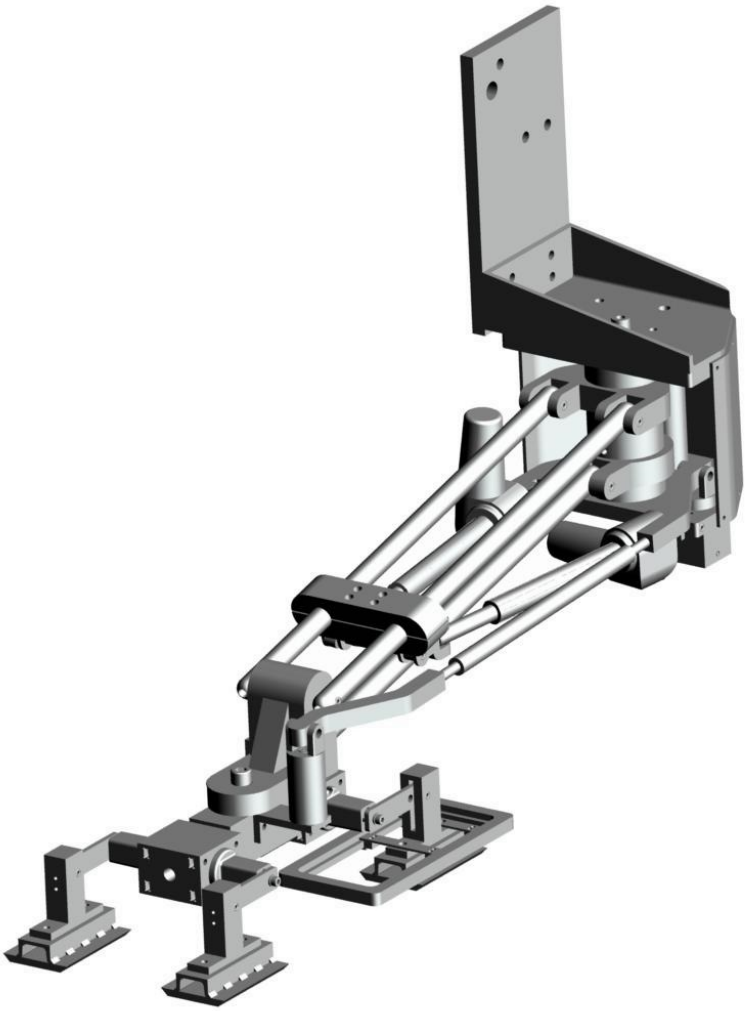


# Modeling and Interaction Control of a Robot Manipulator

Master's Thesis in Systems, Control and Mechatronics

Erik Skoog & David Stenman

Department of Signals and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014

# Modeling and Interaction Control of a Robot Manipulator

*Master's Thesis in Systems, Control and Mechatronics*

Erik Skoog
David Stenman

Department of Signals and Systems
*Division of Automatic Control, Automation and Mechatronics*
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2014

Modeling and Interaction Control of a Robot Manipulator
Erik Skoog, David Stenman



Technical report no: EX064/2014
Department of Signals and Systems
*Division of Automatic Control, Automation and Mechatronics*
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone + 46 (0)31-772 1000

Department of Signals and Systems
Gothenburg, Sweden 2014

# ABSTRACT

This thesis focuses on the modeling and control of a robot arm which is used to transfer energy from electrified tracks in the road to an electric vehicle. The considered robot arm is a highly constrained multi-body mechatronic device with 13 joints but with only two degrees of freedom.

The control objective is to guarantee contact between the robot arm and the electrified tracks whilst maintaining a low contact force. A low contact force is necessary in order to reduce wearing of the the electrified tracks as well as the contact surface of the robot arm, while loss of contact has to be avoided in order to prevent electrical arcs.

A modeling and control framework is presented where a kinematic model is derived and a dynamic model is formulated using the Lagrange method. Control algorithms are proposed, including Impedance control and Force control with inner position loop.

The proposed modeling and control framework has successfully been utilised to model and control the interaction of a robot arm prototype with the electrified tracks. The implemented interaction controllers managed to guarantee constant contact between the robot arm and the electric tracks with a moderate contact force, despite the problem of road unevenness.

This thesis is part of the research project *Slide-in Electric Road System*, the main purpose of which is to study and evaluate the possibilities of using electrified roads for both charging and propulsion of hybrid or fully electrified vehicles during driving.

KEYWORDS: Robot arm, Lagrange mechanics, Exact linearisation, Motion control, Interaction control, Electrified roads.

# ACKNOWLEDGEMENTS

We would like to thank our examiner Paolo Falcone for his guidance during this thesis. We would also like to thank Professor Jonas Sjöberg for taking his time to answer our questions. Last but not least, we would like to thank our friends and families for their support.

Gothenburg, December 2014

Erik Skoog & David Stenman

# Contents

# 1 INTRODUCTION

Vehicle electrification can be regarded as a large step towards a reduction of environmental emissions and dependence of fossil fuel. However, the energy storage devices available today, such as batteries, are expensive and have poor energy density. A large energy storage would also lead to a high energy consumption due to its considerable weight. Therefore, the high amount of energy that is required for propulsion of a heavy vehicle is often not profitable to store within the vehicle. A possible solution is to develop an electrified road system that can continuously transfer energy from the road to the vehicle for propulsion and charging. Electrification of some strategic sections of the road between cities would mean that most of the route could be driven on electricity at low cost and weight of the vehicle. Conductively transfering electricity from the road to the vehicle requires a mechatronic device capable of compensating for the vehicles movement on the road whilst maintaining continuous contact with the electrified tracks.

## 1.1 Background

This thesis is part of a research project, the main purpose of which is to study and evaluate the possibilities of using electrified roads to charge hybrid or fully electrified vehicles.

The research project is led by Volvo GTT and it involves many partners:

- Volvo GTT
- Scania
- Alstom
- Vattenfall
- Swedish Transport Administration
- Projektengagemang (Svenska Elvägar AB)
- Lund University
- KTH Royal Institute of Technology
- Chalmers University of Technology
- Örebro University

The project is partly funded by the Swedish Energy Agency through the program Fordonsstrategisk Forskning och Innovation (FFI) (Viktoria Swedish ICT 2014).

A picture of the robot arm prototype mounted on a truck during testing can be seen in Figure 1.1 below. Note that the two electric tracks are also present in the picture.


Figure 1.1. The robot arm shown in the circle, mounted on truck during testing. The two parallel tracks below the robot arm are the electrified tracks. Image from (Viktoria Swedish ICT 2014).

The robot arm should be mounted under the truck in a similar manner to Figure 1.2

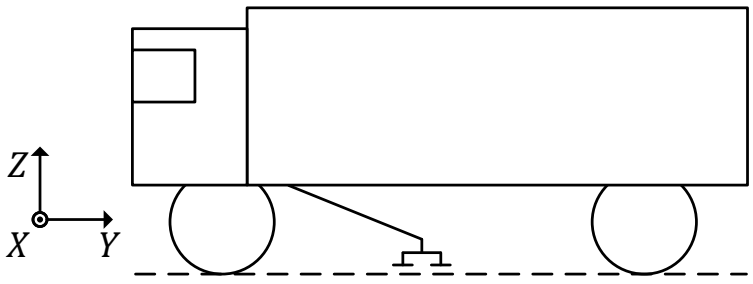



Figure 1.2. Conceptual side view of the robot arm mounted on a truck.

## 1.2 Objective

The purpose of this thesis is to present a modeling and control framework for modeling and control of a mechatronic device, a robot arm, which can be seen in Figure 1.3. The goal is to control the robot arm such that a constant contact is guaranteed between the contact surface of the arm and the electrified tracks whilst maintaining a moderate contact force in order to reduce wearing of the the robot arm and the electrified tracks.

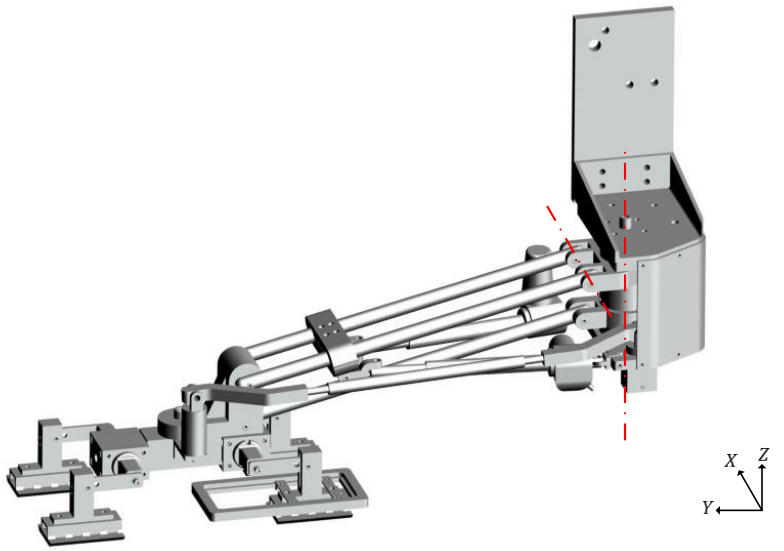



Figure 1.3. CAD model of the robot arm. The dashed lines shows the two axes of rotation representing the two degrees of freedom. The coordinate system shows the orientation of the world frame, which shares orientation with the base frame.

A crucial part of the control problem for the robot arm is the vertical control. Good vertical control ensures that changes in height setpoint are followed quickly whilst maintaining enough contact force to ensure good conductivity. If this is not the case, there is a risk of electrical arcs between the road and the robot arm, which can both be a fire hazard and also disturbing to other road users. The control algorithm currently used by Volvo GTT does not perform as desired, with electric arcs appearing as a result.

## 1.3 Contribution

This thesis contributes to the Slide-in Electric Road System project (Viktoria Swedish ICT 2014) by presenting a systematic modeling and control framework based on the current robot arm prototype. The Lagrange method is applied to formulate a dynamic model of the system. Four different operational space control algorithms is implemented to control the vertical motion of the robot arm. By following the methods described in this thesis, it should be possible to model future robot arm prototypes and implement control algorithms on these with desired performance. Most methods and notations used in this thesis are acquired from the book *Modelling and Control of Robot Manipulators* (Sciavicco and Siciliano 2000).

The mathematical model is formulated using *Wolfram Mathematica* since the mathematical model is too large to deal with by hand. The controllers are implemented and simulated using *MATLAB Simulink*.

## 1.4 Thesis outline

The layout of the report is as follows: Chapter 2 describes the mathematical modeling of the robot arm. The mathematical model consists of expressions describing the kinematics of the robot arm as well as expressions describing the dynamics of the robot which are derived using the Lagrange method. The model of the system is then used in Chapter 3, where four different control strategies for the robot arm are implemented. The results from simulations of the implemented controllers are shown in Chapter 4. Chapter 5 gives the concluding remarks of this thesis.

# 2 MODELING

This chapter covers the formulation of a model describing the kinematic and dynamic properties of the robot arm.

## 2.1 The system

This thesis is based on a robot arm prototype developed by Örebro University (Aldammad *et al.* 2014). The robot arm, as can be seen in Figure 1.3, is a robot manipulator with two degrees of freedom. It rotates around the $Z$-axis of the base frame and a horizontal axis attached to a joint which can be seen in the figure. The motion is controlled by two electrical linear actuators.

The conductive area consists of four contact shoes which can be seen in Figure 2.1. The left and right pair connects to two different electrified tracks respectively. Each shoe is mounted with a spring to ensure good contact. Henceforth, when referring to the end-effector, the contact shoes will be considered.

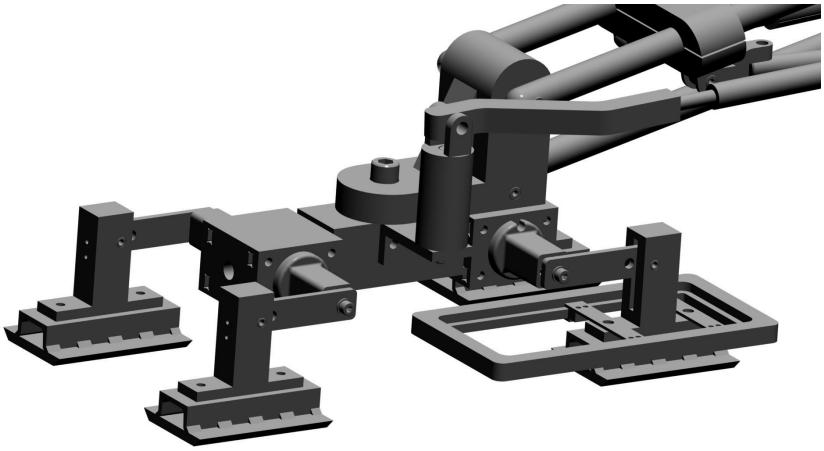
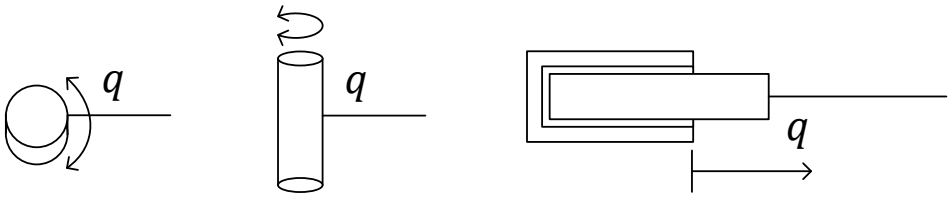


Figure 2.1. Close up of the contact shoes on the robot arm.

The robot arm is constructed in such a way that the orientation of the end-effector with respect to the base is constant. This is achieved by having two different parallelograms in the mechanical structure. One of the parallelograms lies in the vertical plane, spanned by the $Y$- and $Z$-axes of the base frame. The other parallelogram lies in a plane that is perpendicular to the previously mentioned plane, along the length of the robot arm.

The robot arm can be seen as a closed kinematic chain of rigid links, connected by different joints. The two types of joints that are present in the robot arm are revolute and prismatic joints. These can be seen in Figure 2.2.




(a) Revolute joints (b) Prismatic joint

Figure 2.2. The two types of joints used in the system.

Each link, joint and attached coordinate frame is named using a systematic approach which can be seen in Figure 2.3. The frame at each joint is defined with the $Z$-axis as the axis of motion, i.e. an axis perpendicular to the attached link for the revolute joint and an axis parallel with the attached link for prismatic joints.

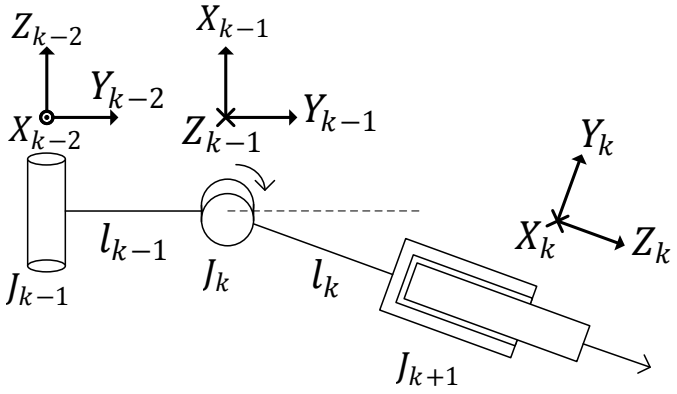



Figure 2.3. Naming of frames, links and joints.

Henceforth, when referring to the $X, Y, Z$-axes without referring to a specific frame, the base frame is always considered.

### 2.1.1 Simplifications

The robot arm has links and joints which, from a mechanical standpoint, give stability and durability but are redundant from a kinematic standpoint, i.e., the links do not add to, or constrict, the range of motion of the robot arm. The system considered in this thesis is a simplified version of a prototype robot arm where several redundant links of the robot arm have been removed.

The robot arm seen in Figure 1.3 can be simplified and abstracted to the schematics seen in Figure 2.4 (Top view) and Figure 2.5 (Side view).

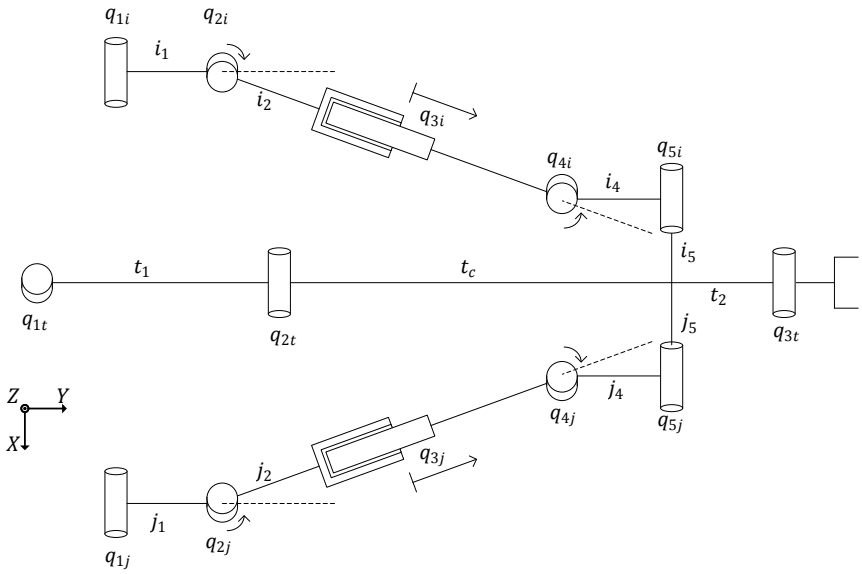



Figure 2.4. Top view of the schematics for the system.

