NTNU – Trondheim
Norwegian University of
Science and Technology

Developing Energy Efficient Scenarios of Work for Industrial Robot Manipulators

Kristian Breistøl

Master of Science in Cybernetics and Robotics
Submission date: July 2015
Supervisor: Anton Shiriaev, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

In industry there is a constant need to make solutions that are more efficient, reduces risk and reduces the impact on the environment. In order to achieve this we see a rapid increase in use of automatic production. Automation of industrial processes typically uses robot manipulators in operations such as drilling, welding, painting, laser cutting or assembly. In the pursuit of improved performance from industrial robot manipulators we see an increasing interest in development of optimal motion on different performance indexes such as velocity, accuracy or energy efficiency. This paper concerns energy efficient scenarios where even minor improvements in performance can lead to substantial impacts on both energy use and profit.

This master thesis includes development of a mathematical model for industrial robot manipulator IRB 1600, by ABB. This includes estimation of masses, centers of mass and inertias. Using a Nikon Metrology camera measurement system the dynamic friction of the robot is characterized, and repeated experiments indicate that friction compensation using this characterization improves performance of the robot in terms of accuracy by **25-50%** for joint 1 thru 3 for different scenarios.

Further, a scheme for development of optimal motion profiles is made, in terms of minimizing mechanical energy consumed by the robot. The motion profiles are made to follow a predefined path, and are derived through numerical optimization. Repeated experiments are performed using an inverse dynamics PD+ controller. By comparing the performance of the robot manipulator to trajectories developed using the standard motion planner (RobotStudio/Rapid), a decrease of about **7%** in consumed energy is obtained.

Sammendrag (Norwegian)

I industriell produksjon er det en konstant søken etter løsninger som er mer effektive, som reduserer risiko og som reduserer påvirkningen på miljøet. For å oppnådette ser vi en økende interesse for automasjon. Automasjon i industiell produksjon involverer typisk bruk av robotmanipulatorer som utfører alt fra drilling, sveising, lakkering, laserkutting til montering. I jakten på forbedret ytelse ved bruk av industriroboter ser vi en økende interesse for utvikling av optimale bevegelsesmønstre i form av raskere, mer nøyaktige eller mer energieffektive baner. Denne masteroppgaven tar for seg energieffektive senarioer hvor selv små ytelsesforbedringer kan gi betydelige resultater i form av både strømforbruk og profitt.

Denne masteroppgaven inkluderer utviklingen av en matematisk modell for industriroboten IRB 1600, produsert av ABB. Modelleringen innebærer estimering av masser, massesentere, og treghetsmomenter. Ved å bruke et Nikon Metrology målesysten basert på kameramålinger har kinetisk friksjon blitt modellert. Ved hjelp av en serie med forsøk er det blitt vist at friksjonskompensering ved foroverkobling av denne karakteristikken gir forbedret nøyaktighet langs en forhndsdefinert bane på **25-50%** for ulike senarier.

Videre er det laget en strategi for å finne optimale hastighetsprofiler med tanke på forbrukt mekanisk energi langs en forhåndsdefinert bane. Disse hastighetsprofilene er kalkulert ved hjelp av numerisk optimering. Det ble så kjørt en serie med forsøk hvor hastighetsprofilene ble implementert med en PD+-kontroller basert på invers dynamikk. Ved åsammenligne energibruken fra banen laget med numerisk optimering med en bane laget med robotprodusentens egen baneplanlegger, ble det vist at banen fra ekstern kontroller reduserte energibruken med omtrent **syv prosent**.

Preface

This master thesis presents the work from the final semester of a Master of Science education at the Norwegian University of Science and Technology (NTNU) at the Department of Engineering Cybernetics. This report is a diploma thesis carried out during the spring of 2015.

I would like to thank my supervisor Professor Anton Shiriaev for thorough guidance and interesting discussions during the work. I would also like to thank postdoctoral fellow Leonid Paramonov for support and highly intersting discussions during the project both regarding teoretical concerns and practical things in the robotics laboratory. I would also like to thank PhD-candidate Stepan Pchelkin for helping me out with practical questions regarding the lab configurations and for transfering valuable experience from robot modelling and providing valuable code. I would like to thank PhD-candidate Sergey Kolyubin for supervision in the laboratory and introduction to the Nikon Metrology equipment. Thanks to Professor Anders Robertson at Lund University for looking trough my program when I was struggling with external control. Finally, I would like to thank Eirik Lie Strandbråten for cooperation during the work on the thesis and for good company during hundreds of hours working in the laboratory.

Trondheim, July 3, 2015

Kristian Breistøl

Table of Contents

List of Tables

List of Figures

Part I

Introduction and Equipment

Chapter 1

Introduction

1.1 Problem Description

The problem description as stated by Professor Anton Shiriaev is given as follows.

The project is aimed at developing specific trajectories for industrial robot manipulators that fulfill the tasks the robots are expected to be performed by a scenario and that require the minimum or close to minimum amount of the mechanical energy spent by the robots to complete the assignment. It is supposed that such optimal trajectories can be found through properly organized model based numerical optimization. The validation of the analytical results will be done based on a series of experiments planned for standard curve subroutines for ABB IRB 1600.

1.2 Motivation

In industry there is a constant need to make solutions that are more efficient, reduces risk and reduces the impact on the environment. It order to achieve this we see a rapid increase in the use of automatic production. Typical tasks for industrial robots involves drilling, welding, laser cutting, grinding, painting or assembly. In automatic production it is sometimes desirable to replace custom machines by industrial robot manipulators such as the ABB IRB1600. The robot manipulators are usually cheaper, more flexible and occupies far less space in a production environment compared to customized machines. The advantage of the custom machines on the other hand is better accuracy, especially at higher

velocities. If able to improve the performance of standard industrial robot manipulators by applying sophisticated motion planning and controls it would widen the number of possible applications for this kind of machines signficantly.

As the use of robot manipulators in industrial production increase world wide the energy consumption by these machines increase with it. Today there are over 190,000 industrial robots in operation in the US alone. Each robot is typically consuming at least 300 kWh per day yielding an expense of $15 billion annually in energy only. By decreasing the energy consumption by just a fraction for each manipulator the total energy savings could be substantial. For instance a reduction of 1% in the energy consumption of US industrial robots would yield the following [5].

+ A reduction in energy consumed by 200,000,000 kWh.

+ Electrical cost reduction of $15,000,000.

+ Greenhouse gas emissions reduced by 140,000 tons.

+ SO_2 and NO_x emissions reduced by 720 tons.

Another aspect that would motivate for development of energy efficient scenarios is that one would expect a energy efficient motion to apply as little torque as possible to the motors of the robot, hence minimizing the wear on the industrial manipulator. Replacing industrial equipment such as robots is an expensive and comprehensive process. Thus, it is an important aspect that energy efficient robot motion can also extend the life span of components in an industrial plant.

1.3 Objectives

The overriding objective of this thesis is to develop specific trajectories for the industrial robot manipulator IRB1600 that fulfill the tasks that the robots are expected to perform by a scenario and that require the minimum or close to minimum amount of mechanical energy spent by the robot to complete the assignment. In order to do this an accurate mathematical model of the system and its properties is needed.

It is suggested to model and compensate for friction at constant velocity in the modeling.

This is to be done by making a friction-velocity mapping of data from several experiments. It is further suggested to perform necessary experiments in order to verify whether feed forward compensation of estimated friction forces improves the performance of the robot in terms of accuracy.

The work on this master thesis is very practically oriented and includes numerous experiments in the industrial robotics laboratory at NTNU. Both training with the equipment/software and the modeling, including the friction compensation, is done in collaboration with Eirik Lie Strandbråten.

Development of robotics system can be broadly divided into two areas. Motion planning and motion control. This thesis primarily concerns the planning and will to a greater degree use controllers made by previous students or employees either at NTNU or at Lund Underversity. The trajectories are applied with a inverse dynamics PD+, which is described in greater detail later in this thesis.

1.4 Outline of Report

the report is divided into three main parts plus appendix. The first part is introduction and Equipment. The second part concerns ingredients in solutions, while the third main part addresses development of energy efficient scenarios.

1.4.1 Introduction and Equipment

This part consists of two chapters. The first chapter presents the problem, objective and outline of the report. The second chapter describes the equipment and software used in this work.

1.4.2 Ingredients in Solutions

Part two includes three chapters. Kinematics where the ingredients in the modeling of the robot manipulator is presented. Modelling of friction forces which deals with friction modeling and validation of the friction model. The third chapter is about the kinematics in path planning.

1.4.3 Development of Energy Efficient Scenarios

This third part consists of four chapters. The first chapter addresses how the optimal motion profiles are calculated. The second consists of the result and discussion on the main results. The last two chapers are conclusion and suggestions for further work.

1.4.4 Appendix

The appendix includes some results from validation of the friction model, which is considered to be too comprehensive to be included in the main part of the report. A brief description of the digital attachments to this thesis can also be found in the appendix.

Chapter 2

Equipment

The hardware can be broadly divided into two parts. The IRB 1600 robot manipulator and the Nikon Metrology camera system. As for software, a lot of different programs have been used to perform tasks of different magnitude. Thus, only the most important software used in the work on this project is described to detail here.

2.1 ABB IRB 1600 Robot Manipulator

The robot manipulator used in the experiments of this thesis is an IRB 1600 by ABB. It can handle a payload of 6kg and has a reach of 1200mm. It has six revolute joints. Serial robots, such as the IRB 1600, usually have six joints, because it requires at least six degrees of freedom to place a manipulated object in an arbitrary position and orientation in the workspace of the robot. The first three joints are used to achieve desired position for the end effector, while the three outermost joint are used to achieve desired orientation. There are encoders in the six joints, that may be used to measure position (angle) of each of the six joints.

ABB robots considered a very stiff and rigid robot compared to e.g. the robots made by the Japanese producer KUKA. On the other hand the cast iron that the links are made from makes the robot manipulator rather heavy. The robot is very flexible in terms of possible applications. It can be mounted in a numerous sort of environments, on walls, upside down and be used to perform various tasks such as welding, painting, mounting and milling. It has a control interface called a Flex Pendant, which is used to run programs on the robot made with ABB's dedicated software RobotStudio/Rapid. It is also possible

to run the robot using external control, which is to be done in this thesis. External control will be further discussed later in the report.

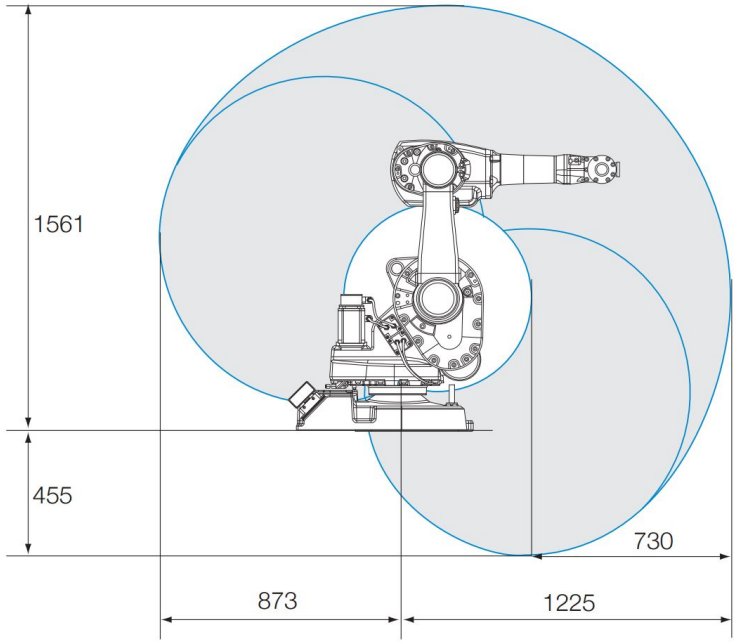


Figure 2.1: ABB IRB 1600 [1].

The IRB 1600 and its reachable workspace in the x, z-plane can be seen in figure 2.1.

2.2 Nikon Metrology Camera System

The metrology equipment consists of several parts and it is used to more accurately measure the position of the robot end effector. The encoders in the joints of the robot are not able to detect certain phenomenas such as backlash in gears, elasticity in the links and effects from static friction. Hence, what is monitored from encoder measurements does not necessarily correspond to the actual motion to a sufficient extent. Therefore it is suggested to use external measurements from Nikon metrology equipment to monitor the robot motion more accurately.

The biggest piece of equipment is the K-610 camera. It is a 3D camera with three lenses, able to track the position of LEDs in space at very high precision. Each lens captures the LEDs position in one plane, and the combined accuracy in a given point is $37\mu m$ [6].

By clustering three LEDs it is possible to define a reference frame, and measure relative position and orientation of the defined coordinate system. This can be seen in figure 2.2.

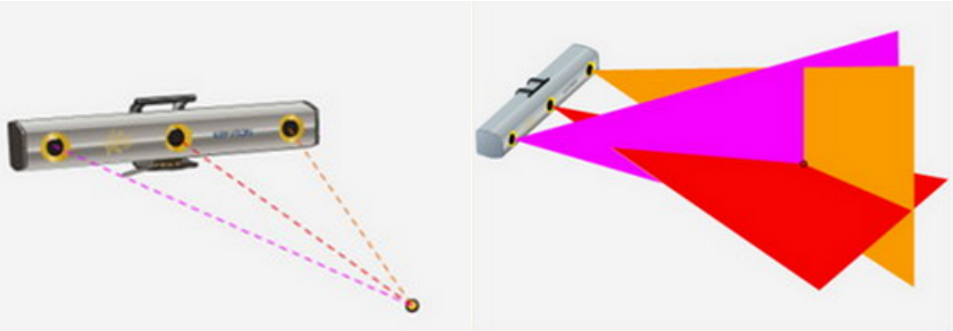


Figure 2.2: LED picked up by camera system [2].

It is desirable to measure the position of the TCP (Tool Center Point) frame of the robot relative to the base frame. Two clusters of at least three LEDs are therefore needed. For each static frame it is possible to define a dynamic frame attached to its respective static frame. For the end effector this is necessary because the frame will be moving during the different experiments. For the base frame it is not strictly necessary, but it is done to get rid of sensitivity in movement of the camera.

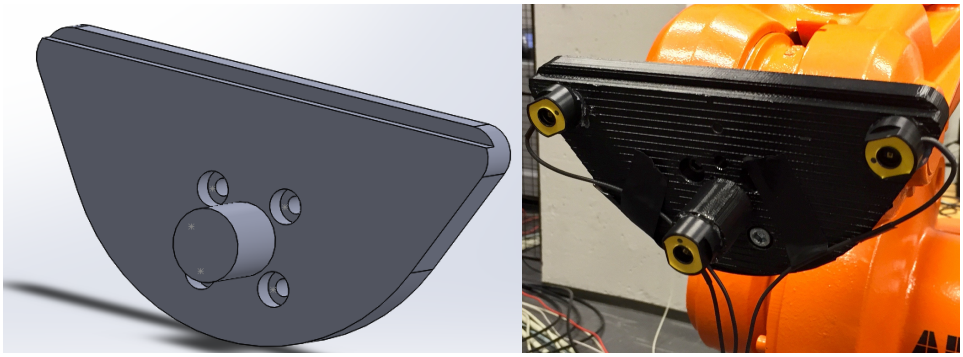
As the metrology system is highly sensitive to changes in the surrounding environment it needs to be calibrated regularly. Due to the precision of this kind of measurements small changes in lightning, temperature, humidity etc. can result in severe errors/noise on the measurement data. Thus, it is crucial to be careful when performing experiments.

One other aspect that is important to take into consideration is that all LEDs needs to be visible for the camera at all times during experiments. If the end effector of the robot is oriented away from the camera or the robot itself shades the LEDs from being visible for the camera the measurement system will not be able to measure its position and orientation.

By using the Nikon Metrology camera system LEDs can be attached to the robot, and three or more LEDs can define a coordinate system. By doing this it is possible to define a dynamic frame attached to one of these static frames, and it is possible to measure relative position and orientation between several different dynamic frames. It is also possible to

measure the position of a dynamic frame relative to a static frame, but this indroduces sensitivity i terms of movement of the measuring equipment.

In this experiment the base frame of the robot manipulator is defined by attaching three LEDs to the socket of the robot, and attaching a dynamic frame to the static frame defined relative to these three LEDs. Regarding the end frame, placement of the LEDs is not as straigh forward as the area of the flange is limited. Therefore it is decided to develop a mounting tool for attachment of the LEDs. This tool is designed using commercial software (SolidWorks), and manufactured in plastic using a 3D printer. Different considerations has to be made in the design process. The tool needs to be efficient in terms of the amount of plastic that goes into the production process. In addition it has to be incredibly rigid not to ruin the measurements. It is desirable to get a decent spread between the different LEDs while not being wider than 20cm, constrained by the printer. In order to simplify the process of defining frames a trace was addet along the y-axis (defined in RobotStudio) of the end effector. The design is shown in figure 2.3 below.



(a) Model from SolidWorks **(b)** Mountig tool with LEDs

Figure 2.3: LED mounting tool

One LED is glued on to the cylinder, while the other two are attached in the upper corners.

There are several different ways to define a coordinate frame. The simplest way is to define a two of the coordinate system axis from different features, and software will align the third axis. These features can be pretty much anyhing from a line, a cylinder to a plane. The features are defined by measuring a bunch of points with a measuring device called a Space Probe. This tool needs to be calibrated. As does the camera itself. Due to the extreme precision of the measurements, the features of the camera changes with changing temperature. Thus, it needs to warm up for approximately 30 minutes before being ready

to use. Occasionaly it also needs to be calibrated. It is time consuming, but very important to do this in order to avoid poor measurements due to changed measuring conditions.

2.3 Software

Numerous programs are used in the work on this thesis. In definition of frames ABB's own software for programming their robot manipulators is used, RobotStudio. This generates code in a language called Rapid. Control schemes implemented in RobotStudio has, allegedly, good performance in specific points, but the error along paths is from experience significantly larger. It is therefore used in the definition of frames. By jogging the robot in known positions using a Rapid program it is possible to make a mapping, and define a frame from this mapping. The definition of frames is a thorough process, and it might introduce severe error to the measurements if done wrong or careless.

