulation of robot dynamics. 14:609–618, 1995.

[8] John J. Craig. *Introduction to Robotics*. Pearson, 2014.

[9] Stefano Stramigioli. *From Differential Manifolds to Interactive Robot Control*. PhD thesis, University of Twente, 12 1998.

[10] Stefano Stramigioli. Modern robotics (02 07)(vanuit eindhoven).

[11] C.A. Cardenas Villa. Development of a safety-aware intrinsically passive controller for multi-dof manipulator. Master's thesis, University of Twente, P.O. Box 217 7500 AE Enschede The Netherlands, 2017.

[12] KUKA Roboter GmbH. *LBR iiwa 7 R800, LBR iiwa 14 R820 Specification)*. KUKA Roboter GmbH, 2015.

[13] Stefano Stramigioli. *Modeling and IPC Control of Interactive Mechanical Systems - A Coordinate-Free Approach*. Springer, 2001.

[14] Bruno Siciliano Oussama Khatib. *Handbook of Robotics*. Springer, 2008.

[15] Luigi Villani Giuseppe Oriolo Bruno Siciliano, Lorenzo Sciavicco. *Robotics: Modelling, Planning and Control*. Springer, 2009.

[16] *Real-time obstacle avoidance for manipulators and mobile robots*, volume 2, 1985.

[17] E. D. Fasse and C. M. Gosselin. On the spatial impedance control of gough-stewart platforms, May 1998.

[18] Oussama Khatib, Luis Sentis, Jaeheung Park, and James Warren. Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 10:29–43, 03 2004.

[19] Christian Ott. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*. Springer, Berlin, Heidelberg, 2008.

[20] Theo J.A. De Vries Stramigioli Stefano Gennaro Raiola, Tadele Shiferaw Tadele. Development of a safety- and energy-aware impedance controller for collaborative robots. 3:1–1, 2018.

[21] Josip Loncaric. *GEOMETRICAL ANALYSIS OF COMPLIANT MECHANISMS IN ROBOTICS EUCLIDEAN GROUP, ELASTIC SYSTEMS, GENERALIZED SPRINGS*. PhD thesis, 1985. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2016-05-13.

[22] Stefano Stramigioli. *Energy in Robotics*, volume 6. Now Publishers Inc, 10 2017.

[23] Cesare Fantuzzi Cristian Secchi, Stefano Stramigioli. *Control of Interactive Robotic Interfaces*. Springer, 2007.

[24] T.S. Tadele. *Human-friendly robotic manipulators: safety and performance issues in controller design*. PhD thesis, University of Twente, 11 2014.

[25] *On the spatial impedance control of Gough-Stewart platforms*, volume 2, May 1998.

[26] KUKA Roboter GmbH. *KUKA Sunrise.FRI 1.14*. KUKA Roboter GmbH, 2017.

Cartesian Acceleration A

This appendix covers the derivation of the initial equation of the instantaneous acceleration $\dot{V}_i$. introduced in subsection 2.2.2.
The starting point being the representation of $\dot{V}_i$ in its matrix form $\dot{\widehat{V}}_i$ seen in Equation A.1.

$$\dot{\widehat{V}}_i = \dot{\mathbf{H}}_i^{i-1}(q)\widehat{V}_i\big(\mathbf{H}_i^{i-1}(q)\big)^{-1}+\mathbf{H}_i^{i-1}(q)\dot{\widehat{V}}_i\big(\mathbf{H}_i^{i-1}(q)\big)^{-1}+\mathbf{H}_i^{i-1}(q)\widehat{V}_i\big(\dot{\mathbf{H}}_i^{i-1}(q)\big)^{-1}+\widehat{\xi}_i\ddot{q}_i \quad \text{(A.1)}$$

Where the time derivative of $\dot{\mathbf{H}}_i^{i-1}(q)$ and its inverse is defined as in Equation A.2 and Equation A.3, respectively.
With $\dot{\mathbf{H}}_i^{i-1}(q)$ being defined as the homogeneous transformation $\mathbf{H}_i^{i-1}(q)$ times the joint rate around the body twist of the ith-joint.

$$\dot{\mathbf{H}}_i^{i-1}(q) = \mathbf{H}_i^{i-1}(0)\widehat{\xi}_i e^{\widehat{\xi}_i q_i}\dot{q}_i = \underbrace{\mathbf{H}_i^{i-1}(0)e^{\widehat{\xi}_i q_i}}_{\mathbf{H}_i^{i-1}(q)}\widehat{\xi}_i\dot{q}_i \quad \text{(A.2)}$$

The inverse of $\dot{\mathbf{H}}_i^{i-1}(q)$ is derived by pre- and post-multiplying the homogeneous transformation $\mathbf{H}_{i-1}^{i}(q)$ in order to achieve the change of coordinates in which the motion is represented in.

$$\begin{aligned}
\big(\dot{\mathbf{H}}_i^{i-1}(q)\big)^{-1} &= -\underbrace{\big(\mathbf{H}_i^{i-1}(q)\big)^{-1}}_{\mathbf{H}_{i-1}^{i}(q)}\dot{\mathbf{H}}_i^{i-1}(q)\underbrace{\big(\mathbf{H}_i^{i-1}(q)\big)^{-1}}_{\mathbf{H}_{i-1}^{i}(q)} \\
&= \underbrace{-\mathbf{H}_{i-1}^{i}(q)\big(\underbrace{\mathbf{H}_i^{i-1}(0)e^{\widehat{\xi}_i q_i}}_{\mathbf{H}_i^{i-1}(q)}}_{\mathbf{I}}\widehat{\xi}_i\dot{q}_i\mathbf{H}_{i-1}^{i}(q) \\
&= -\widehat{\xi}_i\dot{q}_i\mathbf{H}_{i-1}^{i}(q)
\end{aligned} \quad \text{(A.3)}$$

The resulting equation when inserting Equation A.2 and Equation A.3 in Equation A.1 is shown in Equation A.4.

$$\begin{aligned}
\dot{\widehat{V}}_i = {} & \mathbf{H}_i^{i-1}(q)\widehat{\xi}_i\dot{q}_i\widehat{V}_i\mathbf{H}_{i-1}^{i}(q) + \mathbf{H}_i^{i-1}(q)\dot{\widehat{V}}_i\mathbf{H}_{i-1}^{i}(q) \\
& + \mathbf{H}_i^{i-1}(q)\widehat{V}_i\big(-\widehat{\xi}_i\dot{q}_i\mathbf{H}_{i-1}^{i}(q))\big) + \widehat{\xi}_i\ddot{q}_i
\end{aligned} \quad \text{(A.4)}$$

When restructuring Equation A.4 into the format as seen in Equation A.5, one can see that the equation can be split in to 3 separate parts. The first part describes the joint acceleration $\ddot{q}$ around the respective body twist $\widehat{\xi}_i$, the second part represents the Cartesian acceleration of the $i-1$-link represented in the ith-link and the third part being the generalized cross-product of two twist in matrix form.