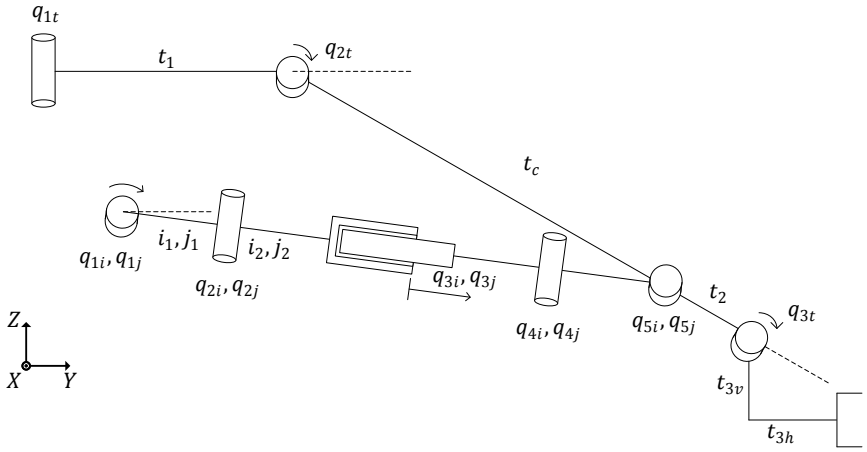



Figure 2.5. Side view of the schematics for the system.

All joints of the abstracted system have been assigned a coordinate frame according to Figure 2.3. This can be seen in Figure 2.6.

The system considered in this thesis has two actuated joints and 11 unactuated joints in total, not taking the end-effector into consideration. This can be seen in Figure 2.4.

The physical system is simplified further to reduce the complexity of the mathematical model. The robot arm can be divided into three different branches which will be called branches $i, j$ and $t$. This is illustrated in Figure 2.7. This denomination

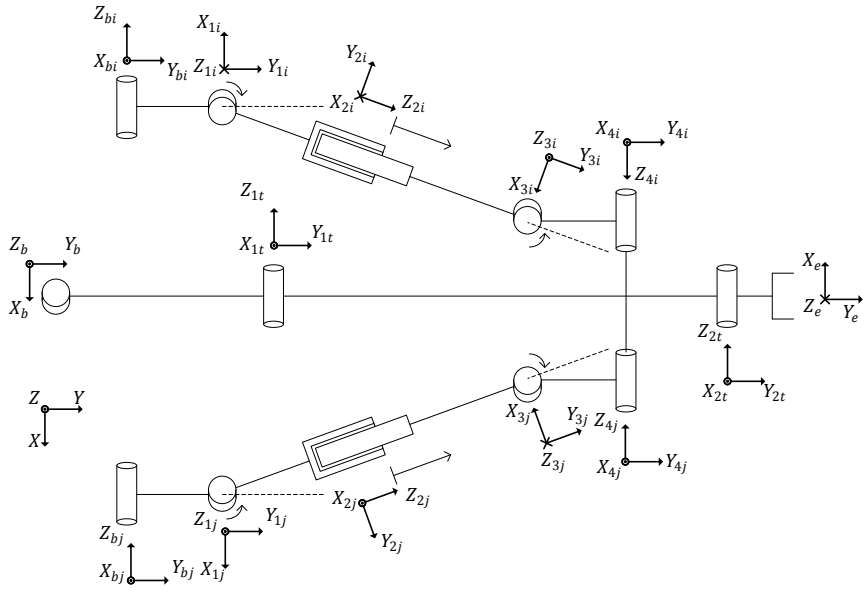



Figure 2.6. Placement and orientation of all the frames in the system.

will be used throughout the thesis for naming variables and links.

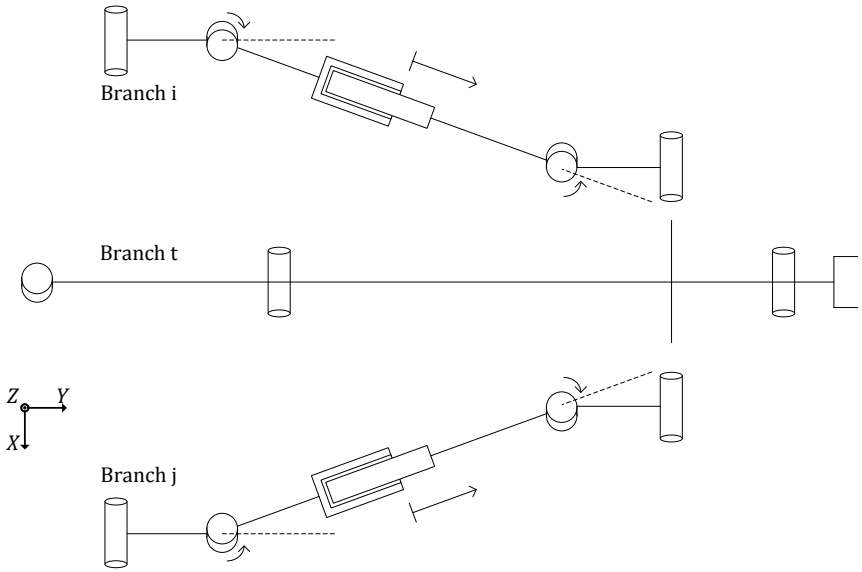



Figure 2.7. The abstracted robot arm divided into branches $i$, $j$ and $t$.

Links $i_1, j_1, i_4, j_4, i_5$ and $j_5$ are considered to be of infinitesimal length and mass. This means that the two joints in each pair of joints $q_{1i}$ and $q_{2i}$, $q_{4i}$ and $q_{5i}$, $q_{1j}$ and $q_{2j}$, $q_{4j}$ and $q_{5j}$, are considered to be located at the same position. This is illustrated in Figure 2.8 (Top view) and in Figure 2.9 (Side view). This implies that links $i_4, j_4, i_5, j_5$ and the corresponding joint variables will neither affect the kinematic nor the dynamic behaviour of branch $i$ and $j$. This property will be

exploited in Section 2.2 and Section 2.3.

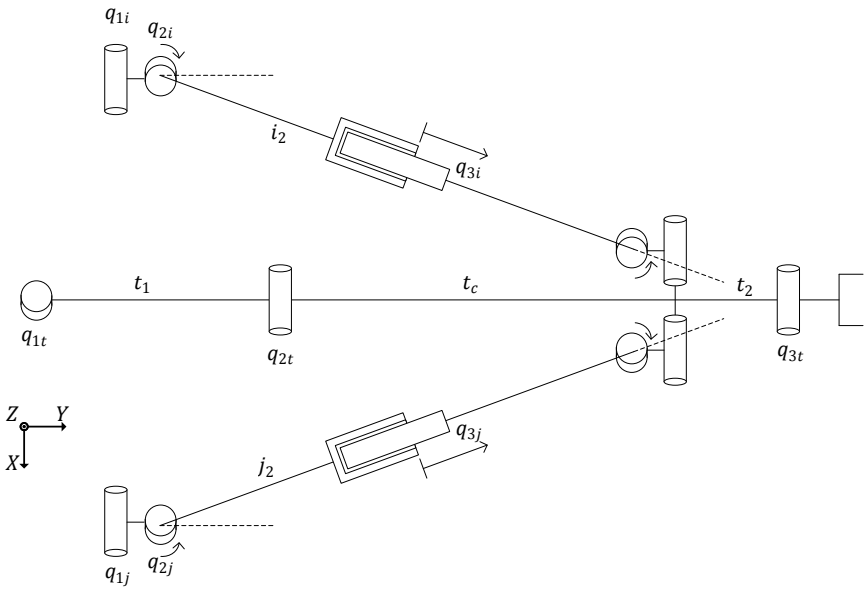



Figure 2.8. Top view of the schematics for the simplified system.

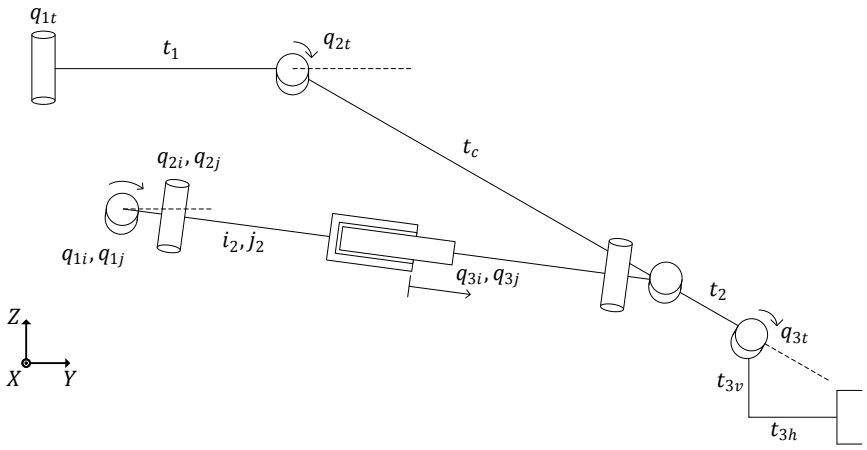



Figure 2.9. Side view of the schematics for the simplified system.

For the considered system, let $\boldsymbol{q}$ denote the vector of all joint variables, both unactuated and actuated. Let $\boldsymbol{q}_u$ denote the vector of all unactuated joint variables and let $\boldsymbol{q}_a$ denote the vector of the actuated joint variables, as presented below.

$$\begin{aligned} \boldsymbol{q} &= \begin{bmatrix} q_{1i} & q_{2i} & q_{3i} & q_{1j} & q_{2j} & q_{3j} & q_{1t} & q_{2t} & q_{3t} \end{bmatrix}^T \\ \boldsymbol{q}_u &= \begin{bmatrix} q_{1i} & q_{2i} & q_{1j} & q_{2j} & q_{1t} & q_{2t} & q_{3t} \end{bmatrix}^T \\ \boldsymbol{q}_a &= \begin{bmatrix} q_{3i} & q_{3j} \end{bmatrix}^T \end{aligned} \tag{2.1}$$

## 2.2 Kinematics

This chapter describes how to find three different relations. First, the relation between the end-effector position and joint variables, the Direct kinematics. Second, the relation between end-effector velocities and joint velocities, the Differential kinematics. The last part of this chapter covers the formulation of constraint equations which describe the relationship between the unactuated joint variables and the actuated joint variables.

The first relation which describe the end-effector as a function of joint variables is

$$\boldsymbol{x} = \boldsymbol{f}_1(\boldsymbol{q}), \tag{2.2}$$

where $\boldsymbol{x}$ is the $6 \times 1$ vector that describes the position and orientation of the end-effector in the following way

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p_e} \\ \boldsymbol{\phi} \end{bmatrix}, \tag{2.3}$$

where $\boldsymbol{p_e}$ is the $3 \times 1$ coordinate vector and where $\boldsymbol{\phi}$ is the $3 \times 1$ vector

$$\boldsymbol{\phi} = \begin{bmatrix} \varphi & \vartheta & \psi \end{bmatrix}^T, \tag{2.4}$$

which describes the orientation of the end-effector with respect to the base frame. $\boldsymbol{p_e}$ and $\boldsymbol{\phi}$ are depicted in Figure 2.11.

The relation between end-effector velocities and joint velocities is described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{2.5}$$

The last relation which describes how the joint variables depend on each other is represented by

$$\boldsymbol{q}_u = \boldsymbol{q}_u(\boldsymbol{q}_a). \tag{2.6}$$

These relationships will be used later on in order to obtain a reduced dynamical model.

### 2.2.1 Direct kinematics

The first step of deriving a kinematic model for a system is to describe the end-effector's position and orientation with respect to the base frame as a function of

the joint variables. This is achieved by first attaching a coordinate frame to every joint in the system as seen in Figure 2.6. The next step is forming a rotation matrix for each pair of consecutive frames $k-1$ and $k$ which describes the rotation between these two frames. By also taking the translation between the two frames into consideration, a transformation matrix can be formed as

$$\boldsymbol{A}_k^{k-1} = \begin{bmatrix} \boldsymbol{R}_k^{k-1} & \boldsymbol{p}_k^{k-1} \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{2.7}$$

where the columns of $\boldsymbol{R}_k^{k-1}(\boldsymbol{q})$ forms the rotation matrix describing frame $k$ with respect to the frame $k-1$ and $\boldsymbol{p}_k^{k-1}(\boldsymbol{q})$ describes the position of frame $k$ with respect to frame $k-1$, i.e., the translation between the frames.

The direct kinematic equations for the robot arm will now be formulated for the selection of links and joints visible in Figure 2.10.

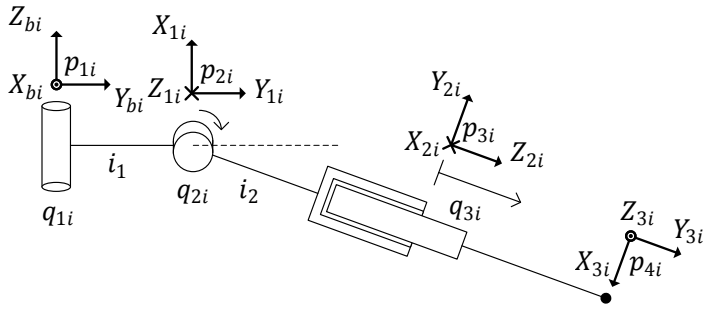



Figure 2.10. A selection of links, joints and their attached frames.

The transformation matrices describing the direct kinematic equations for the frames visible in the figure can be seen below.

$$\boldsymbol{A}_{1i}^{bi} = \begin{bmatrix} 0 & -\sin q_{1i} & -\cos q_{1i} & -i_1 \sin q_{1i} \\ 0 & \cos q_{1i} & -\sin q_{1i} & i_1 \cos q_{1i} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{A}_{2i}^{1i} = \begin{bmatrix} 0 & \cos q_{2i} & -\sin q_{2i} & -i_2 \sin q_{2i} \\ 0 & \sin q_{2i} & \cos q_{2i} & i_2 \cos q_{2i} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\boldsymbol{A}_{3i}^{2i} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & q_{3i} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

These matrices can then be multiplied together as

$$\boldsymbol{A}_{3i}^{bi} = \boldsymbol{A}_{1i}^{bi}\boldsymbol{A}_{2i}^{1i}\boldsymbol{A}_{3i}^{2i} = \begin{bmatrix} \boldsymbol{R}_{3i}^{bi} & \boldsymbol{p}_{3i}^{bi} \\ \boldsymbol{0}^T & 1 \end{bmatrix},$$

where $\boldsymbol{R}_{3i}^{bi}$ describes the orientation of frame $3i$ with respect to the base frame $bi$ and $\boldsymbol{p}_{3i}^{bi}$ describes the position of frame $3i$ with respect to the the base frame $bi$. This leads to

$$\boldsymbol{p}_{3i}^{bi} = \begin{bmatrix} (q_{3i} + i_2)\sin q_{2i} \\ \cos q_{1i}(i_1 + (q_{3i} + i_2)\cos q_{2i}) \\ -\sin q_{1i}(i_1 + (q_{3i} + i_2)\cos q_{2i}) \end{bmatrix}. \tag{2.8}$$

The transformation matrices for the rest of the robot arm are calculated in a similar fashion but are not presented in this thesis due to their cumbersome size. All the kinematic equations are available in the *Wolfram Mathematica* file associated with this thesis (Skoog and Stenman 2014).

By multiplying a set of transformation matrices describing a path from the base frame, $b$, to the frame attached to the end-effector, $e$, as shown in the example, a resulting homogenous transformation matrix is obtained

$$\boldsymbol{T}_e^b = \begin{bmatrix} \boldsymbol{R}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ \boldsymbol{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_e^b(\boldsymbol{q}) & \boldsymbol{Y}_e^b(\boldsymbol{q}) & \boldsymbol{Z}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.9}$$

where $\begin{bmatrix} \boldsymbol{X}_e^b(\boldsymbol{q}) & \boldsymbol{Y}_e^b(\boldsymbol{q}) & \boldsymbol{Z}_e^b(\boldsymbol{q}) \end{bmatrix}$ are the unit vectors and $\boldsymbol{p}_e^b(\boldsymbol{q})$ is the position vector of the frame attached to end-effector with respect to the base frame. This is displayed in Figure 2.11. More information regarding the general case can be found in Appendix A.1.4.

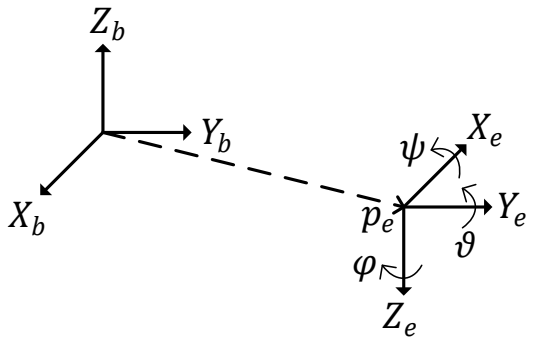



Figure 2.11. Position and orientation of the frame attached to the end-effector.

For the case of the robot arm, transformation matrices are derived for all frames within the system. According to (A.12), the number of consecutive multiplications required to formulate the end-effector transformation matrix is proportional to the number of joints in the utilised chain of links. Therefore, since the $t$ branch contains the least amount of joints, the $t$ branch is selected to describe the path between the base frame and the frame attached to the end-effector (see Figure 2.8).

### 2.2.2 Differential kinematics

The differential kinematic equations describe the mapping between joint velocities and linear and angular velocities of the frames attached to end-effector, links and actuators. This relationship can be described by two different methods; by formulation of both the *Geometric Jacobian* and the *Analytical Jacobian*. These two methods are effectively describing the same relationship but in two different ways. Both of these relations will be utilised when implementing control algorithms later on during this thesis.