In addition to MATLAB with Simulink and the External Control interface, Nikon Metrology Software is used in the camera measurements. In addition programs such as SolidWorks, CAD and Maple etc. are used, but it is considered unnecessary to describe these programs more thorough.

2.4 External Control

It an attempt to achieve improved control compared to the standard ABB controls (i.e. RobotStudio, Rapid) use of External Control is suggested. Using the external control extension it is possible to monitor and control both references and measurements for position, velocity and torques sent from the main controller to the axis controller. This customized controller consists of non commercial software applied only a few places worldwide (including NTNU in Trondheim, Norway and LTH in Lund, Sweden).

This system bypasses some of ABB's integrated security settings. Thus, it is crucial to ensure that the control signals sent to the axis controller via external control makes the robot motion robust and stable. A procedure for testing robustness for a scenario developed for external control can be found in [3]. If this is not considered the robot motion is uncertain and the manipulator can for instance collapse and possibly cause material damage, or even worse, harm humans.

An overall schematic illustration of the signal routing can be seen in figure 2.4 below.

The external control is operated trough an interface called Opcom with four modes corresponding to different signal routing which can be seen in this figure.

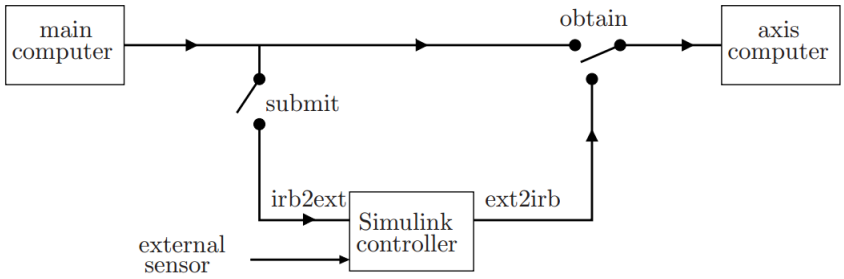


Figure 2.4: Schematic of reference value path from main to axis computer [3].

The opcom interface has four different modes; Unload, Load, Submit and Obtain. The properties of the different modes are as follows

i. **Unload:** In this state the main computer is connected directly to the axis computer, and no program is loaded in the Simulink controller.

ii. **Load:** In this mode the main computer is still connected directly to the axis computer without interference from the external controller, but the Simulink program is loaded into opcom.

iii. **Submit:** In this state the switch denoted ”submit” is switched, and the program can read the signals from the main computer. In this state it is possible to test the logged signals to ensure that the program performes the expected motion control and to ensure stability and robustness.

iv. **Obtain:** When the system enters Obtain mode the signals is routed trough the Simulink controller rather than directly from the main computer to the axis computer. In this mode the experiment can be performed.

The signals from both input to and output from the robot can be logged using an external program (e.g. KST). An overview of how the external control is designed can be seen in figure 2.5.

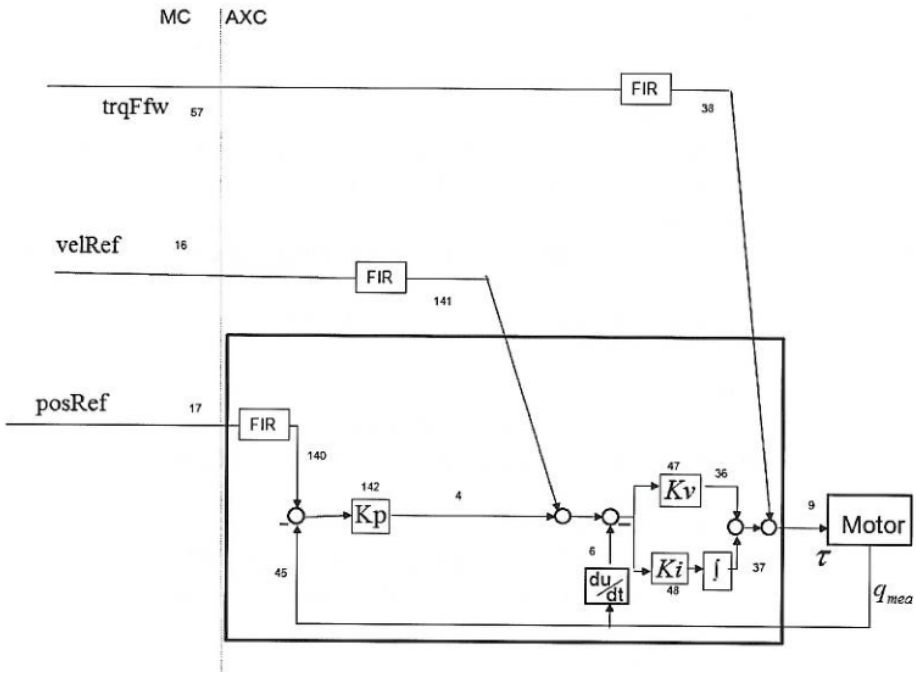


Figure 2.5: Robot control schematics [4].

This is a special kind of controller implemented with a Simulink interface. The control scheme is explained to greater detail in chapter 6.4.

Part II

Ingredients in Solution

Chapter 3

Kinematics

Measurements from the camera are given in cartesian coordinates, while the measurements from the encoders in the robot are in joint space. To be able to compare the two in each domain both forward and inverse kinematics for the robot are necessary. A structured way to define the different reference frames is also needed and is presented in this chapter.

Throughout this project a simplified robot model is used. In this simplified model only the first three joints are actuated. This is due to the scope of this thesis only including the position, not orientation, of the center of the TCP (Tool Center Point) frame. Thus, we get

$$q_{4..6} = 0, \quad \dot{q}_{4..6} = 0, \quad \ddot{q}_{4..6} = 0. \tag{3.1}$$

The kinematics of the simplified model is described to greater detail in chapter 5.

3.1 The Denavit-Hartenberg Convention

The Denavit-Hartenberg convention is used for defining frames. This is a common way to systematize the frames and their relations, known from e.g. [7].

By the DH convention the displacement between two frames is given by four basic transformations. The resulting transformation matrix consists of a rotation matrix, $\mathbf{R}_i$, and a translation vector $\mathbf{O}_i$. For convenience a simplified notation is used. s_{θ_i} denotes $\sin(\theta_i)$,

while c_{θ_i} denotes $\cos(\theta_i)$ etc.

$$A_i = \text{Rot}_{z,\theta_i}\text{Trans}_{z,d_i}\text{Trans}_{x,a_i}\text{Rot}_{x,\alpha_i} \tag{3.2}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{R}_i & \mathbf{O}_i \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{3.3}$$

The robot configuration is given in figure 3.1 below.

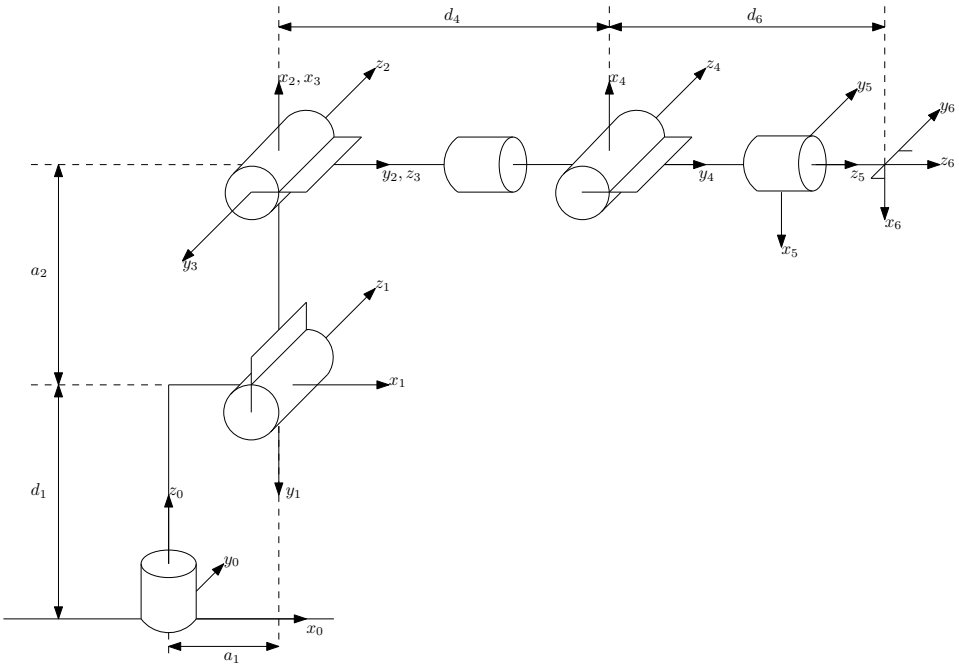


Figure 3.1: Translations and rotations from base frame to end frame for IRB 1600

Its corresponding DH parameters are given in Table 3.1.

Table 3.1: DH Parameters

Link i	a_i	α_i	d_i	θ_i^*
1	a_1	$-90°$	d_1	θ_1^*
2	a_2	0	0	$\theta_2^* - 90°$
3	0	$-90°$	0	θ_3^*
4	0	$90°$	d_4	θ_4^*
5	0	$90°$	0	$\theta_5^* + 180°$
6	0	0	d_6	θ_6^*

By inserting lengths from the datasheet the following DH parameters are retrieved.

Table 3.2: DH Parameters with parameters from data sheet [1].

Link i	$a_i\ [mm]$	$\alpha_i\ [deg]$	$d_i\ [mm]$	$\theta_i^*\ [deg]$
1	150	-90	486.5	θ_1^*
2	475	0	0	$\theta_2^* - 90$
3	0	-90	0	θ_3^*
4	0	90	600	θ_4^*
5	0	90	0	$\theta_5^* + 180$
6	0	0	65	θ_6^*

From the DH parameters the following displacement matrices are retrieved

$$A_1 = \begin{bmatrix} \cos q_1 & 0 & -\sin q_1 & a_1 \cos q_1 \\ \sin q_1 & 0 & \cos q_1 & a_1 \sin q_1 \\ 0 & -1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \sin q_2 & \cos q_2 & 0 & a_2 \sin q_2 \\ -\cos q_2 & \sin q_2 & 0 & -a_2 \cos q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} \cos q_3 & 0 & -\sin q_3 & 0 \\ \sin q_3 & 0 & \cos q_3 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} \cos q_4 & 0 & \sin q_4 & 0 \\ \sin q_4 & 0 & -\cos q_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} -\cos q_5 & 0 & -\sin q_5 & 0 \\ -\sin q_5 & 0 & \cos q_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} \cos q_6 & -\sin q_6 & 0 & 0 \\ \sin q_6 & \cos q_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

By multiplying A_1 trought A_6 the total transformation from the base frame to the TCP frame is given as

$$T_6^0 = A_1 A_2 A_3 A_4 A_5 A_6 \tag{3.4}$$

The symbolic solution to this multiplication is left out of this report as it is comprehensive. It is calculated with Maple, and the script can be found as attachment to this thesis.

For the simplified three degrees of freedom model we get the same matrices for $A_1$and A_2 but a different matrix, denoted A_3', for the combined link consisting of the four outer-

most links in the 6 degrees of freedom model.

$$A_3' = \begin{bmatrix} -\cos q_3 & -\cos q_6 & 0 & -(d_4 + d_6)\sin q_3 \\ \cos q_6 & -\sin q_6 & 0 & (d_4 + d_6)\cos q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

$$T_3^0 = A_1 A_2 A_3' \tag{3.6}$$

3.2 Inverse Kinematics

In addition to retrieving cartesian position from joint measurements it is desirable to derive the inverse kinematics in order to retrieve joint angles from cartesian measurements from the camera. This is considered to be more complicated than the forward kinematics and the solutions are not necessarily unique. The first three joints are considered here as we want to solve the kinematics for the simplified model.

We start by finding the first joint angle. Due to the design of the robot this angle can be calculated using a four-quadrant arctangent function with two arguments, known as `atan2`. It is important to be aware of the order of arguments in this function. There are two different conventions; (x, y) and (y, x). In this thesis the (y, x) convention known from MATLAB is adopted. Note that this convention differs from e.g. [7].

$$\theta_1 = \text{Atan2}(y, x) \tag{3.7}$$

where x, y are measurements of the TCP in the base frame. Finding the next two angles requires use of the law of cosines. The law of cosines is a geometric property given

$$c^2 = a^2 + b^2 - 2ab\cos\gamma \tag{3.8}$$

where the variables are defined as in the left part of figure 3.2.

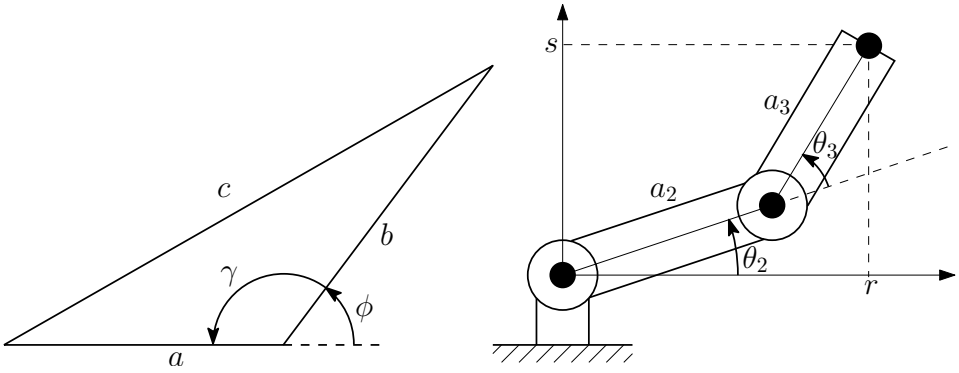


Figure 3.2: Second and third joint of the IRB 1600

This may be written

$$\cos\gamma = \frac{a^2 + b^2 - c^2}{2ab}$$

In this case it is desirable to calculate ϕ rather than γ. Thus, we get

$$\phi = \pi - \gamma \quad \Rightarrow \quad \cos\phi = \cos(\pi - \gamma) = -\cos\gamma = \frac{x^2 + y^2 - a^2 - b^2}{2ab}$$

The corresponding equation for the measures on the right subfigure of figure 3.2 becomes

$$\cos\theta_3 = \frac{r^2 + s^2 - a_2^2 - (a_3')^2}{2a_2a_3'}, \quad a_3' = \sqrt{d_4^2 + a_3^2}$$

Where we have

$$r = \sqrt{(x - a_1\cos\theta_1)^2 + (y - a_1\sin\theta_1)^2}$$
$$s = z - d_1$$

and a_2 is defined explicitly. From this θ_3 can be calculated using `acos`. Notice that $\pi/2$ needs to be subtracted as zero position for the third joint is rotated by $\pi/2$. Once the third joint angle is calculated the second joint angle can be retrieved. This is done by adding the negative angle between the second joint to the end point (joint four in this case) with θ_3, which is the angle of joint three relative to the second link. The second joint, θ_2, is calculated from the following formula

$$\theta_2 = -\text{Atan2}(s, r) + \cdots$$
$$\cdots + \text{Atan2}\left(\sqrt{d_4^2 + a_3^2}\sin\left(-\theta_3 - \frac{\pi}{2}\right),\ a_2 + \sqrt{d_4^2 + a_3^2}\cos\left(-\theta_3 - \frac{\pi}{2}\right)\right) + \frac{\pi}{2}$$

3.3 Estimation of Inertias and Mass Centers

To be able to calculate the required torque to control the robot manipulator at given angles, angular velocities and angular accelerations the standard equation of robot dynamics needs to be derived. The derivation of this equation can be found in most textbooks regarding robot dynamics, for instance [7], and is given as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{3.9}$$

By including the modeled friction this can be written

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \tau_f = \tau \tag{3.10}$$

Estimates of both center of mass and inertia matrices for the respective links are needed. As the simplified model is used the third link and its corresponding center of mass and inertia matrix is calculated from the four outermost links combined. Link three thru six is treated as one body, where only joint three is actuated. The producer of the industrial manipulator does not announce the centers of mass and corresponding inertia. Thus, they need to be estimated. As the internal construction of the links are not available it is uncertain where cavities, gears, wiring etc. are placed within the links. It is therefore suggested to assume that there is close to uniform distribution of mass in the links. The producer claims that the manipulator has a mass of 250 kg. There are different models of this manipulator with different arm lengths and different feasible payload. The producer claims that they all are the same mass. This does not seem very accurate, but it forms the best possible basis for estimating the mass properties of the manipulator. There exists more sophisticated and probably more accurate ways of estimating inertias and centers of mass, but they are considered to be too complicated and time consuming for the scope of this thesis. From experiments with the robot dynamics model it will be possible to tell if the obtained estimates seem reasonable.

Although the producer does not announce the mass properties of their robot manipulators they provide CAD models of the different robots to be downloaded from their website [8]. Commercial software (SolidWorks) is thereby used to estimate mass centers and inertia matrices. The mass of the different parts are scaled in order for the entire robot to have the same mass density and the total mass of 250 kg, as stated in the data sheet. The following masses and center of masses are obtained for the different joints.

Table 3.3: Masses and mass centers of the three links in the simplified robot model

Link i	Mass $[kg]$	Center of mass $[mm]$		
		x	y	z
1	104.3933	52.0	343.3	-11.9
2	20.4521	201.9	-0.2	182.8
3	50.5887	-11.1	15.3	96.8

Intuitively this seem like reasonable results. From this estimates the base should have a mass of approximately 75 kg, which seem reasonable. The mass centers are defined as the distance from the origin of frame $i-1$ to the center of mass of link i in the coordinate system of frame i.