$$
\begin{aligned}
\dot{\widehat{V}}_i &= \widehat{\xi}_i\ddot{q}_i + \mathbf{H}_i^{i-1}(q)\dot{\widehat{V}}_i\mathbf{H}_{i-1}^i(q) + \Big(\mathbf{H}_i^{i-1}(q)\widehat{\xi}_i\dot{q}_i\widehat{V}_i\mathbf{H}_{i-1}^i(q) + \mathbf{H}_i^{i-1}(q)\widehat{V}_i\big(-\widehat{\xi}_i\dot{q}_i\mathbf{H}_{i-1}^i(q)\big)\Big) \\
&= \widehat{\xi}_i\ddot{q}_i + \mathbf{H}_i^{i-1}(q)\dot{\widehat{V}}_i\mathbf{H}_{i-1}^i(q) + \underbrace{\Big(\mathbf{H}_i^{i-1}(q)\widehat{V}_i\mathbf{H}_{i-1}^i(q)\widehat{\xi}_i\dot{q}_i - \widehat{\xi}_i\dot{q}_i\mathbf{H}_i^{i-1}(q)\widehat{V}_i\mathbf{H}_{i-1}^i(q)\big)\Big)}_{[\cdot,\cdot]\rightarrow\text{Lie bracket}}
\end{aligned}
\tag{A.5}
$$

Until now the Cartesian velocity $\dot{V}_i$ was represented in matrix form, however as the vector form of it is needed for the calculations in subsection 2.2.2 it must be reformulated as in Equation A.6.

$$
\begin{aligned}
\dot{V}_i &= \xi_i\ddot{q}_i + Ad_{\mathbf{H}_{i-1}^i(q)}\left(\dot{V}_{i-1}\right) + \left[\underbrace{Ad_{\mathbf{H}_i^{i-1}(q)}V_{i-1}}_{V_i-\xi_i\dot{q}_i}, \xi_i\dot{q}_i\right] \\
&= \xi_i\ddot{q}_i + Ad_{\mathbf{H}_{i-1}^i(q)}\left(\dot{V}_{i-1}\right) + [V_i - \xi_i\dot{q}_i, \xi_i\dot{q}_i] \\
&= \xi_i\ddot{q}_i + Ad_{\mathbf{H}_{i-1}^i(q)}\left(\dot{V}_{i-1}\right) + [V_i, \xi_i\dot{q}_i] - \underbrace{[\xi_i\dot{q}_i, \xi_i\dot{q}_i]}_{0} \\
&= \xi_i\ddot{q}_i + Ad_{\mathbf{H}_{i-1}^i(q)}\dot{V}_{i-1} + [V_i, \xi_i\dot{q}_i] \\
&= \xi_i\ddot{q}_i + Ad_{\mathbf{H}_{i-1}^i(q)}\dot{V}_{i-1} + ad_{V_i}\xi_i\dot{q}_i
\end{aligned}
\tag{A.6}
$$

General equation of motion B

This appendix shows the general derivation of the Mass-matrix $\mathbf{M}(q)$, Coriolis and centrifugal forces $\bar{\mathbf{C}}(q,\dot{q})\dot{q}$ and potential forces $\bar{\mathbf{G}}(q)$ in joint space from the inverse dynamic algorithm discussed in subsection 2.2.2, based on the work done in [5, 7].
It is possible to express the recursive inverse dynamic algorithm in form of a set of matrix equations. This is done by forming out of the different stacked vectors and block-matrices. As starting point one defines the Cartesian Velocities and Wrenches as stacked vectors as seen in Equation B.1 and Equation B.2.

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} \in \mathbb{R}^{6n} \tag{B.1}$$

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} \in \mathbb{R}^{6n} \tag{B.2}$$

With Ξ (Equation B.3) and $\mathbf{M}_c$ (Equation B.4)being defined as constant diagonal block matrices representing the body twist coordinates and mass-inertia's expressed and seen in links own frame frames respectively.

$$\Xi = \begin{bmatrix} \xi_1 & 0 & \cdots & 0 \\ 0 & \xi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \xi_n \end{bmatrix} \in \mathbb{R}^{6n \times n} \tag{B.3}$$

$$\mathbf{M}_c = \begin{bmatrix} \mathbf{M}_{c_1} & 0 & \cdots & 0 \\ 0 & \mathbf{M}_{c_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \mathbf{M}_{c_n} \end{bmatrix} \in \mathbb{R}^{6n \times n} \tag{B.4}$$

And where Equation B.5,Equation B.6 and Equation B.7 are holding the adjoint transformation of the different Cartesian velocities, body twist and transformation matrices respectively.

$$ad_V = \begin{bmatrix} ad_{V_1} & 0 & \cdots & 0 \\ 0 & ad_{V_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & ad_{V_n} \end{bmatrix} \in \mathbb{R}^{6n \times 6n} \tag{B.5}$$

$$ad_{\Xi\dot{q}} = \begin{bmatrix} ad_{\xi_1\dot{q}_1} & 0 & \cdots & 0 \\ 0 & ad_{\xi_2\dot{q}_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & ad_{\xi_n\dot{q}_n} \end{bmatrix} \in \mathbb{R}^{6n\times 6n} \tag{B.6}$$

$$\mathcal{S}(q) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ Ad_{\mathbf{H}_1^2} & 0 & \cdots & 0 & 0 \\ 0 & Ad_{\mathbf{H}_2^3} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & Ad_{\mathbf{H}_{n-1}^n} & 0 \end{bmatrix} \in \mathbb{R}^{6n\times 6n} \tag{B.7}$$

One then defines the Cartesian velocity of the base as Equation B.8,the Cartesian acceleration of the base as Equation B.9 and the wrench acting on the *tcp*-frame as in Equation B.10.

$$V_{base} = \begin{bmatrix} Ad_{\mathbf{H}_0^1} V_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{6n} \tag{B.8}$$

$$\dot{V}_{base} = \begin{bmatrix} Ad_{\mathbf{H}_0^1} \left(\dot{V}_0\right) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{6n} \tag{B.9}$$

$$W_{tcp} = \begin{bmatrix} 0 \\ \vdots \\ Ad_{\mathbf{H}_n^{n+1}}{}^{\top} W_{n+1} \end{bmatrix} \in \mathbb{R}^{6n} \tag{B.10}$$

With the above defined matrices it is possible to represent equation of the recursive inverse dynamic algorithm by the matrix equation seen in Equation B.11.

$$\begin{aligned} V &= \mathcal{S}(q)V + \Xi\dot{q} + V_0 \\ \dot{V} &= \mathcal{S}(q)\dot{V} + \Xi\ddot{q} - ad_{\Xi\dot{q}}\Big(\mathcal{S}(q)V + V_{base}\Big) + \dot{V}_{base} \\ W &= \mathcal{S}^{\top}(q)W + \mathbf{M}_c\dot{V} - ad_V^{\top}\Big(\mathbf{M}_c V\Big) + W_{tcp} \\ \tau &= \Xi^{\top} W \end{aligned} \tag{B.11}$$

With $S(q)$ being a nil-potent matrix of order n ($S^n(q) = 0; n \in \mathbb{N}$), which has the property of $det\big(\mathbf{I}_n + S(q)\big) = 1$ and therefore $\big(\mathbf{I}_n + S(q)\big)$ is invertible as seen in Equation B.12.

$$\mathcal{N}(q) = (\mathbf{I} - \mathcal{S}(q))^{-1} = \mathbf{I} + \mathcal{S}(q)) + \cdots + \mathbf{I} + \mathcal{S}^{n-1}(q)) \tag{B.12}$$

