NTNU

Norwegian University of
Science and Technology

Develop and Implement Force Control Experiments using an Industrial Robot Manipulator

Erik Havikhagen

Master of Science in Cybernetics and Robotics
Submission date: June 2016
Supervisor: Anton Shiriaev, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Preface

This Master thesis presents the final work done in the master program Cybernetics and Robotics at *Norwegian University of Science and Technology* (NTNU). The thesis was carried out in 2016 between January and June. Experiences with robots and the K-series camera system during the project work in fall 2015 are the basis to experiments done in this project.

I would like to thank my supervisor Professor Anton Shiriaev for his support and supervision during the project. I would also like to thank PhD Candidate Sergey Kolyubin for his guidance on the K-610 Series Optical CMM and for help with the IRB 4600. I would like to thank Postdoctoral Leonid Paramonov for interesting discussions and input on the placement of LEDs for the K- 610 Series. Finally i would also like to thank all of my colleagues for their interesting discussions and rants during the numerous hours spent at the office.

June 2016
Erik Havikhagen

Abstract

In the last few decades there has been a large increase in industrial robots used in the machining industry. Industrial robots replaces manual labor in tedious tasks, and increases the level of precision of the process. There are however tasks that are still done by hand or by the use of CNC machines, where the use of industrial robots is limited. One case where the use of industrial robots is limited is in grinding, deburring and drilling of dense metals where a low level of error in the position is required. In these types of operations it is necessary to experiment with force control to discover behaviours that can be compensated, increasing the level of precision when using industrial robots.

The work done in this thesis consists mainly of three different types of experiments, where force control is the main aspect in all of them. All experiments follow a linear path along a metal bar, and forces between 1-30 N are applied on it using *Machining FC* for the IRB 4600. At the end effector a wheel-rod tool is attached, which makes it possible to conduct experiments on the same metal bar without the concern for deformation. The different experiments look at how the behaviour changes when either the velocity or the path is changed. In addition a high precision camera system called the K-610 Series Optical CMM is used to compare deviation of the end effector position with the robot encoders. Different types of filters for use in post processing of the force signals were also investigated.

Conducting force control experiments gave several interesting results regarding the behaviour of the robot during the operation. It was discovered that the impact force increased by large degree when force applied increased. For the first experiment conducted it was discovered a large error offset for force measurements, which was reduced tremendously by experimenting with different approach and withdraw paths of the robot. Investigating the reduction of speed for the robot gave interesting results with how well the force control worked for small forces applied and the reduction of impact force. Oscillations could be seen for all the force measurements produced and reducing the velocity of the robot did little to reduce these oscillations. It was also experimented with a step based velocity method, which proved to be quite dissatisfied. From investigating three different types of filter it was concluded that the Butterworth filter gave the best response in reducing the noise of the measurements.

This thesis contributes with result and discussion of the different experiments done in regard to the use of force control. It also provides a thorough review of everything needed to produce additional force control experiments using this setup. In the next step it would be interesting to experiment with force control when drilling spindle is used, and to create controllers which compensate for undesired behaviour of the robot.

Sammendrag

I de siste tiårene har det vært en stor økning i antallet industrielle roboter som blir brukt i bearbeidings/maskinerings industrien. Industrielle roboter brukes i slitsomme og repeterende operasjoner tidligere utført forhånd, noe som har vært med på å øke effektiviteten og presisjonen til produktene. Det er fremdeles rom for utvikling, og flere oppgaver der mulighetene for industrielle roboter er store gjøres ennå av mennesker og CNC maskiner. En type oppgave som industrielle roboter har potensialet for økt effektivitet er i sliping, drilling og fresing av metaller der høy presisjon er nødvendig. I slike oppgaver er det nødvendig å eksperimentere med kraft kontroll (FC) for å oppdage og kompensere for uønsket oppførsel, som kan være med på å øke graden av presisjon for industrielle roboter.

Arbeidet som er utført i denne oppgaven består i hoveddel av tre forskjellige typer eksperimenter, hvor kraft kontroll er sentralt i alle. Eksperimentene følger en linear vei langs en metalplate, og det blit påført trykkrefter mellom 1-30 N ved å bruke funksjonen *Machining FC* for IRB 4600. Verktøyet som er plassert på robotens ende består av et hjul festet til en stang, som gir muligheten til å utføre samme eksperiment på samme metalplate uten at det skal forekomme deformasjoner i metallet. De forskjellige eksperimentene ser på forskjeller i robotens oppførsel når bane og fart blir forandret. Itillegg til IRB 4600 brukes også et høy presisjons kamera system kalt K-610 Series Optical CMM, som gir mulighet til å oppdage forskjeller i posisjonen til robotens endepunkt fra hva robot enkoderne viser. Det har også blitt sett på bruk av forskjellige typer filtere for etterbehandling av kraftmålinger.

Utførelsen av kraft kontroll eksperimentene ga flere interessante resultater tilknyttet oppførselen til roboten iløpet av operasjonen. Spesielt oppdaget man at slagkraften mellom roboten og metalplaten økte med en økning i kraft benyttet, noe som er høyt uønsket og gir stor feilmargin i starten. Det oppsto også en ganske stor feil forskyvning av kraft i målingene når roboten ikke var i nærheten av metallet i de første eksperimentene. Dette ble redusert kraftig ved å forandre posisjonen til roboten når den nærmer seg og forlater metallplaten. Ved å redusere hastigheten i prosseseringdelen ble det oppdaget at roboten klarte seg bedre når mindre krefter ble brukt, og man fant en reduksjon i slagkraften. For alle kraftmålingene kan man tydelig se at det er påvirket av oscillasjoner. Det ble derfor sett på om reduksjon i hastighet påvirket disse oscillasjonene på noen måte, noe som visste seg å gi lite uttelling. Itllegg ble det prøvd en "step based" metode for hastigheten som også ikke viste forandringer i oppførsel. Ved å se på tre forskjellige filtere ble det konkludert med at Butterworth filteret var den som gav best respons i å redusere støy på målingene.

Denne oppgaven har som bidrag i resultatene av de eksperimenter som ble utført og diskusjonene rundt dette. Det gir også en dyp gjennomgang av de forskjellige prosedyrene nødvendig for å kunne utføre videre eksperimenter med dette oppsettet. Videre kan det være interessant å eksperimentere med kraft kontrol hvor boremotoren er aktivt brukt, itillegg kan det bli lagd kontrollere som kompenserer for den uønskede oppførselen til roboten.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The conception of robotics has been a fascination for uncountable centuries. As early as 400 BC was a wooden bird that could fly created by Archytus of Taremtum [24], which has some resemblance to a robot. In the 18th century alone were countless of brilliant yet impractical automata or robots made. The word robot was first used in a play called *Rossum's Universal Robots* by Karel Capek in the 1920s, where robot is the czech word for slave. It was not until the early 1950s that the type of robot commonly used today was developed. George C. Devol invented and patented a reprogrammable manipulator called the *Unimate* [11], which would later be modified by Joseph Engelberger to create the first industrial robot.

An industrial robot is defined by ISO 8373 as an *automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes which can either be fixed or mobile.* [15]. Research and development of industrial robots increased a lot in the 1970s and 80s, where companies such as ABB and Kuka entered the market. One example of research is the work conducted by Kramer et al. [8] on the investigation deburring using industrial robots. The variety of applications used by industrial robots also increased tremendously in this period. The reason for this ever expanding use of robots in the industry, stems not only from increase in efficiency, but also from reducing health issues caused by manual labor. This is also why the use of industrial robots will most certainly increase in the future.

Using robot manipulators in grinding, deburring, etc operations can be especially hard, when only controlling the position of the manipulators end effector. A small change in the position of the manipulator might contribute to a large force as the robot interacts with the environment. This is why force control has been developed, and makes it possible to control the force applied on the environment. With force control both force and torque is measured directly by a sensor, and then used in the feedback control loops. Still the level of precision and repeatability is not yet high enough for force control with robotic manipulators to be used several high precision industry, where manual labour or other machines are still the most suitable. This is why finding new ways to increase the precision through force control is highly valuable, which lays down the foundation of this thesis.

In this thesis the goal is to experiment with force control using an IRB 4600 robot manipulator from ABB. All the experiments will have a linear path along the surface of the work object, because of limitations with the tool used. In addition the K-Series camera

system by Nikon will be used to measure the position of the tool in respect to the robots base, which is valuable in detecting deviations in the tools location, when comparing the information with the robot encoders. The work is also done to get a better understanding of how force control works by using an industrial robot, and the steps of the experiments can be divided into

1. Planning
2. Execution
3. Measuring
4. Post processing

Also from the data gathered a better understanding of the IRB 4600s behaviour will gained. The experience gained from conducting these experiments gives the possibility to conduct grinding, drilling, deburring, etc experiments with spindle motor actively used.

1.1 Motivation

In the last decade research of robot machining has had a rather small progress, and the potential for improvements are high as pointed out by Chen and Dong [29]. Industrial robots are today mainly used in executing repetitive operations such as welding, painting etc, and robot manipulators for use in machining counts for less than 10 % of the total sale of industrial robots according to IFR [20]. Still the interest for robots used in machining have been increasing in recent years, for example with the COMET project funded by EU in 2008, which was set to discover ways to achieve a cost effiency of 30 % in comparison to machine tools [10].

The effiency of industrial robots in machining are still not fully comparable to CNC machines, which accounts for the most used in machining processes. It is especially for cases involving processing of high density materials that robot machining has greater position error than CNC machines. This comes mainly from the low stiffness of industrial robots, which makes it more affected by reaction forces acting from the environment. A lot of research is done to reduce this position error, and Schneider et al. [27] addresses the issue of stiffness for an industrial robot, where they try to create a model based on stiffness which can be used to compensate for deformation. It was concluded that feed forward model based compensation could be used reduce the error, however there are several issues remaining to make it applicable in the industry.

Industrial robots have many advantages over CNC machines, f.ex higher adaptability, lower cost, better maneuverability and larger workspace, which could make it the better option if precision problems are addressed. By conducting experiments using force control on machining operations it might make it possible to discover new solutions, that could fix the issues with industrial robots in machining. This could open up new areas for industrial robots to be used, which again might lower the cost and increase the effiency on a lot of products created using either manual labor or CNC machines. For these reasons and many more is experimentation of industrial robots used in machining procedures with active use of force control extremely interesting, and benefical to further development.

1.2 Objective

The objective of this thesis is to conduct machining experiments using the industrial robot IRB 4600 by ABB, where force control is a central aspect of the experiments. In addition to gathering measurements using a robot manipulator, the K-610 Series Optical CMM camera system will be used to gather additional position measurements for comparison reasons.

The first and main part of the thesis involves conducting experiment that were earlier done by Petter Kvernberg [17] for validation. The experiment deals with using the IRB 4600 to simulate a grinding procedure on a metal bar along a straight line, which does not take into account friction and vibrations normally found in such procedures. This is possible because a different tool is used, which does not use the spindle motor. Measurements for both force and position will be gathered using external computers connected to the manipulator and the camera system. Two other experiments are also conducted with this setup, that investigates the impact of the velocity and trajectory used for the robot.

It was also planned to conduct experiments using the drilling spindle, however this was not accomplished as the necessary training and security requirements were not met in time. Several of the issues linked to using the drilling spindle for grinding operations are still discussed in this thesis, and possible solutions are given.

From the result of these experiments it is possible to investigate how the behaviour of robot is when using force control, and to see if there are deviations between the camera and the robot. In processing of the measurements several filters have also been looked at, to see which one gives the best response. The effect of collision between the environment and the manipulator is also more clearly understood by looking at the force measurements.

1.3 Contribution

This thesis mainly contributes with the different experiments conducted using force control, and the result provides a validation to the force control experiment conducted by Petter. Furthermore it gives a thorough explanation of the whole setup process for FC experiments, which makes it easier to start conducting experiments with this setup. The thesis also gives a deeper explanation to what filter to be used in post-processing of the signals, and at what velocities and paths of the procedure that works best. While the camera calibration and measurement did not work as wanted it still provides a way in how it should be conducted, and it also gives calibrations settings not to be used.

1.4 Structure of report

The report is divided into six chapters along with a bibliography and an appendix, which includes abbreviations, additional plots and code.*Matlab* and *Maple* code giving all the necessary plots and calculations are provided in the attachements. Also in the attachements are the manuals needed to work with the different equipment located. The different chapters of the report can be summarized as,

1. **Chapter 1** provides an introduction of the report, where the motivation and objective of the thesis are clarified.

2. **Chapter 2** is the mathematical preliminary of the thesis. A basic understanding to the important fields for this thesis is given here. It is necessary to have knowledge in these fields before continuing to the next chapters.

3. **Chapter 3** presents the modeling of the IRB 4600 for both the kinematical and dynamical aspects of the robot.

4. **Chapter 4** gives a thorough explanation of all the equipment used in this thesis. Main focus is given on the IRB 4600 and the K-610 Optical CMM, which comprises the biggest part of the equipment used.

5. **Chapter 5** is the chapter containing all the force control experiments conducted in this thesis. The main result is shown and several aspects with the experiments are discussed.

6. **Chapter 6** gives a conclusion to the project, where the most important discoveries found will be summarized.

Chapter 2

Mathematical Preliminaries

In this chapter an understanding of the different subjects utilized during the time of the thesis will be presented. It is expected that the reader has some knowledge of these topics beforehand. Starting with robot kinematics an explanation of both forward and inverse kinematics will be given for any kind of robot manipulator. Moving on the topic of Jacobian and robot dynamics will be elaborated, followed by a basic understanding of force control. The last section will look upon signal processing with focus on filters, power spectrum density and fourier series.

2.1 Robot kinematics

Robot kinematics involves the study of motion for a robot manipulator, without looking at forces and torques that affects motion. This is done by looking at the relationship between joints of the robot and the position and orientation of the robot. The different joint variables are shown as q_i, where i is the number of joints. It is expected that all joints only have one DOF each, and they can either be revolute or prismatic joints. For a revolute joint $q_i = \theta_i$, which is the angle of rotation. For prismatic joints $q_i = d_i$, which is the joint displacement.

End effector or tcp (tool center point) is a name for the device or tool attached on the robot wrist. In forward kinematics it is possible to find the position and orientation of the end effector given joint variables. Inverse kinematics goes the opposite way by determining the joint variables given the position and orientation. The theory of kinematics is discussed in a general sense, and then in a later chapter modeling of the IRB 4600 with spindle tool will be discussed.

2.1.1 Forward kinematics

Forward kinematics or configuration kinematics problem involves the relationship between individual joints of the robot and position and orientation of the end effector for rigid robots. A good method of selecting frames of reference for all kinds of robotic application is by using the *Denavit - Hartenberg* (DH) convention, Spong et al.[25].

$$A_i = Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \tag{2.1}$$

Equation 2.1 shows the product of four basic transformations used in this convention, where the four quantities θ_i, a_i, d_i, α_i are parameters for link and joint i. A_i is the

homogeneous transformation matrix and changes every time the manipulator changes position and orientation. The IRB 4600, which is used in this project, only has rotational joints, therefore the convention is $A_i = A_i(q_i)$. The transformation matrix A_i can be further derived as shown in equation 2.2.

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

Just by using this equation it is not possible to represent any homogeneous transformation matrix with only the four quantities, and it is therefore necessary that it has unique solution.The following two properties are used to make sure an unique homogeneous transformation matrix exists.