The inertia matrices are calculated directly from SolidWorks from the following formulas

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \tag{3.11}$$

where

$$\begin{aligned}
I_{xx} &= \int\int\int \left(y^2 + z^2\right) \rho(x,y,z)\mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z \\
I_{yy} &= \int\int\int \left(x^2 + z^2\right) \rho(x,y,z)\mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z \\
I_{zz} &= \int\int\int \left(x^2 + y^2\right) \rho(x,y,z)\mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z \\
I_{xy} = I_{yx} &= -\int\int\int xy\rho(x,y,z)\mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z \\
I_{xz} = I_{zx} &= -\int\int\int xz\rho(x,y,z)\mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z \\
I_{yz} = I_{zy} &= -\int\int\int yz\rho(x,y,z)\mathrm{d}x\ \mathrm{d}y\ \mathrm{d}z
\end{aligned}$$

The obtained inertia matrices are denoted

$$I_1 = \begin{bmatrix} 0.0010778147 & 0.0000019394 & 0.000024207 \\ 0.0000019394 & 0.0010108225 & -0.0000131316 \\ 0.000024207 & -0.0000131316 & 0.0002254073 \end{bmatrix} \tag{3.12}$$

$$I_2 = \begin{bmatrix} 0.6887742967 & -0.0021534155 & -0.0000839611 \\ -0.0021534155 & 0.0529500048 & -0.0485985001 \\ -0.0000839611 & -0.0485985001 & 0.7030233365 \end{bmatrix} \tag{3.13}$$

$$I_3 = \begin{bmatrix} 0.0002494443 & 0 & 0 \\ 0 & 0.0003059863 & 0 \\ 0 & 0 & 0.0003883911 \end{bmatrix} \tag{3.14}$$

given in $g \cdot mm^2$. Based on these estimates equation 3.10 can be computed using Maple. The resulting ingredients are extensive, and can be found attached in the digital attachments.

From several experiments the model derived seem quite accurate. It does, however, seem like the gravitational part of the robot equation is a bit inaccurate. This is not too surprising as the mass estimate retrieved from the data sheet seemed very uncertain. The version of the IRB 1600 used in the work on this thesis is probably the lightest of the different configurations, as it is the version with the shortest arm, and smallest feasible payload.

Chapter 4

Modelling of Friction Forces

When applying torque on the robot joints friction forces react in the opposite direction. In order to improve the control of the robot and the motion planning it is desirable to retrieve an estimate of the friction forces acting at different joint velocities and apply a compensation to this. The friction forces acting on the joints can be divided into static and dynamic friction. This set of experiments are done in order to characterize the dynamic friction forces.

The experiments are done by running the robot one joint at the time achieving a trajectory with partially constant joint velocity. The torque at this area of constant velocity motion is then measured. Doing this for several different velocities a friction mapping can be obtained. It is desirable to make friction mapping from both measurements from the encoders in the robot joints and from measurement data from the Nikon Metrology camera. This both works as insurance the experiments are done correct (if the resulting friction mappings are somehow similar) and it may detect dynamics the encoders are unable to detect. Here only the first three joints are considered as the friction model is made for the simplified three degrees of freedom model.

The motion profiles to achieve constant velocity is obtained as stated below.

$$\dot{q}_{i,j} = \begin{cases} a_{1,j}t, & t \leq t_{1,j} \\ a_{2,j}, & t_{1,j} < t \leq \frac{T_j}{2} - t_{1,j} \\ -a_{1,j}t, & \frac{T_j}{2} - t_{1,j} < t \leq \frac{T_j}{2} + t_{j,1} \\ -a_{2,j}, & \frac{T_j}{2} + t_{1,j} < t \leq T_j - t_{1,j} \\ a_{1,j}t, & T_j - t_{1,j} < t \leq T_j \end{cases} \tag{4.1}$$

where T_j is the period of motion, $t_{1,j}$ is chosen time, $a_{1,j}$ and $a_{j,2}$ are parameters of velocity, $j = 1, 2, \ldots, M$, and M is number of experiments.

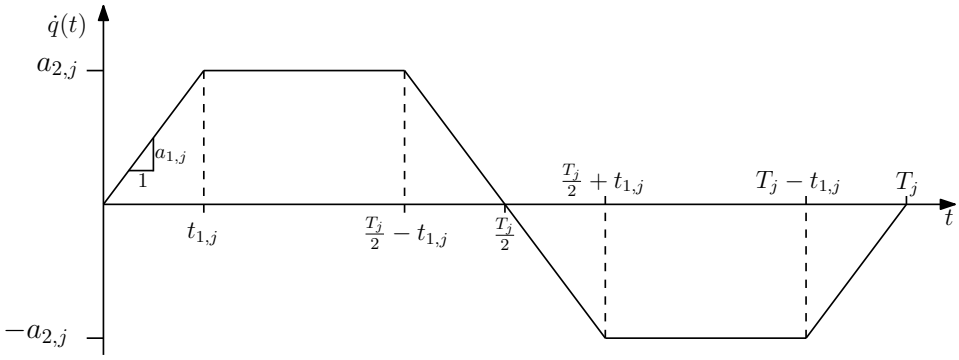


Figure 4.1: Motion profile with constant velocity for static friction experiment

In figure 4.1 the motion profile applied to obtain trajectory segments of constant angular velocity is shown.

4.1 Gravity Compensation

When the robot performs some kind of vertical movement it has to overcome gravity forces as well as friction forces in the joints. This is the case for joint two and three. As only friction forces are considered in this characterization of dynamics it is desirable to extract the contribution from gravity. This is done as stated below. We have the familiar robot equation given as

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau - \tau_f(\dot{q}) \tag{4.2}$$

with the friction force given in the form

$$\tau_f(\dot{q}) = K_1 \text{sign}(\dot{q}) + K_2\dot{q} \tag{4.3}$$

Assuming constant velocity gives

$$\left.\begin{matrix}\dot{q} = \dot{q}_c \\ \ddot{q} = 0\end{matrix}\right| \Rightarrow M(q) \cdot 0 + C(q, \dot{q}_c)\dot{q}_c + G(q) = \tau - \tau_f(\dot{q}_c) \tag{4.4}$$

$$\Rightarrow C(q, \dot{q}_c)\dot{q}_c + G(q) = \tau - \tau_f(\dot{q}_c) \tag{4.5}$$

We get two cases depending on the sign of the constant velocity.

$$(i) \quad \dot{q}_c > 0,\ \dot{q}_c = c \Rightarrow C(q, c)c + G(q) = \tau_+ - \tau_f(c) \tag{4.6}$$

$$\begin{aligned}(ii) \quad \dot{q}_c < 0,\ \dot{q}_c = -c &\Rightarrow -C(q, -c)c + G(q) = \tau_- + \tau_f(c) \\ &\Rightarrow C(q, c)c + G(q) = \tau_- + \tau_f(c)\end{aligned} \tag{4.7}$$

Combining the results, assuming that the impact from gravity is cancelled out, we get

$$\tau_f(c) = \frac{\tau_+ - \tau_-}{2} \tag{4.8}$$

4.2 Friction Characteristics from Experiments

From numerous experiments friction characteristics for the first three joints are retrieved. Both from encoder measurements and from camera measurements. For each joint each experiment done with a certain velocity corresponds to one point in the friction mapping. After doing experiments with different velocities for the first three joints a higher order polynom can be fitted to the data for positive and negative velocity separately. This is done using the `fit`[9] function in MATLAB. Different orders of the curve fitting has been tried out showing minimal impact on the output, but relatively high order polynomials are chosen to ensure that important dynamics are not neglected. In figures 4.2, 4.3 and 4.4 the resulting friction mappings for joint 1, 2 and 3 are shown, respectively.

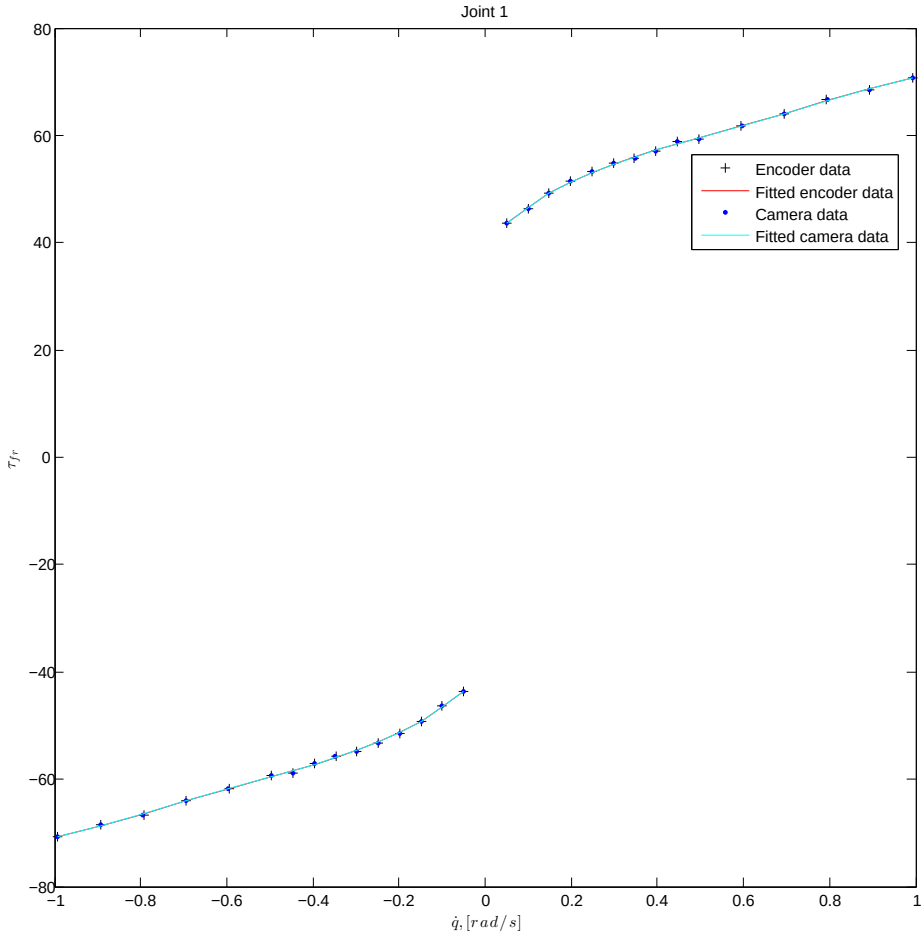
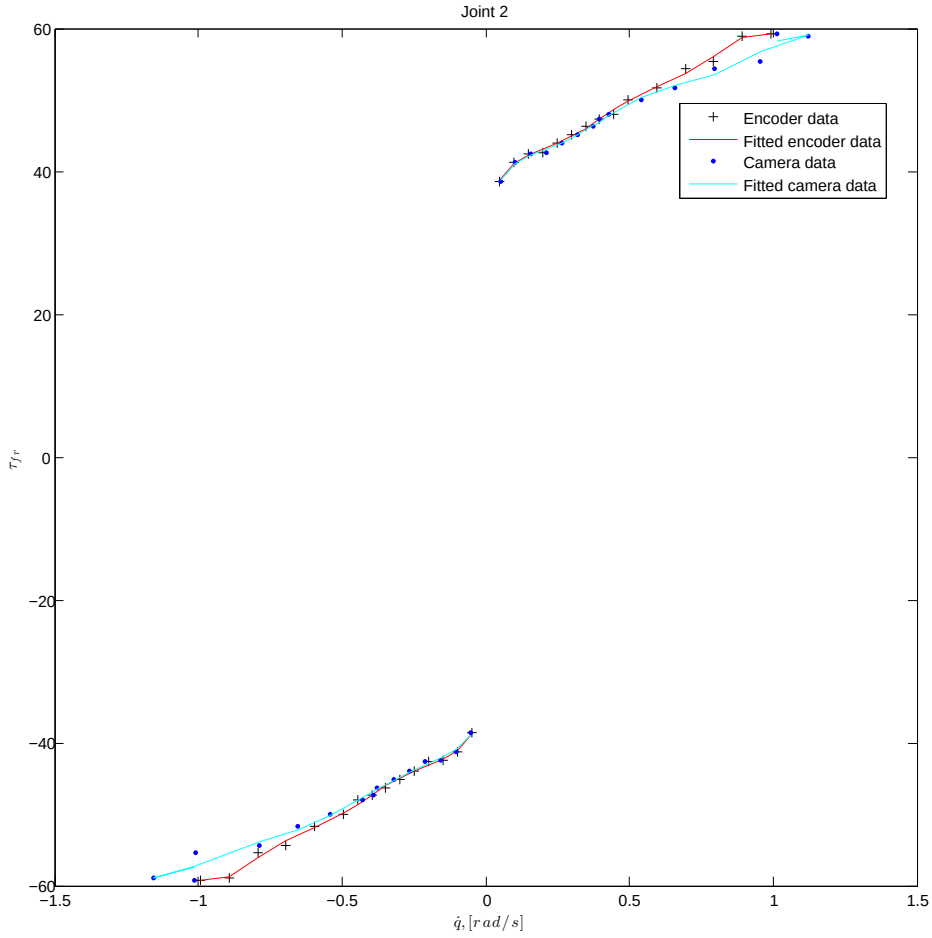


Figure 4.2: Friction Characteristics for first joint



https://open.spotif

Figure 4.3: Friction Characteristics for second joint

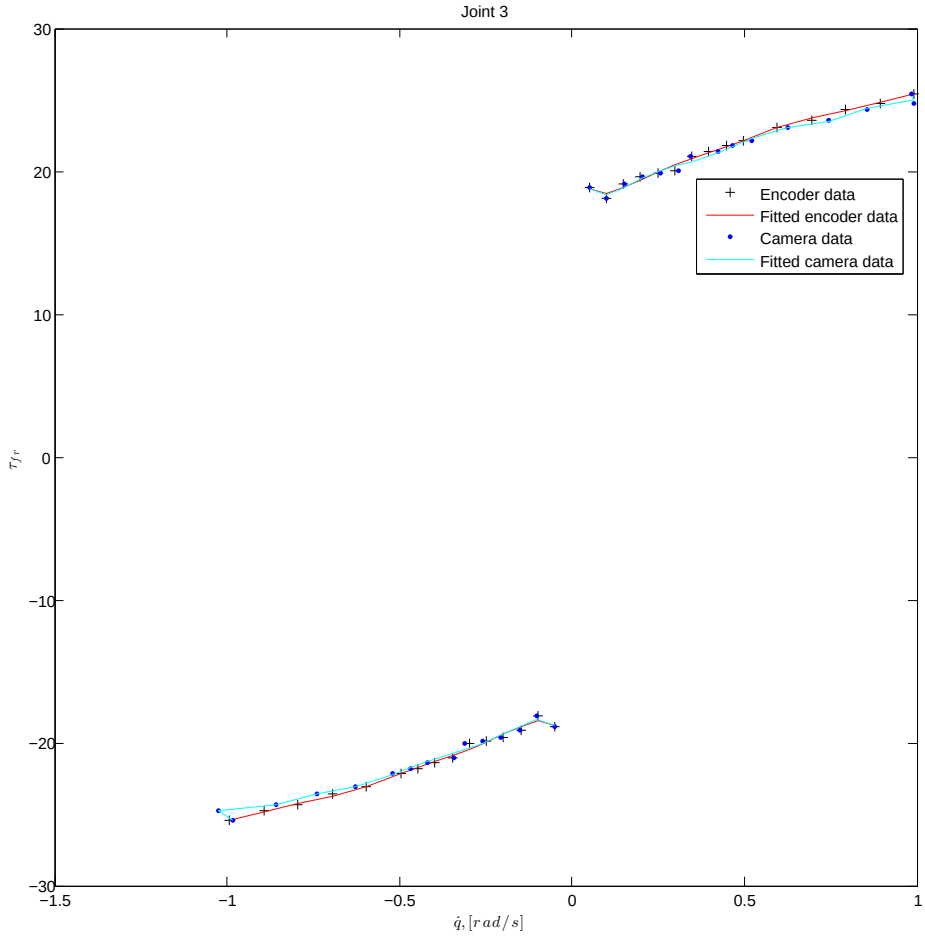


Figure 4.4: Friction Characteristics for third joint

The measurements are marked with dots and crosses for camera measurements and encoder measurements respectively. Here the fitted torque-velocity mappings are polynomials of order five to seven. These friction characteristics are made from velocity measurements on the arm side and not the other side of the gearbox where the torque is applied.

4.3 Velocity Dependent Input to Applied Feed Forward Compensation

The friction characteristics obtained should be quite certain for medium to higher velocities, while they are more uncertain about zero velocity, due to for instance dry friction/stiction. Thus, it is suggested to rely on the velocity reference as input to the calculation of compensation about zero velocity, and to rely on the measured velocity to a greater extent for higher velocities. This is implemented by a bell curve from the given formula.

$$\alpha\left(\dot{q}_{ref}\right) = \exp\left(-\frac{\dot{q}_{ref}^2}{2a^2}\right) \tag{4.9}$$

The resulting joint velocity used in calculation of the friction compensation is then given by

$$\dot{q} = \alpha\left(\dot{q}_{ref}\right)\dot{q}_{ref} + \left(1 - \alpha\left(\dot{q}_{ref}\right)\right)\dot{q}_{meas} \tag{4.10}$$

where $\dot{q}_{ref}$ denotes reference joint velocity, while $\dot{q}_{meas}$ denotes the velocity measurement. Different values for a are tried. This denotes the standard deviation and $a = 1/4$ turns out to suffice. Experiments are also done with a equal to $1/8$, $1/5$, $1/3$ and $1/2$. The resulting gaussian curve takes the form of figure 4.5 with $a = 1/4$.