Which results in an lower triangular block matrix as seen in Equation B.13.

$$\mathcal{N}(\theta) = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ Ad_{\mathbf{H}_1^2} & I & 0 & \cdots & 0 \\ Ad_{\mathbf{H}_1^3} & Ad_{\mathbf{H}_2^3} & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Ad_{\mathbf{H}_1^n} & Ad_{\mathbf{H}_2^n} & Ad_{\mathbf{H}_3^n} & \cdots & I \end{bmatrix} \in \mathbb{R}^{6n\times 6n} \tag{B.13}$$

With the above derived $\mathcal{N}$ it is possible to reformulate Equation B.11 to Equation B.14.

$$\begin{aligned} V &= \mathcal{N}(q)\big(\Xi\dot{q} + V_{\text{base}}\big) \\ \dot{V} &= \mathcal{N}(q)\Big(\Xi\ddot{q} + ad_{\Xi\dot{q}}\big(\mathcal{S}(q)V\big) + ad_{\Xi\dot{q}}\big(V_{\text{base}}\big) + \dot{V}_{\text{base}}\Big) \\ W &= \mathcal{N}^\top(q)\Big(\mathbf{M}_c\dot{V} - ad_V^\top\big(\mathbf{M}_c V\big) + W_{tcp}\Big) \\ \tau &= \Xi^\top W \end{aligned} \tag{B.14}$$

With these definitions it is possible to describe $\mathbf{M}(q)$, $\bar{\mathbf{C}}(q,\dot{q})\dot{q}$ and $\bar{\mathbf{G}}$ as seen in Equation B.15.

$$\begin{aligned} \mathbf{M}(q) &= \Xi^\top\mathcal{N}^\top(q)\mathbf{M}_c\mathcal{N}(q)\Xi \\ \bar{\mathbf{C}}(q,\dot{q})\dot{q} &= -\Xi^\top\mathcal{N}^\top(q)\left(\mathbf{M}_c\mathcal{N}(q)ad_{\Xi\dot{q}}\mathcal{S}(q) + ad_V^\top\mathbf{M}_c\right)\mathcal{N}(q)\Xi\dot{q} \\ \bar{\mathbf{G}}(q) &= \Xi^\top\mathcal{N}^\top(q)\mathbf{M}_c\mathcal{N}(q)\dot{V}_{base} \end{aligned} \tag{B.15}$$

As the matrix equation in Equation B.14 describe a recursive algorithm, the matrix equation in Equation B.15 must also describe a recursive algorithm. Which means they also can be calculated separately in a more time efficient manner than with the Lagrangian formalism.

Implementation C

This chapter describes the general setup and its components as well as the structure of the code for the implemented controller the implementation of the controller previously described in chapter 3.

C.1 Setup overview

This section describes the hardware setup used in this work. The content in this section was written based on the information found in [12, 26]. A visualization of the setup used for the implementation and testing process can be seen in Figure C.1. In the remaining of this section, the different components of the visualized setup will be described briefly.

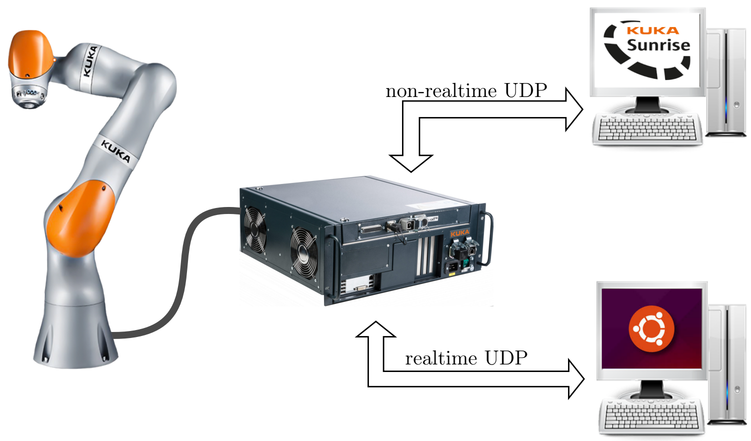

Figure C.1: Visualizes the used harware setup, consisting of an KUKA LBR iiwa R800, KRC unit and two PCs used for implementation purposes.

C.1.1 KUKA LBR iiwa R800

As mentioned previously in this work, the industrial manipulator used in this work is the KUKA LBR iiwa R800. The KUKA LBR iiwa R800 is a collaborative manipulator and was chosen for this project as is equipped with position and torque sensors and torque controlled motors in each joint. In addition the LBR iiwa has compared to more conventional industrial manipulators 7 DOFs. With this extra DOF the KUKA LBR iiwa has an increased dexterity and is able to avoid certain sinuglarties, which more conventional industrial manipulators with only 6 DOF could encounter. The technical specification such as range of motion and maximal joint velocity are given in Table C.1 and Table C.2 respectively.

Range of motion in $^\circ$						
Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7
± 170	± 120	± 170	± 120	± 170	$pm120$	± 175

Table C.1: Shows the range of motion for each joint of the KUKA LBR iiwa.[12]

Maximal joint velocity in $^\circ \backslash s$						
Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7
98	98	100	130	140	180	180

Table C.2: Shows the maximal joint velocity for each joint of the KUKA LBR iiwa.[12]

C.1.2 Control interface

This section gives a short summary of how one can communicate with the KUKA LBR iiwa. As seen in Figure C.1, the LBR iiwa is connected to and get its control signals from the *KUKA Robot Controller*(KRC), which is connected two PCs via a real-time and non-realtime UPD connection respectively.
The PC connected via the non-realtime connection runs Windows and KUKAs *Sunrise.Workbench* programm and is used to upload Robot applications to, as well as download data-recording from the KRC. The other PC runs Ubuntu and communicates the with the KRC by sending and receiving the control inputs/outputs in real-time via KUKAs *Fast Research Interface* (FRI). In the following sections describe the above mentioned different communication components in greater detail.

C.1.3 KUKA Robot Controller

As mentioned above the KUKA LBR iiwa is controlled via the KUKA Robot Controller (KRC), also known as the *KUKA Sunrise Cabinet*. The KRC is responsible for the transmission control inputs as well as the reading the data of the integrated sensors. One has two possibilities to control the LBR iiwa. Firstly one can upload a Java-based applications onto KRC via the KUKA Line interface port (KLI), which is created in the *Sunrise.Workbench*.
This controller runs on the KRC natively. Secondly one can create an FRI client application on the Linux-based PC, which connects to the KRC via a UDP connection with a maximum rate of 1kHz through the KUKA Option Network Interface port (KONI). In this setup, the calculation of the control inputs is outsourced to the PC.

C.1.4 Sunrise.Workbench

As mentioned earlier the Sunrise.Workbench is a tool used to program robot applications in Java, which are loaded onto and are executed on the KRC. With this program, it is possible to define different motion type and patterns which the robot should execute, as well as the integration of external libraries (e.g. sensors, FRI). It offers the possibility to control the LBR iiwa with the following control strategies: Joint impedance Control, Cartesian impedance control, position and velocity control.
Through which the LBR iiwa can execute following motion patterns: spline, point-to-point,

linear and circular motions as well as holding the current position. This Software is also used to configure the KRC as an FRI server.