1. The axis x_1 is prependicular to the axis z_0.

2. The axis x_1 intersects the axis z_0.

Figure 2.1 shows how frames are assigned using the properties of Denavit-Hartenberg. The

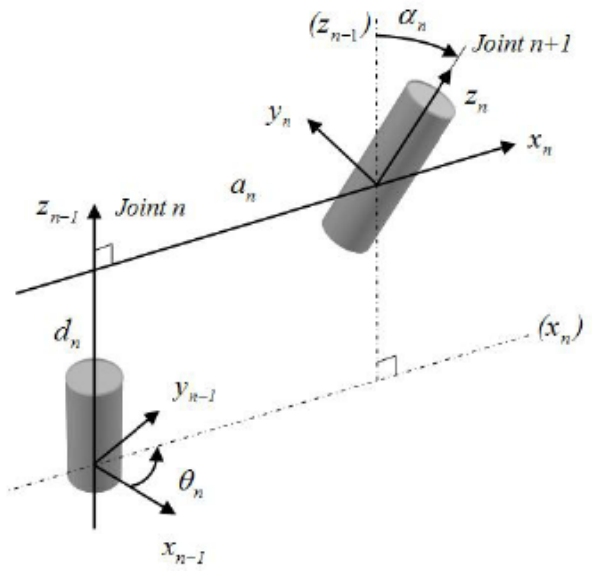


Figure 2.1: Denavit-Hartenberg frame assignement [23]

total transformation can be represented by a sum of multiplication of all joints $A_1 \dots A_n$ and is called the transformation matrix, T_n^0, where n is the last link. The transformation matrix contains the total rotation matrix R_n^0 and the total length o_n^0, which can be seen in equation 2.3.

$$H = T_n^0 = A_1 \dots A_n = \begin{bmatrix} \mathbf{R}_n^0 & \mathbf{o}_n^0 \\ \mathbf{0}_{1\times 3} & \mathbf{1} \end{bmatrix} \tag{2.3}$$

In Spong et al.[25] the whole process of finding the position and orientation of the end effector is summarized in several steps. Starting with choosing the joint axes $z_0 \dots z_n$ to finally getting the transformation matrix T_n^0, as was shown in 2.3.

2.1.2 Inverse kinematics

While forward kinematics deals with determining the position and orientation of the end effector from joint variables, an inverse kinematic problem deals with finding the joint variables in terms of position and orientation of the end effector. This procedure is generally more complex and difficult to handle compared to forward kinematics, but by using the principle of kinematic decoupling we can simplify it. For kinematic decoupling to work it is required that the manipulator has a spherical wrist.

Given **H** from equation 2.3, the inverse kinematics can be stated by finding one or more solutions of the equation $T_n^0(q_1, ..., q_n) = H$. The principle of kinematic decoupling is a method where the inverse kinematic problem is decoupled into two simpler problems. These two problems are known as inverse position kinematics and inverse orientation kinematics. A geometrical approach is mainly used for inverse position kinematics, while Euler angle parameterization is used to solve the orientation problem. To make the problem easier to deal with an algorithm has been created, which can be seen below in three steps.

1. Find q_1, q_2, q_3 such that the wrist center has coordinates on the form

$$o_c^0 = o - d_6 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.4}$$

   where o_c is the wrist center.

2. Calculate R_3^0 from the joint variables found in 1.

3. Find a set of Euler angles that corresponds to

$$R_6^3 = (R_3^0)^{-1} R = (R_3^0)^T R \tag{2.5}$$

This algorithm can be found in Spong [25], where a different approach for the geometric properties has been used for computing step 1.

2.2 Robot Dynamics

This section deals with the dynamics of the robot manipulator, where the relationship between force and motion will be dealt with. Equations of motion is important to look at when designing robot manipulators and control algorithms, it is also important when trying to simulate the robot behaviour. It is for the reason of simulation with force control that in this thesis the robot dynamics are being described. Using the analytic method of Euler- Lagrange to get the equation of motion, it is first needed to find Jacobian matrices for the different joints.

2.2.1 Deriving of the Jacobian

The Jacobian matrix can be split into two, where the first is describing the linear velocity Jacobian J_v and the other describes the angular velocity Jacobian J_ω. These two Jacobians

are also different depending on if the joint is revolute or prismatic. By considering equation 2.3, it is possible to describe linear and angular velocity in the following expressions.

$$v_n^0 = J_v \dot{q}$$

$$\omega_n^0 = J_\omega \dot{q}$$

Here q is a vector of the joint variables, v_n^0 and ω_n^0 are the linear and angular velocity vectors of the end effector respectively. J_v and J_ω are 3x3 matrices that can be combined into J, the manipulator Jacobian. The angular velocity Jacobian can be found for the two cases as

$$J_\omega = \begin{cases} z_{i-1} & for\, revolute\, joint\, i \\ 0 & for\, prismatic\, joint\, i \end{cases} \tag{2.6}$$

, where z_{i-1} can be found from the equation $z_{i-1}^0 = R_{i-1}^0 k$ with $k = z_0^0 = [0,\, 0,\, 1]$.

In the case of linear velocity the terms become a bit more complicate as shown in equation 2.7.

$$J_v = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & for\, revolute\, joint\, i \\ z_{i-1} & for\, prismatic\, joint\, i \end{cases} \tag{2.7}$$

A more detailed description on how to get linear and angular velocity Jacobians is shown in chapter 4.6 of Spong [25].

2.2.2 Euler Lagrange equations

Equation of motion for robot manipulators using Euler-Lagrange can generally be written on the following form,

$$M(q)\ddot{q} + C(q,\, \dot{q})\dot{q} + g(q) = u. \tag{2.8}$$

$M(q) = D(q) + J$ is the Inertia matrix added with the Jacobian, $C(q,\, \dot{q})$ is the Coriolis matrix and g(q) is the gravitational vector. Also note that q is a set of the generalized coordinates for the manipulator, and u is the control vector. This equation is derived from D' Alembert's principle and the principle of virtual work, which is also known by the following equation.

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = u_k \qquad k = 1, \ldots n$$

L is the difference of kinetic and potential energy known as the Lagrangian function $L = K - P$, and n is the DOF. The formula for kinetic energy can be expressed as,

$$K = \frac{1}{2}\dot{q}^T \left(\sum_{i=1}^{n} (m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)) \right) \dot{q} = \frac{1}{2}\dot{q}^T D(q) \dot{q}$$

which gives the Inertia Matrix D(q) as

$$D(q) = \sum_{i=1}^{n} (m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}^T R_i(q) I_i R_i(q)^T J_{\omega_i}(q)).$$

The potential energy is more easily found from,

$$P = \sum_{i=1}^{n} m_i g^T r_c i$$

where m_i is the mass of i link, the vector r_{ci} is the coordinates of the center of mass of link i and g is the gravity vector expressed in the inertial frame. From the potential energy the gravitational vector is found as

$$g(q) = \frac{\partial P}{\partial q}.$$

The Coriolis matrix can be found by computing what is called the **Christoffel symbols**, c_{ijk}, then each [k,j] element of the matrix is defined by

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}(q)\dot{q}_i \tag{2.9}$$

with the Christoffel symbols being

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k}\right) \tag{2.10}$$

Equation 2.8 is created here using a very generalized form, that is applicable for any kind of mechanical system. Compared to the Newton-Euler method the Euler-Lagrange is a lot slower when it comes to computational speed, still it has several properties that can be exploited in regards to control design.

2.3 Force Control

Force control is vital in application where interaction between robot and the environment is critical. Examples of uses are in grinding, deburring, drilling and assembly. The best way FC is implemented is with the use of a force/torque sensor. The force signals from the force sensor is then fed into a controller, that alters the position of the end effector such that the desired force is achieved.

Force control tasks can be seen as putting constraints on the motion of the robot. The number of contraints follows from the number of reaction forces from the environment. The DOF lost for the robot motion is gained in number of forces exerted on the environment from the robot.

Definition 9.1 in Spong. et al [25] defines what is known as the reciprocity condition. In an ideal case where the reciprocity condition ($\xi^T F = 0$) holds, forces of constraints and motion constraints should not work in the same direction and no form of friction is acting between manipulator and the environment. If a material deforms then forces and motion is acting in the same direction, meaning that for this case the reciprocity condition does not hold. For experiments done in chapter 5 this condtion is assumed to hold as friction is neglected, and with the tool used the material should not deform.

The robot equations 2.8 must be modified when contact between the manipulator and the environment is considered. An additional term for the reaction torque is added on the form $J^T F_e$, where J is the Jacobian and F_e is the end effector force. The equation of motion can then be written on the form,

$$M(q)\ddot{q} + C(q,\, \dot{q})\dot{q} + g(q) + J^T(q)F_e = u \tag{2.11}$$

This equation does not say anything about the impact that occurs at the moment the manipulator touches the object, and is only valid when the manipulator is already in contact with the environment. By using a modified version of the inverse dynamics control scheme

$$u = M(q)\ddot{q} + C(q,\, \dot{q})\dot{+}g(q) + J^T(q)a_f$$

the robot equations can be shortened to

$$M(q)(\ddot{q} - a_q) + J^T(q)(F_e - a_f) = 0$$

, where a_q and a_f are the outer loop controls for acceleration and force respectively.

2.4 Signal Processing

Signal Processing deals with the representation, manipulation and transformation of signals and the data that is contained in the signals. An example of signal processing is the separation of two or more signals combined through some kind of operation, or enhancing a component of a signal. For a more detailed view of the concept of signal processing read *Discrete-Time Signal Processing* by Oppenheim. et al [21] and *Digital Signal Processing* by Proakis and Manolakis [22].

This section will look into methods of processing signals, which is central when working with force and position measurements. First Fourier series and Power spectrum density is looked at, discovering their different uses in processing of signals. Then three kinds of filters will be examined, which are the Butterworth, Chebyshev and Elliptic filters. Last cross correlation is looked at, which is used in comparing the similarity of two signals.

2.4.1 Fourier Series

The Fourier series is a linear weighted sum of sinusoids or complex exponentials. For the Fourier series all signals have to be periodic. Examples of periodic signals can be anything from square waves, rectangular waves, sinusoids, complex exponentials, etc. A linear combination of harmonically related complex exponentials can be given on the form

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t}, \tag{2.12}$$

which is a periodic signal with period $T_p = \frac{1}{F_0}$. An issue that arises for representation of periodic signals has to do with whether the series converges to $x(t)$ or not. Through a set of conditions known as the Dirichlet conditions, which is described in *Digital Signal Processing* by Proakis. et al [22], this can be seen as:

$$c_k = \frac{1}{T_p} \int_{t_p} x(t) e^{-j2\pi k F_0 t} dt \tag{2.13}$$

From equation 2.12 and 2.13 the relation of x(t) and Fourier series is summarized, with the c_k being complex valued.

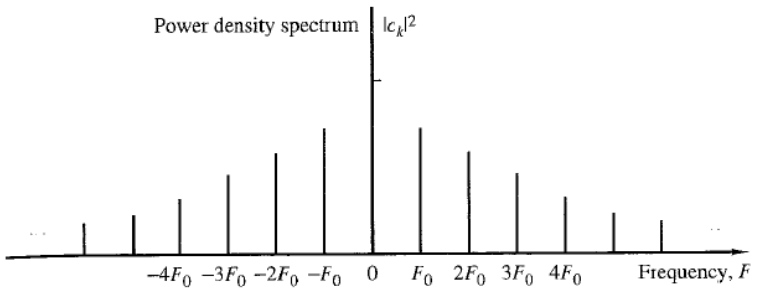


Figure 2.2: Plot of the power density spectrum for a periodic signal [22]

2.4.2 Power Spectrum Density

A periodic signal has infinite energy and a finite average power given by the equation

$$P_x = \frac{1}{T_p}\int_{t_p} |x(t)|^2 dt$$

By using the synthesis equation 2.12 combined with the equation above it is possible to derive the following,

$$\begin{aligned} P_x &= \frac{1}{T_p}\int_{T_p} x(t) \sum_{k=-\infty}^{\infty} c_k^* e^{-j2\pi k F_0 t} dt \\ &= \sum_{k=-\infty}^{\infty} c_k^* \left(\frac{1}{T_p}\int_{T_p} x(t) e^{-j2\pi k F_0 t} dt \right) \\ &= \sum_{k=-\infty}^{\infty} |c_k|^2 \end{aligned} \tag{2.14}$$

The term $|c_k|^2$ represents the power in the kth harmonic component of the signal. From this it is understood that the total average power in a periodic signal can be found as the sum of the average powers in all harmonics. The diagram of figure 2.2 shows what is called the power spectrum density of a signal that is periodic. The shape of the power distribution is depended on the time-domain characteristics of the signal, and the signal has a line spectrum from the fact that the power only exists at discrete values of the frequency.

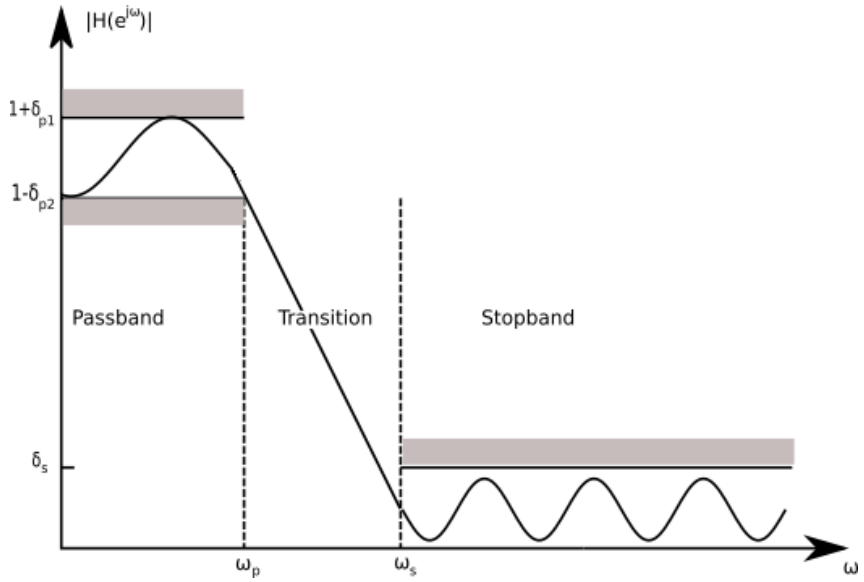


Figure 2.3: Low Pass filter tolerance scheme from p. 524 in Oppenheim [21]

2.4.3 Filter

A filter is a system that let certain frequency components of an input signal pass and rejects all other components. A classification of filters exists based on the frequency component of the filter, and it can be specified as either low pass, high pass, bandpass or bandstop or band-elimination filters. For a filter to be stable there exists two conditions. The first is that the poles of the filters transfer function has to be inside unit circle, and the second is that all complex poles and zeros have to appear in complex conjugate pairs. In design of filters it is easy to transform one type into another type of frequency selective filter.

An ideal frequency selective filter is not attainable in real application, and different kind of approximations can be done. These methods of approximation determine whether the filter is known as Butterworth, Chebyshev, etc. In specification of a low pass filter it is desired to have tolerance scheme as shown in figure 2.3, where ω_p and ω_s are the passband and stopband edge frequencies respectively. In figure 2.4 the different practical filters Butterworth, Chebyshev, etc are compared to eachother.