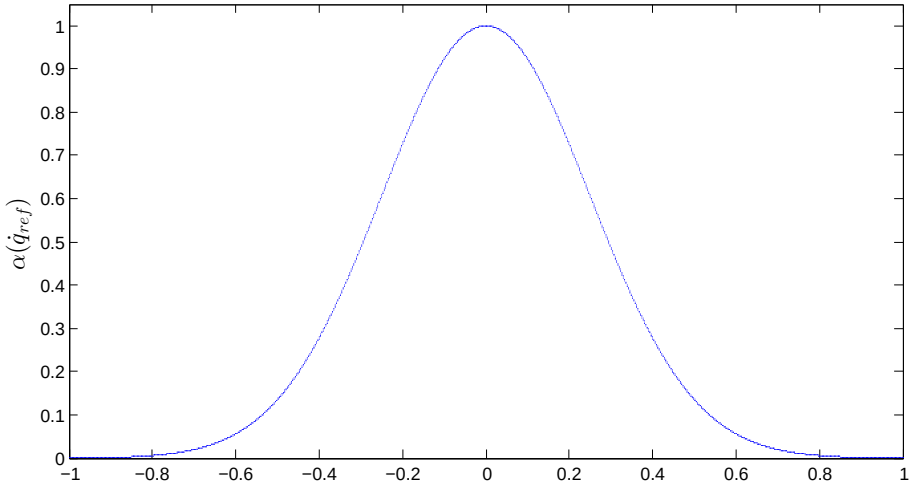


Figure 4.5: Gaussian curve with $a = {}^{1}/_{4}$

4.4 Single Joint Validation of Friction Model

In order to validate whether the friction compensation actually improves the performance by decreasing the error along a given trajectory numerous experiments are performed. It is decided to run the robot along a sine motion profile, and monitor the path deviation at all angles. For each of the first three joints the experiments are to be done with a maximum velocity of 0.1, 0.5 and 1 $[{}^{rad}/_{s}]$. The error at any given instance of time is not of that much interest in oppose to the path error. The angle of the camera measurement is therefore calculated, and then the error is calculated in cartesian coordinates from the cartesian camera measurement to the cartesian position of the reference with the same angle. The camera has a sample period of 0.01 seconds, while the encoders of the robot manipulator has a sample period of 0.004032 seconds. This needs to be considered in comparison of the two measurements, to avoid sampling error. As the measurements are done with completely separate measuring devices they also need to be properly synchronized. As all the different configurations in terms of feed forward applied, velocity, actuated joint etc. makes the number of experiments very large only the experiments performed at a velocity of 0.5 $[{}^{rad}/_{s}]$ are stated here. The reader can find the rest of the results attached in the appendix.

The error retrieved from the experiments where only joint 1 is actuated is presented in

figures 4.6, 4.7 and 4.8.

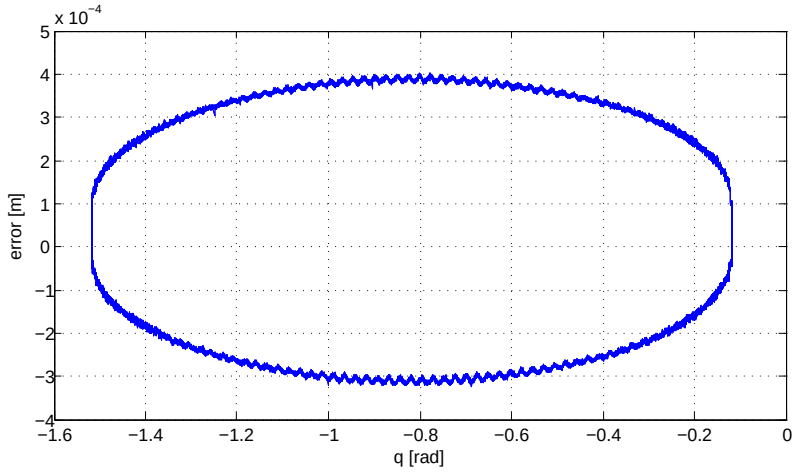


Figure 4.6: Error with no feed forward

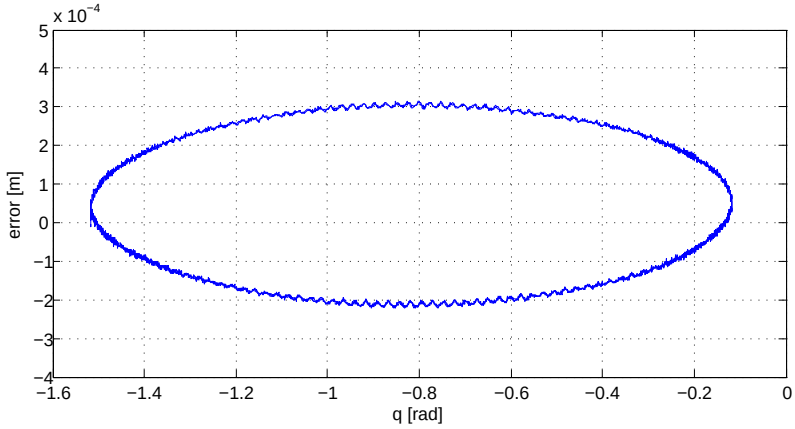


Figure 4.7: Error with feed forward from encoder measurements

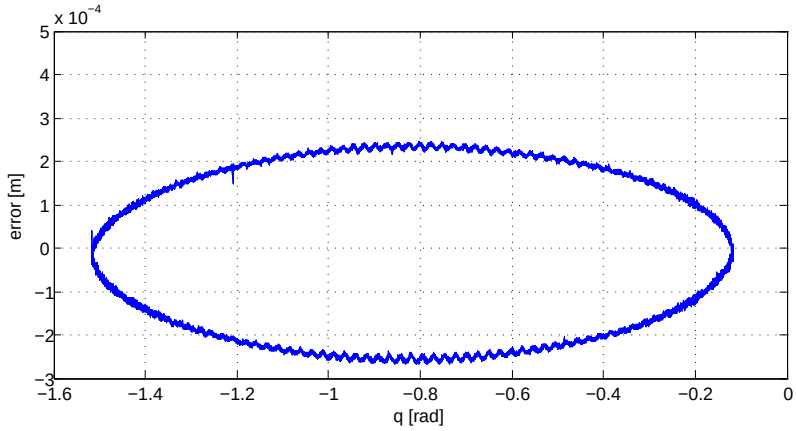


Figure 4.8: Error with feed forward from camera measurements

Here the improvement is about 40%, which is persuasive. The error from the experiment with feed forward from encoder measurements has almost the same magnitude as the error with feed forward from camera measurements.

By actuating the second joint only the results shown in figures 4.9, 4.10 and 4.11 were obtained.

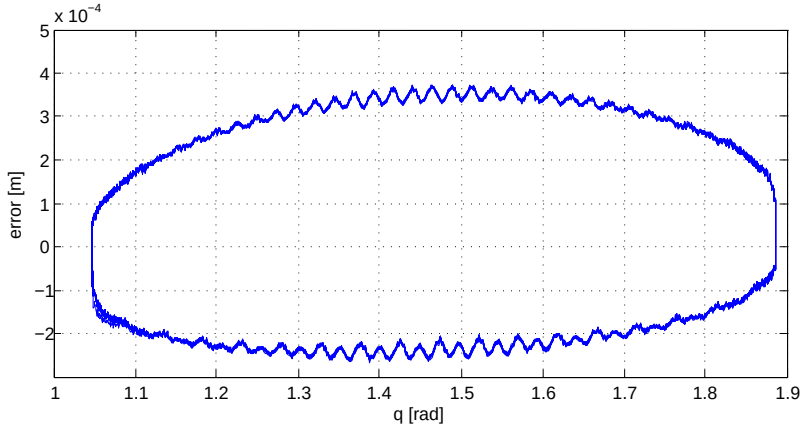


Figure 4.9: Error with no feed forward

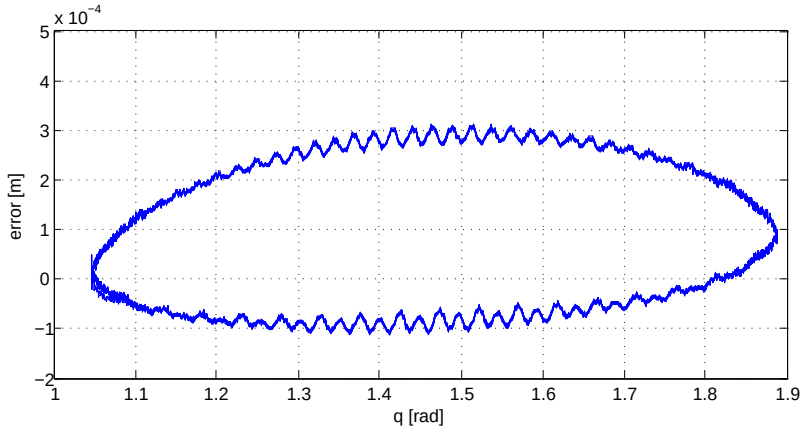


Figure 4.10: Error with feed forward from encoder measurements

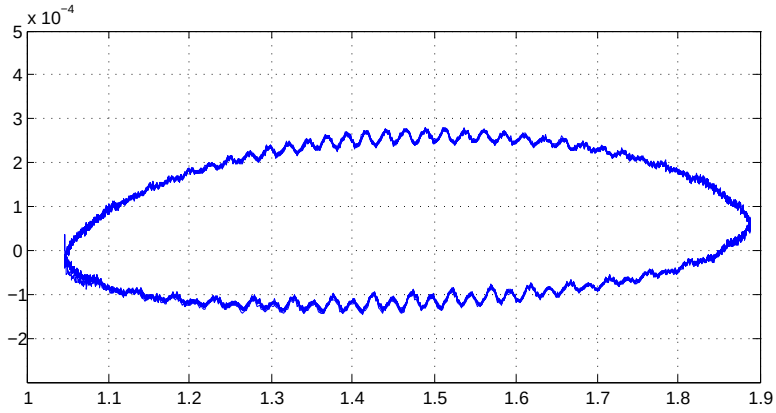


Figure 4.11: Error with feed forward from camera measurements

The improvements by use of feed forward friction compensation for the second joint is also about 40%. For this joint it is worth noticing that the feed forward compensation from camera measurements decreases the error to a greater extent than the feed forward compensation from encoder measurements.

By actuating the third joint only in the same manner the results presented in figures 4.12, 4.13 and 4.14 were obtained.

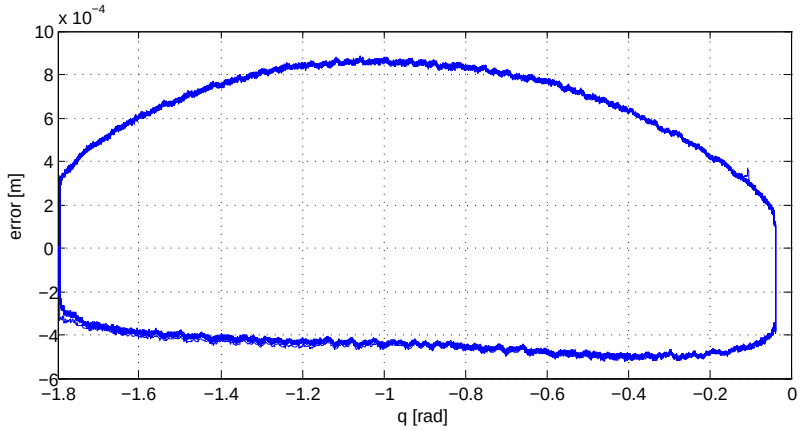


Figure 4.12: Error with no feed forward

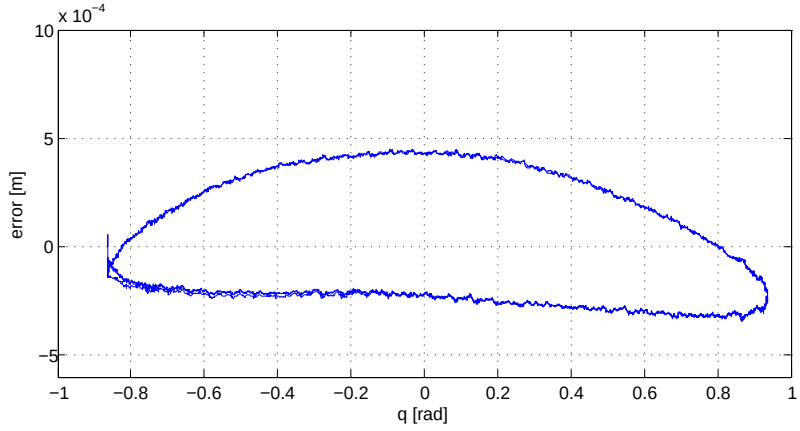


Figure 4.13: Error with feed forward from encoder measurements

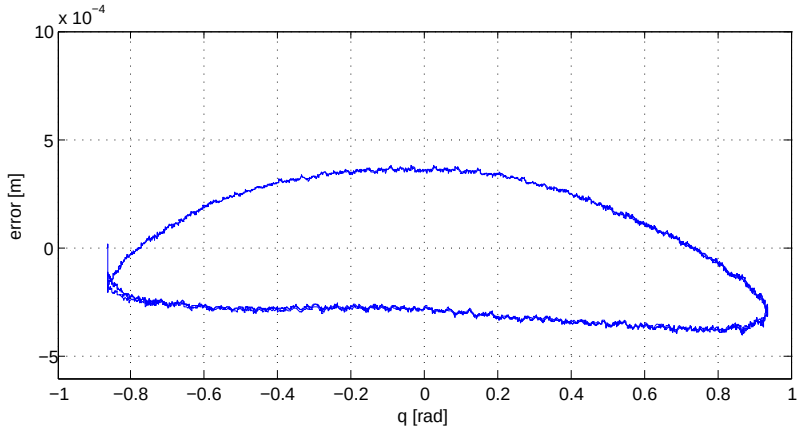


Figure 4.14: Error with feed forward from camera measurements

In this experiment the reduction of error is really persuasive with a magnitude of about 50%. This suggests that the applied feed forward compensation works as expected. This is also confirmed through repeated experiments. The fact that the friction compensation works best for the third joint might be due to this link being the outermost in the kinematic chain. It is also worth noticing that the dynamic friction for this link has about half the magnitude of the other two links.

4.5 Multiple Joint Validation of Friction Model

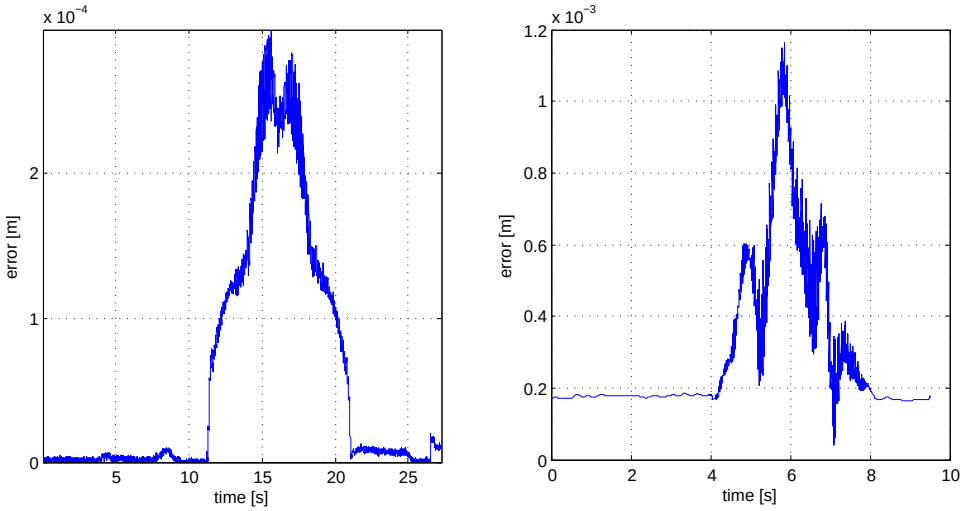
In addition to measuring the error at actuation of each joint separately it is desirable to investigate the total error along a path consisting of motion on all three joints of the simplified model. It is suggested to do this by designing a scenario where the robot runs in a horizontal circle path. The motion planning is done in a similar manner as can be seen in chapter 6, but with a simpler velocity profile from a fourth order polynomial.

In the multidimensional validation of the friction validation two scenarios are performed. The two different scenarios have different time of execution, hence different velocity along the path. In both scenarios the robot starts in the initial position $p_0 = [0 \ -17.7 \ 36 \ 0 \ 0 \ 0]$ in joint space measured in degrees. This initial position is chosen more or less arbitrarily, yet made sure to be feasible. The circular motion is applied with a radius of 20 cm. The first scenario performs the scenario in ten seconds, while the second scenario executes in 15 seconds. Both are performed with no feed forward, with feed forward from encoder

measurements, and with feed forward from camera measurements.

4.5.1 Validation of Friction Model at High Velocity

First the result from the experiment with no feed forward is presented in figure 4.15.



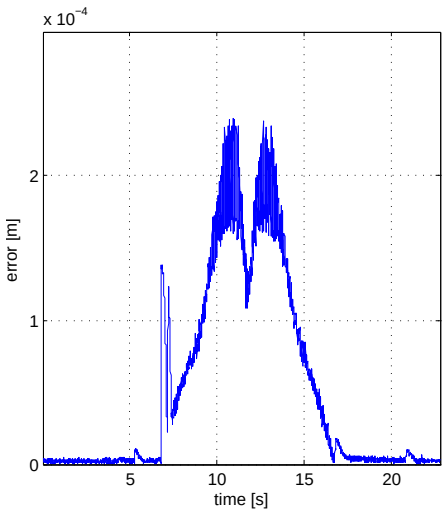
(a) Error form encoder measurements.

(b) Error from camera measurements.

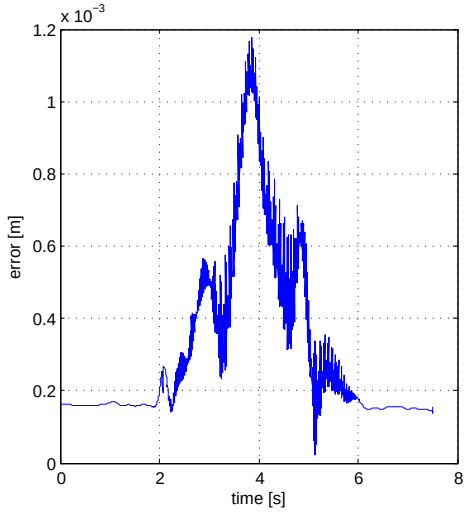
Figure 4.15: Results with no feed forward.

Notice that the measurements are done with separate equipment at different sampling rates. Thus, the time axes are not synchronized.

By applying feed forward from the characterization made from encoder measurements the result in figure 4.16 is achieved.