The first method is based on deriving a configuration dependant matrix called the Geometric Jacobian, $\boldsymbol{J}$, which is defined as

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_O(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.10}$$

where $\boldsymbol{J}_P$ corresponds to the mapping between joint velocities and linear velocity of point $\boldsymbol{p}$ (see (A.23)) and $\boldsymbol{J}_O$ corresponds to the mapping between joint velocities and the angular velocity of point $\boldsymbol{p}$, which can be referred to as the three-dimensional vector $\boldsymbol{\omega}$ (see (A.24)). The derivation of the Geomatrical Jacobian is discussed further in Appendix A.1.5. For a constrained system, the velocity of the end-effector will correspond to

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_O(\boldsymbol{q}) \end{bmatrix} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{q}_a} \dot{\boldsymbol{q}}_a = \boldsymbol{J}(\boldsymbol{q}_a)\dot{\boldsymbol{q}}_a. \tag{2.11}$$

The Geometric Jacobian for the robot arm will now be formulated for the selection of links and joints visible in Figure 2.10. The Geometric Jacobian can be used to find the velocity of the end-point $\boldsymbol{p}_{3i}$ and is described by

$$\boldsymbol{J}^{(3i)} = \begin{bmatrix} \boldsymbol{z}_{bi} \times (\boldsymbol{p}_{3i} - \boldsymbol{p}_{bi}) & \boldsymbol{z}_{1i} \times (\boldsymbol{p}_{3i} - \boldsymbol{p}_{1i}) & \boldsymbol{z}_{2i} \\ \boldsymbol{z}_{bi} & \boldsymbol{z}_{1i} & \boldsymbol{0} \end{bmatrix},$$

which is

$$\boldsymbol{J}^{(3i)} = \begin{bmatrix} 0 & (q_{3i}+i_2)\cos q_{2i} & \sin q_{2i} \\ -\sin q_{1i}(i_1+(q_{3i}+i_2)\cos q_{2i}) & -\cos q_{1i}(q_{3i}+i_2)\sin q_{2i} & \cos q_{1i}\cos q_{2i} \\ -\cos q_{1i}(i_1+(q_{3i}+i_2)\cos q_{2i}) & \sin q_{1i}(q_{3i}+i_2)\sin q_{2i} & -\sin q_{1i}\cos q_{2i} \\ -1 & 0 & 0 \\ 0 & -\cos q_{1i} & 0 \\ 0 & -\sin q_{1i} & 0 \end{bmatrix}.$$

The second method of describing this mapping is by using a so-called Analytical Jacobian, $\boldsymbol{J}_A$. If the orientation of the end-effector can be described by a minimal representation, as explained in Appendix A.1.4, the Analytical Jacobian can be obtained by differentiation of the direct kinematics function. It can be used in the design of operational space control schemes and in describing the relation between forces applied to the end-effector and the resulting torques at the joints. The Analytical Jacobian is expressed by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_\phi(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{2.12}$$

where $\boldsymbol{J}_P$ corresponds to the mapping between the joint velocities and linear velocity of point $\boldsymbol{p}$ and $\boldsymbol{J}_\phi$ corresponds to the mapping between the joint velocities and angular velocity of point $\boldsymbol{p}$. $\boldsymbol{J}_P$ and $\boldsymbol{J}_\phi$ can be formulated as

$$\boldsymbol{J}_P(\boldsymbol{q}) = \frac{\partial \boldsymbol{p}}{\partial \boldsymbol{q}} \tag{2.13}$$

$$\boldsymbol{J}_\phi(\boldsymbol{q}) = \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{q}} \tag{2.14}$$

where $\boldsymbol{\phi}$ is the vector of *Euler Angels* which describes the orientation of the frame attached to the point $\boldsymbol{p}$. This is described further in A.1.4. By time differentiation of (2.12), the acceleration of point $\boldsymbol{x}$ can be expressed by

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}, \tag{2.15}$$

where

$$\dot{\boldsymbol{J}}_A(\boldsymbol{q},\dot{\boldsymbol{q}}) = \frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{q}^2}\dot{\boldsymbol{q}}. \tag{2.16}$$

For the case of a constrained system, the Analytical Jacobian is expressed by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{\phi}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_\phi(\boldsymbol{q}) \end{bmatrix} \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{q}_a} \dot{\boldsymbol{q}}_a = \boldsymbol{J}_A(\boldsymbol{q}_a)\dot{\boldsymbol{q}}_a, \tag{2.17}$$

and (2.15) becomes

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}_A(\boldsymbol{q}_a)\ddot{\boldsymbol{q}}_a + \dot{\boldsymbol{J}}_A(\boldsymbol{q}_a, \dot{\boldsymbol{q}}_a)\dot{\boldsymbol{q}}_a, \tag{2.18}$$

where

$$\dot{\boldsymbol{J}}_A(\boldsymbol{q}_a, \dot{\boldsymbol{q}}_a) = \left( \frac{\partial^2 \boldsymbol{x}}{\partial \boldsymbol{q}^2} \left( \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{q}_a} \right)^2 + \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{q}} \frac{\partial^2 \boldsymbol{q}}{\partial \boldsymbol{q}_a^2} \right) \dot{\boldsymbol{q}}_a. \tag{2.19}$$

The Analytical Jacobian for the robot arm will now be formulated for the selection of links and joints visible in Figure 2.10. The Analytical Jacobian can be used to find the velocity of the end-point $\boldsymbol{p}_{3i}$ and is described by

$$\boldsymbol{J}_P^{(3i)} = \begin{bmatrix} \frac{\partial \boldsymbol{p}_{3i}^{bi}}{\partial q_{1i}} & \frac{\partial \boldsymbol{p}_{3i}^{bi}}{\partial q_{2i}} & \frac{\partial \boldsymbol{p}_{3i}^{bi}}{\partial q_{3i}} \end{bmatrix},$$

where $\boldsymbol{p}_{3i}^{bi}$ is described by (2.8). Hence,

$$\boldsymbol{J}_P^{(3i)} = \begin{bmatrix} 0 & (q_{3i} + i_2)\cos q_{2i} & \sin q_{2i} \\ -\sin q_{1i}(i_1 + (q_{3i} + i_2)\cos q_{2i}) & -\cos q_{1i}(q_{3i} + i_2)\sin q_{2i} & \cos q_{1i}\cos q_{2i} \\ -\cos q_{1i}(i_1 + (q_{3i} + i_2)\cos q_{2i}) & \sin q_{1i}(q_{3i} + i_2)\sin q_{2i} & -\sin q_{1i}\cos q_{2i} \end{bmatrix}.$$

$\boldsymbol{J}_\phi$ is derived by expressing the previously obtained rotation matrix $\boldsymbol{R}_{3i}^{bi}$ as Euler Angles (which is explained in A.1.4) by applying (A.16)-(A.19). This will give the vector of Euler Angels, $\boldsymbol{\phi}$, which is differentiated with respect to the joint variables to obtain $\boldsymbol{J}_\phi$.

Due to the design of the robot arm, with the two parallelograms previously mentioned in Subsection 2.1.1, the end-effector angular velocity around the $X$- and $Z$-axes is equal to zero (see Figure 1.3). Since there are no revolute joints to enable rotation around the $Y$-axis of the robot arm, this leads to the angular velocity around the $Y$-axis being zero. Therefore,

$$\begin{aligned} \boldsymbol{\omega} &= \boldsymbol{0} \\ \dot{\boldsymbol{\phi}} &= \boldsymbol{0}. \end{aligned} \tag{2.20}$$

The fact that there are no rotation around the $X$,$Y$,$Z$-axes of the robot arm leads to the lower three rows of (2.11) and (2.17) are equal to zero.

There exists a relationship between the Geometric Jacobian, $\boldsymbol{J}$, and the Analytical Jacobian, $\boldsymbol{J}_A$, which can be described by

$$\boldsymbol{J} = \boldsymbol{T}_A(\boldsymbol{\phi})\boldsymbol{J}_A, \tag{2.21}$$

where

$$\boldsymbol{T}_A(\boldsymbol{\phi}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}(\boldsymbol{\phi}) \end{bmatrix}. \tag{2.22}$$

For the case of this thesis, since the the lower three rows of $\boldsymbol{J}$ and $\boldsymbol{J}_A$ are zero, (2.21) is equal to

$$\boldsymbol{J} = \boldsymbol{J}_A, \tag{2.23}$$

regardless of the value of $\boldsymbol{T}(\boldsymbol{\phi})$.

There are three ways to find the velocities of the end-effector since the expressions can be derived by considering the structure and motion of any of the three branches $i, j$ or $t$. As explained in the last paragraph of Subsection 2.2.1, the Geometric Jacobian and the Analytical Jacobian for end-effector velocity are derived with respect to the $t$ branch.

### 2.2.3 Constraints

Since the motion of the robot arm is constrained, mathematical expressions have to be formulated which describes the motion of the unactuated joints as functions of the actuated joints.

Each joint variable within a system generates one degree of motion. If the number of degrees of freedom equals the number of degrees of motion, i.e., joint variables, the system is unconstrained. If there are more degrees of motion than degrees of freedom in a system, there exists a number of constraints that may be used to eliminate superfluous joint variables. Suppose that the number of degrees of motion is $M$ and that the number of degrees of freedom is $N$ and that $M > N$, there must exist $M - N$ constraints (Boström 2013). These may be on the form

$$f_k(q_1, \ldots, q_M, t) = 0, \quad k = 1, \ldots, M - N \tag{2.24}$$

which are called *holonomic constraints* since they are a function of joint variables and time.

The design of the robot arm allows rotation of the robot arm around the two axes presented in Figure 1.3. The rotation around the vertical axis is described by the joint variable $q_{1t}$ and the rotation around the lateral axis is described by the joint variable $q_{2t}$ (see Figure 2.8). Since the robot arm has 13 joints, and therefore 13 degrees of motion, and one degree of freedom since only the vertical motion is considered, there must exist $13-1 = 12$ constraints. However, due to the simplifications and assumptions that are made (see Subsection 2.1.1), the number of joint variables that affect the system are reduced by 4. Therefore, the number of constraints are reduced to 8.

There are two methods to obtain constraint equations for a system. One way is to determine the constraints in the operational space, i.e., describing points and axes in space. The second method is to find trigonometric relations by examining the joint space, i.e., analysing the geometry of the robot arm. The latter method provides constraints that may be easier to solve than operational space constraints. Therefore, joint space constraints are used in this thesis.

### Operational space constraints

Consider the position and orientation (the coordinate frames) of two arbitrary points in space. A position constraint can be formed if a relation between the two points can be formulated, e.g., as a constant distance between these two points. An orientation constraint can be formed if a relation between the coordinate frames of the two points can be formed, e.g., if one or more axes of the two frames are parallel with respect to each other.

By deriving direct kinematic expressions for the system depicted in Figure 2.6, operational space constraints can be formulated. The transformation matrix $\boldsymbol{T}_e^b$ can be formulated by considering the structure and motion of any of the three branches $i, j$ or $t$. Since all three transformation matrices describe the position and orientation of the end-effector, they are equivalent. Therefore, equality constraints between these transformation matrices can be imposed.

### Joint space constraints

By projecting the links and joints on two different planes, (one plane being where the two prismatic joints reside and the other being the $Y-Z$-plane), trigonometrical equations can be formed and used to calculate expressions for some joint variables. This can be seen in Figure 2.12 and Figure 2.13.

Some constraints can be considered as trivial when examining the the mechanical structure of the robot arm in the two-dimensional case,

$$q_{1i} = q_{1j}, \quad q_{5i} = q_{5j},$$

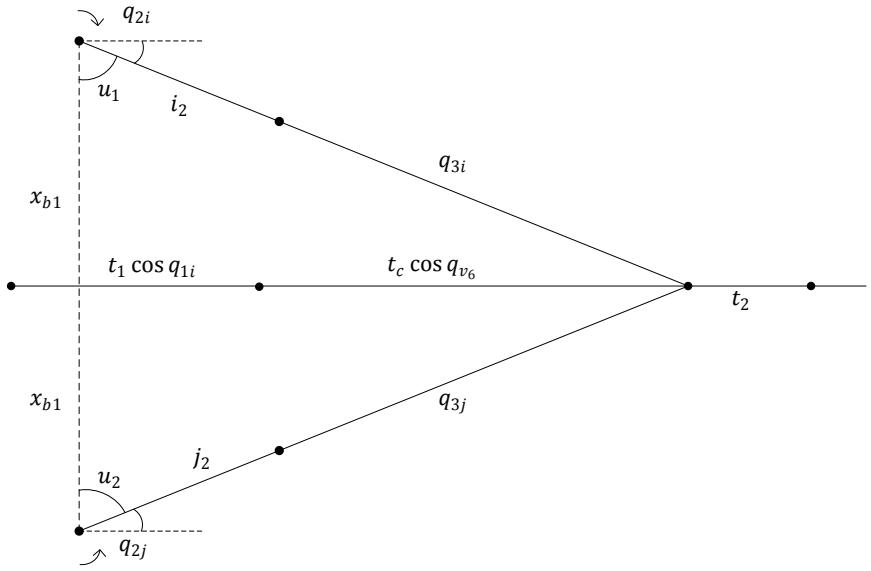



Figure 2.12. Top view for calculations of constraints. Projected in the plane of the two pistons.

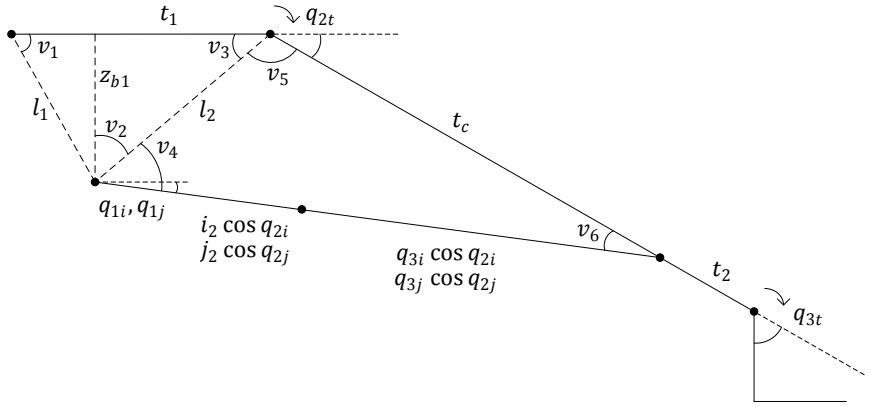



Figure 2.13. Side view for calculations of constraints. Projected in the vertical plane along the $Y - Z$ axes.

seeing that there are no revolute joints that rotate about the longitudinal $Y$-axis.

Some auxiliary equations are introduced in order to derive the constraints. These, together with the constraint equations, are presented below.

$$u_1 = \arccos\left(\frac{q_{3i}^2 + (2x_{b1})^2 - q_{3j}^2}{4q_{3i}x_{b1}}\right)$$
$$u_2 = \arccos\left(\frac{q_{3j}^2 + (2x_{b1})^2 - q_{3i}^2}{4q_{3j}x_{b1}}\right)$$

$$q_{2i} = \frac{\pi}{2} - u_1 \tag{2.25}$$

$$q_{2j} = \frac{\pi}{2} - u_2 \tag{2.26}$$

$$l_2 = \sqrt{l_1^2 + t_1^2 - 2l_1 t_1 \cos v_1}$$

$$v_1 = \arcsin \frac{z_{b1}}{l_1}$$

$$v_2 = \arctan \left( \frac{t_1 - \sqrt{l_1^2 - z_{b1}^2}}{z_{b1}} \right)$$

$$v_3 = \frac{\pi}{2} - v_2$$

$$v_4 = \arccos \left( \frac{(l_2^2 + q_{3j} \cos q_{2j})^2 - t_c^2}{2 l_2 q_{3j} \cos q_{2j}} \right)$$

$$v_5 = \arccos \left( \frac{(l_2^2 + t_c)^2 - (q_{3j} \cos q_{2j})^2}{2 l_2 t_c} \right)$$

$$v_6 = \pi - v_4 - v_5$$

$$q_{1i} = v_2 + v_4 - \frac{\pi}{2} \tag{2.27}$$

$$q_{1j} = v_2 + v_4 - \frac{\pi}{2} \tag{2.28}$$

$$q_{2t} = \pi - v_3 - v_5 \tag{2.29}$$

Two links have been removed that are vital for the kinematics of the robot arm. Both of these links are part of the two parallelograms respectively, but seeing that the thesis only focuses on the vertical motion, one of the parallelograms is ignored. One of the parallelogram links can be removed by adding a very simple constraint to ensure that the system behaves in the same way. This constraint is simply

$$q_{3t} = \frac{\pi}{2} - q_{2t}. \tag{2.30}$$

This will ensure that the $Z$-axis of the frame attached to the end-effector is parallel to the $Z$-axis of the base frame, thus the properties of the parallelogram are fulfilled.

As mentioned in Subsection 2.1.1, the sideways motion about the $Z$-axis of the robot arm is assumed to be zero. Therefore, the displacement of the two actuators are equal. Thus,

$$q_{3i} = q_{3j} \tag{2.31}$$

which leads to

$$\boldsymbol{q}_a = \begin{bmatrix} q_{3i} \end{bmatrix}.$$

Henceforth the actuated joint variable $q_{3i}$ will be referred to as $\boldsymbol{q}_a$, even though the dimension of $\boldsymbol{q}_a$ is one.

By solving the constraint equations presented above, for a reduced set of joint variables, an expression is obtained which describes the unactuated joint variables as a function of the actuated joint variables. Thus, the desired relation shown in (2.6).

Recall the relations expressed by (2.2) and (2.5). The direct kinematic relation is given by (2.9) and the differential kinematic relation is described by (2.10) and (2.12).