Filters can be divided into two classes known as finite impulse response (FIR) filters and infinte impulse response (IIR) filters. FIR filters are designed using polynomial approximation and IIR filters are obtained from an approximating transfer function. In this thesis filters are used for reducing noise on the different signals measured, so having a good understanding of the filter properties is important. An better understanding of practical filters can be found in Practical Analog and Digital Filter Design by Les Thede [26]

Butterworth

The Butterworth filter is known for having a maximally flat response, and no other filter has as smooth transition phase as the Butterworth. It also has the least distortion in the phase response, which makes it desirable in cases where no phase distortion is neces-

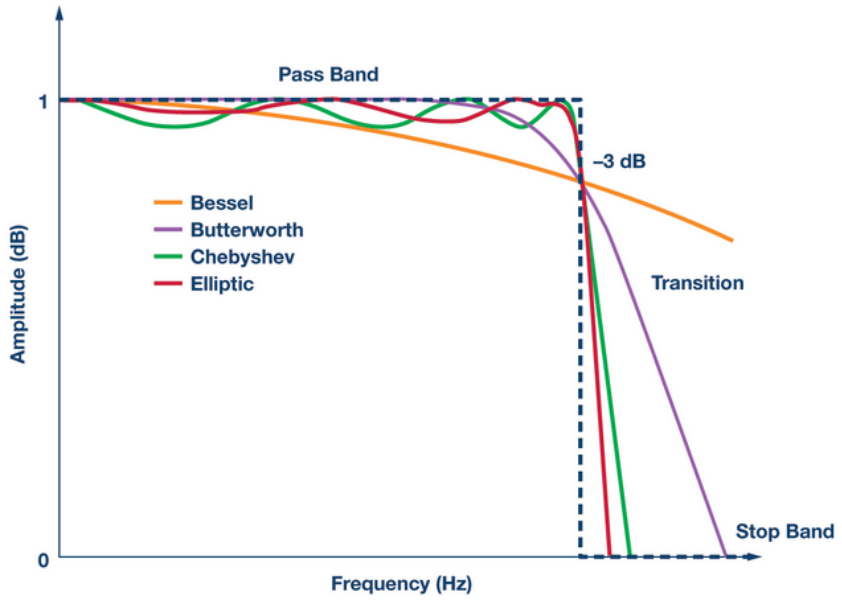


Figure 2.4: Comparison of different filters [28]

sary. Its disadvantage lies in that it has initially a wide transition band from passband to stopband, which also can be changed depending on the filter order used. A comparison of the ideal frequency response to the different Butterworth filter orders can be seen in figure 2.5, where a higher order makes the filter become closer to the ideal filter response.

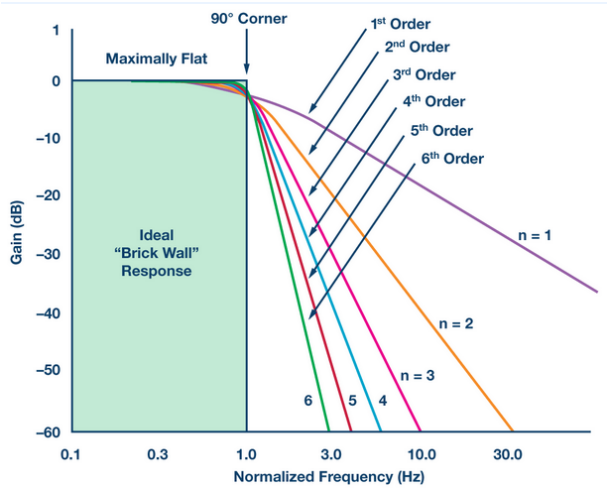


Figure 2.5: Comparison of butterworth filter and ideal filter response [28]

Equation 2.15 shows the magnitude response of the Butterworth filter for an n^{th} order filter, which is taken from Oppenheim. et al [21]. Here Ω_c is the cut-off frequency at -3 dB, n is the order of the filter, and $1 + \epsilon^2$ is the band edge value of $H(\Omega)$.

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 \left(\frac{\Omega}{\Omega_c}\right)^{2n}} \tag{2.15}$$

Chebyshev

The Chebyshev filter is known for having a steep roll off than for the Butterworth filter, but with more ripple in either passband or stopband depending on which type (I or II) is used. This filter is also closer to the ideal frequency response as shown in 2.4 than the Butterworth filter, and may satisfy user specifications at a lower order than the Butterworth. The phase response is not as linear as compared to other filters. The equation of the magnitude response of the Chebyshev filter is seen in equation 2.16, where Ω_c and ϵ are the same as for equation 2.15. C_n is the Chebyshev polynomial of n^{th} order, which can be described as a sequence of orthogonal polynomials.

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 C_n^2 \left(\frac{\Omega}{\Omega_c}\right)} \tag{2.16}$$

Elliptic

The Elliptic filter, which is also known as the Cauer filter, is a filter that is best for its selective characteristics. It is able to provide the lowest filter order for a set of given specifications than any of the other filters discussed. This is achievable because the filter has more ripple in both the passband and stopband than the rest. Designing a elliptic filter is quite difficult compared to the others, where advanced algorithms have been created for this purpose. The magnitude response given by the equation 2.17 is shown for the elliptic filter, where R_n is what is called the Chebyshev rational function of the order n.

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 R_n^2 \left(\frac{\Omega}{\Omega_c}\right)} \tag{2.17}$$

2.4.4 Cross Correlation

Cross correlation of two signals x(n) and y(n) is a sequence $r_{xy}(l)$, that is defined by

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l), \qquad l = 0, \pm 1, \pm 2, ...$$

or equivalently

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n), \qquad l = 0, \pm 1, \pm 2, ...$$

where the index l is the time shift on the lag parameter. It can therefore be used as a measure on how similar two signals are to eachother. The order of the subscripts xy determines the direction of the sequence being shifted relative to the other. This means that a property of cross correlation is that scaling either of the signals applied does not change the shape of the cross correlation.

Chapter 3

Modeling

A part of the goal for this thesis is the ability to track the position of the end effector when experimenting. To do this the robot kinematics described in chapter 2 is not enough, and a more detailed view of the IRB 4600 is needed. Additionally the task space dynamics of the robot will be explored to get a better understanding of the force control task. This is all based on the theory from chapter 2 and from the book by Spong. et al [25], but also on parameters gotten from IRB 4600 manual [2].

The IRB 4600 manipulator has 6 links with all joints being revolute. This makes the task of deriving both the kinematics and the dynamics a lot more complex than for a manipulator containing prismatic joints. At the wrist of the manipulator is a spindle motor with tool attached, which will be described more in chapter 4. As the tool does not increase the number DOF of the robot, one can easily find the position and orientation of the tool by a simple rotation and translation from the last joint of the robot.

3.1 Forward Kinematics modeling

Beginning with acquiring the position and orientation of the tool center point (tcp) using forward kinematics for the IRB 4600. The best way to start is by drawing a model of the manipulator. Here all joints have their frames attached to them and the distances from the joints are shown. The frames are built using the DH convention for existence and uniqueness of solution that was mentioned in chapter 2. This can be seen in figure 3.1, where all parameters are found from [2].

After creating the model the next step is to set up a table that includes the four parameters for joints and links of the manipulator. This is also known as the DH parameters and makes it quite simple in creating homogeneous transformation matrices for the next step. This is shown in table 3.1 and with the values shown in table 3.2. Using equation 2.2 transformation matrices for all links can be created, and from equation 2.3 the homogeneous transformation matrix from base to the last joint can be found. The position and orientation of each link is gotten by extracting the o_i from each of the homogeneous transformation matrices on the form T_i^0 where i is the current link.

As mentioned earlier in the chapter the position and orientation of the tool can be gotten from a simple rotation and translation of the last joint. The transformation matrix shown in equation 3.1 contains the operation necessary for this to work, where all its values have

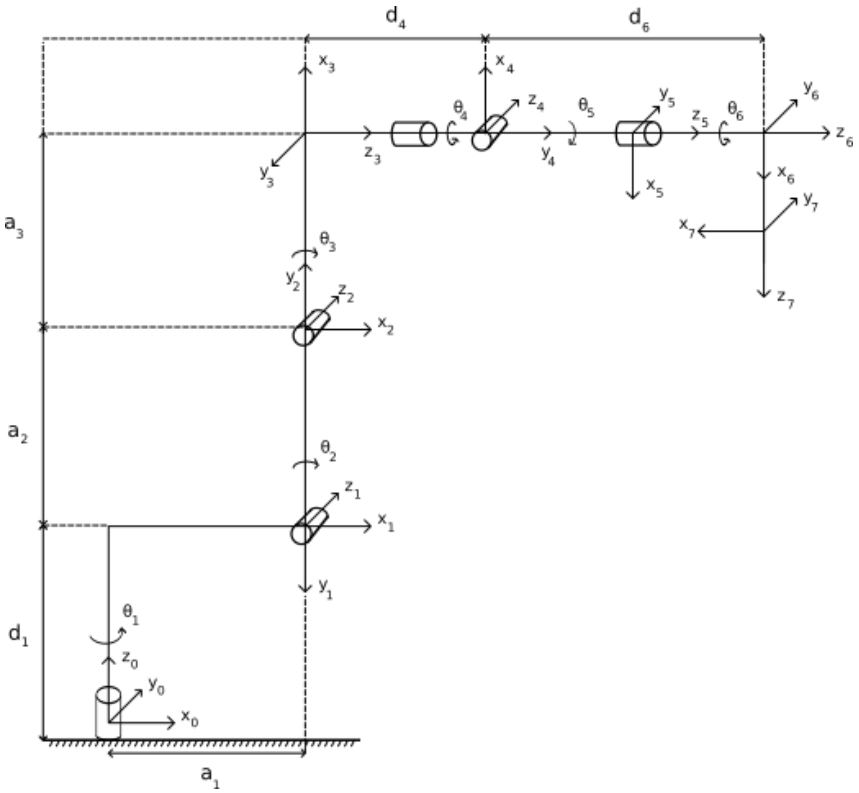


Figure 3.1: Model of an IRB 4600 manipulator

Link	a_i	α_i	d_i	θ_i
1	a_1	$-\frac{\pi}{2}$	d_1	θ_1
2	a_2	0	0	$\theta_2 - \frac{\pi}{2}$
3	a_3	$-\frac{\pi}{2}$	0	θ_3
4	0	$\frac{\pi}{2}$	d_4	θ_4
5	0	$\frac{\pi}{2}$	0	$\theta_5 + \pi$
6	0	0	d_6	θ_6

Table 3.1: DH parameters of the IRB 4600 with tool

Variabel	Value [m]
a_1	0.175
a_2	0.900
a_3	0.175
d_1	0.495
d_4	0.960
d_6	0.135

Table 3.2: Lengths of the different links found from product manual IRB4600 [2]

Variabel	Value [m]
q_{tool_1}	-0.023
q_{tool_2}	0.045
x_{tool}	0.432157
y_{tool}	0.0117039
z_{tool}	0.18366

Table 3.3: Lengths and angles relating to the tcp from joint 6

been found from measurements.

$$A_7^6 = \begin{bmatrix} cos(q_{tool_1}) & -sin(q_{tool_1})cos(q_{tool_2}) & sin(q_{tool_1})sin(q_{tool_2}) & x_{tool} \\ sin(q_{tool_1}) & cos(q_{tool_1})cos(q_{tool_2} & -cos(q_{tool_1})sin(q_{tool_2}) & y_{tool} \\ 0 & sin(q_{tool_2}) & cos(q_{tool_2}) & z_{tool} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

Here q_{tool_1}, q_{tool_2} represent the rotation from the wrist frame to the tool frame, and x_{tool}, y_{tool} and z_{tool} represent the position. With this T_7^0 can be calculated as

$$T_7^0 = T_6^0 A_7^6.$$

The position and orientation of every link and tool can be extracted from the different transformation matrices as o_i^0 and R_i^0. Numerical calculations have been done using Maple to find the different o_i, which can be found in the attachements.

3.2 Inverse Kinematic modeling

Using theory from inverse kinematics the angles of the different joints can be found for IRB 4600. Since the inverse kinematic problem is quite complex and the IRB 4600 has a spherical wrist it is possible to use kinematic decoupling. Starting with a desired position o and orientation R, which is expressed as

$$o = o_6^0(q_1, ..., q_6)$$

$$R = R_6^0(q_1, ..., q_6).$$

It is first necessary to find the wrist center o_c, which is found from equation 2.4.

$$o_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}$$

o_6^0 and R_6^0 can be found by the homogeneous transformation matrix $T_6^0 = T_7^0 A_7^{6^{-1}}$, where

$$T_7^0 = \begin{bmatrix} R_7^0 & o_7^0 \\ 0_{1x3} & 1 \end{bmatrix}$$

is the transformation matrix from base to the tcp and A_6^0 is a simple rotation and translation found in equation 3.1. After finding the wrist center o_c the three first joint angles q_1, q_2 and q_3 can be found using a geometrical approach. q_1 can be found quite simply

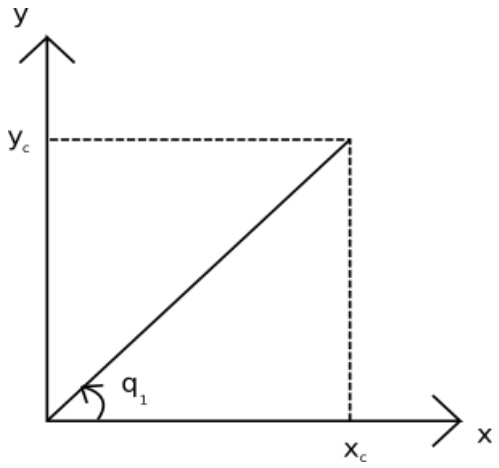


Figure 3.2: Geometrical approach in finding joint angle q_1

from looking at figure 3.2, where the $Atan2(x, y)$ is the two-argument arctangent function found in Spong [25].

$$q_1 = Atan2(y_c, x_c)$$

The elbow of the IRB 4600 is a bit more complex than for different manipulators, which

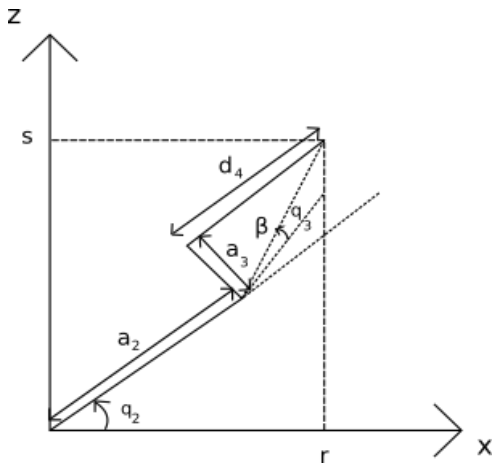


Figure 3.3: Geometrical approach in finding joint angles q_2 and q_3

makes the process of finding the joint angles q_2 and q_3 a bit harder. Looking at figure 3.3 it is easiest to find the joint angle for the third joint using the laws of cosines,

$$q_3 = Atan2(D, \sqrt{1 - D^2}) - \frac{\pi}{2} + Atan2(a_3, d_4)$$

where

$$D = \frac{r^2 + s^2 - a_2^2 - \beta^2}{2a_2\beta},$$

$$r = \sqrt{(x_c - a_1 cos(q_1))^2 + (y_c - a_1 sin(q_1))^2},$$

$$s = z_c - d_1,$$

$$\beta = \sqrt{a_3^2 + d_4^2}.$$

Here β can be described as the distance from third link to the wrist center and were s and r are the length from the second link to the wrist center in the xz-plane. After q_3 is found q_2 can be found as well.

$$q_2 = \frac{\pi}{2} - Atan2(s,r) + Atan2\left(\beta sin(-q_3 - \frac{\pi}{2} + Atan2(a_3, d_4)), a_2 + \beta cos\left(-q_3 - \frac{\pi}{2} + Atan2(a_3, d_4)\right)\right)$$

Now that the joint angles from the base to the wrist has been found, the next step is to find the wrist angles q_4, q_5 and q_6. This is done by first finding T_3^0 for the values of $q_1 ... q_3$.

Then it is possible to find R_6^3 from equation 2.5. Expressing the Euler rotation matrix R with $cos(x) = c_x$ and $sin(x) = s_x$ as,

$$R = \begin{bmatrix} c_\phi c_\theta c_\psi - s_\phi s_\psi & -c_\phi c_\theta s_\psi - s_\phi c_\psi & c_\phi s_\theta \\ s_\phi c_\theta c_\psi - c_\phi s_\psi & -s_\phi c_\theta s_\psi + c_\phi c_\psi & s_\phi s_\theta \\ -s_\theta c_\psi & s_\theta s_\phi & c_\theta \end{bmatrix}$$

where

$$\phi = q_4$$

$$\theta = q_5$$

$$\psi = q_6,$$

and using that R_6^3 is

$$R_6^3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}.$$

Then q_4, q_5 and q_6 are shown to be

$$q_4 = Atan2(r_{23}, r_{13})$$

$$q_5 = Atan2(\sqrt{r_{13}^2 + r_{23}^2}, r_{33})$$

$$q_6 = Atan(r_{32}, -r_{31}) - \pi$$

All joint angles have been expressed based on the position and orientation of the end effector, so by applying forward kinematics it is possible to validate whether the values are correct. Numerical calculations has been in both Maple and Matlab, which can be found in the attachments.