C.1.5 Fast Research Interface

The Fast Research Interface (FRI) as mentioned previously provides a real-time interface between an application FRI server application on the KRC and an FRI client application on an external system via UDP.
However in order to take advantage of this interface, one must first load a Java application onto the KRC via the Sunrise.Workbench. This application must be configured in such a way that it overlays the control signal sent by the FRI C++ client. These control signals can either be positions, torques or wrenches depending on how the user specifies the Java application on the KRC and C++ application on the FRI Client side.

C.2 Code description

This section describes the structure of the program for the implemented control strategy described in chapter 3. The structure of this section is the following: Firstly a general overview over the overall program is given (subsection C.2.1) followed by a detailed description of the different sub-parts of the program in the chronological order as introduced in subsection C.2.1

C.2.1 Overview

The following section discusses the structure of the program of the overall program used for computing the control input τ. It is assumed that needed parameters (e.g. constraints C_j, stiffnesses $K_{t,r,c}$, etc.) are already defined and initialized before the following part of the program is executed. Furthermore, it is assumed that the robot is at the beginning of the program already in its starting configuration $\mathbf{H}^0_{tcp}(0)$ and the program has a predefined desired trajectory/configuration $\mathbf{H}^0_d(t)$.

In Figure C.2 general structure of the program for computing the control input τ can be seen. First *Cartesian constraint control scheme* computes the repelling Wrench $W^{0,i}_{C_j}$ for an constraint C_j for the i^{th} link. Secondly by the *Reactive control scheme*, from which the a motion generating torque τ_{ctrl} based on the displacement H^{tcp}_d and the repelling Wrenches $W^{0,i}_{C_j}$ from active constraints is computed.Followed by the computation of the torques τ_{q_i} in the opposite direction generated by the *Joint limit avoidance* and the computation of the $\tau(q)$ (3.38) before sending them to the manipulator.
In the remain of this chapter the just described steps will elaborated on in greater detail in the following sections, excluding the computation and sending of $\tau(q)$ as it is seen as trivial.

C.2.2 Constraints

This section elaborates on the implementation of the Cartesian constraint control scheme. As mentioned in subsection C.2.1, it is assumed that the constraint are already defined and initialized prior to execution of the described part of the program.

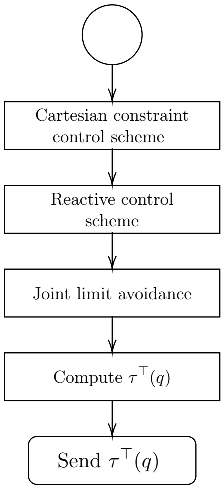

Figure C.2: Illustrates the flowchart of overall project structure.

In the beginning of each iteration the program calculates the projection $p^0_{i,C_j}(q)$ (3.3) of the position p^0_i of each constraint link onto the constraint C_j.

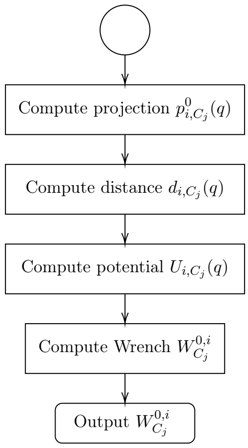

Figure C.3: Visualizes the different calculation steps within the Cartesian Constraint control scheme.

Followed by the computation of the euclidean distance $d_{i,C_j}(q)$ between these to points, which is used as input for the calculation of the gain of the repelling Wrench $W^{0,i}_{C_j}$ based on a potential function $U_{i,C_j}(q)$ (3.5) for the j^{th} constraint. A graphical representation of $U_{i,C_j}(q)$ can be seen in Figure C.4, where the euclidean distance d_{i,C_j} between each constraint joint and their constraints is smaller then a minimum distance x_j. Note that x_j is a design parameter chosen by the user, and represents the distance to the constraint C_j, at which if violated an potential $U_{i,C_j}(q) > 0$ is calculated. Furthermore are κ_j and γ design parameter the user can chose and influence the aggressiveness of $U_{i,C_j}(q)$, when x_j is violated. A comprehensive description of $U_{i,C_j}(q)$ is given subsection 3.1.4.

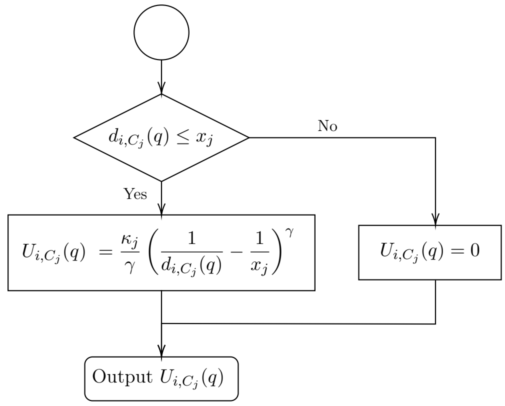

Figure C.4: Illustrates the flowchart for the calculation of the potential U_{i,C_j}.

C.2.3 Reactive Control scheme

The following section discusses the structure of the program, dedicated for reactive control scheme, which is visualized in Figure C.5.
In the first step of each iteration with in this part of the program is the computation of the transformation between the frames Ψ_d and Ψ_{tcp} ($\mathbf{H}_{tcp}^{d}(t)$). In combination with the co-stiffness $G_{t,r,c}$ of the spatial spring is $\mathbf{H}_{tcp}^{d}(t)$ used to compute the total potential energy V_{total} (3.21), based on the described displacement between Ψ_{tcp} and Ψ_d displacement by $\mathbf{H}_{tcp}^{d}(t)$. In addition the kinetic energy exerted by the system T_{total} (3.20) over time is calculated. The calculated potential energy V_{total}, kinetic energy T_{total} and the resulting total energy E_{total} (3.19), are then used as input for the *Energy Scaling* function.

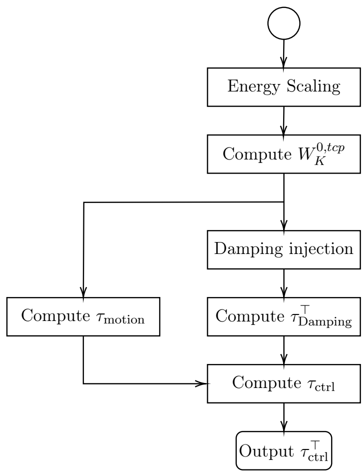

Figure C.5: Illustrates the main structure of the reactive control scheme program.

Based on a user defined maximal energy threshold E_{max}, the scaling factor λ (3.23) is computed. The scaling factor λ is then used to compute new Co-stiffnesses $G_{t,r,c}$(3.25))as seen in Figure C.6.

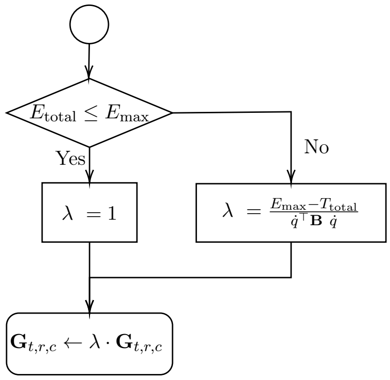

Figure C.6: Illustrates the flowchart of Energy scaling.