## 2.3 Dynamics

The objective of this section is to describe the dynamic properties of the robot arm. This relation may be on the form

$$\ddot{\boldsymbol{q}} = \boldsymbol{f}_3(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{u}, \boldsymbol{h}), \tag{2.32}$$

which describes the acceleration of the joint variables. $\boldsymbol{u}$ is the actuator force and $\boldsymbol{h}$ describes the forces and moments affecting the robot arm at the end-effector.

There are two common methods for derivation of a dynamical model of a robot arm; Newton-Euler formulation and Lagrange formulation. The Newton-Euler formulation employs the forces and moments acting on all the links of a robot arm to model the dynamics of the system. Since the number of calculations required to formulate a dynamical model is proportional to the number of links, the complexity rapidly rises with an increasing number of links within a system.

The second method is the Lagrange formulation, which is the method used in this thesis. The Lagrange method is a systematic approach that relies on the derivation of potential and kinetic energies for a system to formulate a dynamic model.

### 2.3.1 Lagrange formulation

The Lagrangian is formulated as

$$\mathcal{L} = \mathcal{T} - \mathcal{U}, \tag{2.33}$$

where $\mathcal{T}$ is the total kinetic energy and $\mathcal{U}$ is the total potential energy of a system that consists of $n$ rigid links.

A dynamic model on the same form as (2.32) can be derived by formulating the Lagrange's equations according to

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_k} - \frac{\partial \mathcal{L}}{\partial q_k} = \xi_k, \tag{2.34}$$

where $\xi_k$ is the generalised force that is associated with joint variable $q_k$.

The kinetic energy of the system is expressed as

$$\mathcal{T} = \sum_{k=1}^{n} \left( \mathcal{T}_{l_k} + \mathcal{T}_{m_k} \right), \tag{2.35}$$

where $\mathcal{T}_{l_k}$ corresponds to the kinetic energy of link $k$ and $\mathcal{T}_{m_k}$ corresponds to the kinetic energy of the actuator related to link $k$.

The potential energy of the system is expressed as

$$\mathcal{U} = \sum_{k=1}^{n} \left( \mathcal{U}_{l_k} + \mathcal{U}_{m_k} \right), \tag{2.36}$$

where $\mathcal{U}_{l_k}$ corresponds to the kinetic energy of link $k$ and $\mathcal{U}_{m_k}$ corresponds to the potential energy of the actuator related to link $k$.

The kinetic energy of the links for each branch corresponds to

$$\mathcal{T}_l = \frac{1}{2} \sum_{k=1}^{n} \left( m_{l_k} \dot{\boldsymbol{p}}_{l_k}^T \dot{\boldsymbol{p}}_{l_k} + \boldsymbol{\omega}_k^T \boldsymbol{R}_k \boldsymbol{I}_{l_k}^k \boldsymbol{R}_k^T \boldsymbol{\omega}_k \right), \tag{2.37}$$

where $\boldsymbol{p}_{l_k}$ describes the center of mass of link $k$ and $m_{l_k}$ describes the mass of the same link. The kinetic energy of the actuators for each branch corresponds to

$$\mathcal{T}_m = \frac{1}{2} \sum_{k=1}^{n} \left( m_{m_k} \dot{\boldsymbol{p}}_{m_k}^T \dot{\boldsymbol{p}}_{m_k} + \boldsymbol{\omega}_k^T \boldsymbol{R}_k \boldsymbol{I}_{m_k}^i \boldsymbol{R}_k^T \boldsymbol{\omega}_k \right), \tag{2.38}$$

where $\boldsymbol{p}_{m_k}$ describes the center of mass of joint $k$ and $m_{m_k}$ describes the mass of the same joint. Inserting (2.10) into (2.37) and (2.38) forms

$$\mathcal{T}_l = \frac{1}{2} \sum_{k=1}^{n} \left( m_{l_k} \dot{\boldsymbol{q}}^T \boldsymbol{J}_P^{(l_k)T} \boldsymbol{J}_P^{(l_k)} \dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T \boldsymbol{J}_O^{(l_k)T} \boldsymbol{R}_k \boldsymbol{I}_{l_k}^k \boldsymbol{R}_k^T \boldsymbol{J}_O^{(l_k)} \dot{\boldsymbol{q}} \right) \tag{2.39}$$

and

$$\mathcal{T}_m = \frac{1}{2}\sum_{k=1}^{n}\left(m_{m_k}\dot{\boldsymbol{q}}^T\boldsymbol{J}_P^{(m_k)T}\boldsymbol{J}_P^{(m_k)}\dot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^T\boldsymbol{J}_O^{(m_k)T}\boldsymbol{R}_k\boldsymbol{I}_{m_k}^k\boldsymbol{R}_k^T\boldsymbol{J}_O^{(m_k)}\dot{\boldsymbol{q}}\right). \tag{2.40}$$

The potential energy of the links and the actuators in the system corresponds to

$$\mathcal{U} = -\sum_{k=1}^{n}(m_{l_k}\boldsymbol{g}_0^T\boldsymbol{p}_{l_k} + m_{m_k}\boldsymbol{g}_0^T\boldsymbol{p}_{m_k}) \tag{2.41}$$

where $\boldsymbol{g}_0$ is the $(3 \times 1)$ vector describing the gravitational acceleration in three dimensions.

For the case of the robot arm, (2.39)-(2.41) are utilised to obtain the total kinetic and potential energy of all the links and joints within the system. As specified in Section 2.1, the number of links in all three branches of the abstracted model is equal to three, which implies that $k \in \{1, 2, 3\}$ regarding the sums associated with the energies of the links of each branch $i, j$ and $t$. Since there are only two actuators, one in each of the side branches and both attached to link 3, $k = 3$ in the sums that are associated with the energies of the actuators in branch $i$ and $j$. Thus

$$\{\mathcal{T}_{m_k}, \mathcal{U}_{m_k}\} = 0, \quad k \in \{1, 2\}.$$

Formulating the Lagrangian described by (2.33) will render a Lagrangian which is a function of both unactuated and actuated joints, $\boldsymbol{q}_u$ and $\boldsymbol{q}_a$. However, since the system is constrained, the Lagrangian must depend on actuated joints only. Consider the constraint equations derived in Subsection 2.2.3. By applying

$$\boldsymbol{q}_u = \boldsymbol{q}_u(\boldsymbol{q}_a),$$

and its derivative

$$\dot{\boldsymbol{q}}_u = \frac{\partial \boldsymbol{q}_u}{\partial \boldsymbol{q}_a}\frac{d\boldsymbol{q}_a}{dt}, \tag{2.42}$$

the Lagrangian can now be expressed as a function of the actuated joint variables according to

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{L}(\boldsymbol{q}_a, \dot{\boldsymbol{q}}_a). \tag{2.43}$$

According to (2.34), the Lagrange's equations can be calculated as

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}_a} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}_a} = \boldsymbol{\xi}. \tag{2.44}$$

(2.43) together with (2.44) gives the final Lagrange equation

$$\frac{\partial^2 \mathcal{L}}{\partial \dot{\boldsymbol{q}}_a^2}\ddot{\boldsymbol{q}}_a + \frac{\partial}{\partial \boldsymbol{q}_a}\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{q}}_a}\dot{\boldsymbol{q}}_a - \frac{\partial \mathcal{L}}{\partial \boldsymbol{q}_a} = \boldsymbol{\xi}. \tag{2.45}$$

By identifying terms, (2.45) can be rewritten as the second order differential equation

$$\boldsymbol{B}(\boldsymbol{q}_a)\ddot{\boldsymbol{q}}_a + \boldsymbol{C}(\boldsymbol{q}_a, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_a + \boldsymbol{g}(\boldsymbol{q}_a) = \boldsymbol{\xi}, \tag{2.46}$$

where $\boldsymbol{B}$ is a $(N \times N)$ *intertia matrix* where $N$ are the degrees of freedom of the system. The index $b_{ij}$ describes the effect of acceleration of joint $i$ on joint $j$. The inertia matrix is a symmetric, positive definite matrix, whose indices are configuration dependent. $\boldsymbol{C}$ can be interpreted as the dampening term in the model. $\boldsymbol{g}$ corresponds to the moment generated by gravitational pull at each link and actuator.

The generalised force can be expressed as

$$\boldsymbol{\xi} = \boldsymbol{u} - \boldsymbol{F}\dot{\boldsymbol{q}}_a - \boldsymbol{J}^T(\boldsymbol{q}_a)\boldsymbol{h}, \tag{2.47}$$

where $\boldsymbol{u}$ is the vector of input forces applied by the actuators and $\boldsymbol{F}$ is a diagonal matrix that contains friction coefficients for each joint. $\boldsymbol{h}$, which will be discussed in Chapter 3, denotes the vector of external forces and moments exerted on the end-effector. $\boldsymbol{J}$ (see (2.11)) represents the Geometric Jacobian for end-effector velocity. This gives

$$\boldsymbol{B}(\boldsymbol{q}_a)\ddot{\boldsymbol{q}}_a + \boldsymbol{C}(\boldsymbol{q}_a, \dot{\boldsymbol{q}}_a)\dot{\boldsymbol{q}}_a + \boldsymbol{g}(\boldsymbol{q}_a) = \boldsymbol{u} - \boldsymbol{F}\dot{\boldsymbol{q}}_a - \boldsymbol{J}^T(\boldsymbol{q}_a)\boldsymbol{h}, \tag{2.48}$$

which will have a number of rows equal to the number of degrees of freedom, $N$. In the case of this thesis, the degree of freedom is one (see Subsection 2.1.1). (2.48) describes the desired dynamic relation sought by (2.32). Due to the sheer size of the expression described by (2.48), it will not be presented here but it is available in the *Wolfram Mathematica* file associated with this thesis (Skoog and Stenman 2014).

# 3 CONTROL

This chapter presents the motion and interaction control of the robot arm. Four different control algorithms in the operational space are presented and implemented.

The first two controllers, *PD Control with Gravity Compensation* and *Inverse Dynamics Control* are both motion control schemes that do not handle interaction with the environment. The other two control algorithms, *Impedance Control* and *Force Control with Inner Position Loop*, are both Interaction controllers, i.e., they handle interaction between the robot arm and the environment.

As mentioned in Subsection 2.2.2, the orientation of the end-effector is assumed to be the same as, and invariant with respect to, the base frame. Therefore, the focus of the four implemented controllers is controlling the position of the end-effector and not its orientation. Hence, there will be no moment exerted on the end-effector by the environment.

The position of the end-effector, the environment and the desired position of the end-effector are given by the $3 \times 1$ vectors $\boldsymbol{x}$, $\boldsymbol{x}_e$ and $\boldsymbol{x}_d$ respectively. The $Z$-coordinates of these vectors are explained in Figure 3.1.

Henceforth, the notation for actuated joint variables $\boldsymbol{q}_a$ is changed. In lieu of (2.1), when referring to the vector $\boldsymbol{q}$, the actuated joints is considered.

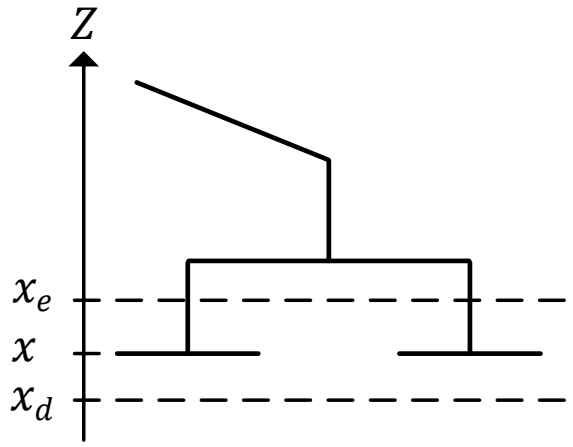



Figure 3.1. Demonstration of the different notations for position. $\boldsymbol{x}_e$ describes the position of the environment, $\boldsymbol{x}$ the end-effector position and $\boldsymbol{x}_d$ describes the desired position of the end-effector.

## 3.1 PD Control with Gravity Compensation

The PD controller with gravity compensation controls the position, $\boldsymbol{x}$, of the robot arm. By the final value theorem, a steady state error will be present since the considered controller does not include any integral action. However, by compensating for the effect of gravity on the links, the robot arm will reach a constant reference position without error.

Given a desired constant end effector position $\boldsymbol{x}_d$, and that the end effector contact force $\boldsymbol{h}$ is equal to zero, the task is to find a suitable control algorithm that will lead the operational space position error

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}(\boldsymbol{q}) \tag{3.1}$$

towards zero. Consider now (2.48) with $\boldsymbol{h} = \boldsymbol{0}$, rewritten as

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \left(\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F}\right)\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u}. \tag{3.2}$$

By proposing a Lyapunov function candidate (Sciavicco and Siciliano 2000, p. 252)

$$V(\dot{\boldsymbol{q}}, \tilde{\boldsymbol{x}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{B}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\tilde{\boldsymbol{x}}^T\boldsymbol{K}_P\tilde{\boldsymbol{x}} > 0 \quad \forall \dot{\boldsymbol{q}}, \tilde{\boldsymbol{x}} \neq \boldsymbol{0}, \tag{3.3}$$

it can be shown that the following control law

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} - \boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{3.4}$$

with positive definite diagonal matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$, will lead to

$$\dot{V} = -\dot{\boldsymbol{q}}^T\boldsymbol{F}\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}^T\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_D\boldsymbol{J}_A(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{3.5}$$

which will be less than zero for any $\dot{\boldsymbol{q}} \neq \boldsymbol{0}$. Thus, the Lyapunov function will decrease as long as $\dot{\boldsymbol{q}} \neq \boldsymbol{0}$. Hence, the system will reach the following equilibrium position

$$\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{0}. \tag{3.6}$$

In the case of full rank Analytical Jacobian, $\boldsymbol{J}_A$,

$$\tilde{\boldsymbol{x}} = \boldsymbol{x}_d - \boldsymbol{x}(\boldsymbol{q}) = \boldsymbol{0}, \tag{3.7}$$

i.e., the desired position is achieved. However, in the case of $\boldsymbol{h} \neq \boldsymbol{0}$, in lieu of (3.6),

$$\boldsymbol{J}_A^T(\boldsymbol{q})\boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}. \tag{3.8}$$

In the case of full rank Analytical Jacobian, $\boldsymbol{J}_A$, the steady state error will correspond to

$$\tilde{\boldsymbol{x}} = \boldsymbol{K}_P^{-1}\boldsymbol{T}_A^T\boldsymbol{h}. \tag{3.9}$$

The simulation model of the robot arm with implemented PD Control with Gravity Compensation is illustrated in Figure 3.2.

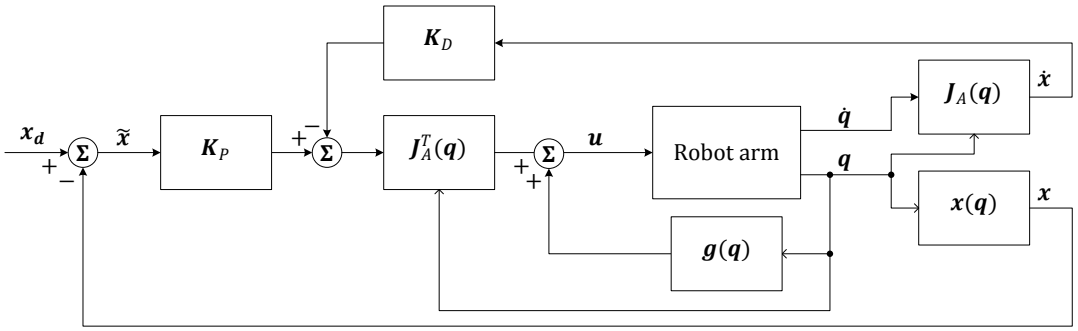



Figure 3.2. Block diagram for PD Control with Gravity Compensation.

The equation underlying the robot arm block in Figure 3.2 is derived from (3.2) as

$$\ddot{\boldsymbol{q}} = \boldsymbol{B}^{-1}\boldsymbol{u} - \boldsymbol{B}^{-1}\left(\boldsymbol{C} + \boldsymbol{F}\right)\dot{\boldsymbol{q}} - \boldsymbol{B}^{-1}\boldsymbol{g}.$$

The robot arm block contains two integrators to obtain $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$.

## 3.2 Inverse Dynamics Control

Inverse dynamics control can be seen as an enhanced version of the PD control with gravity compensation. This control algorithm is based on the exact linearisation of all nonlinear dynamics of the system. It includes additional terms, actively providing the system with a spring-damper behaviour.

Consider the task of tracking a given operational space trajectory, given that $\boldsymbol{h} = \boldsymbol{0}$. (3.2) can be rewritten as

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{u}, \tag{3.10}$$

where

$$\boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{F})\,\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}). \tag{3.11}$$

An inverse dynamics linearising control control signal $\boldsymbol{u}$ can be chosen as

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{3.12}$$

which will lead to a system of double integrators

$$\ddot{\boldsymbol{q}} = \boldsymbol{y}. \tag{3.13}$$

The new control signal $\boldsymbol{y}$ can now be designed in such way that the state $\boldsymbol{x}$ will follow a given trajectory $\boldsymbol{x}_d$. Recall (2.15)

$$\ddot{\boldsymbol{x}} = \boldsymbol{J}_A(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}},$$

that suggests a control law

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) \left( \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} - \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \right), \tag{3.14}$$

with positive definite diagonal matrices $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$. The error dynamics can be described by

$$\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_D\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P\tilde{\boldsymbol{x}} = \boldsymbol{0}, \tag{3.15}$$

with the solution

$$\boldsymbol{\chi} = \boldsymbol{\chi}_0 e^{\boldsymbol{A}t}, \quad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K}_P & -\boldsymbol{K}_D \end{bmatrix}, \quad \boldsymbol{\chi} = \begin{bmatrix} \tilde{\boldsymbol{x}} \\ \dot{\tilde{\boldsymbol{x}}} \end{bmatrix}. \tag{3.16}$$

The properties of (3.15) and (3.16) will ensure that the desired trajectory $\boldsymbol{x}_d$ will asymptotically be reached. By tuning of the control parameters $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$, different dynamic behaviour of the system is obtained. By studying the eigenvalues of $\boldsymbol{A}$, this behaviour can be evaluated.