3.3 Dynamic modeling with force control

In this section the dynamics of the IRB 4600 will be explored using Euler-Lagrange method. It is desired to express the motion of the robot on the form shown in equation 2.11. Before this method can be applied the Jacobian have to be calculated for every joint using equations 2.7 and 2.6 for linear and angular velocity.

These equations depend on the position and orientation of every joint, which was found from using forward kinematics earlier in this chapter. In addition the distances to the center of mass of every link along with its inertia is found from looking at CAD files for

the different links provided by ABB [5]. The mass for the different links was also found using [5].

$$J = \begin{bmatrix} J_{v_i} \\ J_{\omega_i} \end{bmatrix} = \begin{bmatrix} z_{i-1} \times (o_{n_i} - o_{i-1}) \\ z_{i-1} \end{bmatrix}$$

where n_i is the length from the base to the center of mass of the different links. The next step is to find the potential energy for the whole manipulator by calculating the potential energy of the different links and summing them together. The potential energy of the different links are found from,

$$P_i = m_i g r_{ci}$$

with r_{ci} being the length to center of mass and m_i being the mass found earlier. When the potential energy is found $G(q)$, which is the gravitational vector in equation 2.11, can be calculated as,

$$G(q) = \begin{bmatrix} \frac{\partial P}{\partial q_1} \\ \frac{\partial P}{\partial q_2} \\ \frac{\partial P}{\partial q_3} \\ \frac{\partial P}{\partial q_4} \\ \frac{\partial P}{\partial q_5} \\ \frac{\partial P}{\partial q_6} \end{bmatrix}.$$

After finding the potential energy the next part involves finding the kinetic energy used by the manipulator, where the M(q) matrix is written on the form

$$M(q) = \sum_{i=1}^{6} M_i(q)$$

with each $M_i(q)$ being,

$$M_i(q) = m_i(J_{v_i}^T J_{v_i}) + J_{\omega_i}^T I_i J_{\omega_i}.$$

Finding the Coriolis matrix $C(q, \dot{q})$ is done by first calculating the Christoffel symbols from equation 2.10 and from defining the matrix in equation 2.9. For the IRB 4600 this is quite a heavy numerical task to calculate, because of the sheer number of DOFs involved.

$$C(q, \dot{q}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

The last expression added to the robotic equation 2.11 is the reaction torque. The reaction torque can be found from Jacobian of the last joint J and the end effector forces and torques acting on the environment.

$$J^T F_e = \begin{bmatrix} J_v & J_\omega \end{bmatrix} \begin{bmatrix} Fx \\ Fy \\ Fz \\ tx \\ ty \\ tz \end{bmatrix}$$

Numerical calculations for the robotic equation has been made in Maple, and can be found in the attachements.

Chapter 4

Equipment

To be able to carry out the different experiments in this project, the use of several different types of equipment are necessary. It is therefore valuable to learn as much as possible about them, as will be elaborated in this chapter. First the robot manipulator with tooling will be looked at and discussed. Additionally the use of both main function package for force control and position control will be explained.

After that an understanding of how the camera system works will be made, and how to best create frames will be discussed. At the end of this chapter a section is dedicated looking at issues or errors with the equipment that were found during the project.

4.1 ABB IRB 4600 Robot Manipulator

The IRB 4600 is a 6 - DOF robot manipulator from ABB, which can perform a range of heavy duty industrial tasks. Its producer ABB is one of the leading suppliers for industrial robots, and they have robot manipulators for a wide range of applications. In the robotics lab at NTNU three different ABB robots are present, the one used being the IRB 4600 while the other two are the IRB 140 and 1600.

The IRB 4600 comes in several different payload capacity and lengths, with this particular one having a capacity of 60 N and a reach lenght of 2.05 m. The manipulator consists of a base standing on conveyor platform, and six links with rotary joints inbetween. It is mainly used in material handling, cutting and arc welding, and its working range can be seen in the appendix. A picture of the IRB 4600 manipulator to be used in this project can be seen in figure 4.1.

For all robot manipulators from ABB the manipulator is accompanied with a controller cabinet and a flexpendant. The controller acts as a power and signal source for the robot, and every command the robot gets comes from the controller. This is also where the manipulator can be started or stopped.There exist emergency switches on both the cabinet and the flexpendant that cuts the power supply for the robot should an uncontrollable situation ever occur.

The flexpendant is the users way of communicating with the robot. This device can be seen in figure 4.2. Its main features can be used by working with touch screen. As the controller is set to manua,l an enabling button has to be pushed every time the robot is

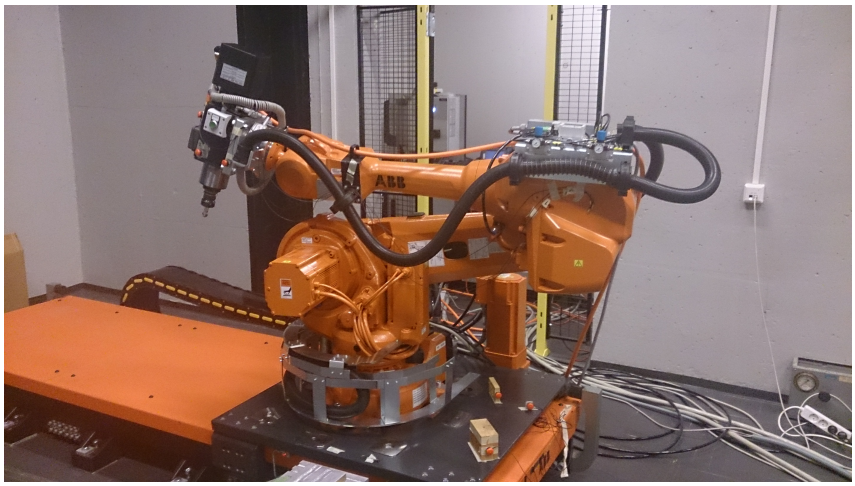


Figure 4.1: An IRB 4600 manipulator with tool on conveyor belt

moved. The robot can be moved by hand with the use of the joystick, or a path can be programmed on the flexpendant. In addition robot programs for position control can be created using RobotStudio, which is further described in section 4.3.

Figure 4.2: Flexpendant used in control of robot manipulator [4]

4.1.1 Creating paths using Flexpendant

The procedure of using the joystick to move the robot in different directions is called jogging. For this particular manipulator it is possible to either jog each of the joints independently, where the arm and wrist of the robot can be moved separately. It can also be moved linearly in a three dimensional space with the base as origin, or using reorientation around the tcp.

Creating paths can be programmed using program editor in the ABB menu, which uses a high level language called *Rapid*. Here several built in functions are used, and one only needs to specify parameters such as target location, velocity of the robot etc to make it work. Some of the easiest functions are the MoveL and MoveJ commands, which makes the robot move either linearly or using the joints independently.

A target can be also found by jogging the robot to the exact position and using modify path option in program editor. This method however depends on how accurate the user is, and it is not recommended. Each target value along with its speed and other values can be changed in program data.
The created path can be executed using buttons on the flexpendant, and it is always smart to use a lower speed when initially executing a path to avoid unnecessary.

4.1.2 Calibrating the robot

Every time the manipulator is turned off completely the encoders doesn't know exactly where it is at start up, and it is therefore to calibrate the encoders. This is done by jogging each of the joints for the manipulator manually in a set of positions. To understand where to position each of the joints, look for markings of the kind that can be seen in figure 4.3. After every joint is put into position, it is required to run the calibration from the ABB menu on the flexpendant.

Figure 4.3: Calibration mark for the fourth joint on the iRB 4600

4.1.3 Tools and sensors

From the end of the last link of the manipulator there is a tool with force sensor attached. The tool is a spindle motor created by Colombo for the purpose of removing material. The spindle can achieve speeds of up to 18 000 rpm, see [9] for more information of the spindle motor. The spindle motor has the possibility of changing tool end, such that alternative tools can be attached on the device.

Currently a rod wheel tool was created to fit into the spindle motor, and is to be used for force control experiments. This tool end does not depend on the use of the spindle motor, and has therefore no use in chapter 5. Figure 4.4 shows a close up of both the spindle motor and rod wheel tool used for experimentation.

The sensor attached between the robot and the spindle motor is a 6 DOF force/torque sensor that is called Omega 160 IP65, and is created by ATI Industrial Automation. The sensor consists mainly of a transducer and power supply with cables. The transducer is a compact structure that converts force and torque into analog strain gauge signals. For specifications of the sensor see manual in [7].

Figure 4.4: A spindle motor with a wheel rod tool attached

4.1.4 External control

In chapter 5 where force control is to be experimented with, the method from external control is used for extracting position measurement. External control makes it possible to both read and modify different properties such as positions and velocities. Its main usage lies in integrating controllers modeled in simulink to the robot control through a real time workshop and an interface called opcom. Modeling these kinds of controllers are beyond the scope of the thesis, for more information specified to this topic see Force Control Interface, Dressler [12].

Logging signals from the robot controller is run independently from control execution,

and when the program is ran all signals specified by the control model is logged. This data needs to be converted to .txt file before it can viewed, and before logging it is necessary to specify both sampling time and time duration of the experiment. An example of code for this procedure is shown in [12].

4.2 Machining FC

Robotic machining is used on all kinds of material removal processes, surface processes and surface finishing. In the robotics lab at the faculty grinding procedures has been done earlier with the IRB 4600. Most of the time using robotic machining, a constant applied force is desired. Therefore several force control applications by different companies have been developed to handle this.

One of these force control applications is the Robotware Machining FC created by ABB, which is compatible with several types of industrial manipulators. Some examples of where Machining FC function is used in the industry are for grinding, drilling, milling and polishing operations, see figure 4.5 for an example of a grinding procedure. The function package consists mainly of the two parts called FC Pressure and FC SpeedChange.

Figure 4.5: Example of an grinding operation using robotic manipulator [13]

FC Pressure makes the robot sensitive to contact forces, meaning that it can change its position in order to apply a constant force on a surface, even if this position is not currently known. The path created when FC Pressure is used can be both linear and circular. Most common uses are in grinding, polishing and deburring processes. FC Pressure will be mainly used in this thesis, since a grinding tool which applies a constant force along a linear path is to be used for experimentation, see chapter 5.

FC SpeedChange is used in processes where a certain level of accuracy for the path is important, and it works by reducing the path speed when a certain value of the machining forces are exceeded. As mentioned above FC SpeedChange is not important for the kind

of experiments considered in this project.

Before force control can be activated using Machining FC it is required to calibrate the force sensor. This is done automatically by the software using a function called FCCalib, which also compensate for sensor offset and gravitational force. For FCCalib to work as intended, it is necessary to specify data for the load used. Load data can be easily retrieved by performing a load identification with the function FCLoadID. To achieve the least possible offset error one needs to perform the load identification as close as possible to the workspace without causing contact between tool and the environment. In addition tool data needs to be specified before creating paths.

4.2.1 Creating procedures using Machining FC

A force control procedure using Machining FC can be divided into four different steps before it can be executed. It all begins with the user having to teach the robot the desired path. Figure 4.6 shows the screen for teaching a path. This function includes the possibility to view the path in three different views.

It can be seen from figure 4.6 that the path is divided into the three phases, where

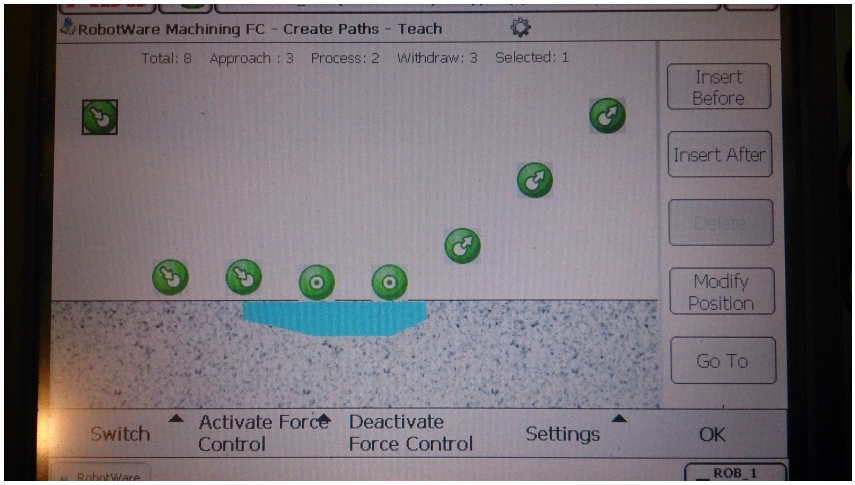


Figure 4.6: Interface for the teaching in FC Machining

the first is approaching the workobject, the second is processing workobject and the last is withdrawing from the workobject. How teaching basically works is by jogging the robot along the desired path, and then at different points update the position. By following the recommendations listed below one should be able to create a functioning path for the force control procedure.

1. Jog linearly along the whole path.
2. Get close to the workobject early in the path, so that the force control is triggered correctly.
3. Always keep the tool in an orientation perpendicular to the workobject.

4. Modify the speed such that it minimizes impact forces when approaching the workobject.

5. Apply forces only in the direction wanted, and be aware of which way each direction represents.

From the process of teaching the robot its desired path, the next step is to make the manipulator learn the taught path. This is an automatic process that uses the data from teaching procedure to record a series of targets along the path. These targets are chosen to optimize the performance of the procedure. In the same format as with teaching, learning works by making the manipulator run the new optimized path.

When the new path is learned the next step is to export the learned path for execution. In this operation a rapid module is created that contains the whole path with all the parameters included. An example of the Rapid code produced for a FC procedure can be seen in the appendix. It is also possible in this operation to compensate for wear and tear that might have happened.

The next step is testing the learned path that was exported, and while this step is

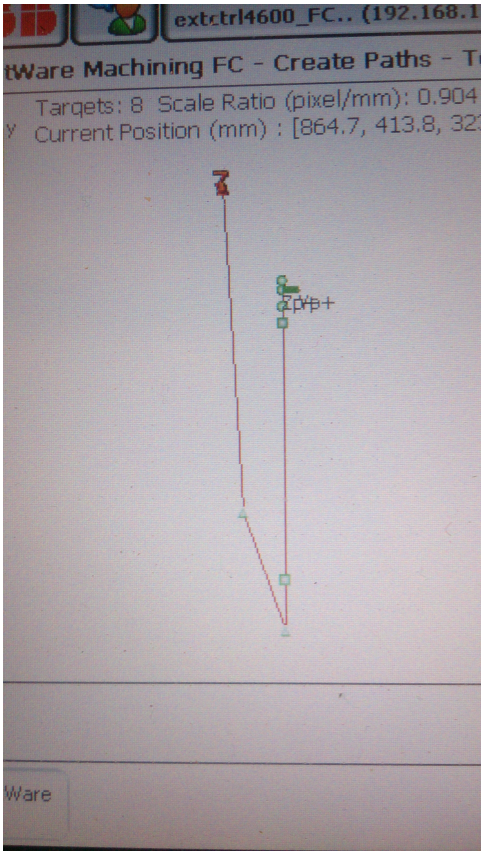


Figure 4.7: The path tested viewed in HMI

not needed for the executing of the rapid module, it is still quite important to conduct. The testing is usually carried out with a lower speed than needed, and indicates how well the path will execute. Figure 4.7 shows the path that is to be followed in a human machine interface (HMI).