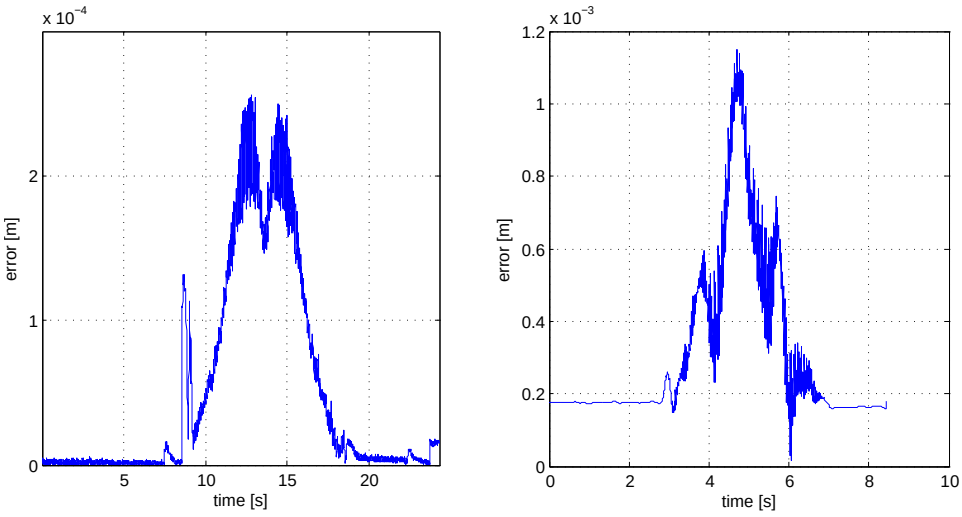
(a) Error from encoder measurements.



(b) Error from camera measurements.

Figure 4.16: Results with feed forward from encoder characteristics.

With feed forward compensation from the characteristics made from camera measurements the results presented in figure 4.17 are obtained.



(a) Error from encoder measurements.

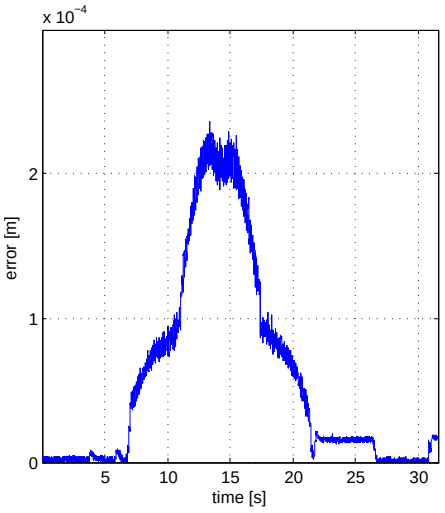
(b) Error from camera measurements.

Figure 4.17: Results with feed forward from camera characteristics.

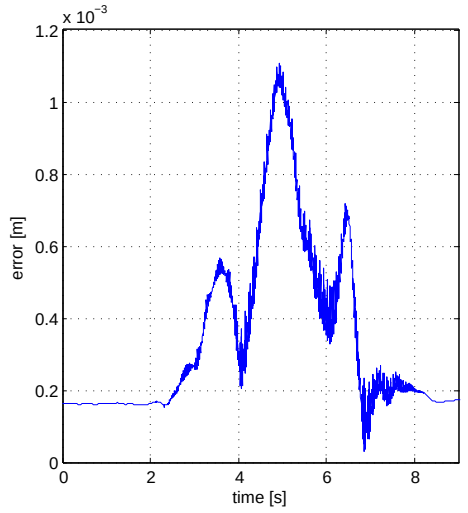
Here it is possible to see some improvement using the feed forward compensation, as expected. The magnitude of improvement is about 20%, which might be less persuasive than expected, although it it significant. The two feed forward schemes do not show any significant difference in terms of improvement. The error from the encoders is generally smaller. This is expected, as the controller uses encoder measurements as input, and certain phenomenas are not captured by the encoders.

4.5.2 Validation of Friction Model at Lower Velocity

Another set of experiments are performed to investigate the possible improvement at smaller velocity. In these experiments the execution time is set to 15 seconds. The results in figures 4.18, 4.19 and 4.20 are obtained.

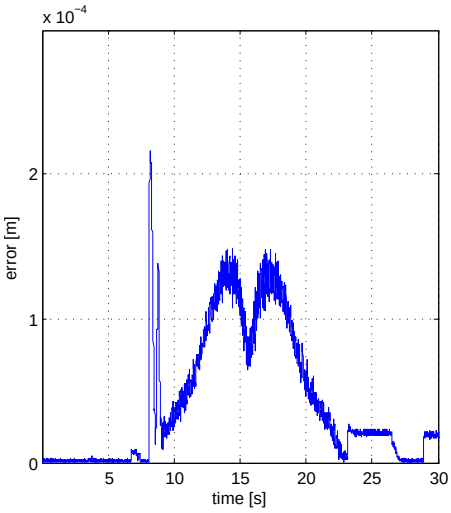


(a) Error from encoder measurements.

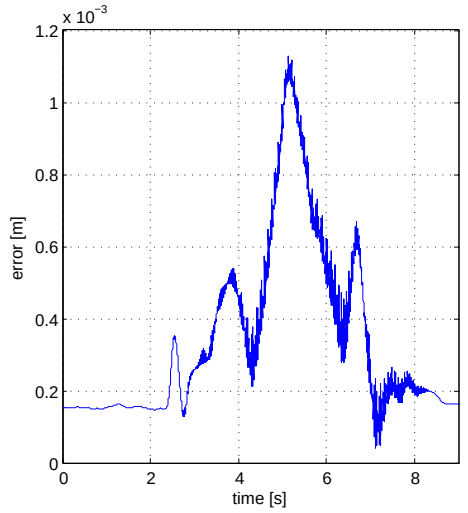


(b) Error from camera measurements.

Figure 4.18: Results with no feed forward.

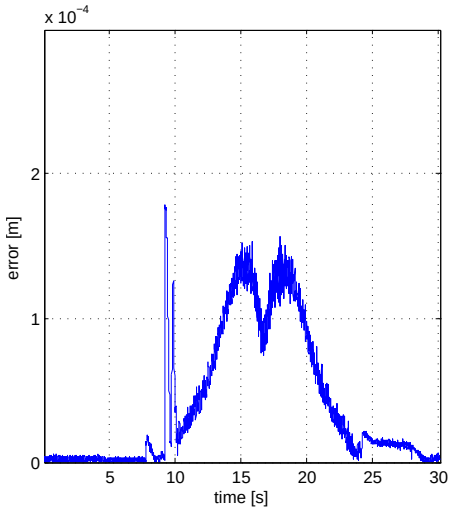


(a) Error from encoder measurements.

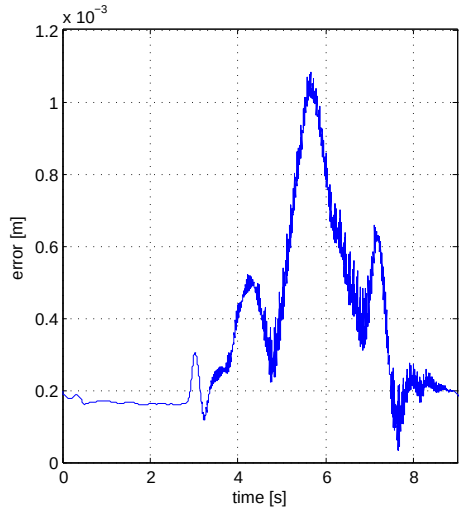


(b) Error from camera measurements.

Figure 4.19: Results with feed forward from encoder characteristics.



(a) Error from encoder measurements.



(b) Error from camera measurements.

Figure 4.20: Results with feed forward from camera characteristics.

These results suggests that the friction compensation works better for lower velocities. An improvement of about 35% is obtained, witch is satisfying.

The friction compensation is proven to improve performance for the robot manipulator in terms of accuracy. This is shown both by doing validation on separate joints and by validation on more complex paths. The experiments suggests that the friction compensation is most effective for lower, although not close to zero, velocities.

Using measurements from the Nikon Metrology camera system served as approval of the friction model obtained from encoder measurements, but it was not possible to see any hidden dynamics from these measurements.

Chapter 5

Path Planning

Paths are developed in order to investigate the performance, and to be used in the validation of the friction model derived in chapter 4. The first path to be made is for a circular motion in the x, y-plane (*i.e.* constant in z-direction of the TCP frame).

The robot manipulator has six degrees of freedom, and thus both position (x, y and z) as well as orientation (roll, pitch and yaw) can be controlled. In this thesis the orientation of the end effector is not considered. Therefore, the model is simplified to a three degrees of freedom simplified configuration, where there are no actuation on the three outermost links ($q_{4..6} = 0$, $\dot{q}_{4..6} = 0$). The simplified model can be seen in figure 5.1.

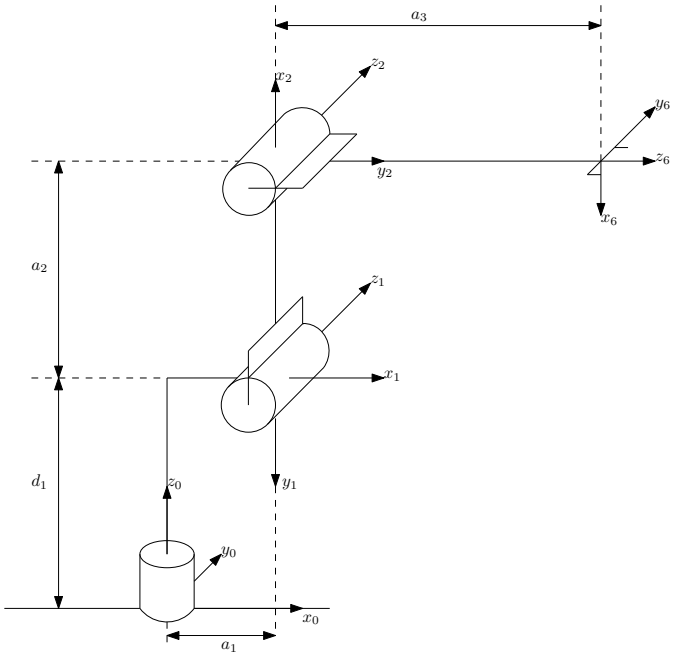


Figure 5.1: Simplified 3 DOF Model

As the velocity of the motion is not the measure of investigation in this thesis the controller is detached from explicit dependence on time, and is rather controlled by a different measure (*i.e.* θ).

We start by parametrizing a 2D circle path in the x, y-plane. For this path θ denotes the angle along the circular path.

$$P^0 = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r\cos\theta + x_c \\ r\sin\theta + y_c \\ z_c \end{bmatrix} \tag{5.1}$$

where x_c, y_c and z_c are the cartesian coordinates of the center of the circle in the TCP frame, and r is the radius of the 2D circle. x_c, y_c, z_c, r can be chosen arbitrarily and $\theta \in [-\pi, \pi]$. Although the initial position and radius can be chosen more or less at random it is important to ensure that the resulting circle path is feasible for the robot.

In joint space the path can be retrieved by solving the inverse kinematics problem as stated

in chapter 3 giving

$$q = \begin{bmatrix} q_1^*(x,y,z) \\ q_2^*(x,y,z) \\ q_3^*(x,y,z) \end{bmatrix} \tag{5.2}$$

By parametrizing the $q = q^*(t)$ as a function of the angle, θ, the following is obtained

$$q = \begin{bmatrix} q_1^*(t) \\ q_2^*(t) \\ q_3^*(t) \end{bmatrix} = \Phi(\theta)|_{\theta=\theta^*(t)} = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \\ \phi_3(t) \end{bmatrix} \Bigg|_{\theta=\theta^*(t)} \tag{5.3}$$

From this the joint velocities are given

$$\dot{q} = \Phi'(\theta)\dot{\theta} \tag{5.4}$$

and the joint acceleration is obtained by the formula

$$\ddot{q} = \Phi''(\theta)\dot{\theta}^2 + \Phi'(\theta)\ddot{\theta} \tag{5.5}$$

It is desirable to have the path not being explicitly dependent of time. Thus, we write

$$\theta = \text{Atan2}(y - y_c, x - x_c) \tag{5.6}$$

where $y_c = 0$. A graphic explanation on how the angle of the first joint is retrieved from θ can bee seen in figure 5.2 where the horizontal circle is approached from above.

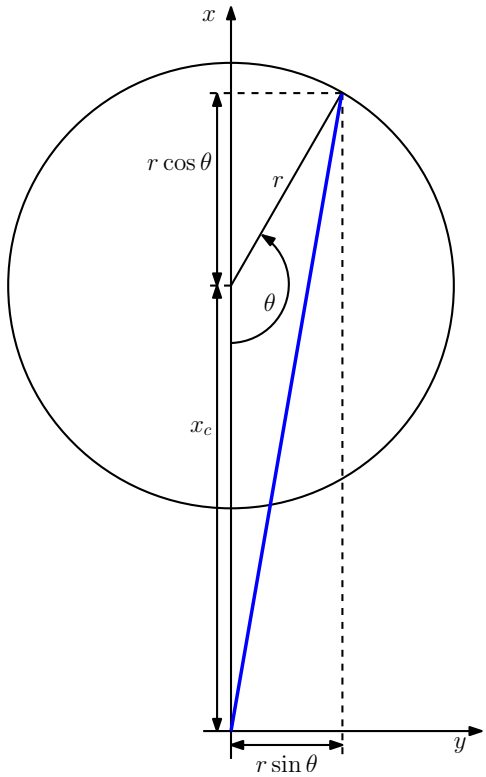


Figure 5.2: The circular path seen from above, and corresponding x and y coordinates.

The blue line denotes the arm of the robot manipulator. From this the first joint angle can be found as follows

$$q_1 = \text{Atan2}(r\sin\theta + y_c, r\cos\theta + x_c) \tag{5.7}$$

The geometry of the motion can be seen in figure 5.3 below.

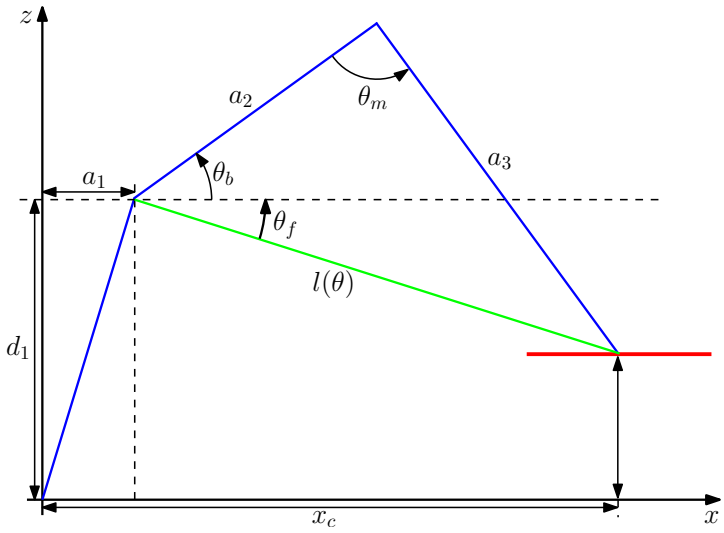


Figure 5.3: Robot geometry seen in the x, z-plane

Here The red line denotes the circle path seen in the x, z-planet. The links are shown with blue lines, while the distance from the second link to the end effector, $l(\theta)$, is shown with a green line.

The last two joint angles of the simplified model can be derived from different properties shown below. The length $l(\theta)$ is calculated from the Pythagorean theorem.

$$l(\theta) = \sqrt{(d_1 - z_c)^2 + (x_c - a_1 + r\cos\theta)^2} \tag{5.8}$$

$$\theta_m = \arccos\left(\frac{a_3^2 + a_2^2 - l^2(\theta)}{2a_2a_3}\right) \tag{5.9}$$

$$\theta_f = \text{Atan2}\left(d_1 - z_c,\ x_c - a_1 + r\cos\theta\right) \tag{5.10}$$

$$\theta_b = \arcsin\left(\frac{a_3\sin\theta}{l(\theta)} - \theta_f\right) \tag{5.11}$$

$$\tag{5.12}$$

Due to the initial position of both joint two and three not being aligned with link one the joint angles are offset by $\frac{\pi}{2}$. Using the identities above the following joint angles are

obtained

$$q_2 = \frac{\pi}{2} - \theta_b \tag{5.13}$$

$$q_3 = \frac{\pi}{2} - \theta_m \tag{5.14}$$

The resulting motion for the simplified three link manipulator with initial position

$$p^0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0 \\ 0.13 \end{bmatrix} \tag{5.15}$$

measured in meters, and with radius $r = 0.15\ m$ is shown in figure 5.4.

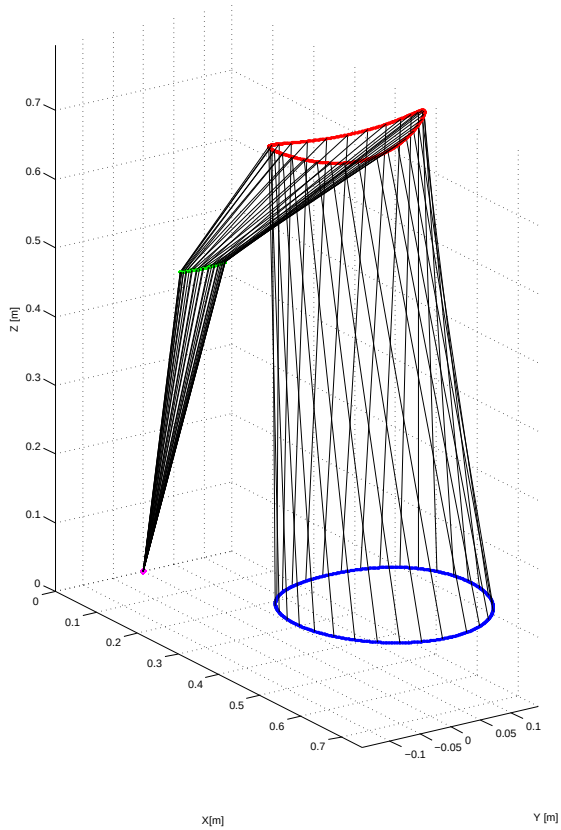


Figure 5.4: Three link configurations for horizontal circle

In figure 5.4 the dot colored in magenta is the position of link 1. The green line is the position of joint 2, while the red line is the position of the third joint along the circular trajectory. The blue line is the position of the end effector of the simplified thee link model along the motion. The black lines intersecting the joints are present for the sake of illustration of the links.