The simulation model of the robot arm with implemented Inverse Dynamics Control is illustrated in Figure 3.3.

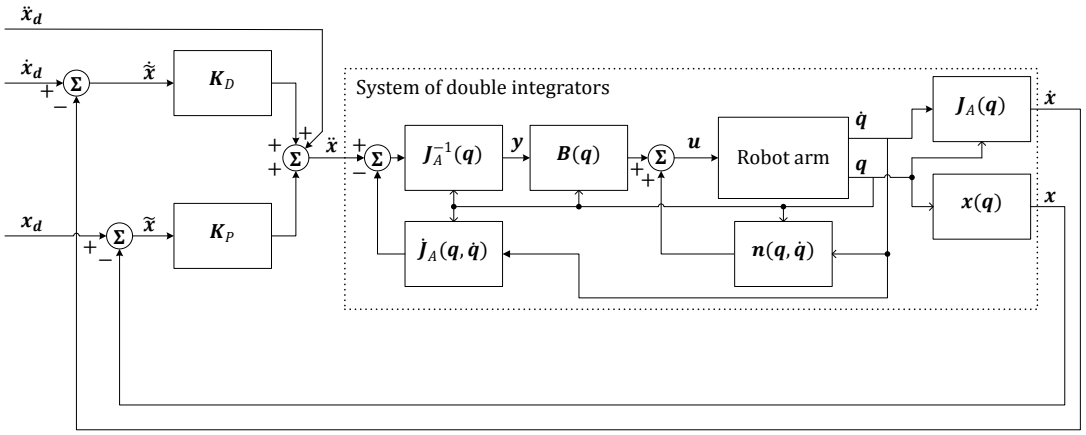



Figure 3.3. Block diagram for Inverse Dynamics Control.

## 3.3 Impedance Control

Impedance control is based on the Inverse dynamics control with the addition that it also addresses the interaction between the robot arm and the environment. It also actively add a mass dynamic to the system, providing the system with a mass-spring-damper behaviour. A contact force is created by the robot arm due to its given spring behaviour when the setpoint is set below the surface of the environment, as shown in Figure 3.1.

Consider the system described by (2.48) and the inverse dynamics control law

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}),$$

with $\boldsymbol{n}$ as in (3.11). The controlled robot arm will now be described by

$$\ddot{\boldsymbol{q}} = \boldsymbol{y} - \boldsymbol{B}^{-1}(\boldsymbol{q})\boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}, \tag{3.17}$$

where $\boldsymbol{h}$ is the previously mentioned contact force and moment affecting the end effector. In the case of this thesis, the environment has been defined as an elastically compliant plane. $\boldsymbol{h}$ is therefore calculated as

$$\boldsymbol{h} = \boldsymbol{K}\boldsymbol{T}_A(\boldsymbol{x} - \boldsymbol{x_e}), \tag{3.18}$$

where $\boldsymbol{K}$ is the stiffness matrix which corresponds to the reaction forces exerted on the end effector by the environment. $\boldsymbol{K}$ is assumed to be very large. The

derivation of $\boldsymbol{T}_A$ is expressed by (2.21) and $\boldsymbol{x} - \boldsymbol{x_e}$ represents the operational space displacement. $\boldsymbol{x}$ is the position of the end-effector and $\boldsymbol{x_e}$ is the position of the undeformed surface, as explained in Figure 3.1.

The stiffness matrix can be used in the following way to form the environmental stiffness matrix, $\boldsymbol{K}_A$.

$$\boldsymbol{K}_A = \boldsymbol{T}_A^T \boldsymbol{K} \boldsymbol{T}_A. \tag{3.19}$$

$\boldsymbol{K}_A$ is then used to describe the equivalent forces affecting the robot arm

$$\boldsymbol{h}_A = \boldsymbol{K}_A(\boldsymbol{x} - \boldsymbol{x_e}). \tag{3.20}$$

The control signal $\boldsymbol{y}$ will be chosen conceptually analogous to (3.14) as

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) \boldsymbol{M}_d^{-1} \left( \boldsymbol{M}_d \ddot{\boldsymbol{x}}_d + \boldsymbol{K}_D \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P \tilde{\boldsymbol{x}} - \boldsymbol{M}_d \dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}}) \dot{\boldsymbol{q}} \right), \tag{3.21}$$

where $\boldsymbol{M}_d$ is a positive definite diagonal matrix that corresponds to the equivalent mass of the system. This will yield the operational space system

$$\boldsymbol{M}_d \ddot{\boldsymbol{x}} + \boldsymbol{K}_D \dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}_P \tilde{\boldsymbol{x}} = \boldsymbol{M}_d \boldsymbol{B}_A^{-1}(\boldsymbol{q}) \boldsymbol{h}_A, \tag{3.22}$$

where

$$\boldsymbol{B}_A(\boldsymbol{q}) = \boldsymbol{J}_A^{-T}(\boldsymbol{q}) \boldsymbol{B}(\boldsymbol{q}) \boldsymbol{J}_A^{-1}(\boldsymbol{q})$$

is the equivalent configuration dependant inertia matrix of the robot arm in the operational space. (3.22) gives an expression for a generalised *mechanical impedance* between the resulting forces $\boldsymbol{M}_d \boldsymbol{B}_A^{-1} \boldsymbol{h}_A$ and the operational space displacement $\tilde{\boldsymbol{x}}$. This impedance can be seen as a mechanical mass-spring-damper system whose coefficients are defined by the diagonal matrices $\boldsymbol{M}_d, \boldsymbol{K}_D$ and $\boldsymbol{K}_P$. The relation between the spring constant $\boldsymbol{K}_P$ and the environment stiffnes can be expressed by

$$\boldsymbol{K}_P(\boldsymbol{x}_d - \boldsymbol{x}) = \boldsymbol{K}_A(\boldsymbol{x} - \boldsymbol{x}_e). \tag{3.23}$$

At equilibrium, the end-effector position is given by

$$\boldsymbol{x}_\infty = \left( \boldsymbol{I} + \boldsymbol{K}_P^{-1} \boldsymbol{K}_A(\boldsymbol{x}) \right)^{-1} \left( \boldsymbol{x}_d + \boldsymbol{K}_P^{-1} \boldsymbol{K}_A(\boldsymbol{x}) \boldsymbol{x}_e \right) \tag{3.24}$$

and the resulting contact force can be expressed by

$$\boldsymbol{h}_{A\infty} = \left(\boldsymbol{I} + \boldsymbol{K}_A(\boldsymbol{x})\boldsymbol{K}_P^{-1}\right)^{-1} \boldsymbol{K}_A(\boldsymbol{x})(\boldsymbol{x}_d - \boldsymbol{x}_e). \tag{3.25}$$

The simulation model of the robot arm with implemented Impedance Control is illustrated in Figure 3.4.

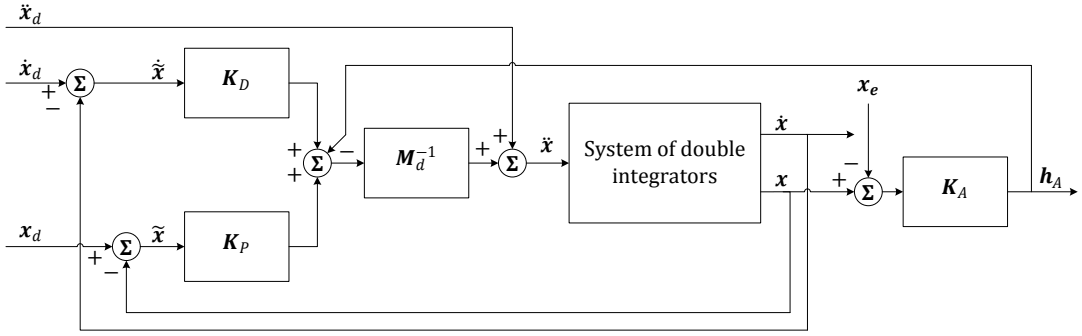



Figure 3.4. Block diagram for Impedance Control.

The system of double integrators is presented in Figure 3.5.

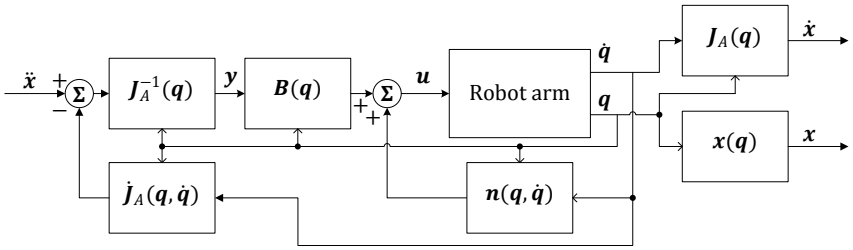



Figure 3.5. System of double integrators

## 3.4 Force Control with Inner Position Loop

The force controller relies on the same principles as the Impedance controller. However, instead of fixing the position setpoint to some constant distance below the environment surface, by the use of an inner position loop, a suitable position setpoint based on the contact force error is calculated. The task for the controller is to maintain a constant end-effector contact force. Hence, the availability of a force sensor at the end-effector is implicitly assumed.

Assuming error-free measurements of the contact force acting on the robot arm, a new linearising control law can be chosen as

$$\boldsymbol{u} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{y} + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{h}. \tag{3.26}$$

The control signal will now be chosen as

$$\boldsymbol{y} = \boldsymbol{J}_A^{-1}(\boldsymbol{q})\boldsymbol{M}_d^{-1}\left(-\boldsymbol{K}_D\dot{\boldsymbol{x}} + \boldsymbol{K}_P(\boldsymbol{x}_F - \boldsymbol{x}) - \boldsymbol{M}_d\dot{\boldsymbol{J}}_A(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\right), \tag{3.27}$$

where $\boldsymbol{x}_F$ is related to a force error. This will yield the operational space system

$$\boldsymbol{M}_d\ddot{\boldsymbol{x}} + \boldsymbol{K}_D\dot{\boldsymbol{x}} + \boldsymbol{K}_P\boldsymbol{x} = \boldsymbol{K}_P\boldsymbol{x}_F, \tag{3.28}$$

which will lead $\boldsymbol{x}$ to $\boldsymbol{x}_F$. Let $\boldsymbol{h}_{Ad}$ denote a desired constant force reference where the relation between $\boldsymbol{x}_F$ and the force error can be expressed as

$$\boldsymbol{x}_F = \boldsymbol{C}_F\left(\boldsymbol{h}_{Ad} - \boldsymbol{h}_A\right), \tag{3.29}$$

where $\boldsymbol{C}_F$ corresponds to a diagonal matrix where the elements are specifying the control action of the force controller. (3.28) and (3.29) reveal that the force control relies on a preexisting inner position control loop. Assuming that an elastic environment is affecting the robot arm, (3.28) together with (3.29) becomes

$$\boldsymbol{M}_d\ddot{\boldsymbol{x}} + \boldsymbol{K}_D\dot{\boldsymbol{x}} + \boldsymbol{K}_P\left(\boldsymbol{I} + \boldsymbol{C}_F\boldsymbol{K}_A\right)\boldsymbol{x} = \boldsymbol{K}_P\boldsymbol{C}_F\left(\boldsymbol{K}_A\boldsymbol{x}_e + \boldsymbol{h}_{Ad}\right). \tag{3.30}$$

With a proportional $\boldsymbol{C}_F$ there will exist an undesired steady state error. However, if $\boldsymbol{C}_F$ contains an integral action according to

$$\boldsymbol{C}_F = \boldsymbol{K}_F + \boldsymbol{K}_I\int^t(\cdot)dt, \tag{3.31}$$

it will be possible to achieve $\boldsymbol{h}_A - \boldsymbol{h}_{Ad} = \boldsymbol{0}$.

The simulation model of the robot arm with implemented Force Control with inner Position loop is illustrated in Figure 3.6.

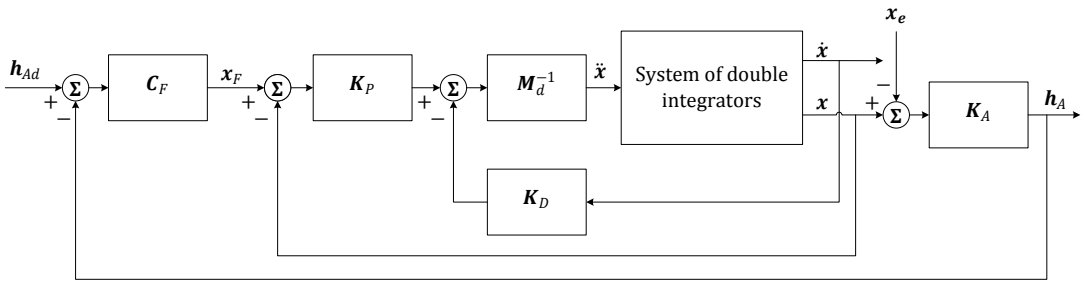



Figure 3.6. Block diagram for Force Control with inner Position loop.

# 4 RESULTS

Four different control algorithms have been implemented for the simplified system. The first two controllers are motion controllers without any interaction with the environment. The latter two are interaction controllers that will be used to interact with the environment to control the contact force between the robot arm and the road.

Each controller has been simulated for two different cases. These cases are different for the motion and interaction controllers and will be explained in more detail for each controller.

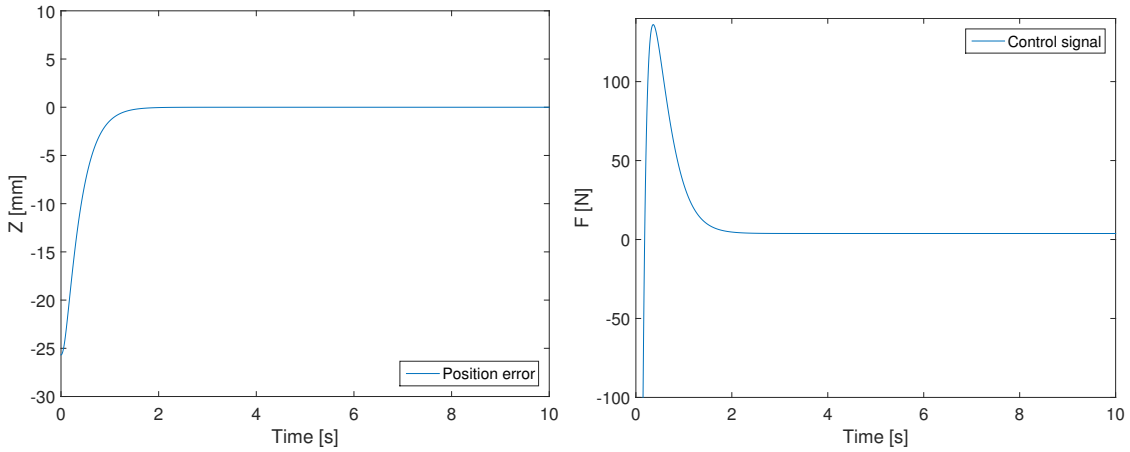
## 4.1 PD Control with Gravity Compensation

The first simulation case represents that the robot arm end-effector should maintain a fixed position relative to the base frame.

The parameters used in the first case can be seen in Table 4.1 and the results of the simulation are presented in Figure 4.1. Since the controller compensates for the effect of gravity on the links, there is no remaining steady state error. This can be seen in the figure.

Table 4.1. Case 1, constant setpoint: Parameters for PD control with gravity compensation.

| Setpoint [mm] | $K_P$ | $K_D$ |
|---|---|---|
| -70 | 30 | 6 |




(a) Position error (b) Control Signal

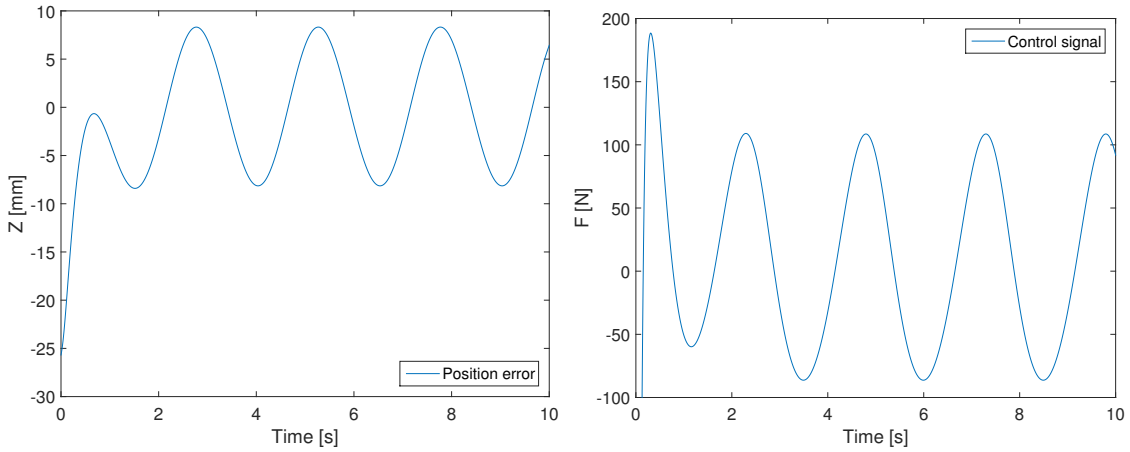
Figure 4.1. Case 1, constant setpoint: Simulation results, PD control with gravity compensation.

The second simulation case represents the robot arm tracking a sinusoidal position reference, without any interaction.