4.2.2 Logging force measurement

The following section looks at the way of gathering measurements from the force sensor. First of all a computer needs to be connected to the robot controller, so that it has the possibility to collect force signals from the sensor. ABB has also created a software called Test Signal Viewer (TSV) that can store or modify all signals collected from the robot controller. Test signal identities for the different forces and torques signals can be found in Force Control Manual [3].

TSV also has the means of applying simple filters on-line, which can be used for comparison purposes. The software has a sampling time of 0.504 ms, and will need to be downsampled compared to the position measurements for robot and camera. Figure 4.8 shows how the interface of the TSV looks like. The force signals are stored as text files that can be read by Matlab. More information about the TSV can be found in Test Signal

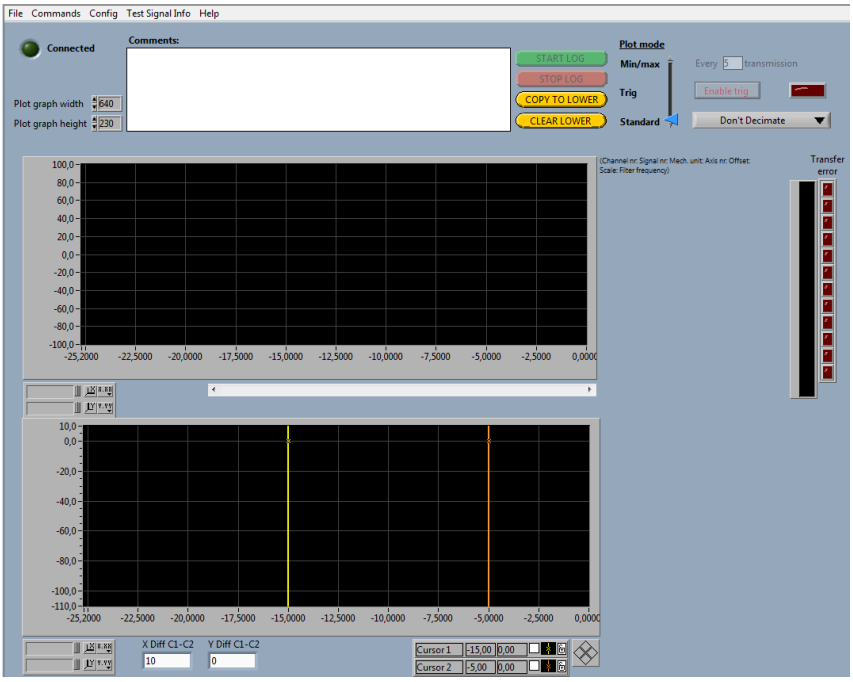


Figure 4.8: Interface of TestSignalViever software

Viewer Manual by ABB [1].

4.3 RobotStudio

RobotStudio is a simulation and offline programming software created by ABB robotics, that can be used to simulate different scenarios for a robot manipulator. The software is visually based, meaning that most tasks and actions performed can be created just moving in the view screen. Robotstudio uses RAPID as programming language, which is quite useful as you have the means of editing the code used by the flexpendant.

Robotstudio has an extensive library of different manipulators, tools and other equipment that can be used, additionally it is possible to create objects and tools using the software. While the software is not directly used in this thesis and is more used with the

topic of position control, it still valuable to have an understanding of it. There also exists an add-on called Machining PowerPacs that has the possibility to simulate scenarios with force control, which has not been used in any experimentation here.

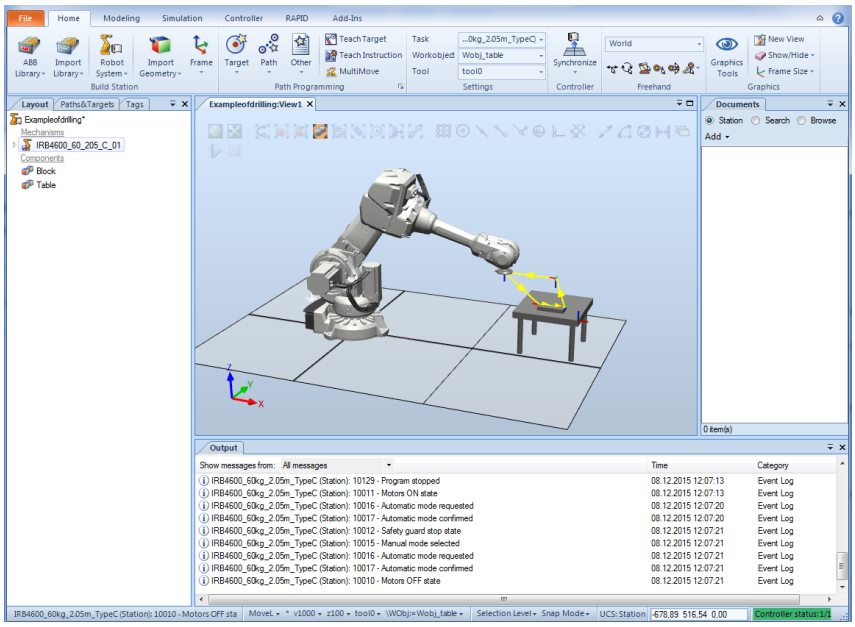


Figure 4.9: An example of a RobotStudio program

4.3.1 Modeling in RobotStudio

As mentioned RobotStudio also includes the possibility to create objects similar to that of an AutoCAD software. These objects can then be used to create a workspace for the robot manipulator to operate in. There are several ways for an object to be created, for example building solids or extruding a surface. From these objects tools can also be made, where it is important to look into the precision needed from the tool. When both tool and workspace has been created that satisfies the desired requirements then the station is complete.

4.3.2 Creating trajectories and paths

After a robot manipulator with desired tool and workobject is chosen it is possible to move the manipulator to different positions and orientations either by jogging and/or creating trajectories. Jogging works by choosing one of the several jog options in the freehand menu, and then just freely moving the different joints on the manipulator by hand.

Creating trajectories and paths can be done by first creating targets on the desired surface. A path can then be made by using several targets and programming the robot to move between them in a desired order. A configuration is also needed between moving to each target as this is done to create a reachable trajectory. Same as when using the flexpendant one has to differ between linear movement and joint movement, where in several cases using linear movement will not work while the opposite is true for joint movement.

4.3.3 Simulation

An important reason for using Robotstudio is that one has the possibility to simulate the motion of the manipulator as it moves along a path. Synchronizing the station with the trajectory wanted to a virtual controller gives the means of simulating the manipulator, a RAPID code of the path is produced and can further be exported to the flexpendant. It is important to note that a well modeled station with real physical parameters is needed if we want to study collision, path precision etc in the simulation.

4.4 Nikon K-Series Optical CMM

The K-Series Optical CMM is a highly accurate camera system used for tracking motions, and in this case the motion of a robot manipulator. The camera itself consists of three lenses, where each lens can track several LEDs used for the creation of dynamical frames. The K610 camera has the possibility to cover 17 m^3 from a distance of up to 6 m.

The camera is connected to a device called the controller, which provides the camera with power and communication signals. The controller is the main hub of the K-Series system, where it also synchronises all LEDs through strobers and communicates with the measurement PC. Figure 4.10 shows the whole system, and for more details see K-Series Training manual [19].

Before the camera is able to track any kind of motion it needs to be calibrated to get any valid measurements. If neither the camera nor the space probe has been calibrated for a while, it is best to start with the camera calibration before space probe calibration.

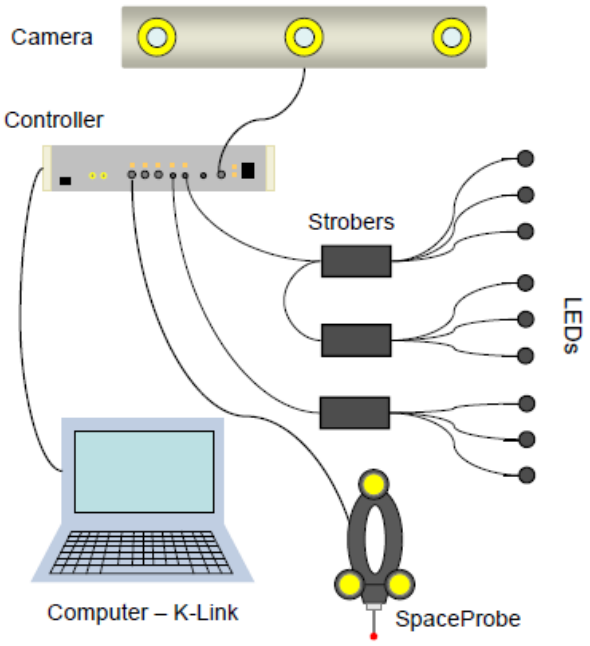


Figure 4.10: View of the whole K-Series system

4.4.1 Camera Calibration

Starting with the calibration of the CCD camera it is possible to adjust for the camera geometry, and take into account changes in temperature and orientation. The calibration process depends on the use of a software called K-CMM and the K - Reference application. In camera calibration a tool called the K - Ref Bar is used. The K - Ref Bar is basically a rod with built in LEDs at both ends, where the camera has the possibility to detect the different LEDs. The LEDs have to be visible for the camera every time a measurement is taken. The calibration requires in total 23 measurements before it is finished, and the bar has to be held in multiple orientations and positions. The interface of the application as shown in figure 4.11, shows three indicator sections which can be listed as:

1. Shows error in position (up, down, left, right, forward, backward.
2. Shows error in orientation (rotation)
3. Shows where the LED indicators show the error in LED angle to the camera.

When the different errors are below a certain threshold value the application will automatically start measuring.

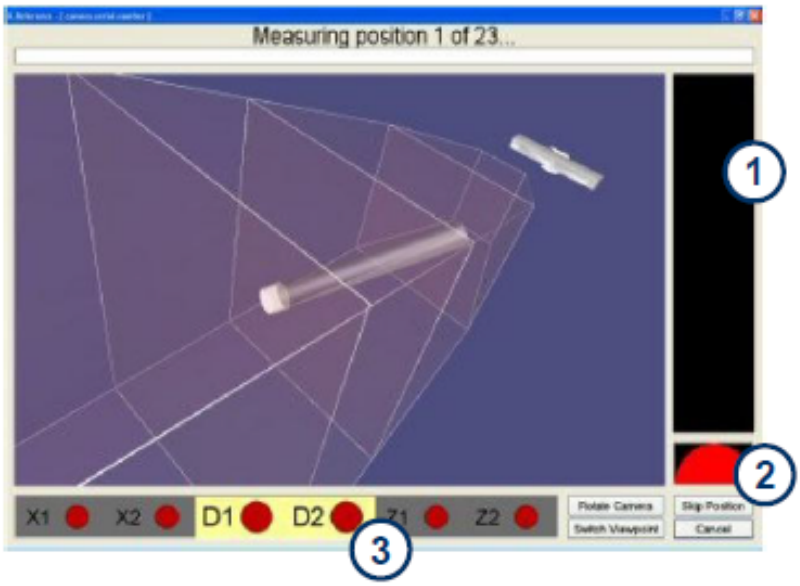


Figure 4.11: K - Reference application for camera measurements

4.4.2 Probe Calibration

After finishing calibrating the camera it is desired to calibrate for the space probe. This is done to achieve smallest possible error margin when measuring the different frames. To get the best results in calibrating the use of a tripod is necessary.

Starting K-CMM and Probe calibration begins the procedure of calibration. Figure 4.12 shows the probe calibration application, and the most important features are listed below.

1. Shows how many measurements that are taken

2. Shows which LEDs on the space probe that are visible to the camera
3. Shows the movement from front to back, side to side and also the total error.

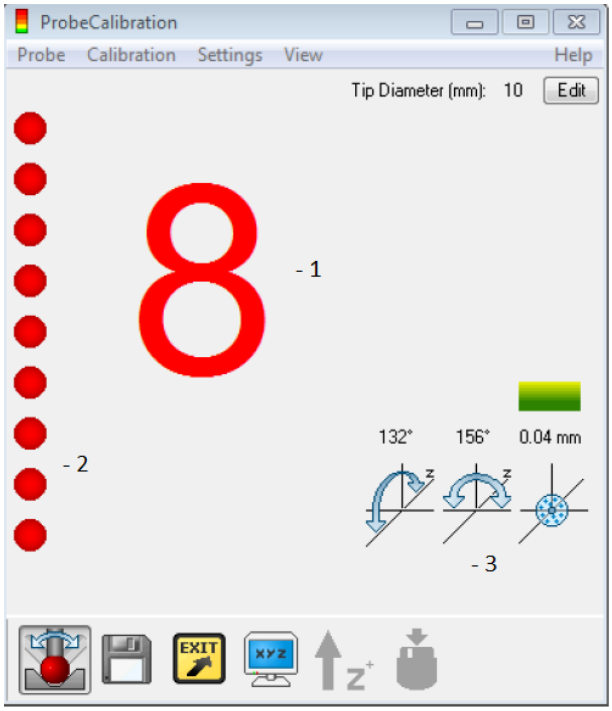


Figure 4.12: K - Reference application for camera measurements

A total error of less than 0.04 mm is ideal when the movement bars are fully decreased. Measurements can be taken holding the probe and tip in steady position and pushing one of the inner buttons on the probe. A minimum of atleast 8 measurements should be taken, each in the different movement direction. When all the measurements are taken the procedure can be ended by pushing the outer button on the space probe. If necessary one can delete a single measurement by holding down the outer button for a longer period of time.

4.4.3 Creating frames

Before camera measurements can be taken, it is required to create frames that can be measured. This is done using a software called Geoloc, where different geometrical shapes, ie circle, line, point, can be created using the space probe. Figure 4.13 shows the geoloc interface for measurements, and by combining the different shapes created it is possible to define frames.

The frames can be chosen either as dynamical or static frames, where dynamical frames will be used most in this thesis. It is also required that each frame connected to atleast three LEDs, which are visible to camera at all times. All frames created needs also to be calibrated before they can be used for measuring, as it is very hard to define a perfect frame only by measuring with space probe.

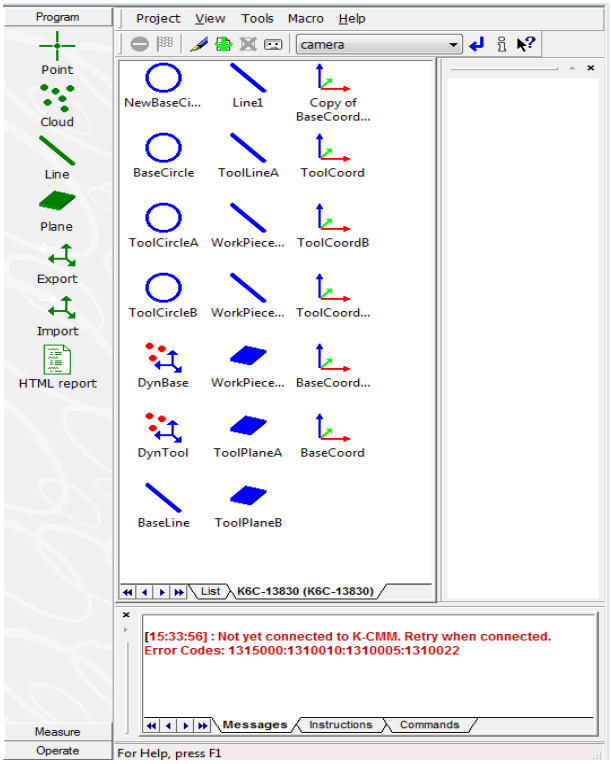


Figure 4.13: Interface of the geoloc software used in creating frames

Below shows a list of the step for step procedure in how to create the necessary frames for measuring the position of the IRB 4600. For best possible measurements of geometrical shapes, it is important to keep the space in the same orientation towards camera during the whole process.