Part III

Development of Energy Efficient Scenarios

Chapter 6

Optimal Motion Profiles

6.1 Ingredients in Development of Optimal Scenarios

The overriding objective of this thesis is to plan motion for given scenarios that results in the robot manipulator consuming as little mechanical energy as possible. In the pursuit of such trajectories it is desirable to develop motion profiles that are not explicitly dependent on time.

The first path that is considered is a circular path in the x, y-plane. The position along this path is given by a generalized coordinate $\theta \in [0, 2\pi]$. From an initial position and the generalized coordinate it is possible to retrieve the cartesian coordinate of the end effector in the base frame coordinate system, which in turn can be used to retrieve joint angles through forward kinematics. Thus we get functions of θ rather than time.

$$q_i = \varphi_i(\theta), \quad i = 1, 2, 3 \tag{6.1}$$

By differentiating this both once and twice angular velocity and acceleration for the three first joints are retrieved.

$$\dot{q}_i = \varphi_i'(\theta)\dot{\theta} \tag{6.2}$$

$$\ddot{q}_i = \varphi_i''(\theta)\dot{\theta}^2 + \varphi_i'(\theta)\ddot{\theta} \tag{6.3}$$

To make the velocity profile of θ periodic it is suggested to make it as a sum of sines and cosines gained by a set of different coefficients. These coefficients are to be the parameters

of optimization. The velocity profile is given

$$\dot{\theta} = h(\theta) = \sum \left(a_i \sin(i\omega\theta) + b_i \cos(i\omega\theta) \right) \tag{6.4}$$

where ω is the angular frequency and i is an index. a_i and b_i are the coefficients used as minimizers. $i \in [1..n]$, where n denotes the order of the velocity profile.

In this approach it is decided to keep the order relatively small. Scenarios of order 3, 5 and 7 are implemented in order to investigate the possible gain in performance. Note that the indices start at zero. At an angular frequency of $\omega = 1$ all components of order higher than zero will integrate to zero over $\theta \in [0,\ 2\pi]$. We have

$$a_1 \sin(0 \cdot \theta) = 0, \quad b_1 \cos(0 \cdot \theta) = b_1, \quad \forall\ \theta \in \mathbb{R}^1 \tag{6.5}$$

Thus b_1 constitutes the average angular velocity along θ. Also note that the average velocity along θ generally is different from the average velocity of θ along time.

To obtain the acceleration profile the derivative with respect to time of this function $h(\theta)$ is then calculated.

$$\begin{aligned}
\ddot{\theta} &= \frac{\mathrm{d}}{\mathrm{d}t} h(\theta) \\
&= \frac{\mathrm{d}}{\mathrm{d}\theta} h(\theta) \underbrace{\frac{\mathrm{d}\theta}{\mathrm{d}t}}_{h(\theta)} \\
&= h'(\theta) h(\theta) \\
\ddot{\theta} &= \Big(\sum i\omega a_i \cos(i\omega\theta) - i\omega b_i \sin(i\omega\theta) \Big) \Big(\sum a_i \sin(i\omega\theta) + b_i \cos(i\omega\theta) \Big)
\end{aligned} \tag{6.6}$$

θ and the functions for $\dot{\theta}$ and $\ddot{\theta}$ can now be plugged into (6.1), (6.2) and (6.3). Both $\varphi'(\theta)$ and $\varphi''(\theta)$ are calculated using Maple. The result is very comprehensive and is therefore excluded from the report. The script used for calculating $\varphi'(\theta)$ and $\varphi''(\theta)$ can be found in the digital attachments.

By plugging θ, $\dot{\theta}$ and $\ddot{\theta}$ into the formulas for q, $\dot{q}$ and $\ddot{q}$ and then plug the resulting angles, velocities and accelerations into the robot equation one can retain the corresponding torque, τ, at any given point on the trajectory. The robot equation, as stated before, is given

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + \tau_f = \tau \tag{6.7}$$

At this point a scheme for obtaining the torque along a motion profile given by the predefined coefficients in $h(\theta)$ is developed.

The total mechanical energy consumed along the path is given by the integral of power along the extension of time along the path. The power consumed by the robot at any given instance of time is given

$$P = |\tau \dot{q}| \tag{6.8}$$

Thus, the total energy consumpion becomes

$$E = \int_0^T |\tau \dot{q}| \mathrm{d}t \tag{6.9}$$

In order to calculate the energy consumpiton the scheme for calculating power is inserted into a Simulink model with an integrator. Then the model is simulated with a variable-step ode45 solver until θ reaches 2π. This Runge-Kutta solver is a fifth order method that performs a fourth-order estimate of the error, and it is the default solver in Simulink. The Simulink model can be found in the digital attachments.

In order to find coefficients that ensure a minimum use of mechanical energy an interior-point optimization function in MATLAB, `fmincon`[10], is used. This is a constrained optimization function, and in this approach several constraints has to be present. The angular velocities are constrained to lie inside the region where the friction is identified. It is assumed that the unconstrained solution would lie inside this region anyways, but the friction characteristics from curve fitting are uncertain outside this region. Also the velocity along the circular path is constrained to be non-negative by some margin, δ. As the controller is independent of time it would keep standing still forever if, at some point, the velocity reference is equal to zero.

The optimization scheme uses the Simulink model as objective function and the coefficients a and b as function variables, where a and b are vectors. The performance index is total energy consumed by the robot.

6.2 Dry Friction

As the friction compensation model derived in chapter 4 is made for constant velocity it is not able to capture static friction, called stiction. As a result of this strange behavior is observed at close to zero joint velocity. These effects results in motion profiles of very small velocities and with a couple of spikes where most of the movement is present. The time of these profiles extends to a couple of minutes which seems unreasonable.

In an attempt to avoid error from unmodeled stiction the friction compensation is modified to penalize velocities close to zero as can be seen in figure 6.1. The modification is shown for the third joint, and is done in a similar manner for the first two joints.

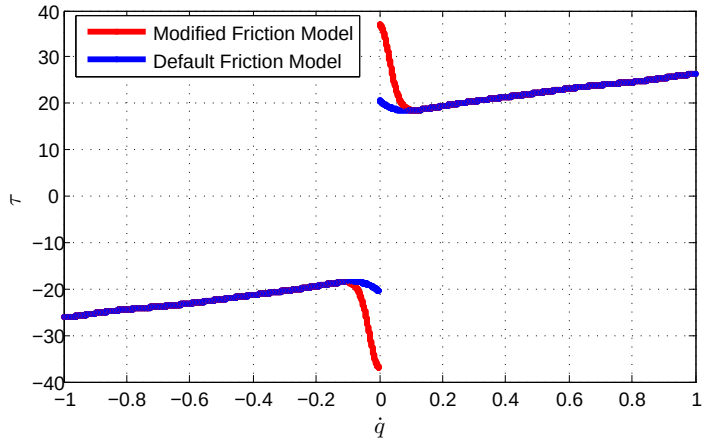


Figure 6.1: Modified friction compensation shown for joint 3.

This modification seems to make the solutions from the numerical optimization more reasonable.

6.3 Optimal Motion Profiles

By optimizing the energy consumption with coefficients of order 7 the velocity profile for θ shown in figure 6.2 is obtained.

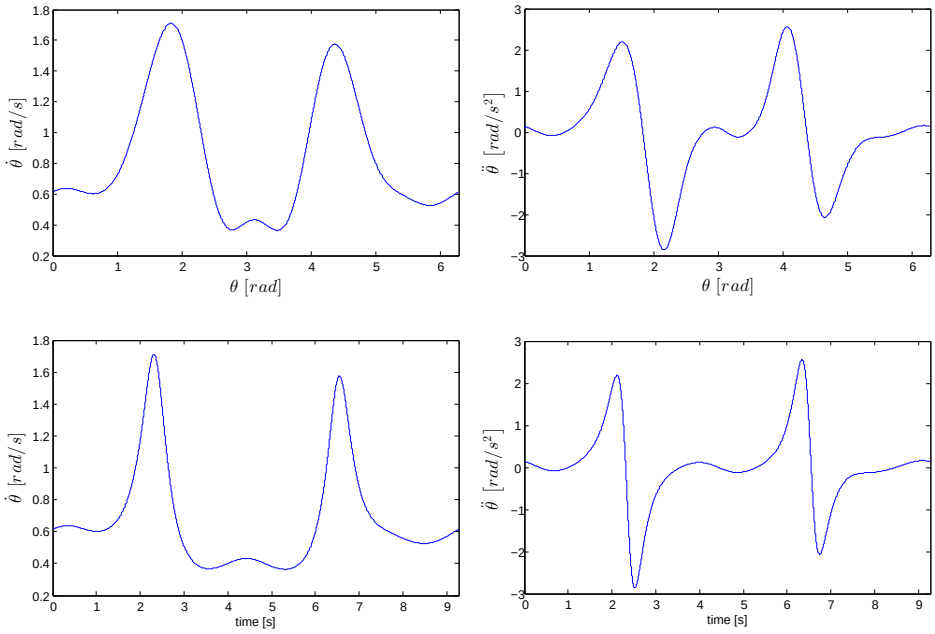


Figure 6.2: Motion profiles for $\dot{\theta}$ and $\ddot{\theta}$ along θ and time.

The angular velocity seem rather swift, but not unreasonable. The corresponding profiles in joint space are given in figure 6.3.

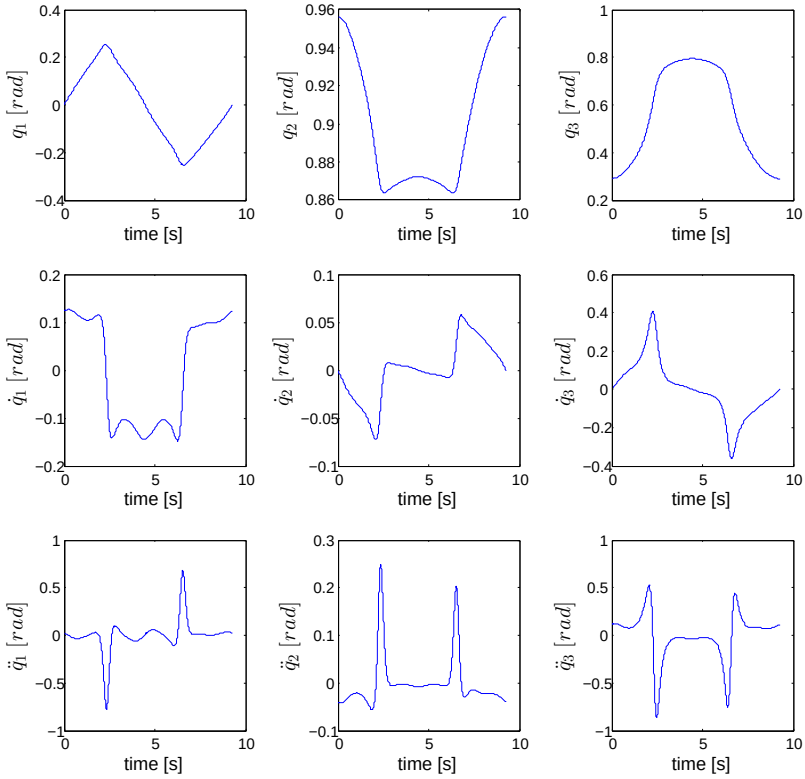


Figure 6.3: Joint motion profiles for trajectory of order 7.

Intuitively this seems reasonable. The time extends to almost 10 seconds, at angular velocities that looks reasonable. The angular velocity for joint 3 is generally higher. This seem reasonable, as it is at the end of the kinematic chain and thus requires the least torque to move. The power along the path is given in figure 6.4.

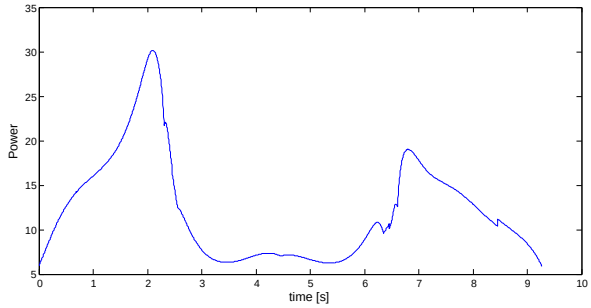


Figure 6.4: Power consumption along trajectory of order 7.

6.4 Control scheme

The optimized trajectories are not explicitly dependent on time. Similarly the trajectories are implemented with a special kind of controller independent of time. The controller is implemented with the External Control Simulink interface as shown in figure 6.5. It is an inverse dynamics controller, and the details can be seen in [7] or [11]. A block diagram describing the architecture of the controller is presented in figure 2.5.

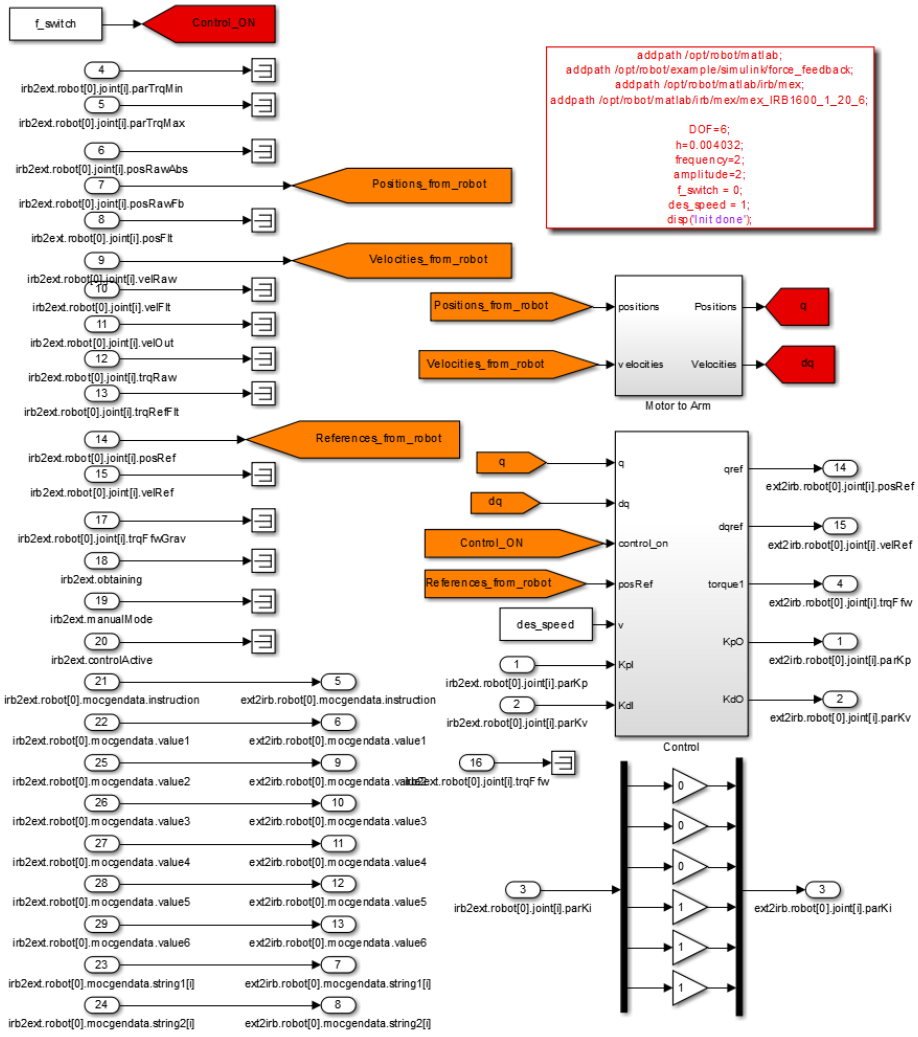


Figure 6.5: Simulink External Control - Main Interface.

This interface mainly describes the inputs to and outputs from the axis c omputer. The main functionality of the control scheme lies inside the subsystem denoted ”Control”. Its architecture can be seen in figure 6.6.

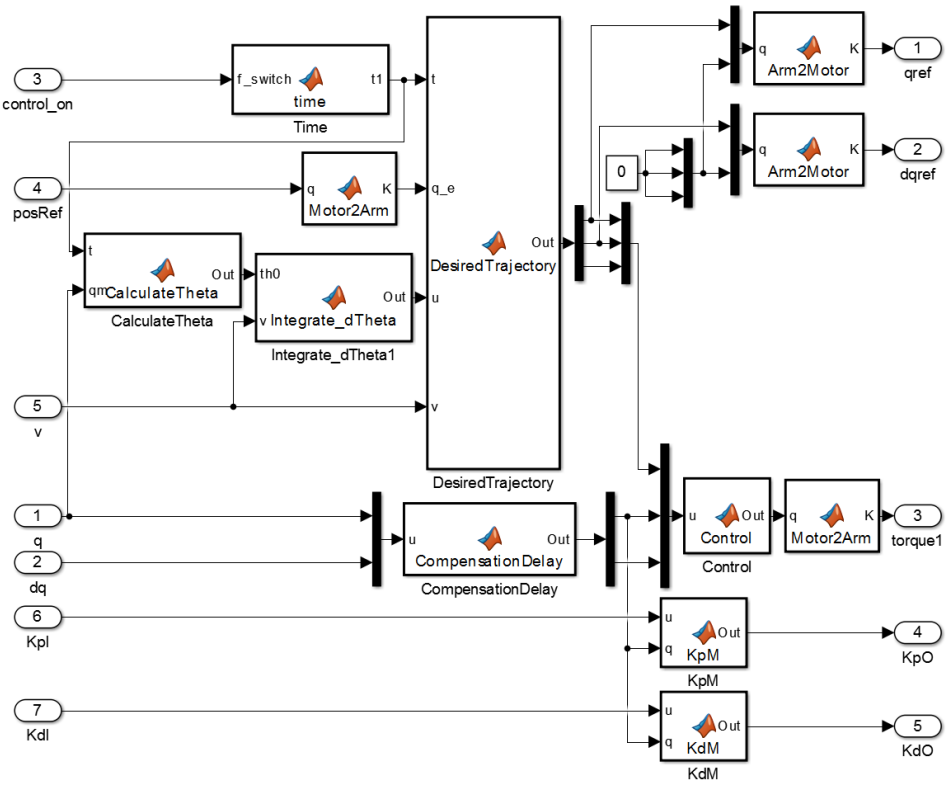


Figure 6.6: Simulink External Control - Control Architecture.