The parameters used in the simulation of the second case can be seen in Table 4.2. The simulation result can be seen in Figure 4.2. As expected, a steady state error cannot be avoided.

Table 4.2. Case 2, sinusoidal setpoint: Parameters for PD control with gravity compensation.

| Setpoint [mm] | $K_P$ | $K_D$ |
|---|---|---|
| $-70 \pm 5$ | 30 | 6 |




(a) Position error (b) Control Signal

Figure 4.2. Case 2, sinusoidal setpoint: Simulation results, PD control with gravity compensation.

## 4.2 Inverse Dynamics Control

The first simulation case represents that the robot arm should maintain a fixed position relative to the base frame. The parameters used in this simulation can be seen in Table 4.3.

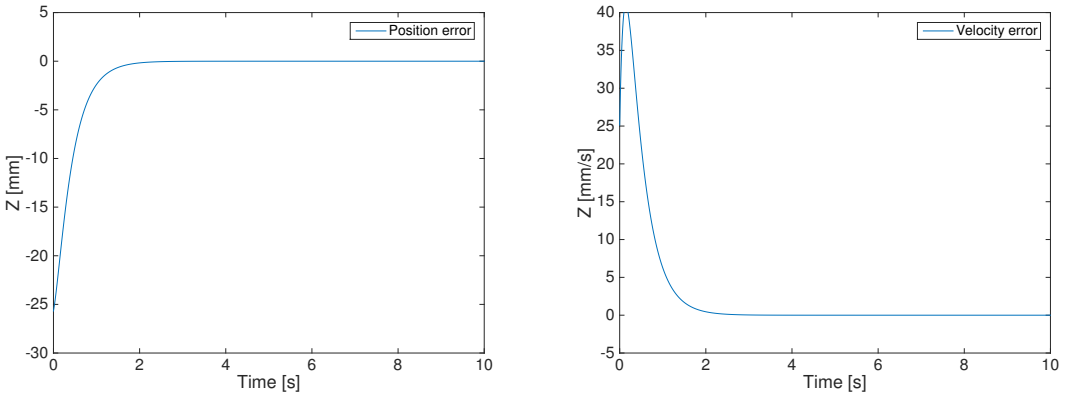
As explained in Section 3.2, the error dynamics can be calculated by (3.16). The results of this calculation, which are presented in Figure 4.3, can be compared to the simulation results which are presented in Figure 4.4. Clearly, the results are equivalent. Due to the controller perfectly compensating for all the dynamics of the system, zero steady state error is successfully achieved. The poles of the system can be calculated from the matrix $\boldsymbol{A}$ in (3.16) where the values of $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are shown in Table 4.3. The obtained poles are

$$p = \{-8.3723, -2.6277\},$$

which clearly resides in the left half of the complex plane thus the system is asymptotically stable.

Table 4.3. Case 1, constant setpoint: Parameters for Inverse dynamics control.

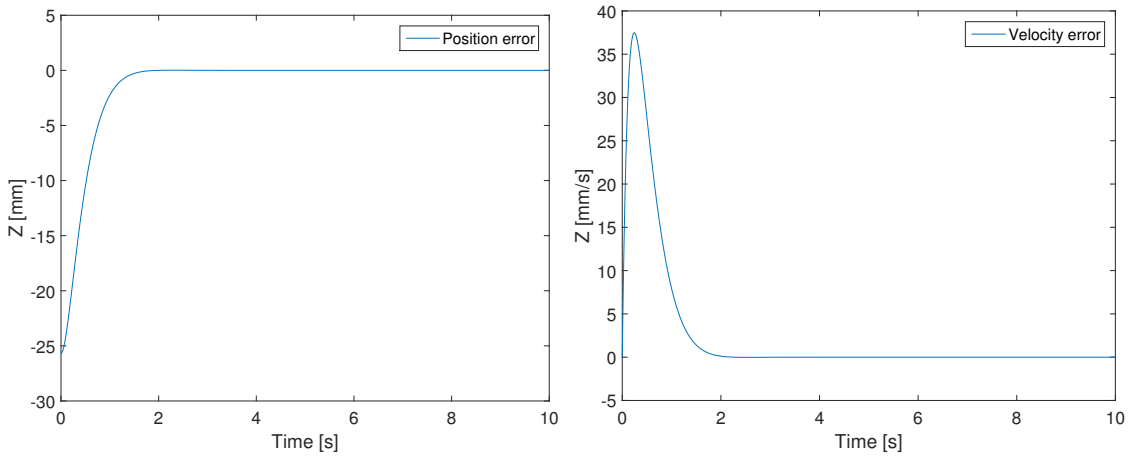
| Setpoint [mm] | $K_P$ | $K_D$ |
|---|---|---|
| -70 | 30 | 10 |




(a) Position error (b) Velocity error

Figure 4.3. Error dynamics for an equivalent spring-damper system.

The control signal for the first case is presented in Figure 4.5.




(a) Position error (b) Velocity error

Figure 4.4. Case 1, constant setpoint: Simulation results, Inverse dynamics control.

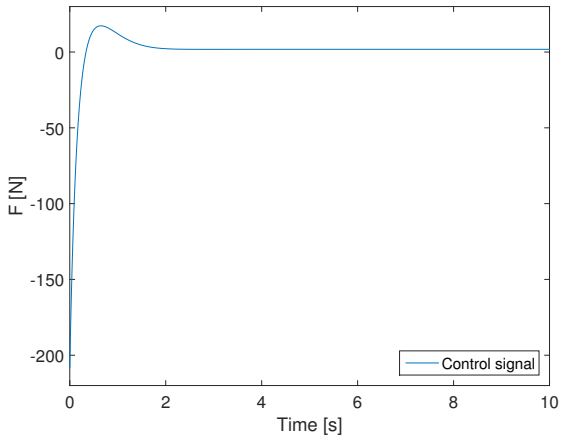


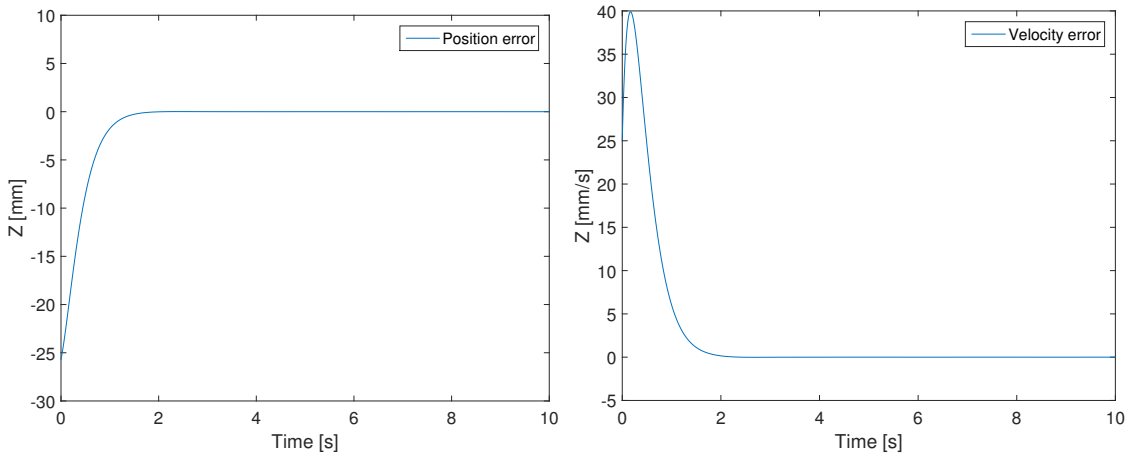

Figure 4.5. Case 1, constant setpoint: Control signal for Inverse dynamics control.

The second simulation case represents the robot arm tracking a sinusoidal position and its two time derivatives as reference, without any interaction. The parameters used in the simulation for the second case can be seen in Table 4.4. The result of this simulation is presented in Figure 4.6. Due to the controller perfectly compensating for all the dynamics of the system, zero steady state error is successfully achieved.

Table 4.4. Case 2, sinusoidal setpoint: Parameters for Inverse dynamics control.

| Setpoint [mm] | $K_P$ | $K_D$ |
|---|---|---|
| $-70 \pm 5$ | 30 | 10 |

The control signal for the second case is presented in Figure 4.7.




(a) Position error (b) Velocity error

Figure 4.6. Case 2, sinusoidal setpoint: Simulation results, Inverse dynamics control.

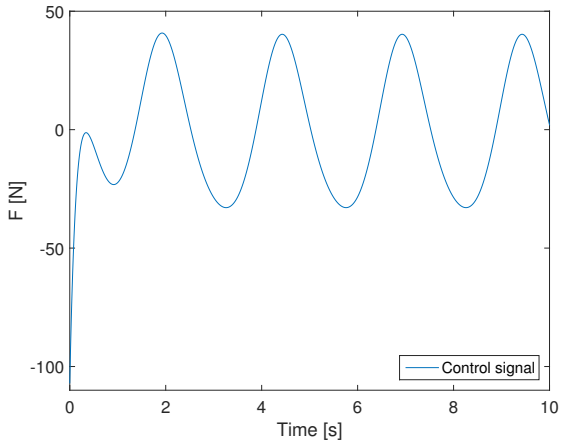



Figure 4.7. Case 2, sinusoidal setpoint: Control signal for Inverse dynamics control.

## 4.3 Impedance Control

The task for this controller is to have a low, yet persistent, contact force to ensure contact between the contact shoes and the electrified tracks.

In the first case, the vehicle travels at a constant speed on a flat road. This implies a constant distance between the vehicle and the road. It also implies a constant distance between the setpoint and the road.

The parameters used in the simulation of the first case can be seen in Table 4.5. The results of this simulation are presented in Figure 4.8 - Figure 4.10. Due to the ground stiffness there will be a displacement between the end-effector and the road according to (3.23), which can be seen in Figure 4.8.

Since the end-effector is in contact with the road, there will be a contact force which can be seen in Figure 4.10b.

Table 4.5. Case 1, road unevenness represented by a constant: Parameters for Impedance control.

| Setpoint [mm] | Displacement [mm] | Mass [Kg] | $K_P$ | $K_D$ |
|---|---|---|---|---|
| -70 | 0.25 | 1 | 75 | 25 |

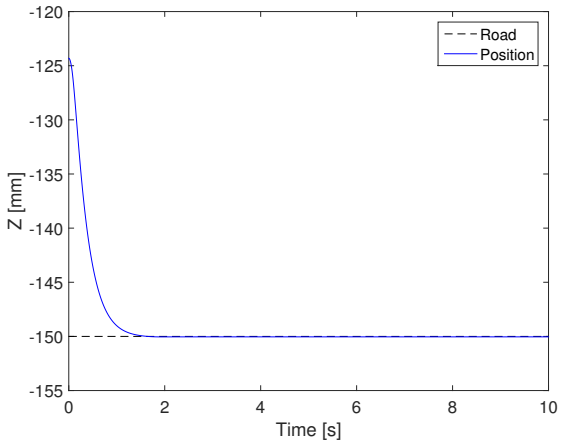
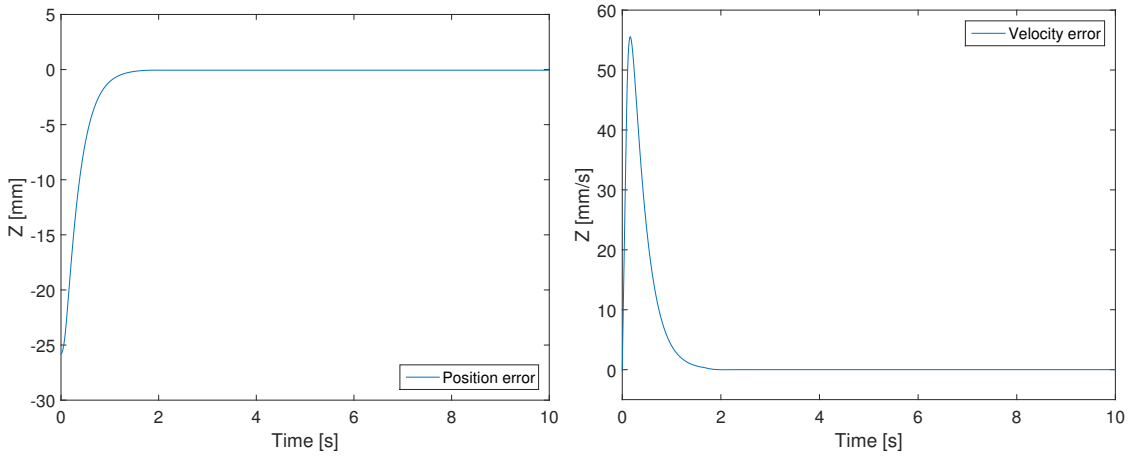
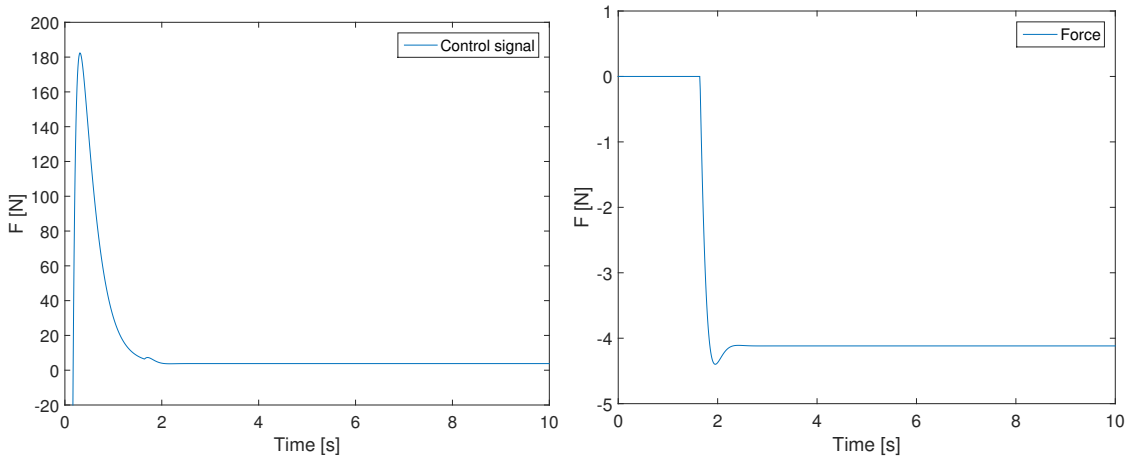



Figure 4.8. Case 1, road unevenness represented by a constant: Position of the end-effector for Impedance control.




(a) Position error (b) Velocity error

Figure 4.9. Case 1, road unevenness represented by a constant: Simulation results, Impedance control.




(a) Control Signal (b) Contact Force

Figure 4.10. Case 1, road unevenness represented by a constant: Simulation results, Impedance control.

In the second case, the vehicle travels with varying speed on a non-flat road, i.e., the vehicle's suspension will be compressed and extended as it drives on the road. This means that the distance between the vehicle and road will vary and the setpoint position will be set lower than the road to ensure contact force. The alteration of compression in the vehicle's suspension will be regarded as a sinusoidal change.

The parameters used in the simulation of the second case can be seen in Table 4.6. The results of this simulation are presented in Figure 4.10 - Figure 4.11. Due to the ground stiffness there will be a displacement between the end-effector and the road according to (3.23).

Since the end-effector is in contact with the road, there will be a contact force which can be seen in Figure 4.13b.

Table 4.6. Case 2, road unevenness represented by a sinusoid: Parameters for Impedance control.

| Setpoint [mm] | Displacement [mm] | Mass [Kg] | $K_P$ | $K_D$ |
|---|---|---|---|---|
| $-70 \pm 5$ | 0.25 | 1 | 75 | 25 |

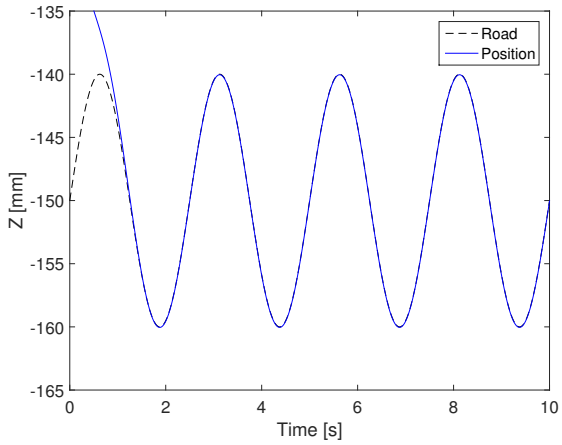
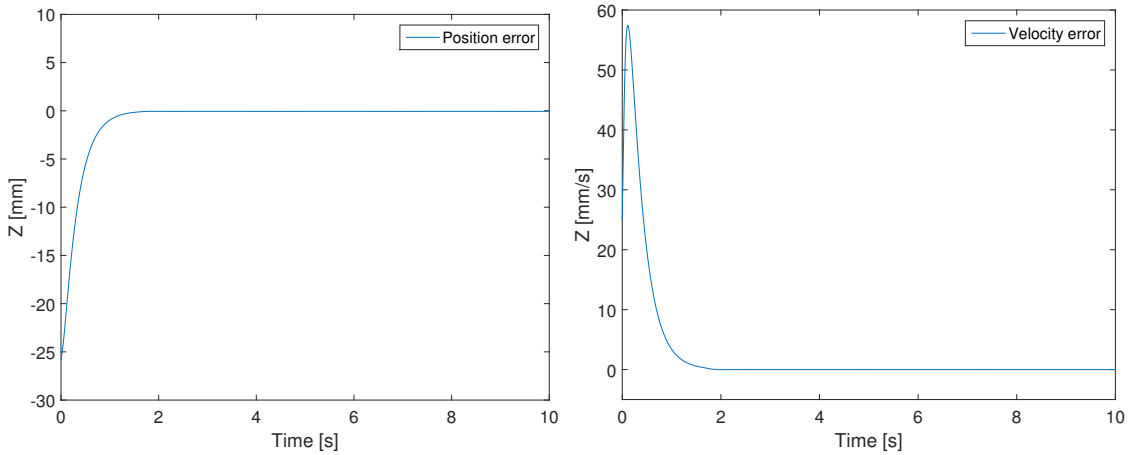
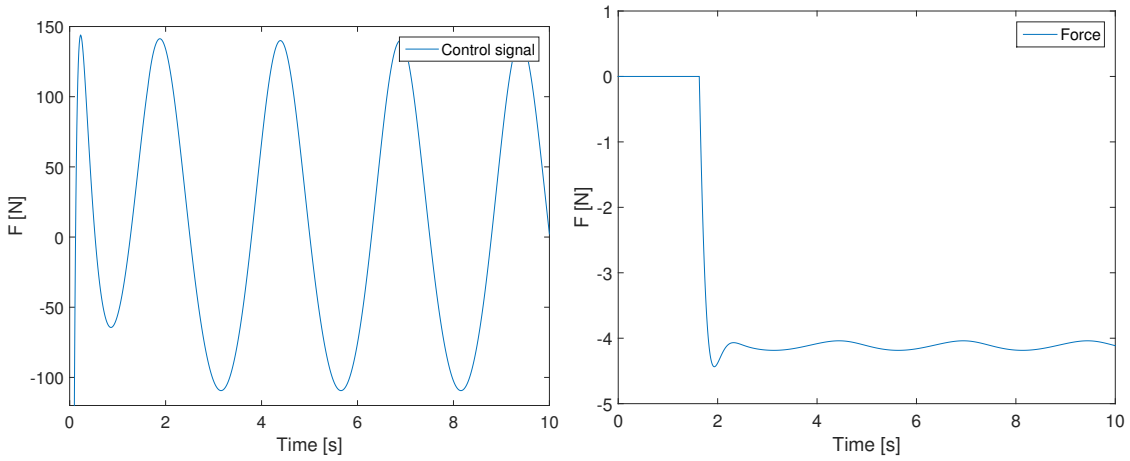



Figure 4.11. Case 2, road unevenness represented by a sinusoid: Position of the end-effector for Impedance control.