Here a frame for both the tcp and the base is created, where the base will be declared as the dynamical reference frame. Each of the geometrical shape or figure can be measured in any given order. Figure 4.14 shows the desired way in creating the frames.

1. Starting with the base, the first geometrical shape to be created in this thesis is the circle. This is used to define the origin of the frame, which will be the same as the center of the circle. It can be done by measuring atleast three points in space, such that a half circle is defined. For IRB 4600 this is not a very good solution as the measurements gathered with the space probe would be quite inaccurate. In this thesis the circle is instead defined by taking measurements while jogging the first joint of the manipulator which has circular motion.

2. Defining a plane for the base can be done by taking four measurements with the space probe in a square form on the base platform. Experimentation done by Petter [17] however found that the best way to define the base frame with highest accuracy was to create the plane on the workpiece. Measurements should then be taken by measuring at edge of each side of the metal plate. This is done because the workpiece is more leveled than the base platform, and also because the base platform is not totally rigid. The plane is used to get the z- axis, which is supposed to be perpendicular with the surface of the plane. Because the workpiece is used for the plane, a translation has to be applied to get it at the base of the robot, which is applied in the z axis for 280 mm downwards.

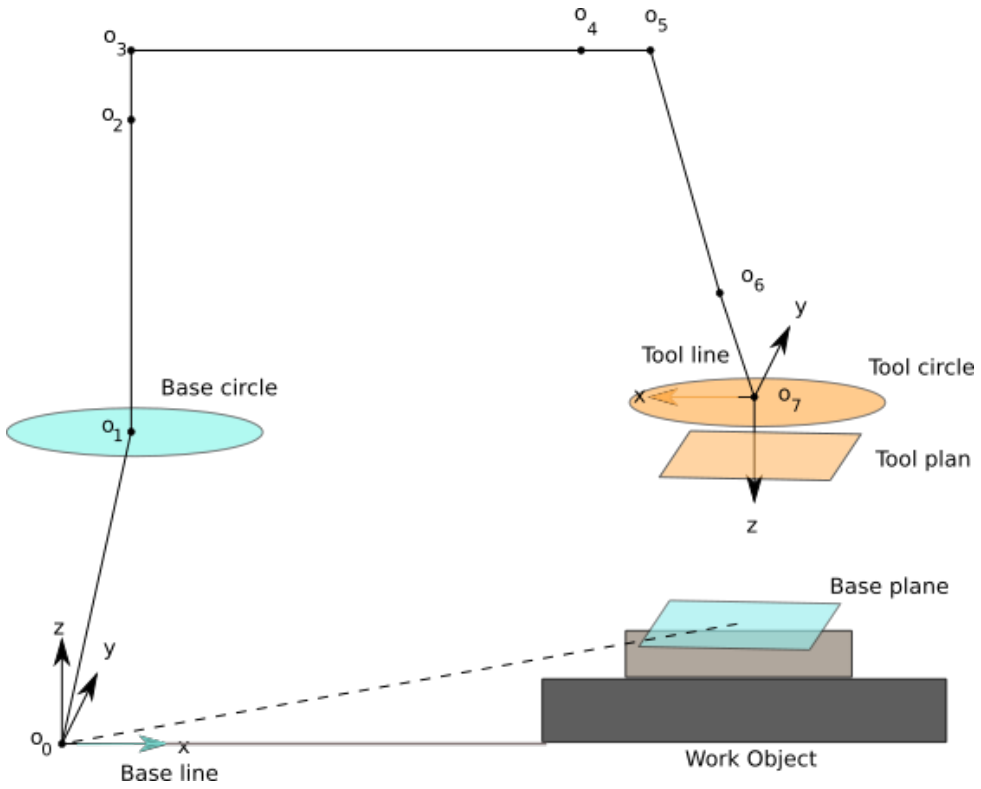


Figure 4.14: Shows how to assign frames for the tool and base of IRB 4600

3. Remaining for the base frame is to get the x- and y- axis, and the way to get them is by defining a line for either of the directions. The last direction can then direcly be taken by using the right hand rule, which is done automatically by geoloc. For this project the line is chosen to be in the x- axis, that is the same x- axis as measured directly from the robot. The line needs atleast three points measured with the space probe to be feasible. As the line only represent one axis of the frame it was most sufficient to measure it along linear path on workpiece.

4. The tool frame has almost the same procedure as the base, with defining a circle, a plane and a line. The circle is best taken by measuring several points with the space probe around the circular shape of the spindle motor.

5. A plane for tool frame can be gotten from measuring four points atop of the spindle motor, and then translating such that the frame is at the tcp.

6. The last geometrical shape created is a line for the tool frame. This can also be done either for the x- or y- axis, and as with the base the line was defined in the x-axis. For simplicity and optimal accuracy the same line for the base was used.

7. Frames for the base and tool are formed by creating two reference frames containing the geometrical shapes that were defined in the earlier steps. Here it is important to know which direction the different axis points, so that the frames are defined right. As mentioned earlier the line is used to produce the x- axis, while the plane defines the z- axis and the origin is defined by the circle figure.

8. After the reference frames are created the next step is to create dynamic frames, which uses the reference frames created in the last step. It is also possible to create

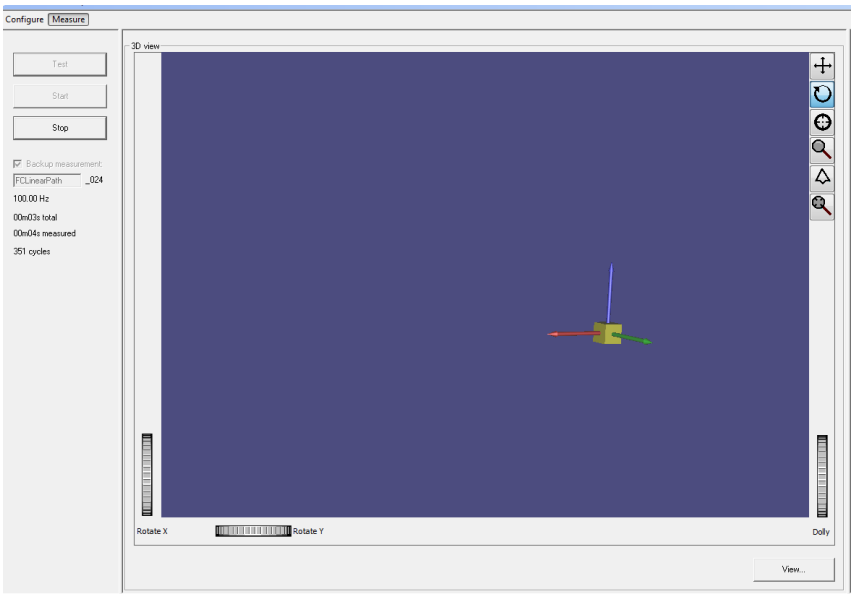


Figure 4.15: DMM interface when measuring frames by using the camera

a static frame for the base, however this is not done as the base is not totally static. At this stage one should not move the robot in any direction, as the dynamic frames are created by the camera in comparison with the LEDs placed on both tool and base.

9. The frames for both the base and tool is now properly defined. A last step before using the frames for measurements is to calibrate. This calibration is done so that the frames fit perfectly with the robot base and tcp. If the frames still does not fit for the tool and base consider doing the step 1-6 again. An other way of increasing the accuracy of the measurements is to add additional LEDs to the different frames.

Now that the frames are perfectly defined and fit the following statement should hold:

$$o_0 = Camera\,base\,frame$$

$$o_7 = Camera\,tool\,frame$$

The frames can now be used for measurements which is done in DMM software. This software loads frames created in geoloc and places them in the three dimensional space relative to the LEDs. One always has to specify the number of LEDs used at each strober when creating a new project. Initially a camera reference frame is used as world reference, so this needs to be changed to the dynamic base frame.

4.4.4 Logging using the camera system

This is easily done using the DMM software, where frames are defined as in section 4.4.3. After measuring a .mat file is created storing measurements for both the base and tool frame. The sampling time of the camera is equal to 10 ms, which is a lot lower than for the other two signals. The .mat can be easily viewed in matlab. Figure 4.15 shows the interface when measurements from the camera is taken.

4.5 Troubleshooting

During the time spent working with both the camera and robot several issues occured that resulted in delay of the experimentation. Solutions to these issues were hard to be found in any of the different manuals related to the equipment used. In this section these issues will be discussed, and ways to avoid or fix them will be presented. The first and most important step to do is to always check manuals supplied by the manufacturer and to contact those responsible for the equipment whenever a problem occurs.

4.5.1 Issues concerning robot manipulator

1. The robot controller is set in emergency stop, even if none of the emergency stop switches were turned on and restarting the system doesn't return it to running mode.

There are many reasons for the emergency stop event to happen, usually it should happen if the user deems the robot motion as not safe. In this case however it happened from pushing the enabling device repeately in short duration of time. This was not known at the time causing quite some confusion, as the robot was still within its working range and not close to any kind of object.

The problem was resolved by help from the technical responsible at the lab, and the element that appeared to stop it from functioning again was that the IRB 4600 had an additional emergency switch for the tool that had not been turned off. The best way to avoid this kind of problem is to not rush anything when doing operations that require the enabling device.

2. Using FC: The robot won't follow the path taught when learning.

What happened here was that the robot didn't follow the path during the phase between approaching the metal and processing along the metal. Instead of following its taught path the robot went in a different direction, and ultimately stopping while producing an error message. The reason for this happening was that the robot tcp was not close enough to the workobject, when force control was started. It therefore made it impossible for the robot to sense a force in the right direction. When this was established it was easy enough to fix it by letting the manipulator be closer at an earlier point.

3. Issues when loading models in opcom.

An issue that occured several times when trying to load models to the robot controller was that the drive module stopped communicating with the flexpendant. Fixing it was easy enough as one only needed to warm start the flexpendant, but this issue seemed to occur on models that had worked earlier. One of the ways to stop this from ever happening again is to always restart the external computer, when performing new measurements.

4.5.2 Issues concerning camera system

1. Connection to camera is lost after using DMM and Geoloc for a while.

This seemed to happen from time to time and the only way of solving it was by restarting

the computer. It was most probable caused by the change in temperature at the lab, and is therefore quite hard to avoid.

2. Camera is not able to see LEDs at all times.

After a time of not being used at all, the camera started to flicker when measuring LEDs. This could also be seen from the space probe, which would flicker between green and red when used. Most likely the cause of this is from being inactive for awhile without being turned off and the lenses were not covered causing dust to settle.

Best way to avoid this is to always remember to turn off the controller when the system is not to be used for an amount of time, which was not done in this case. When the problem has already occured the way to fix it is by using a cleaning kit that comes with the camera.

Chapter 5

Force Control Experiments

This chapter is a dedicated to force control experiments that has been done earlier by Petter Kvernberg [17], where the behaviour of the IRB 4600 with force control will be investigated. Several FC experiments with different forces will be conducted, and results achieved will be compared to earlier work. Additional experiments will be conducted and processed to give a more wide perspective of force control use. The differences of these additional experiments mainly lies in using different paths and velocities.

The main idea behind doing all the experiments is to study the performance of the manipulator while in contact with the surface of a metal object without deformation. Instead of using the spindel tool a different tool will be used, that have the capability of being somewhat similar to machining processes while also achieving a better case of repeatability. Data from both the robot and camera will be used for analysis. The goal is to verify the results of experiments done earlier, and to further investigate a way to compensate for unwanted behaviour of the robot in a machining process. This study will focus more on the aspects bound with force control, and topics such as impact will be included.

This chapter consists of three parts, where the first one looks at the planning and the execution of the experiments, the second part looks at the results of the processed data, and the last is a discussion of some of the most important aspects in this chapter. Everything done and experimented with in this chapter was supposed to be a precursor to the next experiments that included the use of a drilling spindle. This was however cancelled as the necessary training and safety concerns using the drilling spindle was not provided for during the time of the project.

5.1 Planning and Execution

Similar to earlier experiments, a scenario has to be developed before implementation of the experiment can be done. Using the same tool created for the last experiment a new linear path will be made, and tested for different pressure forces. Measurements will be gathered from the robot position, which will be done using external control, camera position and pressure force signals taken from Test Signal Viewer. As the different measurements have different sampling time they have to be downsampled, thus both robot position and pressure force signal sampling time will be downsampled to the camera position sampling time of 100 Hz.

	Path 1	Path 2
Nr	x, y, z	x, y, z
	Approach	
1	835 mm, 410 mm, 330 mm	835 mm, 410 mm, 330 mm
2	924 mm, 449 mm, 234 mm	924 mm, 449 mm, 234 mm
3	930 mm, 449 mm, 173 mm	930 mm, 449 mm, 173 mm
4	940 mm, 449 mm, 173 mm	940 mm, 449 mm, 173 mm
	Process	
5	949 mm, 449 mm, 168 mm	949 mm, 449 mm, 168 mm
6	1103 mm, 449 mm, 168 mm	1103 mm, 449 mm, 168 mm
	Withdraw	
7	1080 mm 420 mm 230 mm	1130 mm, 449 mm, 220 mm
8	835 mm, 410 mm, 330 mm	1080 mm, 420 mm, 230 mm
9		835 mm 410 mm 330 mm

Table 5.1: The different paths used for experimentation with force control

Planning and execution of the additional experiments follow a rather similar procedure to the initial experiment, and only certain variables are different. Friction will not be taken into account, because of minimized friction forces caused by having the tool face the direction of motion. When it comes to execution of the experiments it is necessary to look at the right way of controlling the robot without causing damage or errors, which wil alsol be elaborated on in this part of the chapter.

5.1.1 Trajectory planning

The process will only consist of linear motion, where a constant pressure force is applied in the direction downwards on the metal plate. This means that the force applied will work in the same direction as gravity. For all the different experiments done the approach and process part of the path is almost the same, while when it comes to the withdraw part it differs in how abrupt the trajectory goes back to starting position. Table 5.1 shows the different points in space referenced from robot base, that are to be used in creating the different trajectories, and is shown for two different paths.

From the table it is necessary to state that force control is onsly used in the process phase, while for the approach and withdraw phase position control is used. The two paths will be tested for several different forces ranging from 1 to 30 N. In addition along the trajectory different velocities will be used, and as such a slower velocity is used at the process phase.

It is desired to approach and withdraw exactly 1 cm from the edge of the metal object from opposite sides, assuring that the wheel is in touch with the objects flat surface at all times during process. This was also done in order to know exactly at what points in space impact between the robot and the environemnt is supposed to happen.

Creation of trajectories are done simply by using Machining FC software on the flexpendant, where it is required to first teach the path to the robot before being able to execute it. Chapter 4.2.1 describes the whole process from teaching to execution, and in

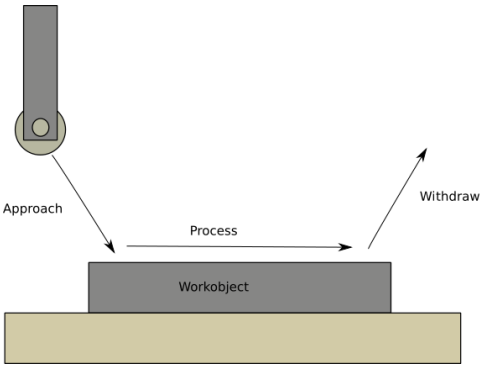


Figure 5.1: The path of the robot along the workobject split into several phases

the appendix is an example of the Rapid code generated from Machining FC [3].

When creating the path it was essential to move the robot as close as possible to the workobject during the approach. This ensured that the robot moved towards the metal plate in right way when force control was activated in the learning the robot.