The blocks denoted `Motor2Arm` and `Arm2Motor` are multiplication operations corresponding to the gearing ratios of the different joints. The Simulink file with corresponding MATLAB functions can be found as digital attachments to this thesis.

6.4.1 Time

The time block works as a clock. The iteration time of the robot manipulator control system is 0.004032 seconds. Thus, this block increments the time with this amount for each iteration.

6.4.2 Calculate Theta

The block denoted `CalculateTheta` gets the current joint positions as input. The position of the TCP frame origin is calculated through `ForwardKinematics`, and θ is calculated using the four-quadrant arctangent function `atan2`.

6.4.3 Integrate dTheta and Compensation Delay

The system has a delay of about three samples from input to output, i.e. $\Delta t \approx 0.012\ s$. Therefore an estimate of θ three samples ahead is made. This is done using the velocity profile for $\dot{\theta}$ previously derived.

In a similar manner `CompensationDelay` is predicting q and $\dot{q}$ three time samples ahead.

6.4.4 Desired Trajectory

This block consists of huge vectors of references for θ as well as $q,\ \dot{q}$ and $\ddot{q}$. It takes the current control variable, θ, as input and calculates the corresponding joint position, velocity and acceleration references using interpolation.

A scheme for ramping the initial state to the reference initial position in a small amount of time (here 0.5 seconds is used) is also included in the functionality of this block.

Chapter 7

Main Results

Experiments are performed as described in chapter 6, by applying the retrieved velocity profiles of about 10 seconds with the inverse dynamics PD+ controller. From the experiments the following results are obtained. The motion starts in the most stretched out position of the path denoted $p_0(x, y, z) = [0.75, 0, 0.13]^\top \ [m]$. Figure 7.1 indicates that the experiments with trajectory developed from numerical optimization saves energy compared to the experiment with trajectory developed from the standard ABB planner at the area where the end effector is at its least stretched out position.

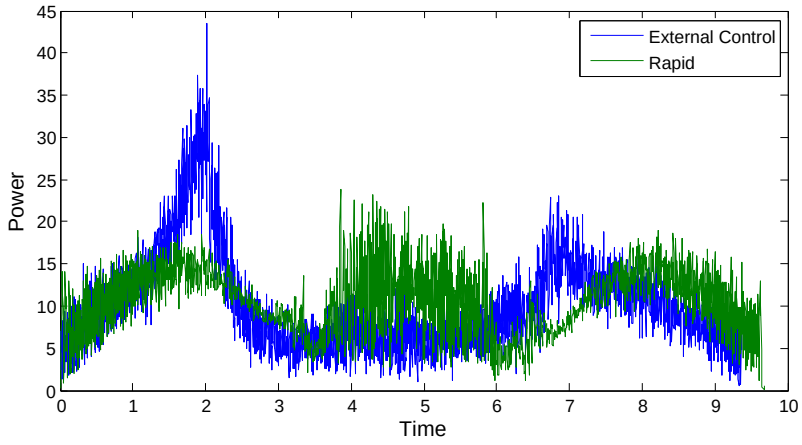


Figure 7.1: Power consumed by the robot manipulator along the trajectory for the External Control scheme and the trajectory obtained from Rapid.

It is also worth noticing that the power consumed by the robot from the optimized path has a spike in power consumption at about two seconds. This spike can also be seen in the power reference for the optimal motion (figure 6.4), but is hard to explain intuitively. The interior-point numerical optimization tool used in the development of this motion is highly sensitive to initial point, step-size limits etc. These parameters might therefore be looked into in order to avoid such costly behavior.

The integrated power, i.e. total consumed energy is given in table 7.1. The manipulator consumes **6.98%** less energy along the trajectory using external control motion compared to the trajectory made with the standard ABB controller, Rapid. It is also worth noticing that the external control motion is about 0.22 seconds faster than the motion from the standard controller. Repeated experiments shows that these results are consistent.

Table 7.1: Power Consumption

Controller	Power Consumption
External Control	95.4023
Rapid	102.0584

As the trajectory for external control is developed for a simplified model only actuating the inner three joints the three outermost joints are not considered in the calculation of power, i.e. $q_{4..6} = 0,\ \dot{q}_{4..6} = 0,\ \ddot{q}_{4..6} = 0$. The power comsumption along the ABB Rapid trajectory is also calculated from the first three joints, but the three outermost joints are not neccesarily unactuated in this case. Hence, it might be the case that torque applied to link 4 through 6 increases the power consumption for the Rapid motion even more, and thus, the improvement might be even better than what stated here. From observation the movement of the three outermost joints is minimal along the ABB Rapid trajectory, but including it could improve the result further.

The torque and joint velocity for the three inner joints along the circular motion can be seen in figure 7.2.

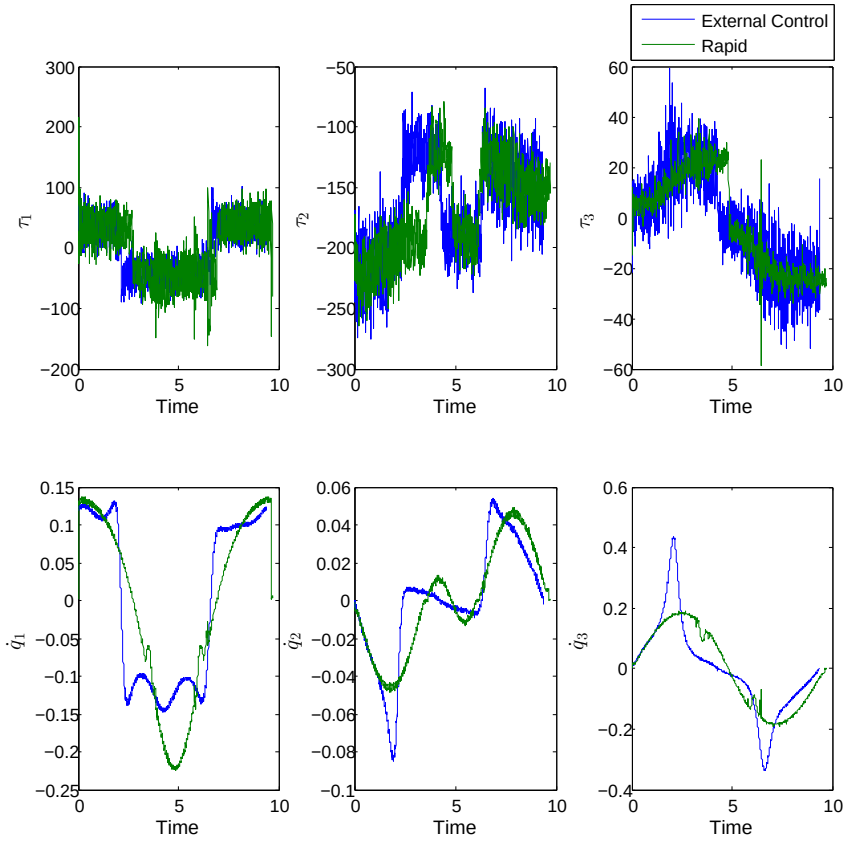


Figure 7.2: Torques and joint velocities for joint 1 trough 3 for both the external control and the Rapid experiment.

Figure 7.1 suggests that the gain in efficiency occurs from about 3.5 to 6 seconds. Figure 7.2 suggests that this gain is due to the more constant velocity of joint 1.

The level of noise on the torque applied to the third joint suggests that the tuning has room for improvements. It would be desirable to be able to avoid the spikes in angular velocity on joint three at about two and seven seconds. However, the magnitude on the torque applied to the third joint is small compared to the other two, one should not expect severe improvements from avoiding this.

Although 7% may seem insignificant, it could, as stated in chapter 1, lead to severe im-

provements in terms of profits and consumption of energy if applied on enough manipulators over an extended period of time.

Chapter 8

Conclusion

This thesis considers modeling of robot dynamics as well as development and implementation of energy efficient scenarios for industrial robot manipulator ABB IRB 1600.

The robot dynamics are extremely complex, and although certain phenomenas such as backlash in the gears, joint stiffness and static friction are not considered, the robot model seems to be quite accurate. Friction characterizations for constant velocity for the first three joints are developed, and repeated experiments indicates that a feed forward compensation applied to the axis computer improves the manipulator performance in terms of accuracy by 25-50% for different scenarios.

A scheme for making optimal trajectories in terms of consumption of mechanical energy along a predefinded path is made. The optimal scenarios are applied to the robot manipulator using an inverse dynamics PD+ controller. Through repeated experiments it is shown that a trajectory derived from numerical optimization decreases the energy consumption along a circular path of radius 15 cm in the x, y-plane by $\approx \mathbf{7\%}$ compared to trajectories from the standard ABB planner.

Chapter 9

Further Work

Further work includes making the dynamic model of the robot model more sophisticated by modeling phenomenas not considered in this thesis. Such phenomenas may include static friction, backlash of the gearbox and joint stiffness.

As the inverse dynamics PD+ controller is very sensitive in terms of controller gains it would be reasonable to look more into tuning of K_p and K_d. Due to construction work in the industrial robotics laboratory the time to experiment with different controller gains was limited, and small changes here could lead to severe improvements in performance.

It would also be reasonable to investigate the possible improvements in performance by moving the path in the workspace of the manipulator. In addition investigation of development of different energy optimal or sub-optimal routines for path shapes such as lines or connected path segments with corners etc. would be reasonable to include to further work.

Bibliography

[1] ABB Robotics. *IRB 1600 Data Sheet*. https://library.e.abb.com/public/3b0491a94bd700a248257c71004ef393/PR10282EN_R8.pdf, 2014. PR10282 EN R8.

[2] NIKON Metrology. *Nikon K-610 Key Features*. http://www.nikonmetrology.com/en_EU/Products/Portable-Measuring/Optical-CMM/K-Series-Optical-CMM/(key_features), 2015. Technical Specifications.

[3] Isolde Dressler. Force control interface for abb s4/irc5. /home/isolde/doc/extctrl.pdf, July 2009.

[4] Anders Robertsson Denis Stökle and Björn Olofsson. External control - extctrl. 2011.

[5] Willian Robert Brown. *Maneuver Based Design of Passive Assist Devices for Active Joints*. PhD thesis, University of Michigan, 2013.

[6] NIKON Metrology. *Nikon K-610 Technical Specifications*. http://www.nikonmetrology.com/en_EU/Products/Portable-Measuring/Optical-CMM/K-Series-Optical-CMM/(specifications), 2015. Technical Specifications.

[7] Seth Hutchinson Mark W. Spong and M. Vidyasagar. *Robot Modeling and Control*. Wiley, 2006.

[8] ABB. *CAD Model, IRB1600*. http://new.abb.com/products/robotics/industrial-robots/irb-1600/irb-1600-cad, 2015. Documentation.

[9] MathWorks. *Documentation - fit*. http://www.mathworks.com/help/releases/R2015a/curvefit/fit.html, 2015. Documentation.

[10] MathWorks. *Documentation - fmincon*. http://se.mathworks.com/help/optim/ug/fmincon.html, 2015. Documentation.

[11] Anders Robertsson Leonid B. Freidovich Stepan S. Pchelkin, Anton S. Shiriaev. Analysing and comparison of 'pd+' and inverse dynamics controls for orbital stabilization of a full-actuated system. Norwegian University of Science and Technology, 2014.

[12] Marius Nordheim Røv. Time optimal motion planning and motion control for industrial manipulators. Norwegian University of Science and Technology, 2014.

Appendix

The appendix consists of two parts. In A.1 some additional results from validation of the friction model is presented. In A.2 brief descriptions of the attached digital appendix can be found.

A.1 Validation of Friction Compensation

Validation of the friction compensation applied to the robot manipulator is done by performing different scenarios one link at the time for the three joints of the simplified robot model. Experiments are done for all three joints at three different maximum velocities and with three different feed forwards (i.e. no feed forward, feed forward from encoder measurements and feed forward from camera measurements). As this generates a whole lot of plots the particularly curious can investigate the results here in the appendix rather than within the main part of the thesis.

A.1.1 Actuation on Joint 1

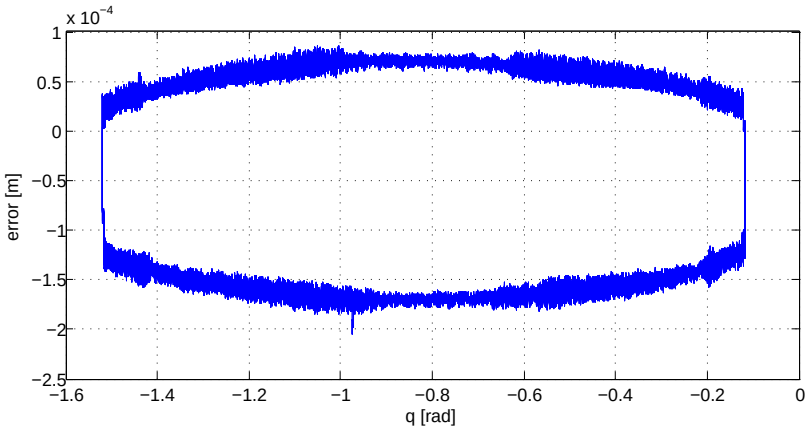


Figure 1: Error with no feed forward and maximum velocity of 0.1 $[^{rad}/_{s}]$

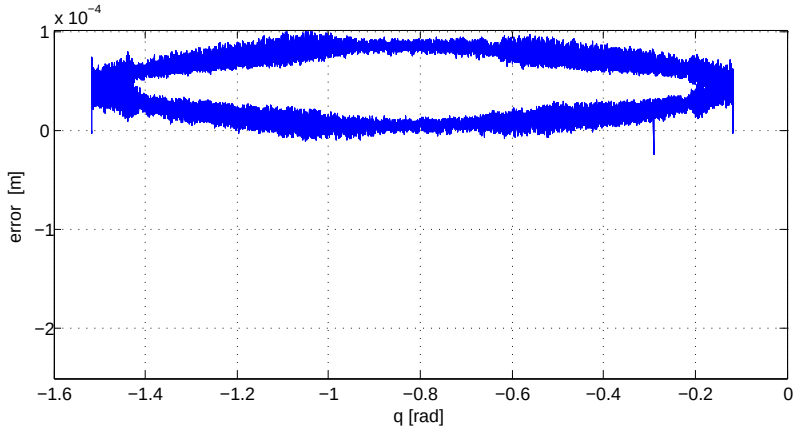


Figure 2: Error with feed forward from encoder measurements and maximum velocity of 0.1 $[^{rad}/_{s}]$

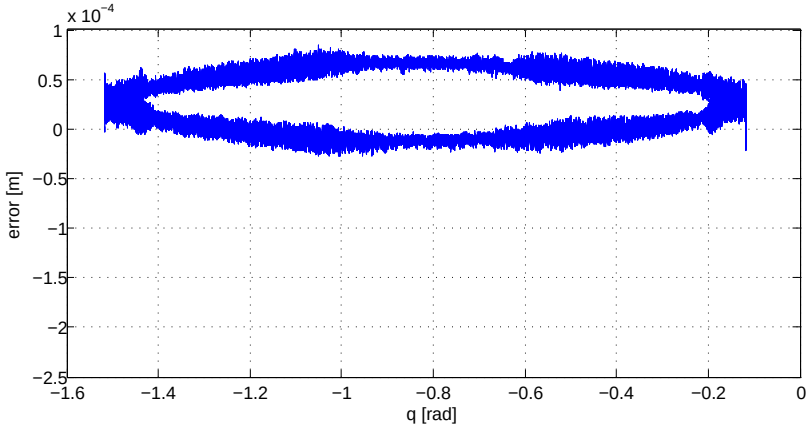


Figure 3: Error with feed forward from camera measurements and maximum velocity of 0.1 $[^{rad}/_{s}]$

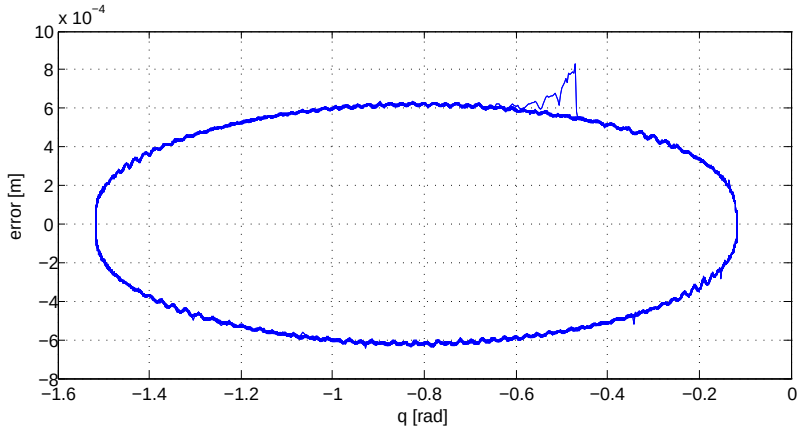


Figure 4: Error with no feed forward and maximum velocity of 1.0 $[^{rad}/_{s}]$

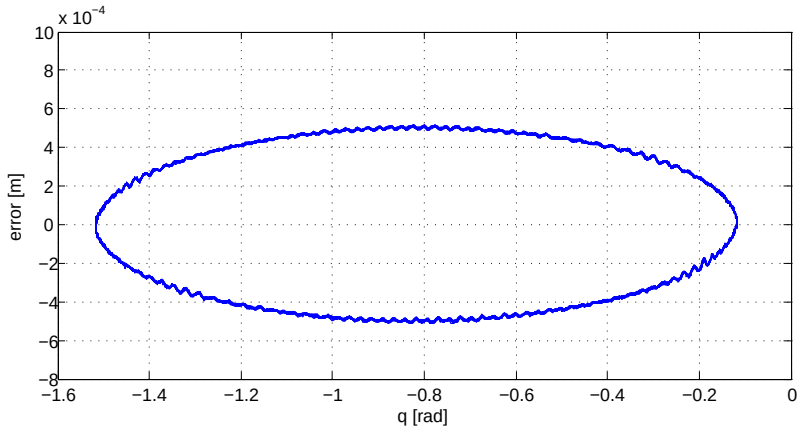


Figure 5: Error with feed forward from encoder measurements and maximum velocity of 1.0 $[^{rad}/_{s}]$