The next step in the program is as seen in Figure C.3, the computation of the motion generating Wrench $W_K^{0,tcp}$ (3.16) with the newly calculated Co-stiffnesses $G_{t,r,c}$. The newly calculate elastic Wrench $W_K^{0,tcp}$ is then in combination with the in subsection C.2.2 described Wrenches $W_{C_j}^{0,i}$(3.10) to calculate the torques $\tau_{motion}^\top$ (3.18) and of power P_{ctrl} (3.29) which the controller is capable of transferring to the robot. This is then used as input to *Damping injection* function visualized in Figure C.7 and described in subsection 3.4.2. The *Damping injection* function limits P_{ctrl}, by increasing the Damping coefficients $\mathbf{B}$ with the scaling parameter β (3.33). The computation of β is based on the robots energy exchanged with its environment due to its motion and a by the user predefined maximal power threshold $P_{\max}$.

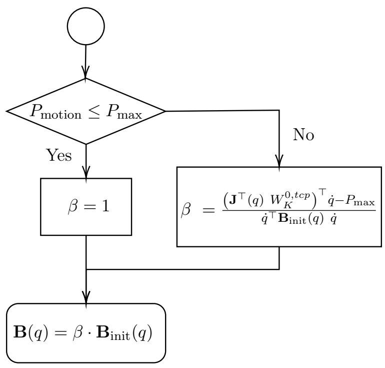

Figure C.7: Shows the flowchart of Damping injection.

From this newly computed $\mathbf{B}$ (3.32), resulting damping torque τ_{Damping} (3.26) is computed. The summation of τ_{motion} and $\tau_{Damping}$ is equal to τ_{ctrl} as defined in Equation 3.38.

C.2.4 Joint limit avoidance

The following section gives an inside into how the implementation of the Joint limit avoidance, described in section 3.5 and visualized in Figure C.8. In the first step the angular distances $\bar{q}_i$ and $\underline{q}_i$, which describe the angular displacement between the joints current angular position q_i and its the upper ($\bar{q}_{i,\text{limit}}$) and lower ($\underline{q}_{i,\text{limit}}$) physical limit. In the next step it is checked if angular displacement $\bar{q}_i$ is smaller than a user defined upper threshold $\bar{q}_{i,J}$. If this is the case a torque $\tau_{q_i} = \tau_{\bar{q}_i}$ as defined in Equation 3.35 is generated.

In case this $\bar{q}_i$ is not smaller than $\bar{q}_{i,J}$ it is check if $\underline{q}_i$ is smaller than a user defined lower threshold $\underline{q}_{i,J}$. If this is true a torque $\tau_{q_i} = \tau_{\underline{q}_i}$ as defined in Equation 3.35 is generated. In the event that none of these two conditions no torque is generated as it does not violate the joint constraint set by the upper and lower thresholds.

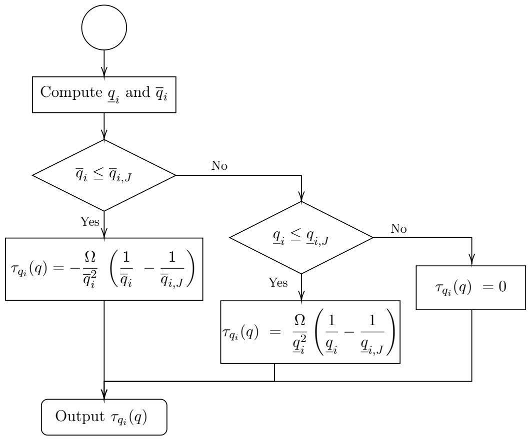

Figure C.8: Illustrates the flowchart of Joint avoidance.

Trajectory and Contraints D

In this part of the report the trajectory used for the validation of the implemented control strategy (chapter 4) are described.

D.1 Cartesian Constraint Test

The trajectory chosen to verify the features of the controller described in chapter 3 is defined as an translation $p_d^0(t)$, (D.1) and a fixed orientation defined by the Rotation matrix $\mathbf{R}_d^0(t)$ (D.2).

$$p_d^0(t) = \begin{bmatrix} 0.129 + 0.5 \cdot \sin(t) \\ 0.4 \cdot \cos(t) \\ 0.74 + 0.35 \cdot \sin(2 \cdot t) \end{bmatrix} \tag{D.1}$$

$$\mathbf{R}_d^0(t) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{D.2}$$

When merging (D.1) and (D.2) with each other they result in the homogeneous transformation matrix $\mathbf{H}_d^0(t)$ (D.3).

$$\mathbf{H}_d^0(t) = \begin{bmatrix} 0 & 0 & 1 & 0.129 + 0.5 \cdot \sin(t) \\ 1 & 0 & 0 & 0.4 \cdot \cos(t) \\ 0 & 1 & 0 & 0.74 + 0.35 \cdot \sin(2 \cdot t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{D.3}$$

This transformation is then used as an anchor point for the spring pulling on the end-effector of the robot. An illustration of the trajectory is given in Figure D.3

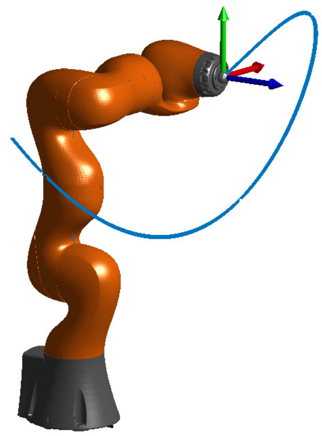

Figure D.1: Illustrated the KUKA LBR iiwa in the configuration $\mathbf{H}^0_{tcp}$ at $t = 0$s and the sline trajectory.

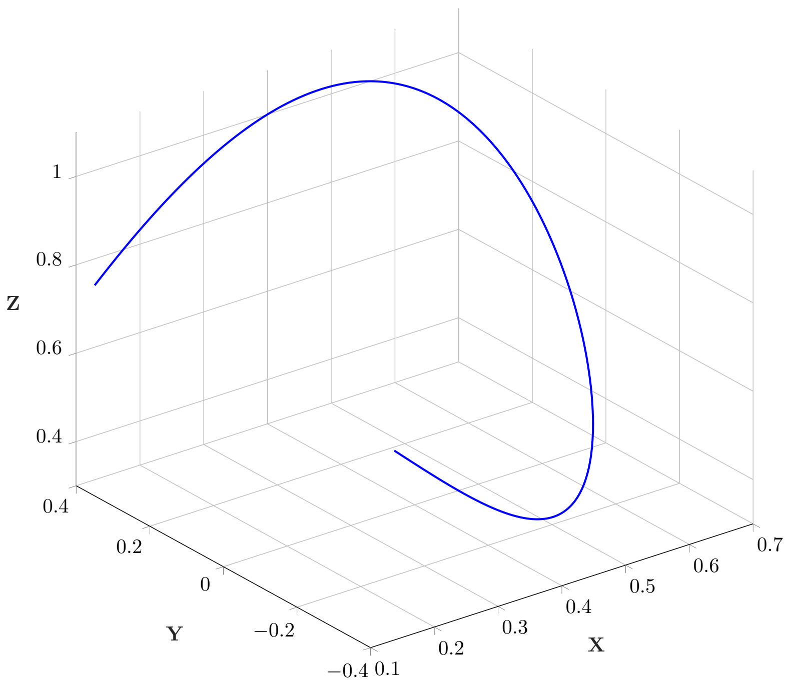

Figure D.2: Illustrated the sline trajectory $p^0_d(t)$.