(a) Position error (b) Velocity error

Figure 4.12. Case 2, road unevenness represented by a sinusoid: Simulation results, Impedance control.




(a) Control Signal (b) Contact Force

Figure 4.13. Case 2, road unevenness represented by a sinusoid: Simulation results, Impedance control.

Based on the results presented above, it can be concluded that the Impedance controller fulfills the control objective.

## 4.4 Force Control with Inner Position Loop

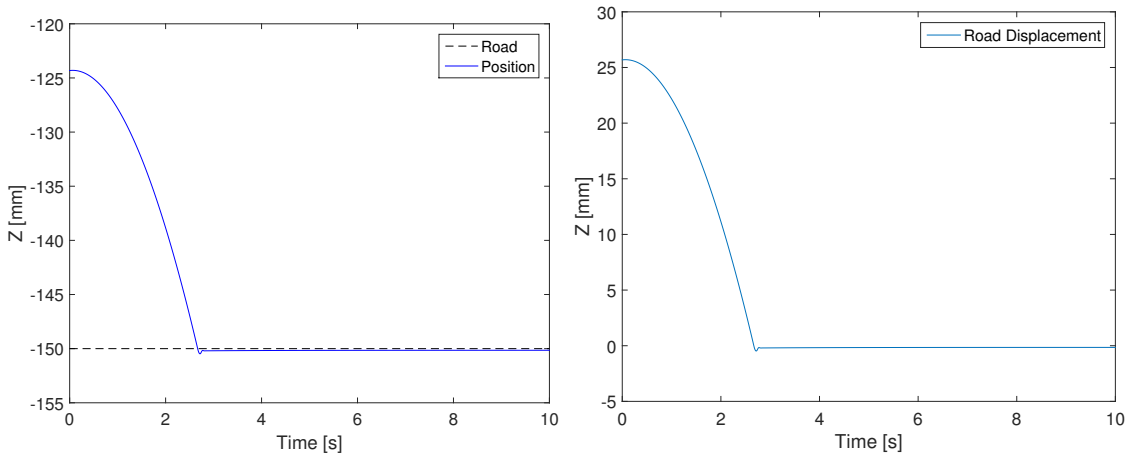
The force controller takes a reference force as setpoint which is used to calculate a suitable position setpoint via an inner position loop. The goal for this controller is to ensure contact between the end-effector and the electrified tracks by maintaining a constant end-effector contact force corresponding to the force setpoint.

In the first case, the vehicle travels at a constant speed on a flat road. This implies a constant distance between the vehicle and the road. The force setpoint is constant.

The simulation parameters used in the first case can be seen in Table 4.7. The results of the simulation are presented in Figure 4.14 and Figure 4.15. Figure 4.14b presents the displacement between the end-effector and the road. The figure shows the contact force error successfully declining towards zero.

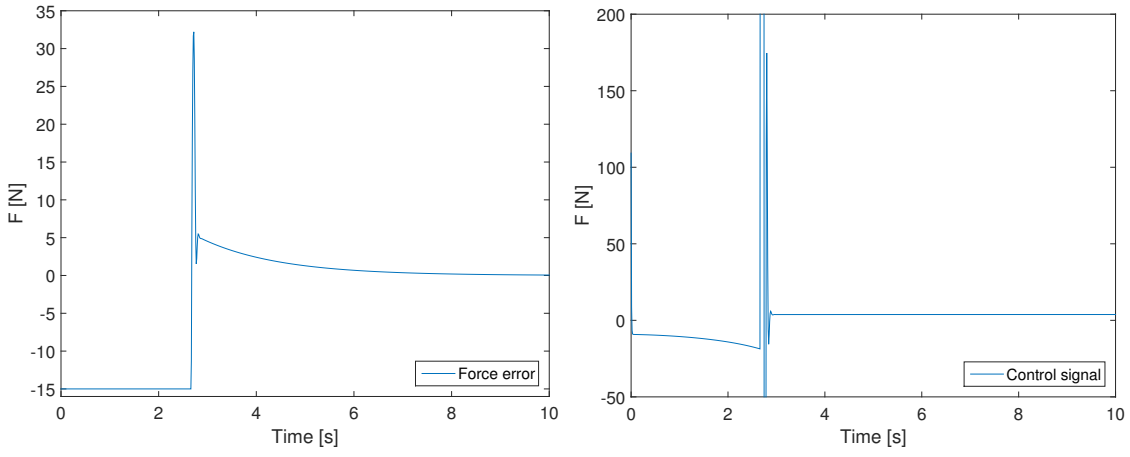
Table 4.7. Case 1, road unevenness represented by a constant: Parameters for Force control.

| Setpoint [N] | Mass [Kg] | $K_P$ | $K_D$ | $K_F$ | $K_I$ |
|---|---|---|---|---|---|
| -10 | 1 | 10 | 100 | 20 | 20 |




(a) Position (b) Displacement

Figure 4.14. Case 1, road unevenness represented by a constant: Simulation results, Force control.




(a) Contact force error (b) Control Signal

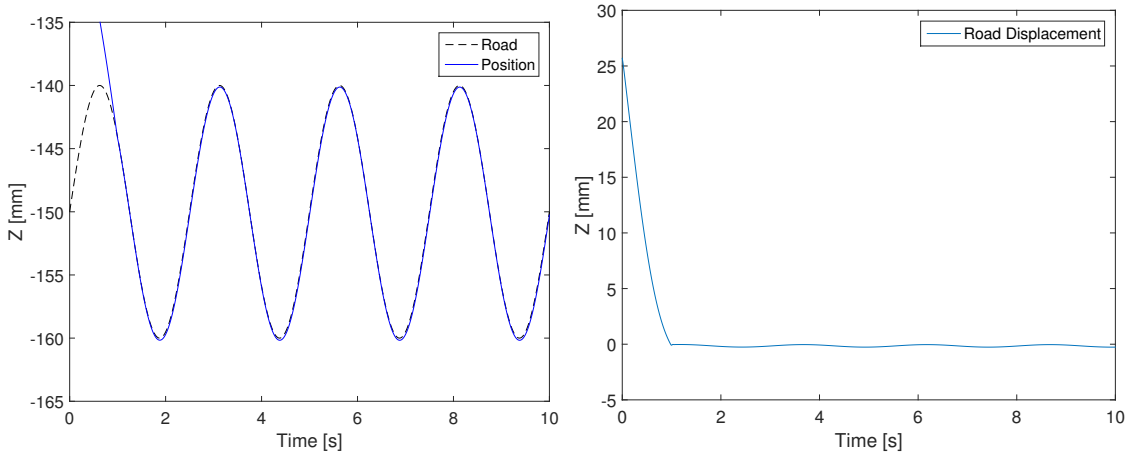
Figure 4.15. Case 1, road unevenness represented by a constant: Simulation results, Force control.

In the second case, the vehicle travels with varying speed on a non-flat road, i.e., the vehicle's suspension will be compressed and extended as it drives on the road. This means that the distance between the vehicle and road will vary. The alteration of compression in the vehicle's suspension will be regarded as a sinusoidal change. The force setpoint is constant.

The simulation parameters used in the second case can be seen in Table 4.8. The results of the simulation are presented in Figure 4.16 and Figure 4.17a. Figure 4.16b presents the displacement between the end-effector and the road.

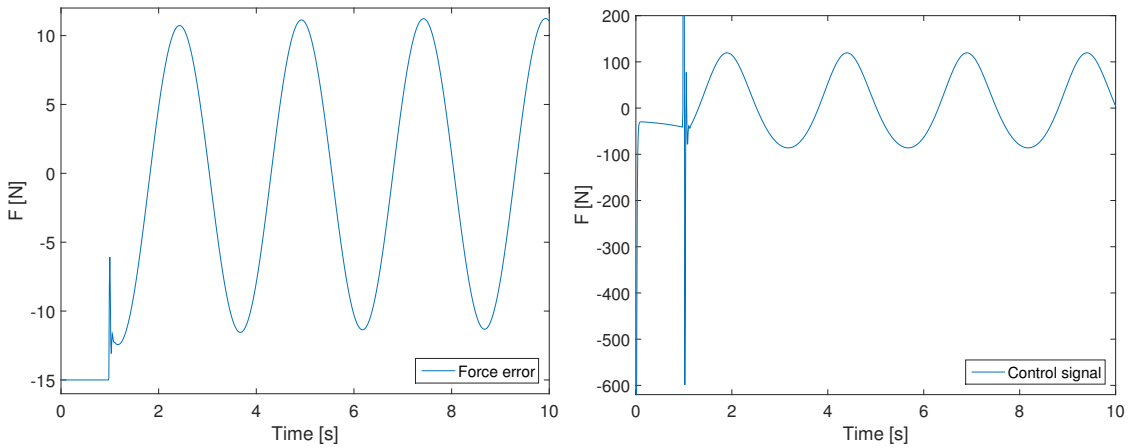
Table 4.8. Case 2, road unevenness represented by a sinusoid: Parameters for Force control.

| Setpoint [N] | Mass [Kg] | $K_P$ | $K_D$ | $K_F$ | $K_I$ |
|---|---|---|---|---|---|
| $-10$ | 1 | 10 | 100 | 20 | 20 |




(a) Position (b) Displacement

Figure 4.16. Case 2, road unevenness represented by a sinusoid: Simulation results, Force control.




(a) Contact force error (b) Control Signal

Figure 4.17. Case 2, road unevenness represented by a sinusoid: Simulation results, Force control.

As can be seen in the results presented above, the contact force error does not reach zero at steady-state but a constant contact is achieved, thus, the Force controller with inner position loop fulfills the control objective.

# 5 DISCUSSION

The use of the Lagrange method to derive a dynamic model has proven advantageous since the considered system consists of a large number of links and joints. This method allows for a systematic approach where the complexity of the model is kept at a manageable level.

One challenge during the modeling phase of this thesis was the coupling between the vertical and lateral motion of the robot arm, i.e., the individual actuators affect the motion of the robot arm around both the vertical and the lateral axes. All constraint equations necessary to employ three-dimensional motion was not found during this project.

An alternative way of constructing the robot arm could simplify the modeling of the system. If the two degrees of freedom were decoupled it would be easier to model the robot arm. A possible way of doing this would be to have a rotating actuator mounted in the axis of rotation of the arm. A linear actuator could then be mounted between stated rotating axis and the robot arm to elevate the arm. Possible drawbacks could be that the actuators might individually have to take larger loads, meaning larger actuators. The design might also allocate too much space underneath a vehicle to be a viable solution.

The purpose of the two motion controllers is not to fulfill the control objective, but to show how it is possible to add complexity to an initially rather simple controller to achieve a complex end result. The two implemented interaction controllers however are possible candidates to fulfill the control objective since both of them can theoretically maintain a constant contact force.

All four control algorithms are based on exact compensation of some, or all, nonlinear dynamics of the system. The PD controller compensates for the effect of gravity, but the other controllers compensate for all undesired nonlinear dynamics whilst also actively adding a desired dynamic behaviour to the system. A comparison between the error dynamics of the system with Inverse dynamics control and the error dynamics of the equivalent linear system was made. The results can be seen in Figure 4.3 and Figure 4.4. Both figures display the exact same behaviour which leads to the conclusion that the controllers manage to perfectly compensate for all nonlinear dynamics of the system.

The results of the Impedance controller show that it manages to follow the given reference trajectory in the second simulation case to a certain extent as is shown in Figure 4.12a. Seeing that the controller manages to follow the given trajectory, the contact force will fluctuate in the near vicinity of the desired contact force. This can be seen in Figure 4.13b.

The results of the Force controller with inner position loop show that for the first simulation case, the force error declines towards zero. This can be seen in Figure 4.15a. For the second simulation case there is a trade-off between the magnitude of the initial impact force and the magnitude of the force error. Tuning the controller such that the resulting impact force is low will lead to a large deviation of force error. Tuning the controller for a small force error deviation will result in a high impact force. The tuning used for the simulation of the second case for force control allows for a somewhat large deviation for the force error, whilst guaranteeing contact with the electrified tracks. This can be seen in Figure 4.17a.

Both interaction controllers seem to sufficiently fulfill the control objective, i.e., guaranteeing constant contact with the environment whilst simultaneously maintaining a moderate contact force.

There are some advantages of the Impedance control compared to the Force control. It requires less to implement than the Force control and the Force controller also needs an accurate measurement of the contact forces and moments applied on the end-effector to function. The Impedance controller follows a trajectory based on position, velocity and acceleration, which the Force control does not, seeing that the latter can only have a vector of desired forces and moments as reference. This gives that in the case of lateral motion of the truck, the Impedance controller could easily turn the robot arm to follow the powered rails, whilst the Force controller would rely on the resulting lateral friction forces and moments on the contact shoes caused by the truck's deviation from the electrified tracks. The latter seems like a less robust way of following the tracks, seeing that it relies on disturbance-free contact rather than a measured position signal from a sensor.

The are also some disadvantages of the Impedance control compared to the Force control. The Impedance control relies on a good knowledge and measurement of the vertical distance, velocity and acceleration of the road relative to the base frame. If any of these measurements cannot be obtained, the controller will not be able to control the robot arm as desired. Even if all measurements can be obtained, the Impedance controller also relies on a estimated homogeneous stiffness in the road. Should the road become more or less stiff compared to the stiffness when the controller was originally tuned, the contact force would change without any indication. The Force control on the other hand only desires a distance measurement between the road and the base frame, and of course the previously mentioned force measurement, to function. Seeing that the controller has a force as reference, the force will be sufficiently controlled regardless of road stiffness. This also means that tuning the impedance control to a certain force will be a somewhat tedious task compared to the Force control.

To evaluate which of the two interaction controllers is most suitable to control the robot arm. The Force control relies only on two measurements, instead of three as for the Impedance control, and is easier to tune to obtain a certain contact force. However, one of the disadvantages of Impedance control, that a change will occur in the road stiffness, can be deemed unlikely to occur. Therefore, due to the slightly less advanced implementation and the lack of force sensors one can argue that the Impedance control is more suitable to implement for the robot arm.

The control of a device such as the robot arm may be scheduled and divided into several control modes. One mode can be implemented for the localization of the electrified tracks, one for the actual interaction with the rails and one for lifting the robot arm up to a resting position. By scheduling the controller in this way, it is possible to tune the controller for each separate mode to maximise performance.

# Bibliography

Aldammad, M., Ananiev, A. and Kalaykov, I. (2014). Current collector for heavy vehicles on electrified roads.

Boström, A. (Ed.) (2013). *Rigid Body Dynamics*. Gothenburg.

Sciavicco, L. and Siciliano, B. (Eds.) (2000). *Modelling and Control of Robot Manipulators*. Advanced Textbooks in Control and Signal Processing series. 2nd edn. Springer. London.

Skoog, E. and Stenman, D. (Eds.) (2014). *Modeling of a Robot Arm,* Mathematica Notebook File. Gothenburg.

Viktoria Swedish ICT (2014). Slide-in electric road system, *Conductive project report*. Technical report.