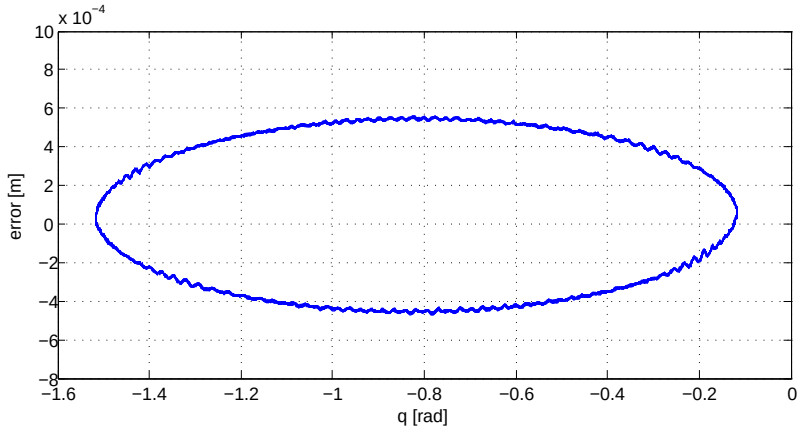


Figure 6: Error with feed forward from camera measurements and maximum velocity of 1.0 $[^{rad}/_{s}]$

A.1.2 Actuation on Joint 2

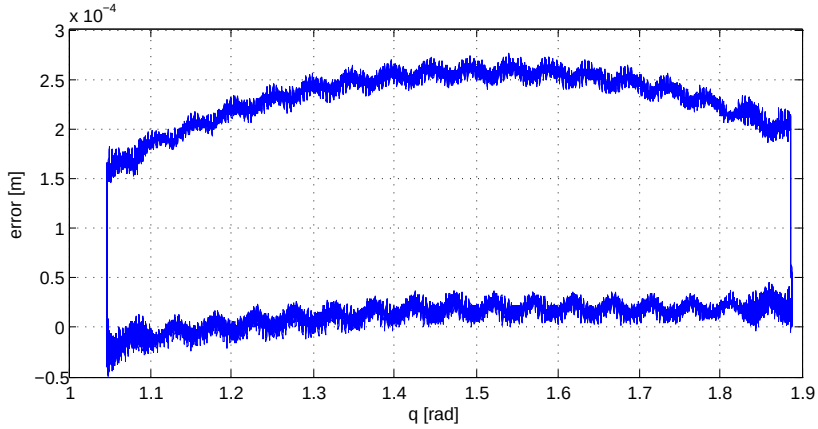


Figure 7: Error with no feed forward and maximum velocity of 0.1 $[^{rad}/_{s}]$

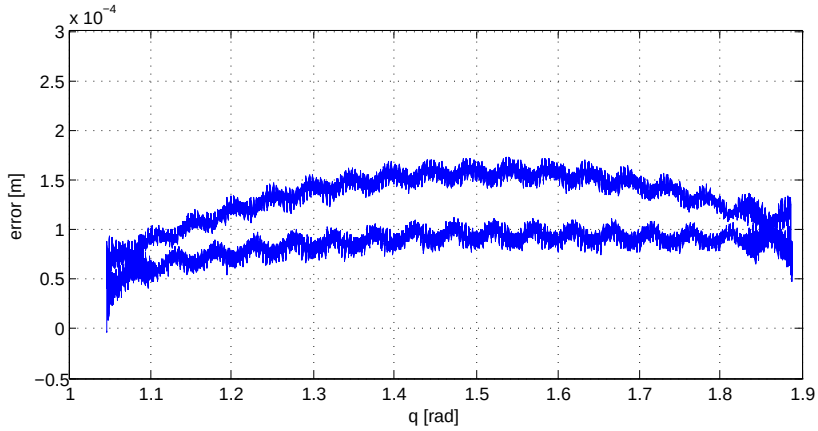


Figure 8: Error with feed forward from encoder measurements and maximum velocity of 0.1 $[^{rad}/_{s}]$

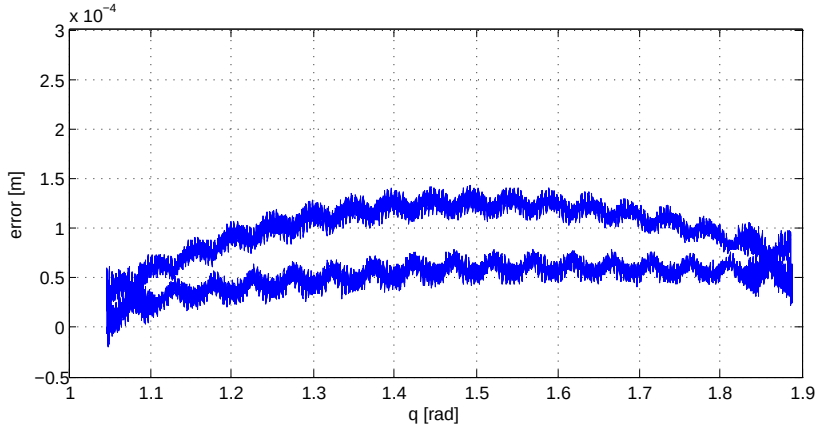


Figure 9: Error with feed forward from camera measurements and maximum velocity of 0.1 $[^{rad}/_{s}]$

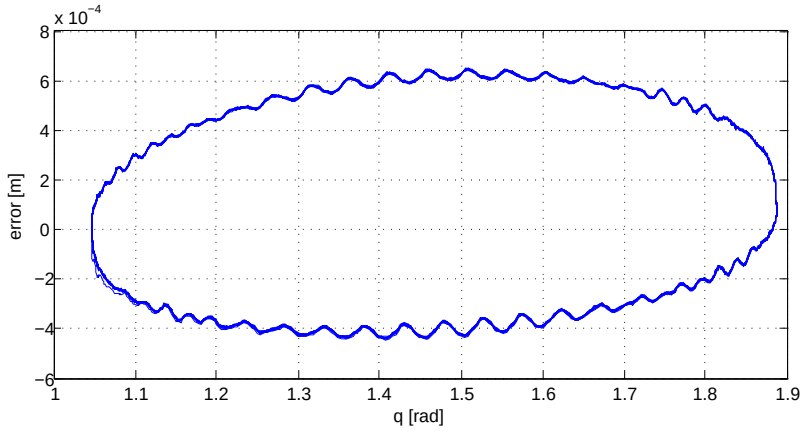


Figure 10: Error with no feed forward and maximum velocity of 1.0 $[^{rad}/_{s}]$

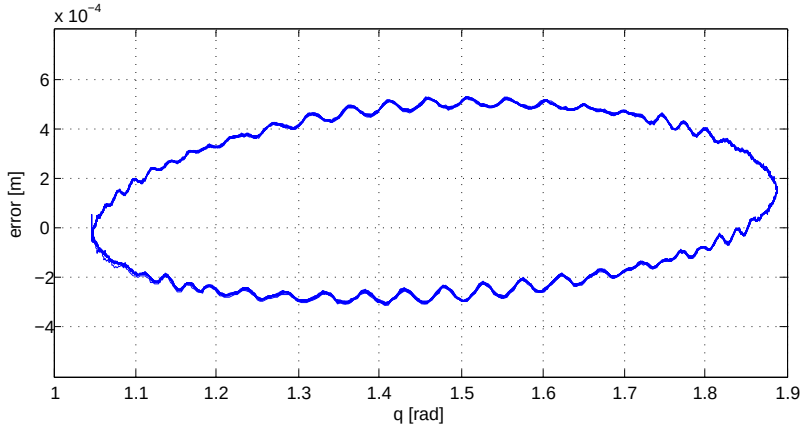


Figure 11: Error with feed forward from encoder measurements and maximum velocity of 1.0 $[^{rad}/_{s}]$

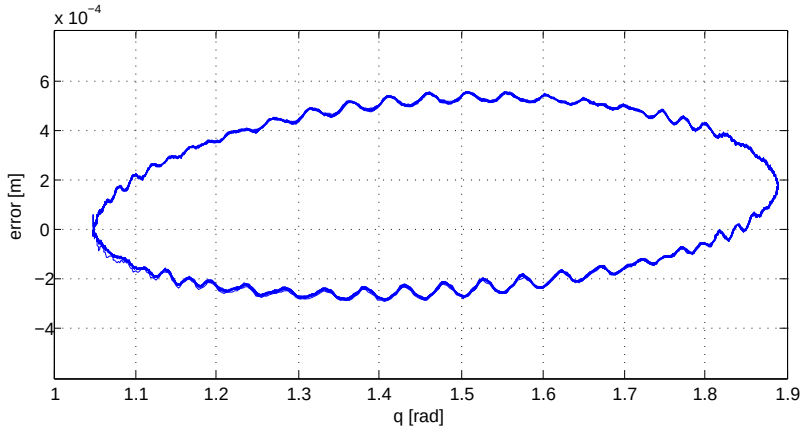


Figure 12: Error with feed forward from camera measurements and maximum velocity of 1.0 $[^{rad}/_{s}]$

A.1.3 Actuation on Joint 3

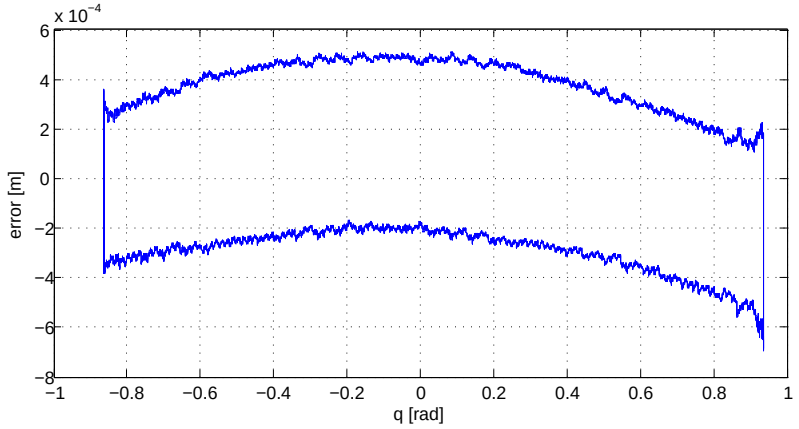


Figure 13: Error with no feed forward and maximum velocity of 0.1 $[^{rad}/_{s}]$

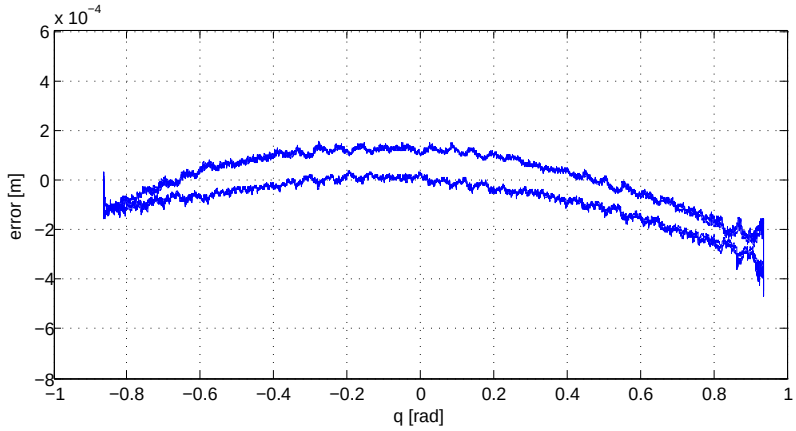


Figure 14: Error with feed forward from encoder measurements and maximum velocity of 0.1 $[^{rad}/_{s}]$

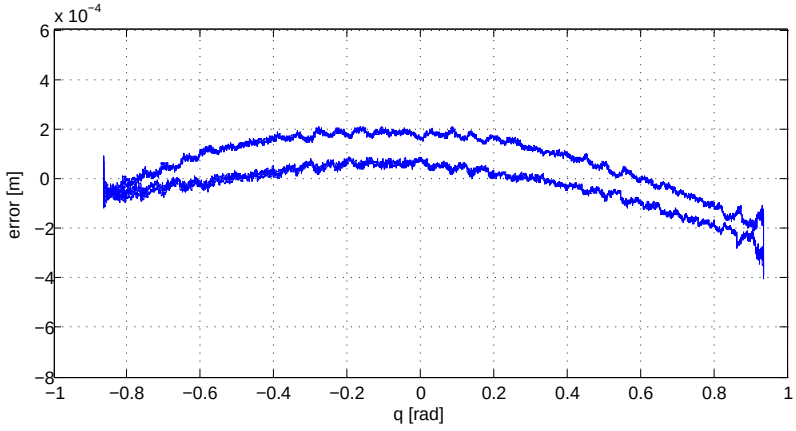


Figure 15: Error with feed forward from camera measurements and maximum velocity of 0.1 $[^{rad}/_{s}]$

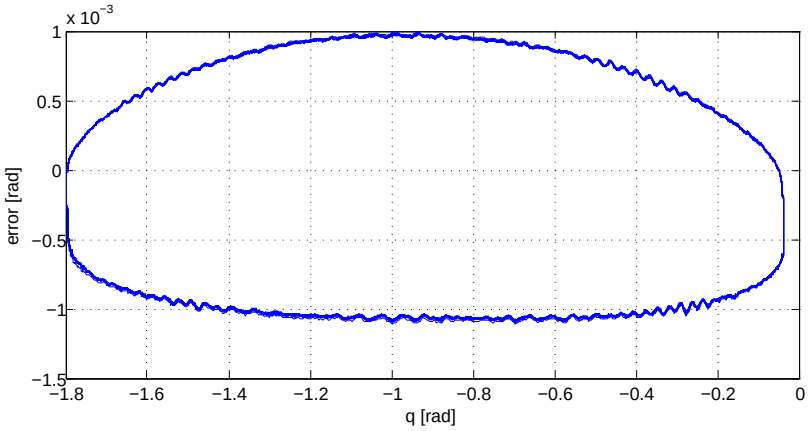


Figure 16: Error with no feed forward and maximum velocity of 1.0 $[^{rad}/_{s}]$

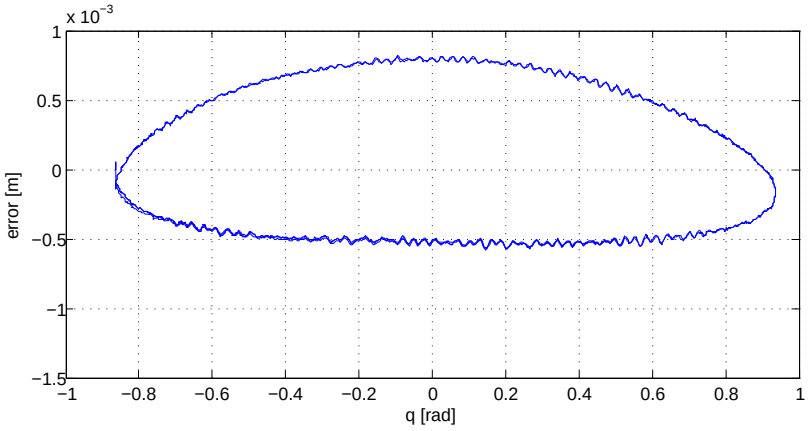


Figure 17: Error with feed forward from encoder measurements and maximum velocity of 1.0 $[^{rad}/_{s}]$

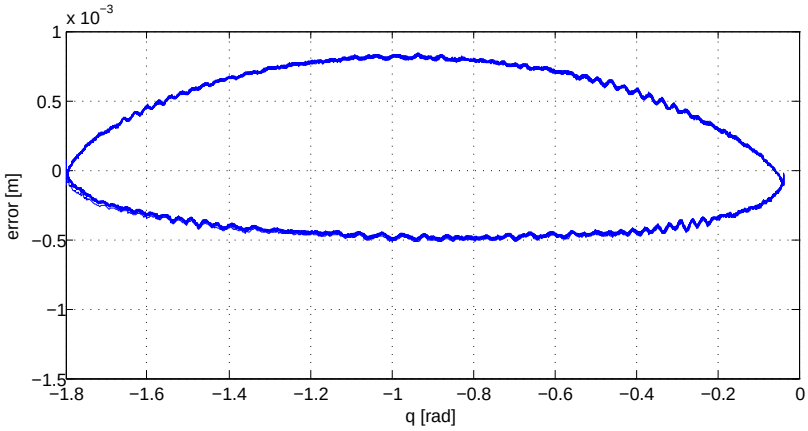


Figure 18: Error with feed forward from camera measurements and maximum velocity of 1.0 $[^{rad}/_{s}]$

A.2 Description of Attached Files

A.2.1 MATLAB and Simulink

`Arm2Motor.m` and `Motor2Arm.m` are matrices with gear ratios to go from the arm side of the joint to the motor side and vice versa.

`ForwardKinematics.m` and `InverseKinematics.m` maps joint coordinates to cartesian coordinates and vice versa as derived in chapter 3.

`smooth.m` and `smoothing.m` are not developed as part of this thesis, but used a lot to filter noisy signals. The scripts are provided by Mark Tobenkin at MIT, and included for convenience of whoever will look into further work on this thesis.

`IdentificationConstantVelocity.mdl` is the Simulink scheme applied to obtain data for the friction model.

`optimizeCoefficients.m` is a script performing the numerical optimization in order to retrieve velocity profiles that minimizes consumption of mechanical energy along a predefined path.

`simOpt.m` is the objective function to the optimization problem, and it is a function

that runs a simulation of the Simulink model `robotOptim.slx`. `constraints.m` is a function providing constraints to the numerical optimization problem.

`robotModel.slx` is pretty much similar to `robotOptim.slx`, but provides more output data to be investigated with optimal coefficients as input.

`InvDynControl_2012_vCircle.mdl` is the Simulink model containing the inverse dynamics PD+ control scheme.

A.2.2 SolidWorks

`tedmounting_tool_v2.SLDPRT` is the design file for the part 3D printed to be able to attach LEDs to the flange of the robot manipulator.

A.2.3 Maple

`getDerivatives.mw`, `motionProfiles.mw` and `robotEquation.mw` are based on work from [12], and used to derive symbolic expressions for joint velocities and accelerations as well as the components of the robot equation, and the fourth order motion profile used in validation of the friction model validation.