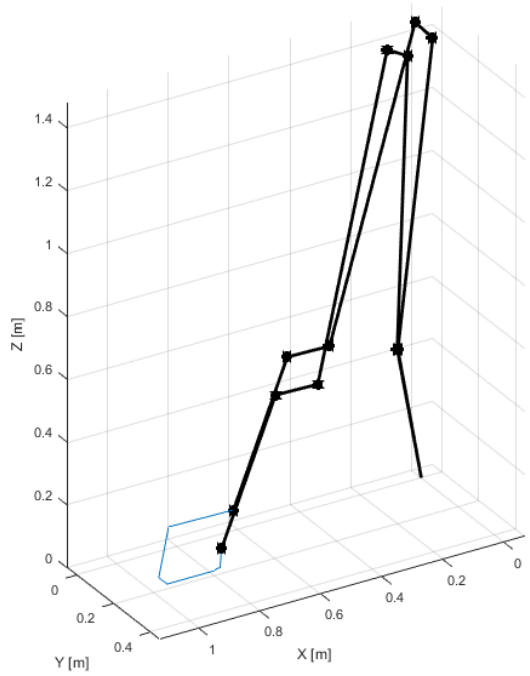


Figure 5.2: The whole path of the force control experiment for path 2

5.1.2 Camera calibration and setup

The camera is a vital part in the experiments for examining the behavior of the industrial robot during grinding processes. For more information about the K-Series camera see chapter 4.4. There was only one position available for the camera, that made it able to capture the whole workspace, which can be seen in figure 5.3.

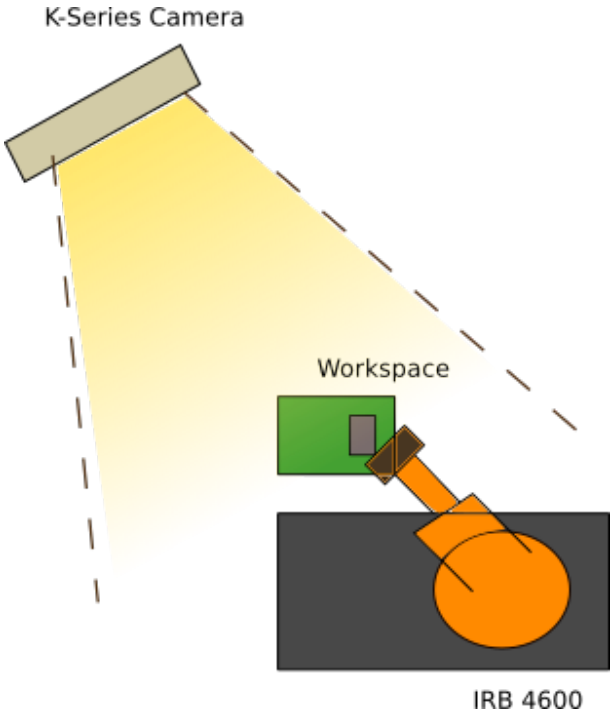


Figure 5.3: Position of the camera at the robot lab

When the camera is setup in the right position and everything is working, the first thing to do is to calibrate it. This is done by following the procedure in chapter 4.4.1, and ensures that the camera is measuring with as small error margins as possible. In addition to calibrating the camera it is also necessary to calibrate the space probe. This device is used for measuring points in space where the camera is operable, and with the help of a certain type of special LEDs, it gives the possibility to create coordinate frames. Figure 5.4 and 5.5 shows the placement of these LEDs. The fourth LED on the tool frame in an additional experiment to see if it could improve the quality of measurement.

How to create frames for both tool and base are shown in chapter 4.4.3. To summarize all that is needed to create a frame is a line, plane and circle. The first two are used to create the axes, while the last defines the origin of the frame. The alteration in creating base plane at the work object is important to note. It was done based on earlier experiments on the subject, and was supposed to give more accurate measurements in accordance with the robot measurements. The frames with this change can be seen in figure 4.14.

It is difficult to get the frames to fit the tool and base immediately, therefore it is necessary to translate frames in the different axis, which is known as calibrating the frames. Calibrating the frames for the tool was done at the exact point where the tool hits the workobject. By doing this it ensures that the measurements are optimal for the space around the workspace where force control is used for processing.

Figure 5.4: LED placement for the tool, note additional LED

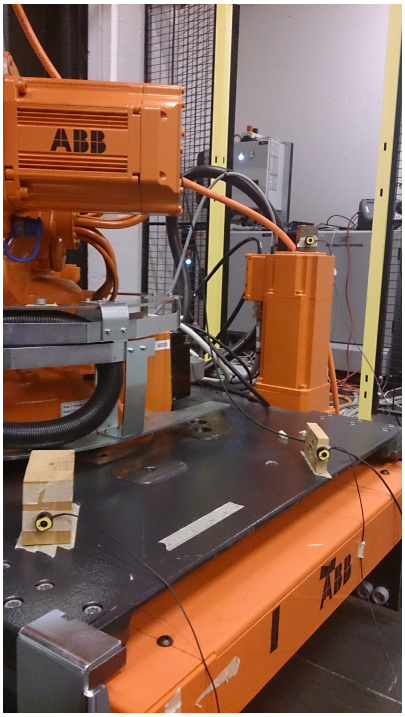


Figure 5.5: LED placement for the base

5.1.3 Measurement setup

Being able to collect and process measurement data requires a good setup, and it is important to know exactly what is being measured at all times. There are three different kinds of measurement data that are necessary in this thesis. The data for pressure force is collected from force sensor mounted on the tool, and the force sensor collects data for pressure force in x, y and z direction. As the z direction is the direction that the tool is pushing on metal plate, which makes the pressure force in the z direction is the most needed force measurement. The software used in gathering pressure force measurements is called TSV, and is more thoroughly described in chapter 4.2.2.

To achieve best possible result for force measurements it is necessary to know the exact load data of the tool. Using the function LoadID on the Machining FC package close to the workspace, gives the weight of the tool and location of the center of gravity. This can then be used to update the load data.

In addition to force measurements it is also desired to collect position measurements. This gives the opportunity to show how much force is used at different positions, which is important when looking at the result of the force control procedure. Both camera and robot encoders are used in gathering position measurements, such that it can be compared to one another. Also the camera and the robot have different sampling time, which will be needed to take into account for when processing.

Camera measurements are gathered from DMM software included in the K-Series package. For more information about the software and the camera see chapter 4.4.4. In DMM it is necessary to specify which frame is used as reference, so changing this from default

frame (camera frame) to base frame is vital.

Position measurements for the robot are not taken directly from robot encoders, instead the angle of each joint is gathered from the encoders through the controller. Forward kinematics is therefore used in calculating the tool center point, which was described in chapter 3.1.

Currently the way to get data from the controller is by using opcom, and the computer needs to be directly connected to the robot controller. Opcom is mainly used to load controller models, and while a model is loaded gives the opportunity to either read or write directly to the controller. Only submit mode is needed when gathering measurement data.

5.1.4 Execution

Executing the path created is done in Machining FC. The same path is repeated twice for each force in the experiment. This is done to acquire force measurements with or without low pass filter used. All the different measurement data are gathered at the same time with a slight delay between them. When experimenting it was important to work between observing the robot and collecting the data, so that one could easily stop the robot if something unexpected happens. Nothing unexpected should happen as the path should always be tested before running the force measurements.

During the execution of the first experiments it was seen that the force sensor measured a relatively higher force than expected. This occured when the force control function was not active, and was much higher than the resolution of the sensor. Several methods have been tried to reduce this error, which will be discussed in section 5.3.5.

5.2 Results

In this section the results from the different force control experiments will be shown. Differences between camera and robot positions will be looked at. Additionally the choice of filter in processing the measurement data will be discussed. The initial experiment show a somewhat similar result as with Petter's force control experiment, and the other experiments gives additional wanted information.

5.2.1 Force measurements

Much of the reason for using force control instead of position control was that one could insert a desired force and then make the robot press with this force on the workobject. It is therefore necessary to look at the measurement data gathered from TSV, such that this can be verified. The experiments was conducted for five different forces as mentioned earlier, which was in the range of 1 - 30 N. The robot is pressing downwards on the metal plate, which is the same as the positive z direction of the force frame.

Force measurements are therefore displayed for positive z forces. Figure 5.6 shows the force measurements for the experiment associated with path 2 in table 5.1. For force measurements associated with path 1 look at figure 1 in the appendix.

The force measurements shows both the raw and low pass filtered force data, where a 5 Hz low pass filter was used. For the first path it can be seen that especially for 10 N plot a large initial offset error is present. This offset error which will be discussed in section 5.3.5 is also present for the other force measurements of path 1 aswell.

The force plot for path 1 in the appendix is only meant for illustration, and is quite unscientific as the measurement data for 1 and 5 N had to be altered to give meaningful plots. A plot of the raw data for path 1 is shown in figure 5.28, where the offset error is seen to increase with the less force used. It is therefore concluded that the second path path 2 is the one to be used in further experimentation.

Most of the force plots for path 2 is seen to follow its desired force, except for 1 N which is around initial value at all times. By examining the plots for the position closer, see figure 5.10, it is seen that the tool doesn't touch the metal plate at all during process.

The reason for this unexpected behaviour could be explained by looking at the resolution of the force sensor, which is around 1 - 2 N. Therefore the robot might not be able to sense if it touches the metal plate before the robot reaches the withdraw phase. Still it was expected that the robot would be able to touch the metal plate, as was the case in Petter's experiments [17]. This was one of the reason a new experiment was done, where the measurements gave a different result. This is shown in figure 5.7 , where the speed of the robot was decreased during the process phase.

Another undesired behaviour seen in figure 5.6 is that there exist high oscillations for all the different forces applied. These oscillations will be further discussed in section 5.3.7.

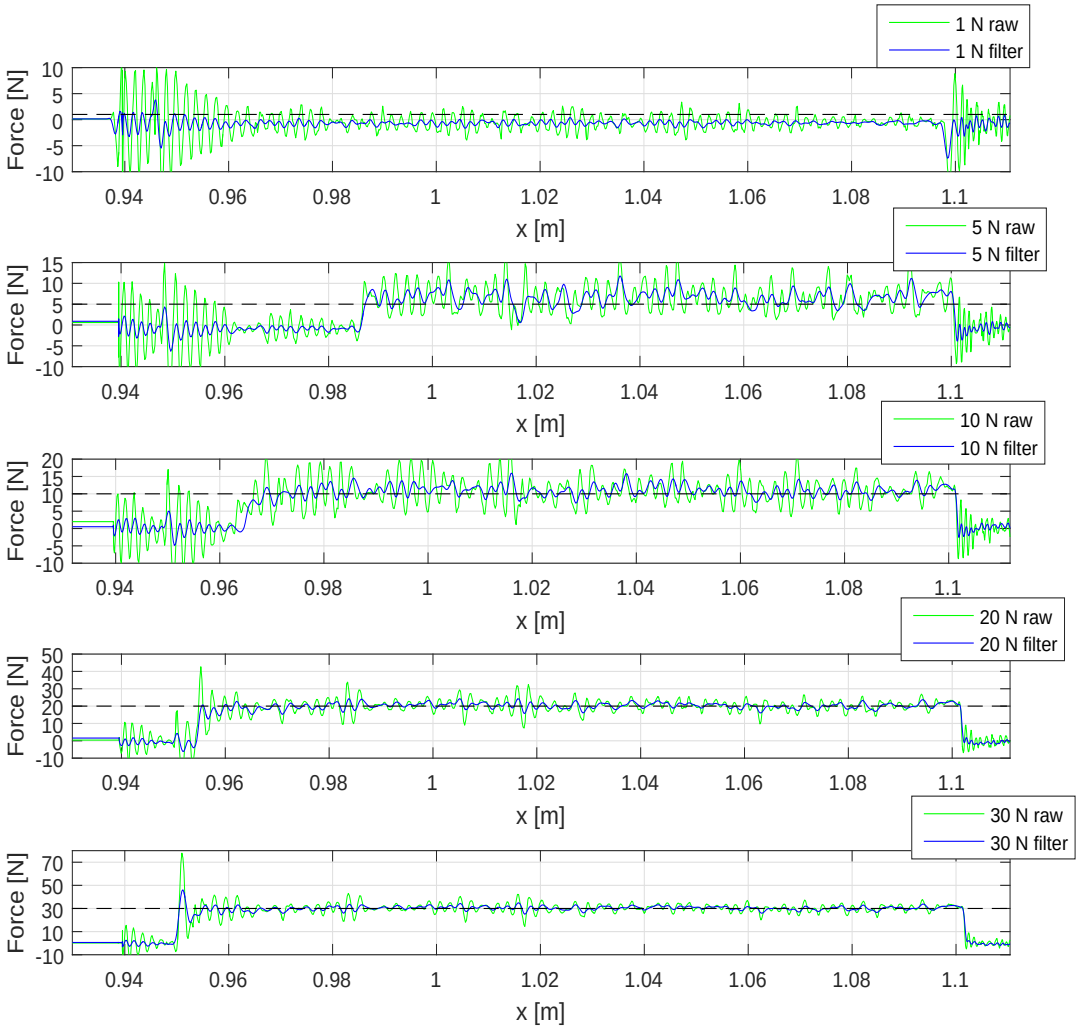


Figure 5.6: Force measurements from 1-30 N for path 2

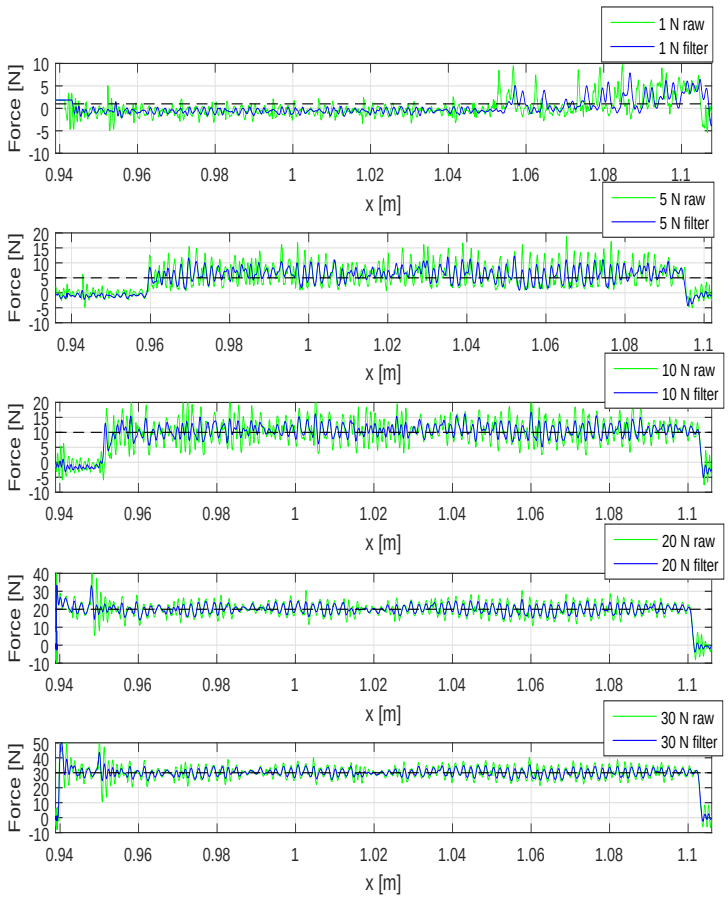


Figure 5.7: Force measurements from 1-30 N using path 2 when velocity along the path is decreased to 10 mm/s

Examining the force measurement plot it can be seen an overshoot for the force at around $x = 0.9494m$. This is also the same area where the tool is supposed to collide with the metal plate, which indicates how big effect the collision poses. The overshoot is seen to increase linearly with the pressure force that is measured. In section 5.3.1 the effect of collisions and what can be done to reduce its effect will be look at.

It takes the IRB 4600 approximately 8 seconds to run from $x = 0.9494m$ to $x = 1.1030m$, as can be seen in figure 5.8. It takes approximately 12 seconds for force control to be deactivated, and at this point the signal spikes as seen in the plot. This is a highly undesired behavior that needs further to be looked at more. Other than that the robot performs quite well when in contact with the environment, and a mean of the force can be found to be at $F_{mean} = 20.2N$ for this particular run..