# A Modeling

## A.1 Kinematics

### A.1.1 Representation of a Rigid Body

A rigid body can be described by its position and orientation in space with respect to the reference frame. Let $O - xyz$ be the reference frame and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ be the unit vectors of the reference frame. A point $\boldsymbol{o}^{'}$ can be written with respect to the frame in $O - xyz$ in the following way

$$\boldsymbol{o}^{'} = o_x^{'}\boldsymbol{x} + o_y^{'}\boldsymbol{y} + o_z^{'}\boldsymbol{z},$$

where $o_x^{'}, o_y^{'}, o_z^{'}$ are the components of the vector $\boldsymbol{o}^{'}$ along the frame axes of $O-xyz$. The position of the point $\boldsymbol{o}^{'}$ can be described by the vector

$$\boldsymbol{o}^{'} = \begin{bmatrix} o_x^{'} \\ o_y^{'} \\ o_z^{'} \end{bmatrix}. \tag{A.1}$$

In order to describe the orientation of $\boldsymbol{o}^{'}$ with respect to the frame $O - xyz$ an orthonormal frame is attached to the point $\boldsymbol{o}^{'}$ with the unit vectors $\boldsymbol{x}^{'}, \boldsymbol{y}^{'}, \boldsymbol{z}^{'}$. The resulting frame will be $O - x^{'}y^{'}z^{'}$. The relation between the unit vectors of frame $O - x^{'}y^{'}z^{'}$ and the reference frame $O - xyz$ can be described by

$$\begin{aligned} \boldsymbol{x}^{'} &= x_x^{'}\boldsymbol{x} + x_y^{'}\boldsymbol{y} + x_z^{'}\boldsymbol{z} \\ \boldsymbol{y}^{'} &= y_x^{'}\boldsymbol{x} + y_y^{'}\boldsymbol{y} + y_z^{'}\boldsymbol{z} \\ \boldsymbol{z}^{'} &= z_x^{'}\boldsymbol{x} + z_y^{'}\boldsymbol{y} + z_z^{'}\boldsymbol{z}. \end{aligned} \tag{A.2}$$

### A.1.2 Rotation matrices

The rotational matrix is a $(3\times3)$ matrix consisting of the three unit vectors in (A.1), according to

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{x}^{'} & \boldsymbol{y}^{'} & \boldsymbol{z}^{'} \end{bmatrix} = \begin{bmatrix} x_x^{'} & y_x^{'} & z_x^{'} \\ x_y^{'} & y_y^{'} & z_y^{'} \\ x_z^{'} & y_z^{'} & z_z^{'} \end{bmatrix}. \tag{A.3}$$

A point $\boldsymbol{p}$ is represented by

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix},$$

with respect to the frame $O-xyz$, but it can also be represented by

$$\boldsymbol{p}^{'} = \begin{bmatrix} p_x^{'} \\ p_y^{'} \\ p_z^{'} \end{bmatrix},$$

in the coordinate frame $O-x^{'}y^{'}z^{'}$. Both $p$ and $p^{'}$ represent the same point in space, $P$. Therefore, one can write

$$\boldsymbol{p} = p_x^{'}\boldsymbol{x} + p_y^{'}\boldsymbol{y} + p_z^{'}\boldsymbol{z} = \begin{bmatrix} \boldsymbol{x}^{'} & \boldsymbol{y}^{'} & \boldsymbol{z}^{'} \end{bmatrix} \boldsymbol{p}^{'},$$

which can be rewritten according to (A.3) as

$$\boldsymbol{p} = \boldsymbol{R}\boldsymbol{p}^{'}. \tag{A.4}$$

Let $\boldsymbol{p}^i$ denote a vector describing the vector of coordinates of a point $P$ in the $i^{th}$ frame. Let $\boldsymbol{R}_j^i$ be the rotation matrix describing frame $i$ with respect to frame $j$. Let there be three vectors, $\boldsymbol{p}^0$, $\boldsymbol{p}^1$ and $\boldsymbol{p}^2$, all describing the same point for the corresponding frame $O-x_0y_0z_0$, $O-x_1y_1z_1$ and $O-x_2y_2z_2$ respectively. $\boldsymbol{p}^2$ written in the coordinate frame 1 can be written as

$$\boldsymbol{p}^1 = \boldsymbol{R}_2^1\boldsymbol{p}^2. \tag{A.5}$$

Writing the equations for describing $\boldsymbol{p}^0$ and $\boldsymbol{p}^1$ and also $\boldsymbol{p}^0$ and $\boldsymbol{p}^2$ in the same manner and inserting these into (A.5) will result in

$$\boldsymbol{R}_2^0 = \boldsymbol{R}_1^0\boldsymbol{R}_2^1. \tag{A.6}$$

### A.1.3 Transformation matrices

The previous section only looked at the rotation between frames. If a translation were also to be included, it can be expressed in the following way

$$\boldsymbol{p}^0 = \boldsymbol{o}_1^0 + \boldsymbol{R}_1^0\boldsymbol{p}^1, \tag{A.7}$$

where $\boldsymbol{o}_1^0$ denotes the vector describing the origin of frame 1 with respect to frame 0. By simply adding a fourth component to the generic coordinate vector it is possible to write the translation + rotation in a more compact form,

$$\tilde{\boldsymbol{p}} = \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}. \tag{A.8}$$

Using this new way of describing vectors one can write a translation + rotation as

$$\boldsymbol{A}_1^0 = \begin{bmatrix} \boldsymbol{R}_1^0 & \boldsymbol{o}_1^0 \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{A.9}$$

which is a $(4 \times 4)$ matrix called the *homogeneous transformation matrix*. Just like the rotation matrix, one can write consecutive rotations and translations by stacking homogeneous transformation matrices, as seen below.

$$\tilde{\boldsymbol{p}}^0 = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \dots \boldsymbol{A}_n^{n-1} \tilde{\boldsymbol{p}}^n \tag{A.10}$$

$$\boldsymbol{A}_n^0 = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \dots \boldsymbol{A}_n^{n-1} \tag{A.11}$$

In the special case of a homogenous transformation matrix from the base frame $b$ and the end-effector $e$, the matrix is denoted

$$\boldsymbol{T}_e^b = \boldsymbol{A}_1^b \boldsymbol{A}_2^1 \dots \boldsymbol{A}_e^{e-1}. \tag{A.12}$$

### A.1.4 Direct kinematics

The purpose of direct kinematics is to calculate the end effector's position and orientation as a function of the joint variables in a manipulator. The homogeneous transformation matrix in the previous section can be written as

$$\boldsymbol{T}_e^b = \begin{bmatrix} \boldsymbol{R}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ \boldsymbol{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_e^b(\boldsymbol{q}) & \boldsymbol{s}_e^b(\boldsymbol{q}) & \boldsymbol{a}_e^b(\boldsymbol{q}) & \boldsymbol{p}_e^b(\boldsymbol{q}) \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{A.13}$$

where $\boldsymbol{q}$ is the vector of joint variables. $\begin{bmatrix} \boldsymbol{n}_e^b(\boldsymbol{q}) & \boldsymbol{s}_e^b(\boldsymbol{q}) & \boldsymbol{a}_e^b(\boldsymbol{q}) \end{bmatrix}$ are the unit vectors of the frame attached to the end effector at position $\boldsymbol{p}_e^b$. All of these are in this case expressed in the base frame, hence the $b$ superscript. These vectors are all a function of $\boldsymbol{q}$.

#### Euler Angles

Rotation matrices are characterised by nine non-independent elements. This implies that the rotation matrices give a redundant description of the frame orientation. The elements are, due to orthogonality, related by six constraints which implies that it is sufficient to describe the orientation of a rigid body in space by only three parameters. Representation of an orientation using a set of three angles according to

$$\boldsymbol{\phi} = \begin{bmatrix} \varphi & \vartheta & \psi \end{bmatrix}^T, \tag{A.14}$$

will lead to a *minimal representation*. Consider an arbitrary rotation around one of the coordinate axes with a corresponding rotation matrix that is a function of a single angle. Then, a composition of three such rotations, assuming that two

successive rotations are not made around parallel axes, will correspond to a generic rotation matrix $\boldsymbol{R}(\boldsymbol{\phi})$.

Given a rotation matrix $\boldsymbol{R}_e^b(\boldsymbol{q})$, which represents an arbitrary orientation of the end effector, obtained through consecutive rotations as described in appendix A.1.3, there exists a set of Euler Angles, $\boldsymbol{\phi}$, which will lead to

$$\boldsymbol{R}(\boldsymbol{\phi}) = \boldsymbol{R}_e^b(\boldsymbol{q}), \tag{A.15}$$

for a given rotation matrix

$$\boldsymbol{R}_e^b = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \tag{A.16}$$

For the case of Euler Angles describing ZYX rotation, the solution to (A.15) corresponds to

$$\varphi = \text{Atan2}(r_{21}, r_{11}) \tag{A.17}$$

$$\vartheta = \text{Atan2}\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \tag{A.18}$$

$$\psi = \text{Atan2}(r_{32}, -r_{33}). \tag{A.19}$$

## A.1.5 Differential kinematics

The mapping between the joint velocities $\dot{\boldsymbol{q}}$ and the velocity of a point $\boldsymbol{p}$ is described by the $(6 \times N)$ Geometric Jacobian matrix $\boldsymbol{J}$ such that

$$\boldsymbol{v} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_O(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}}. \tag{A.20}$$

The velocity of an arbitrary frame $i$ corresponds to

$$\dot{\boldsymbol{p}}_i = \boldsymbol{J}_{P1}^{(i)}\dot{q}_1 + \ldots + \boldsymbol{J}_{Pi}^{(i)}\dot{q}_i = \boldsymbol{J}_P^{(i)}\dot{\boldsymbol{q}} \tag{A.21}$$

$$\boldsymbol{\omega}_i = \boldsymbol{J}_{O1}^{(i)}\dot{q}_1 + \ldots + \boldsymbol{J}_{Oi}^{(i)}\dot{q}_i = \boldsymbol{J}_O^{(i)}\dot{\boldsymbol{q}}, \tag{A.22}$$

which can be written in vector form

$$\boldsymbol{J}_P^{(i)} = \begin{bmatrix} \boldsymbol{J}_{P1}^{(i)} & \cdots & \boldsymbol{J}_{Pi}^{(i)} & \boldsymbol{0} & \cdots & \boldsymbol{0} \end{bmatrix} \tag{A.23}$$

$$\boldsymbol{J}_O^{(i)} = \begin{bmatrix} \boldsymbol{J}_{O1}^{(i)} & \cdots & \boldsymbol{J}_{Oi}^{(i)} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}. \tag{A.24}$$

To find the Geometric Jacobian for the frame attached to an arbitrary point $\boldsymbol{p}_i$, the columns of (A.23),(A.24) correspond to

$$\boldsymbol{J}_{Pj}^{(i)} = \begin{cases} \boldsymbol{z_{j-1}} & \text{for a prismatic joint } j \\ \boldsymbol{z_{j-1}} \times (\boldsymbol{p}_i - \boldsymbol{p}_{j-1}) & \text{for a revolute joint } j. \end{cases} \tag{A.25}$$

Further,

$$\boldsymbol{J}_{Oj}^{(i)} = \begin{cases} \mathbf{0} & \text{for a prismatic joint } j \\ \boldsymbol{z}_{j-1} & \text{for a revolute joint } j. \end{cases} \tag{A.26}$$

To find the Geometric Jacobian for the frame attached to link $i$, the columns of (A.23), (A.24) correspond to

$$\boldsymbol{J}_{Pj}^{(l_i)} = \begin{cases} \boldsymbol{z_{j-1}} & \text{for a prismatic joint } j \\ \boldsymbol{z_{j-1}} \times (\boldsymbol{p}_{l_i} - \boldsymbol{p}_{j-1}) & \text{for a revolute joint } j, \end{cases} \tag{A.27}$$

where $\boldsymbol{p}_{l_i}$ is the center of mass for link $i$. Further,

$$\boldsymbol{J}_{Oj}^{(l_i)} = \begin{cases} \mathbf{0} & \text{for a prismatic joint } j \\ \boldsymbol{z}_{j-1} & \text{for a revolute joint } j. \end{cases} \tag{A.28}$$

With some modifications, the notion used in (A.27)-(A.28) can be adopted for computing the Geometric Jacobian for an actuator $i$ according to

$$\boldsymbol{J}_{Pj}^{(m_i)} = \begin{cases} \boldsymbol{z_{j-1}} & \text{for a prismatic joint } j \\ \boldsymbol{z_{j-1}} \times (\boldsymbol{p}_{m_i} - \boldsymbol{p}_{j-1}) & \text{for a revolute joint } j, \end{cases} \tag{A.29}$$

where $\boldsymbol{p}_{m_i}$ is the center of mass for link $i$. Further,

$$\boldsymbol{J}_{Oj}^{(m_i)} = \begin{cases} \boldsymbol{J}_{Oj}^{(l_i)} & j = 1, \ldots, i-1 \\ k_r \boldsymbol{z}_{m_i} & j = i, \end{cases} \tag{A.30}$$

where $k_r$ is the gear reduction ratio of the actuators.

For the robot arm considered in this thesis, the Geometric Jacobians for the links and actuators in the three branches $i, j$ and $t$ are presented below. The Geometric Jacobian for end-effector velocity is also presented.

$$
\begin{aligned}
\boldsymbol{J}_i^{l_1} &= \begin{bmatrix} \boldsymbol{z}_{bi} \times (\boldsymbol{p}_{l_1} - \boldsymbol{p}_{bi}) & 0 & 0 \\ \boldsymbol{z}_{bi} & 0 & 0 \end{bmatrix} \\
\boldsymbol{J}_i^{l_2} &= \begin{bmatrix} \boldsymbol{z}_{bi} \times (\boldsymbol{p}_{l_2} - \boldsymbol{p}_{bi}) & \boldsymbol{z}_{1i} \times (\boldsymbol{p}_{l_2} - \boldsymbol{p}_{1i}) & 0 \\ \boldsymbol{z}_{bi} & \boldsymbol{z}_{1i} & 0 \end{bmatrix} \\
\boldsymbol{J}_i^{l_3} &= \begin{bmatrix} \boldsymbol{z}_{bi} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{bi}) & \boldsymbol{z}_{1i} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{1i}) & \boldsymbol{z}_{2i} \\ \boldsymbol{z}_{bi} & \boldsymbol{z}_{1i} & 0 \end{bmatrix} \\
\boldsymbol{J}_i^{m_3} &= \begin{bmatrix} \boldsymbol{z}_{bi} \times (\boldsymbol{p}_{m_3} - \boldsymbol{p}_{bi}) & \boldsymbol{z}_{1i} \times (\boldsymbol{p}_{m_3} - \boldsymbol{p}_{1i}) & \boldsymbol{z}_{2i} \\ \boldsymbol{z}_{bi} & \boldsymbol{z}_{1i} & k_r \boldsymbol{z}_{2i} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{J}_j^{l_1} &= \begin{bmatrix} \boldsymbol{z}_{bj} \times (\boldsymbol{p}_{l_1} - \boldsymbol{p}_{bj}) & 0 & 0 \\ \boldsymbol{z}_{bj} & 0 & 0 \end{bmatrix} \\
\boldsymbol{J}_j^{l_2} &= \begin{bmatrix} \boldsymbol{z}_{bj} \times (\boldsymbol{p}_{l_2} - \boldsymbol{p}_{bj}) & \boldsymbol{z}_{1j} \times (\boldsymbol{p}_{l_2} - \boldsymbol{p}_{1j}) & 0 \\ \boldsymbol{z}_{bj} & \boldsymbol{z}_{1j} & 0 \end{bmatrix} \\
\boldsymbol{J}_j^{l_3} &= \begin{bmatrix} \boldsymbol{z}_{bj} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{bj}) & \boldsymbol{z}_{1j} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{1j}) & \boldsymbol{z}_{2j} \\ \boldsymbol{z}_{bj} & \boldsymbol{z}_{1j} & 0 \end{bmatrix} \\
\boldsymbol{J}_j^{m_3} &= \begin{bmatrix} \boldsymbol{z}_{bj} \times (\boldsymbol{p}_{m_3} - \boldsymbol{p}_{bj}) & \boldsymbol{z}_{1j} \times (\boldsymbol{p}_{m_3} - \boldsymbol{p}_{1j}) & \boldsymbol{z}_{2j} \\ \boldsymbol{z}_{bj} & \boldsymbol{z}_{1j} & k_r \boldsymbol{z}_{2j} \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{J}_t^{l_1} &= \begin{bmatrix} \boldsymbol{z}_{bt} \times (\boldsymbol{p}_{l_1} - \boldsymbol{p}_{bt}) & 0 & 0 \\ \boldsymbol{z}_{bt} & 0 & 0 \end{bmatrix} \\
\boldsymbol{J}_t^{l_2} &= \begin{bmatrix} \boldsymbol{z}_{bt} \times (\boldsymbol{p}_{l_2} - \boldsymbol{p}_{bt}) & \boldsymbol{z}_{1t} \times (\boldsymbol{p}_{l_2} - \boldsymbol{p}_{1t}) & 0 \\ \boldsymbol{z}_{bt} & \boldsymbol{z}_{1t} & 0 \end{bmatrix} \\
\boldsymbol{J}_t^{l_3} &= \begin{bmatrix} \boldsymbol{z}_{bt} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{bt}) & \boldsymbol{z}_{1t} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{1t}) & \boldsymbol{z}_{2t} \times (\boldsymbol{p}_{l_3} - \boldsymbol{p}_{2t}) \\ \boldsymbol{z}_{bt} & \boldsymbol{z}_{1t} & \boldsymbol{z}_{2t} \end{bmatrix}
\end{aligned}
$$

$$
\boldsymbol{J}_t^{e} = \begin{bmatrix} \boldsymbol{z}_{bt} \times (\boldsymbol{p}_{e} - \boldsymbol{p}_{bt}) & \boldsymbol{z}_{1t} \times (\boldsymbol{p}_{e} - \boldsymbol{p}_{1t}) & \boldsymbol{z}_{2t} \times (\boldsymbol{p}_{e} - \boldsymbol{p}_{2t}) \\ \boldsymbol{z}_{bt} & \boldsymbol{z}_{1t} & \boldsymbol{z}_{2t} \end{bmatrix}
$$