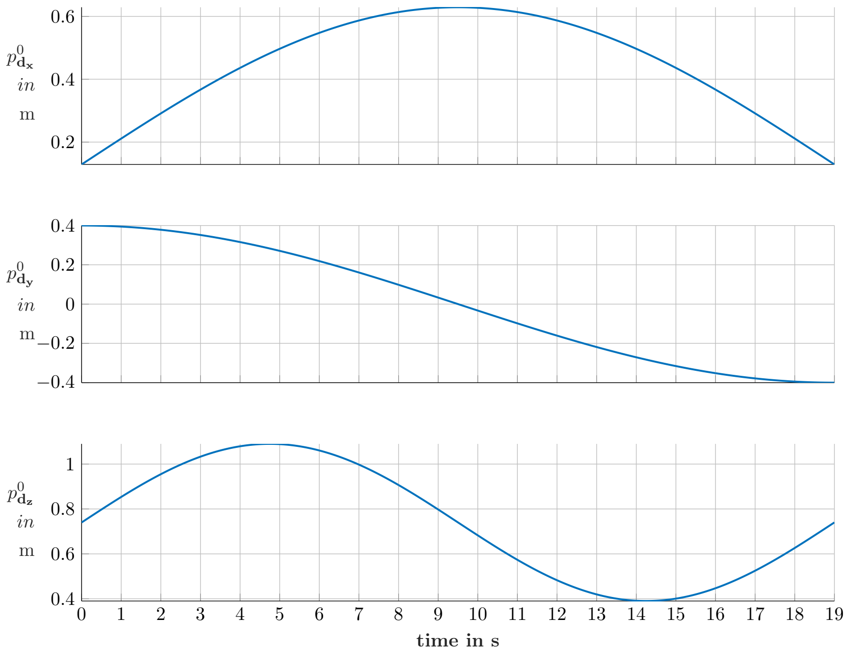

Figure D.3: Illustrated the x,y and z coordinate specific components of the sline trajectory $p^0_d(t)$.

LBR Model E

This appendix lists the symbolically solved **spatial twists** $\tilde{\xi}$, the **Forward Kinematic** equation for H^0_{tcp} and the **Body Jacobian** in **??**,**??** and **??** respectively. In this appendix the following notation for the use of the trigonometric functions will be used used.

$$\begin{aligned}
\sin(\theta_1) &= s_1 \\
\cos(\theta_1) &= c_1 \\
\sin(\theta_1 + \theta_2) &= s_{1,2} \\
\cos(\theta_1 + \theta_2) &= c_{1,2}
\end{aligned}$$

E.1 Spatial Twist

$$\xi_1^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \xi_2^0 = \begin{bmatrix} -0.340 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \xi_3^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \xi_4^0 = \begin{bmatrix} 0.74 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} \quad \xi_5^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \xi_6^0 = \begin{bmatrix} -1.14 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \quad \xi_7^0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

E.2 Forward Kinematic

This part of the appendix shows the symbolic representation of the homogenoues transformation from $\mathbf{H}^0_{tcp}$.

E.2.1 Rotation Matrix

$$
\begin{aligned}
\mathbf{R}^0_{tcp} &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \\
r_{11} &= s_7\left(s_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) - c_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right) \\
&\quad - c_7\left(s_6\left(s_4\left(s_1 s_3 - c_1 c_2 c_3\right) + c_1 c_4 s_2\right) + c_6\left(c_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) + s_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right)\right) \\
r_{12} &= s_7\left(s_6\left(s_4\left(s_1 s_3 - c_1 c_2 c_3\right) + c_1 c_4 s_2\right) + c_6\left(c_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) + s_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right)\right) \\
&\quad + c_7\left(s_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) - c_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right) \\
r_{13} &= c_6\left(s_4\left(s_1 s_3 - c_1 c_2 c_3\right) + c_1 c_4 s_2\right) - s_6\left(c_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) + s_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right) \\
r_{21} &= c_7\left(s_6\left(s_4\left(c_1 s_3 + c_2 c_3 s_1\right) - c_4 s_1 s_2\right) + c_6\left(c_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) + s_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right)\right) \\
&\quad - s_7\left(s_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) - c_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right) \\
r_{22} &= -s_7\left(s_6\right)\left(s_4\left(c_1 s_3 + c_2 c_3 s_1\right) - c_4 s_1 s_2\right) + c_6\left(c_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) + s_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right)\right) \\
&\quad - c_7\left(s_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) - c_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right) \\
r_{23} &= s_6\left(c_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) + s_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right) - c_6\left(s_4\left(c_1 s_3 + c_2 c_3 s_1\right) - c_4 s_1 s_2\right) \\
r_{31} &= c_7\left(c_6\left(c_5\left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) - s_6\left(c_2 c_4 + c_3 s_2 s_4\right)\right) - s_7\left(s_5\left(c_2 s_4 - c_3 c_4 s_2\right) - c_5 s_2 s_3\right) \\
r_{32} &= -c_7\left(s_5\left(c_2 s_4 - c_3 c_4 s_2\right) - c_5 s_2 s_3\right) - s_7\left(c_6\left(c_5\left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) - s_6\left(c_2 c_4 + c_3 s_2 s_4\right)\right) \\
r_{33} &= s_6\left(c_5\left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) + c_6\left(c_2 c_4 + c_3 s_2 s_4\right)
\end{aligned}
\tag{E.1}
$$

E.2.2 Position Vector

$$
\begin{aligned}
p^0_{tcp} &= \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \\
p_x &= l_7\left(c_6\left(s_4\left(s_1 s_3 - c_1 c_2 c_3\right) + c_1 c_4 s_2\right) - s_6\left(c_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) + s_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right)\right) \\
&\quad - l_5 s_4\left(s_1 s_3 - c_1 c_2 c_3\right) - l_2 c_1 s_2 + l_8 s_6\left(c_5\left(c_4\left(s_1 s_3 - c_1 c_2 c_3\right) - c_1 s_2 s_4\right) + s_5\left(c_3 s_1 + c_1 c_2 s_3\right)\right) \\
&\quad - l_8\left(c_6 - 1\right)\left(s_4\left(s_1 s_3 - c_1 c_2 c_3\right) + c_1 c_4 s_2\right) - l_5 c_1 s_2\left(c_4 - 1\right) \\
p_y &= l_8\left(s_4\left(c_1 s_3 + c_2 c_3 s_1\right) - c_4 s_1 s_2\right)\left(c_6 - 1\right) \\
&\quad - l_7\left(c_6\left(s_4\left(c_1 s_3 + c_2 c_3 s_1\right) - c_4 s_1 s_2\right) - s_6\left(c_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) + s_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right)\right) \\
&\quad + l_5 s_4\left(c_1 s_3 + c_2 c_3 s_1\right) - l_2 s_1 s_2 - l_8 s_6\left(c_5\left(c_4\left(c_1 s_3 + c_2 c_3 s_1\right) + s_1 s_2 s_4\right) + s_5\left(c_1 c_3 - c_2 s_1 s_3\right)\right) \\
&\quad - l_5 s_1 s_2\left(c_4 - 1\right) \\
p_z &= l_7\left(s_6\left(c_5\left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) + c_6\left(c_2 c_4 + c_3 s_2 s_4\right)\right) - l_2\left(c_2 - 1\right) \\
&\quad - l_8\left(c_6 - 1\right)\left(c_2 c_4 + c_3 s_2 s_4\right) - l_8 s_6\left(c_5\left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) - l_5 c_2\left(c_4 - 1\right) - l_5 c_3 s_2 s_4
\end{aligned}
\tag{E.2}
$$

E.3 Body Jacobian

$$J_b = \begin{bmatrix} j_{11} & j_{12} & j_{13} & j_{14} & j_{15} & j_{16} & j_{17} \\ j_{21} & j_{22} & j_{23} & j_{24} & j_{25} & j_{26} & j_{27} \\ j_{31} & j_{32} & j_{33} & j_{34} & j_{35} & j_{36} & j_{37} \\ j_{41} & j_{42} & j_{43} & j_{44} & j_{45} & j_{46} & j_{47} \\ j_{51} & j_{52} & j_{53} & j_{54} & j_{55} & j_{56} & j_{57} \\ j_{61} & j_{62} & j_{63} & j_{64} & j_{65} & j_{66} & j_{67} \end{bmatrix} \tag{E.3}$$

$$\begin{aligned} j_{11} = &-(c_7(s_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) + c_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3))) \\ &-s_7(s_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) - c_5(c_3s_1 + c_1c_2s_3))) \\ &(l_9(c_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - s_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))) \\ &-l_8(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2)(c_6 - 1) - l_5s_4(c_1s_3 + c_2c_3s_1) \\ &+l_2s_1s_2 + l_8s_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3)) + l_5s_1s_2(c_4 - 1)) \\ &-(c_7(s_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) + c_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))) \\ &-s_7(s_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) - c_5(c_1c_3 - c_2s_1s_3)))(l_5s_4(s_1s_3 - c_1c_2c_3) \\ &-l_9(c_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) - s_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3))) \\ &+l_2c_1s_2 - l_8s_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3)) \\ &+l_8(c_6 - 1)(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) + l_5c_1s_2(c_4 - 1)) \end{aligned}$$

$$\begin{aligned} j_{12} = &(s_7(s_5(s_2s_4 + c_2c_3c_4) + c_2c_5s_3) - c_7(c_6(c_5(s_2s_4 + c_2c_3c_4) - c_2s_3s_5) - s_6(c_4s_2 - c_2c_3s_4))) \\ &(l_2(c_2 - 1) - l_9(s_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) + c_6(c_2c_4 + c_3s_2s_4)) + l_8(c_6 - 1)(c_2c_4 + c_3s_2s_4) \\ &+l_8s_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) + l_5c_2(c_4 - 1) + l_5c_3s_2s_4) \\ &+ l_2(s_7(s_5(s_2s_4 + c_2c_3c_4) + c_2c_5s_3) - c_7(c_6(c_5(s_2s_4 + c_2c_3c_4) - c_2s_3s_5) - s_6(c_4s_2 - c_2c_3s_4))) \\ &-(s_7(s_5(c_2s_4 - c_3c_4s_2) - c_5s_2s_3) - c_7(c_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) - s_6(c_2c_4 + c_3s_2s_4))) \\ &(l_2s_2 - l_9(s_6(c_5(s_2s_4 + c_2c_3c_4) - c_2s_3s_5) + c_6(c_4s_2 - c_2c_3s_4)) \\ &+l_8(c_6 - 1)(c_4s_2 - c_2c_3s_4) + l_8s_6(c_5(s_2s_4 + c_2c_3c_4) - c_2s_3s_5) + l_5s_2(c_4 - 1) - l_5c_2c_3s_4) \end{aligned}$$

$$\begin{aligned} j_{13} = &l_5c_5s_4s_7 - l_8c_5s_4s_7 + l_8c_7s_4s_5 - l_8c_4s_6s_7 - l_9c_7s_4s_5 + l_9c_4s_6s_7 + l_5c_6c_7s_4s_5 \\ &+ l_8c_5c_6s_4s_7 - l_8c_6c_7s_4s_5 - l_9c_5c_6s_4s_7 \end{aligned}$$

$$j_{14} = l_8c_5c_7 - l_9c_5c_7 - l_5s_5s_7 + l_8s_5s_7 + l_5c_5c_6c_7 - l_8c_5c_6c_7 - l_8c_6s_5s_7 + l_9c_6s_5s_7$$

$$j_{15} = -s_6s_7(l_8 - l_9)$$

$$j_{16} = -c_7(l_8 - l_9)$$

$$j_{17} = 0$$

$$
\begin{aligned}
j_{21} = {} & (s_7 (s_6 (s_4 (s_1 s_3 - c_1 c_2 c_3) + c_1 c_4 s_2) + c_6 (c_5 (c_4 (s_1 s_3 - c_1 c_2 c_3) - c_1 s_2 s_4) + s_5 (c_3 s_1 + c_1 c_2 s_3))) \\
& + c_7 (s_5 (c_4 (s_1 s_3 - c_1 c_2 c_3) - c_1 s_2 s_4) - c_5 (c_3 s_1 + c_1 c_2 s_3))) \\
& (l_9 (c_6 (s_4 (c_1 s_3 + c_2 c_3 s_1) - c_4 s_1 s_2) - s_6 (c_5 (c_4 (c_1 s_3 + c_2 c_3 s_1) + s_1 s_2 s_4) + s_5 (c_1 c_3 - c_2 s_1 s_3))) \\
& - l_8 (s_4 (c_1 s_3 + c_2 c_3 s_1) - c_4 s_1 s_2) (c_6 - 1) - l_5 s_4 (c_1 s_3 + c_2 c_3 s_1) + l_2 s_1 s_2 \\
& + l_8 s_6 (c_5 (c_4 (c_1 s_3 + c_2 c_3 s_1) + s_1 s_2 s_4) + s_5 (c_1 c_3 - c_2 s_1 s_3)) + l_5 s_1 s_2 (c_4 - 1)) \\
& + (s_7 (s_6 (s_4 (c_1 s_3 + c_2 c_3 s_1) - c_4 s_1 s_2) + c_6 (c_5 (c_4 (c_1 s_3 + c_2 c_3 s_1) + s_1 s_2 s_4) + s_5 (c_1 c_3 - c_2 s_1 s_3))) \\
& + c_7 (s_5 (c_4 (c_1 s_3 + c_2 c_3 s_1) + s_1 s_2 s_4) - c_5 (c_1 c_3 - c_2 s_1 s_3))) \\
& (l_5 s_4 (s_1 s_3 - c_1 c_2 c_3) - l_9 (c_6 (s_4 (s_1 s_3 - c_1 c_2 c_3) + c_1 c_4 s_2) \\
& - s_6 (c_5 (c_4 (s_1 s_3 - c_1 c_2 c_3) - c_1 s_2 s_4) + s_5 (c_3 s_1 + c_1 c_2 s_3))) \\
& + l_8 (c_6 - 1) (s_4 (s_1 s_3 - c_1 c_2 c_3) + c_1 c_4 s_2) + l_5 c_1 s_2 (c_4 - 1))
\end{aligned}
$$

$$
\begin{aligned}
j_{22} = {} & (c_7 (s_5 (s_2 s_4 + c_2 c_3 c_4) + c_2 c_5 s_3) + s_7 (c_6 (c_5 (s_2 s_4 + c_2 c_3 c_4) - c_2 s_3 s_5) - s_6 (c_4 s_2 - c_2 c_3 s_4))) \\
& (l_2 (c_2 - 1) - l_9 (s_6 (c_5 (c_2 s_4 - c_3 c_4 s_2) + s_2 s_3 s_5) + c_6 (c_2 c_4 + c_3 s_2 s_4)) \\
& + l_8 (c_6 - 1) (c_2 c_4 + c_3 s_2 s_4) + l_8 s_6 (c_5 (c_2 s_4 - c_3 c_4 s_2) + s_2 s_3 s_5) + l_5 c_2 (c_4 - 1) + l_5 c_3 s_2 s_4) \\
& + l_2 (c_7 (s_5 (s_2 s_4 + c_2 c_3 c_4) + c_2 c_5 s_3) + s_7 (c_6 (c_5 (s_2 s_4 + c_2 c_3 c_4) - c_2 s_3 s_5) - s_6 (c_4 s_2 - c_2 c_3 s_4))) \\
& - (c_7 (s_5 (c_2 s_4 - c_3 c_4 s_2) - c_5 s_2 s_3) + s_7 (c_6 (c_5 (c_2 s_4 - c_3 c_4 s_2) + s_2 s_3 s_5) - s_6 (c_2 c_4 + c_3 s_2 s_4))) \\
& (l_2 s_2 - l_9 (s_6 (c_5 (s_2 s_4 + c_2 c_3 c_4) - c_2 s_3 s_5) + c_6 (c_4 s_2 - c_2 c_3 s_4)) + l_8 (c_6 - 1) (c_4 s_2 - c_2 c_3 s_4) \\
& + l_8 s_6 (c_5 (s_2 s_4 + c_2 c_3 c_4) - c_2 s_3 s_5) + l_5 s_2 (c_4 - 1) - l_5 c_2 c_3 s_4)
\end{aligned}
$$

$$
\begin{aligned}
j_{23} = {} & l_5 c_5 c_7 s_4 - l_8 c_5 c_7 s_4 - l_8 c_4 c_7 s_6 + l_9 c_4 c_7 s_6 - l_8 s_4 s_5 s_7 + l_9 s_4 s_5 s_7 \\
& + l_8 c_5 c_6 c_7 s_4 - l_9 c_5 c_6 c_7 s_4 - l_5 c_6 s_4 s_5 s_7 + l_8 c_6 s_4 s_5 s_7
\end{aligned}
$$

$$j_{24} = l_8 c_7 s_5 - l_8 c_5 s_7 - l_5 c_7 s_5 + l_9 c_5 s_7 - l_5 c_5 c_6 s_7 + l_8 c_5 c_6 s_7 - l_8 c_6 c_7 s_5 + l_9 c_6 c_7 s_5$$

$$j_{25} = -c_7 s_6 (l_8 - l_9)$$

$$j_{26} = s_7 (l_8 - l_9)$$

$$j_{27} = 0$$

$$
\begin{aligned}
j_{31} = {} & l_2 c_6 s_2 s_3 s_4 - l_2 c_3 s_2 s_5 s_6 - l_5 c_6 s_2 s_3 s_4 + l_5 c_3 s_2 s_5 s_6 - l_5 c_5 s_2 s_3 s_6 + l_5 c_2 s_4 s_5 s_6 \\
& + l_8 c_5 s_2 s_3 s_6 - l_8 c_2 s_4 s_5 s_6 - l_2 c_4 c_5 s_2 s_3 s_6 - l_5 c_3 c_4 s_2 s_5 s_6 + l_5 c_4 c_5 s_2 s_3 s_6 + l_8 c_3 c_4 s_2 s_5 s_6
\end{aligned}
$$

$$
\begin{aligned}
j_{32} = {} & l_2 c_3 c_6 s_4 - l_5 c_3 c_6 s_4 - l_5 c_3 c_5 s_6 + l_8 c_3 c_5 s_6 + l_2 s_3 s_5 s_6 - l_5 s_3 s_5 s_6 \\
& - l_2 c_3 c_4 c_5 s_6 + l_5 c_3 c_4 c_5 s_6 + l_5 c_4 s_3 s_5 s_6 - l_8 c_4 s_3 s_5 s_6
\end{aligned}
$$

$$j_{33} = s_4 s_5 s_6 (l_5 - l_8)$$

$$j_{34} = c_5 s_6 (l_5 - l_8)$$

$$j_{35} = 0$$

$$j_{36} = 0$$

$$j_{37} = 0$$

$$j_{41} = c_7 (c_6 (c_5 (c_2 s_4 - c_3 c_4 s_2) + s_2 s_3 s_5) - s_6 (c_2 c_4 + c_3 s_2 s_4)) - s_7 (s_5 (c_2 s_4 - c_3 c_4 s_2) - c_5 s_2 s_3)$$

$$j_{42} = c_7 (c_6 (c_3 s_5 + c_4 c_5 s_3) + s_3 s_4 s_6) + s_7 (c_3 c_5 - c_4 s_3 s_5)$$

$$j_{43} = -c_7 \left(c_4 s_6 - c_5 c_6 s_4\right) - s_4 s_5 s_7$$

$$j_{44} = -c_5 s_7 - c_6 c_7 s_5$$

$$j_{45} = -c_7 s_6$$

$$j_{46} = s_7$$

$$j_{47} = 0$$

$$j_{51} = -c_7 \left(s_5 \left(c_2 s_4 - c_3 c_4 s_2\right) - c_5 s_2 s_3\right) - s_7 \left(c_6 \left(c_5 \left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) - s_6 \left(c_2 c_4 + c_3 s_2 s_4\right)\right)$$

$$j_{52} = c_7 \left(c_3 c_5 - c_4 s_3 s_5\right) - s_7 \left(c_6 \left(c_3 s_5 + c_4 c_5 s_3\right) + s_3 s_4 s_6\right)$$

$$j_{53} = s_7 \left(c_4 s_6 - c_5 c_6 s_4\right) - c_7 s_4 s_5$$

$$j_{54} = c_6 s_5 s_7 - c_5 c_7$$

$$j_{55} = s_6 s_7$$

$$j_{56} = c_7$$

$$j_{57} = 0$$

$$j_{61} = s_6 \left(c_5 \left(c_2 s_4 - c_3 c_4 s_2\right) + s_2 s_3 s_5\right) + c_6 \left(c_2 c_4 + c_3 s_2 s_4\right)$$

$$j_{62} = s_6 \left(c_3 s_5 + c_4 c_5 s_3\right) - c_6 s_3 s_4$$

$$j_{63} = c_4 c_6 + c_5 s_4 s_6$$

$$j_{64} = -s_5 s_6$$

$$j_{65} = c_6$$

$$j_{66} = 0$$

$$j_{67} = 1$$