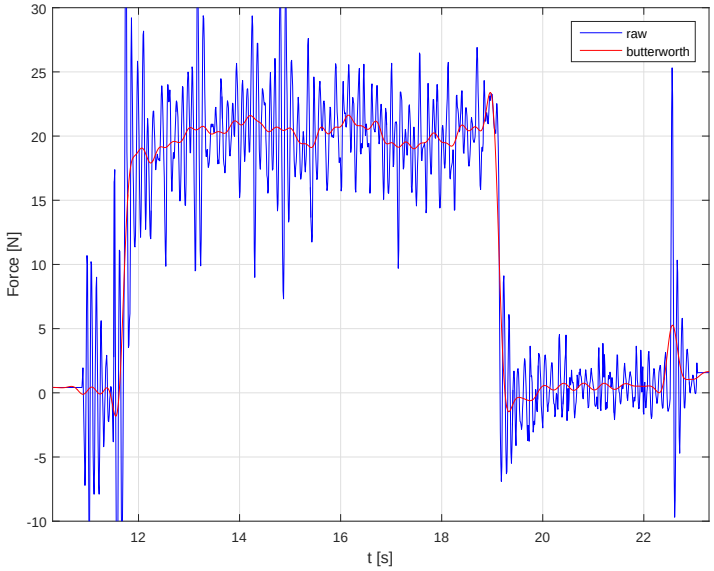


Figure 5.8: Plot of the 20 N force where a buttworth filter has been applied, the x axis shows the time sampled from the measurements

For comparison reasons a plot of the PSD is shown in figure 5.9 where 10 N was applied to the force control. Here the power density is compared from the measurements using raw data and low pass filter applied in TSV. In addition a Butterworth filter has been used on the raw data. Between approximately 15 - 30 Hz there appears to be noise of high amplitude for the raw data. Using the different filters this noise can be reduced to a lot less, where the Butterworth filter is the best at reducing the noise. The Butterworth filter and several other IIR types of filters will be discussed later.

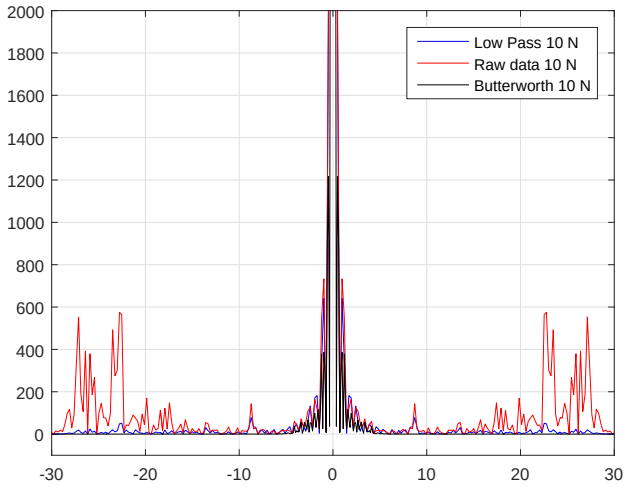


Figure 5.9: Comparing the power spectrum density of the raw data, low pass filter applied on-line and Butterworth filter applied on raw data

5.2.2 Position measurements

This section will look into the position measurements gathered from the robot encoders and the camera, for when the robot was running the force control procedure. First they will be examined separately for each force applied, and then it will be seen how they deviate from eachother. As a a high precision camera system is used, the measurements gathered from it should be quite reliable. The only issue to this is that the frames have to be defined perfectly and that the base of the robot is fully static. Figure 2 and 5.10 shows the position measurements gathered from the robot encoders for both path 1 and path 2 in table 5.1. See appendix for position measurement plots for path 1.

Comparing the two plots shows little difference overall in how the robot moves along

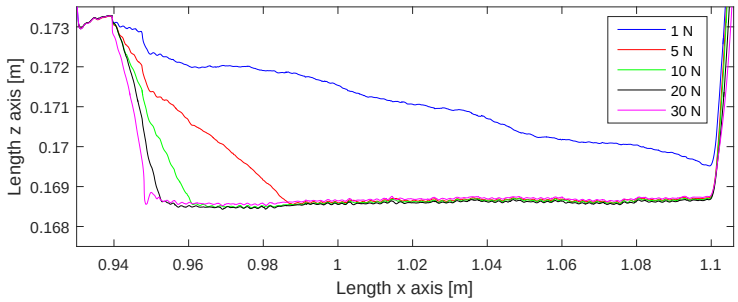


Figure 5.10: Position measurements from 1-30 N for path 2 measured by robot encoders

the path, and differ only at the withdraw stage as expected. The plot of the robot position shows that there is a link between how fast the robot reaches the metal plate and how much pressure force is used. At 30 N the robot touches the metal plate at around $x = 0.948m$, which is around the place where it is supposed to touch. While for 1 N the robot doesn't reach the workobject in both paths, which has already been understood from the plots of the force measurements in figure 5.6. For the withdraw point it can be seen that the robot withdraws ($x = 1.101$) almost at the taught point ($x = 1.103m$).

Figure 3 in the appendix and 5.11 shows the plot of the robot position according to

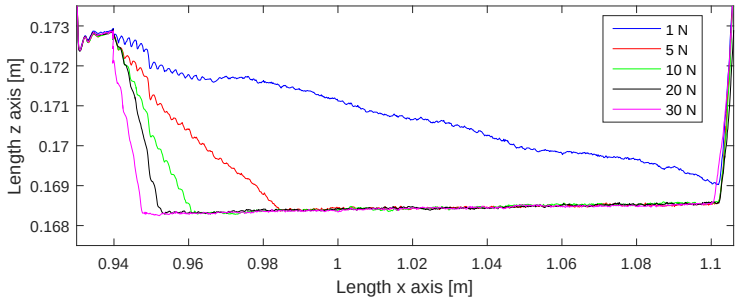


Figure 5.11: Position measurements from 1-30 N for path 2 camera system

what the camera measures for the two different paths. By comparing the measurements from camera and robot encoder it can be found that they differ especially in the process phase, which is seen in figure 5.12. The plot for 1 N is not shown here as it doesn't follow the desired path at all for both measurement systems. One of the main reasons for this shift in the x direction could be from the frames not fitting perfectly. Another reason

could come from the fact that the camera measures most accurate close to the point where it is calibrated, such that it follows the robot best close to the approach point.

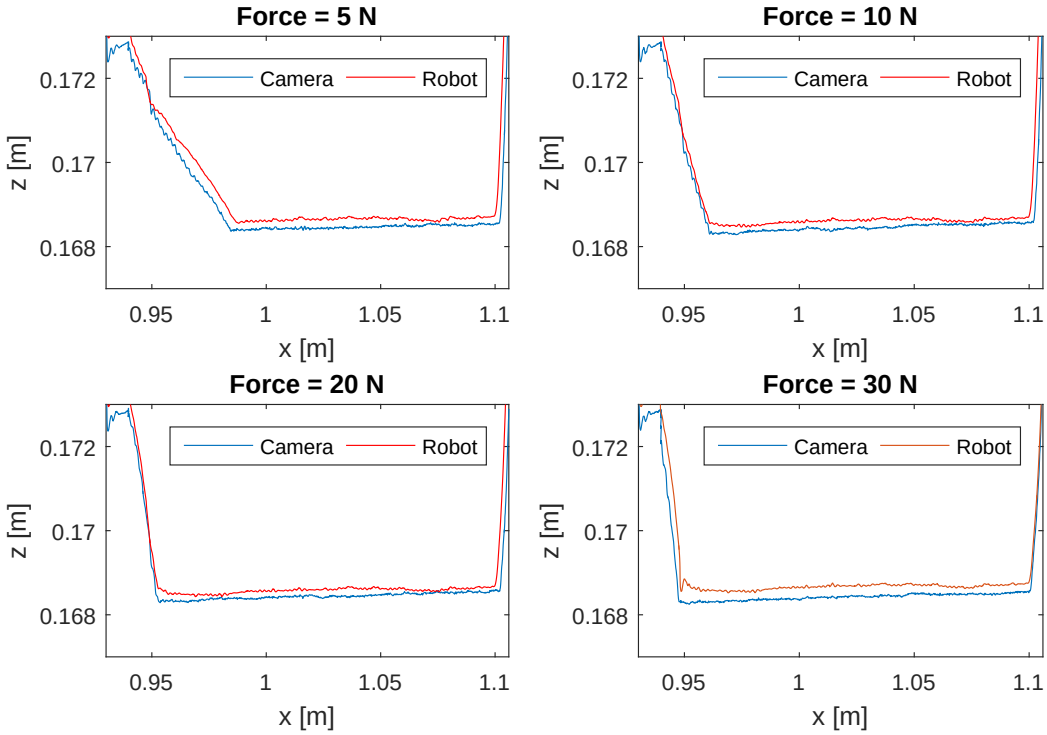


Figure 5.12: Comparison of the position measurements from 5, 10, 20 and 30 N for path 2 using both systems

By looking at the plot in figure 5.13 it can be seen that the error for position in z direction is somewhere at $z_d = 0.5$ mm close to the point of calibration. The error increases closer to the withdraw point, which could be contributed from the fact it increases its distance from the point where it was calibrated.

The error is not sufficiently low for what was desired of the result from the camera measurements, which can be seen from figure 5.13 containing the errors in the different direction. Especially the error in the y direction is particularly bad, where the displacement is an average of $y_d = 1.4$ mm during the procedure. The error will be compensated for using an additional LED placed on the tool, in addition it will be attempted to calibrate the frames such that it fits better. Still this might not help as the robot base slightly moves on, which was discussed in Petter's thesis [17]. The result of this is discussed and compared to the earlier experiment in section 5.2.4.

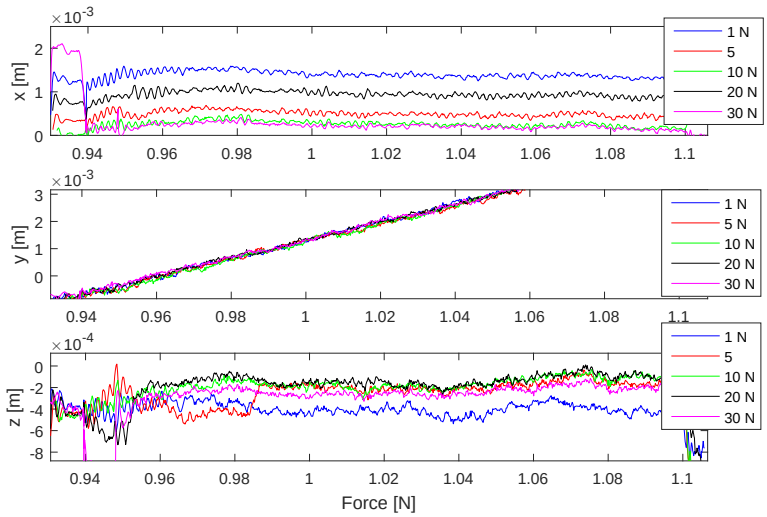


Figure 5.13: Error margins in the direction of the x-,y- and z-axis for the IRB 4600 measured by the camera

5.2.3 Choice of filter

A relatively high noise can be seen for the force measurements in figure 5.9, even using low pass filter in TSV doesn't minimize this noise fully. It is therefore necessary to design a suitable filter for the force measurements, so that precisely the force used will be shown. Three different types of IIR filters are used and processed using zero phase filtering. These are the Butterworth, Chebyshev Type II and Elliptic filters.

Use of a Butterworth filter has already been experimented with earlier by Petter, the two others however has not been looked at and they could give a better result. To get a better understanding of filters and signal processing check chapter 2.4. For the Butterworth filter the best result was achieved using a fourth order filter with cross rate of 7 Hz, which gave the flattest possible magnitude response without ruining the signal. Using a higher order filter than the one used with Butterworth did not reduce the noise any better. The plot using a force of 20 N with Butterworth filter apllied be seen in figure 5.14. Here the force sensor resolution has also been taking into account.

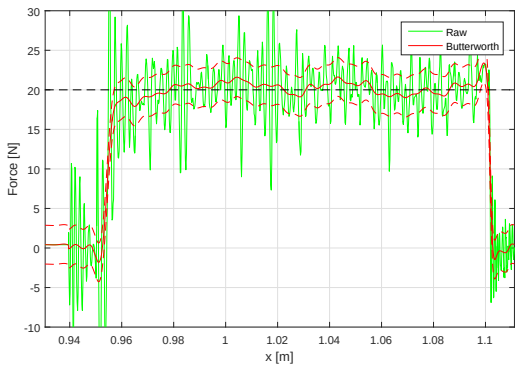


Figure 5.14: Plot of the force measurements for 20 N with Butterworth filter applied to the raw data

For the elliptic filter a lower cross rate of 5 Hz was used. The order of this filter however needs to increase by one and in addition attenuation for both passband and stopband has to be introduced. This was set to 1 dB and 75 dB for the passband and stopband respectively, and the result from this is plotted in figure 5.15. Compared to the Butterworth filter this is not as flat, but contains equiripple through the whole signal. This is to be expected for a Elliptic filter, as it has equiripple in both passband and stopband.

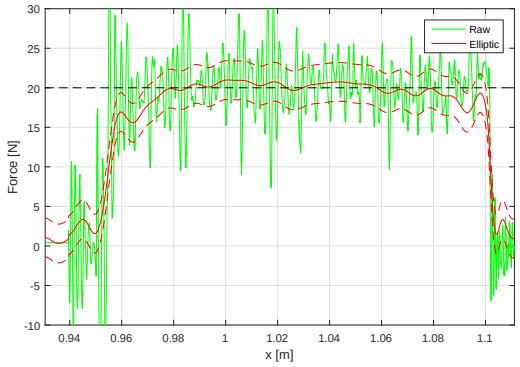


Figure 5.15: Plot of the force measurements for 20 N with Elliptic filter applied to the raw data

The last plot shows the use of the Chebyshev Type II filter, and can be seen in figure 5.16. Here an order of 5 and a cross rate of 17 Hz was used, in addition it was chosen to have a passband attenuation of 1 dB and a stopband attenuation of 80 dB. The plot of the Chebyshev Type II shows that it has equiripple, which is not as fluctuating as with the Elliptic filter. All the different filters applied reduces the overshoot present in the force signal by a large amount, as one would expect assuming that the impact between the tool and metal bar causes vibrations.

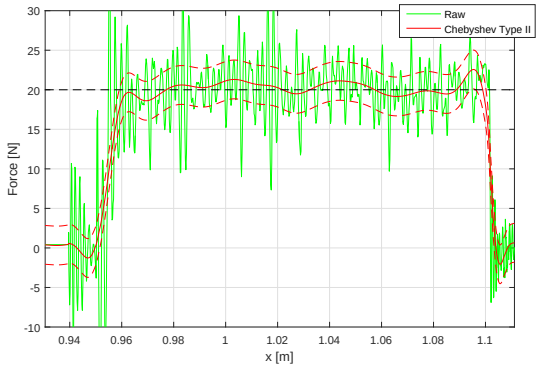


Figure 5.16: Plot of the force measurements for 20 N with Chebyshev Type II filter applied to the raw data

Looking at PSD of the different filters in figure 4 in the appendix it can be seen that they all reduce the noise from the raw data quite well. For this reason and the fact that Butterworth filter doesn't have equiripple, it was chosen to use the Butterworth filter. Still it requires a lower cross rate than for the Chebyshev Type II filter.

5.2.4 Camera calibration and setup

From the measurements concerning the position of the robot it was made pretty clear that the calibration of the different frames were not satisfying. It was therefore decided to run an additional experiment with new set of calibration for the camera. In addition an extra LED was placed on the tool.

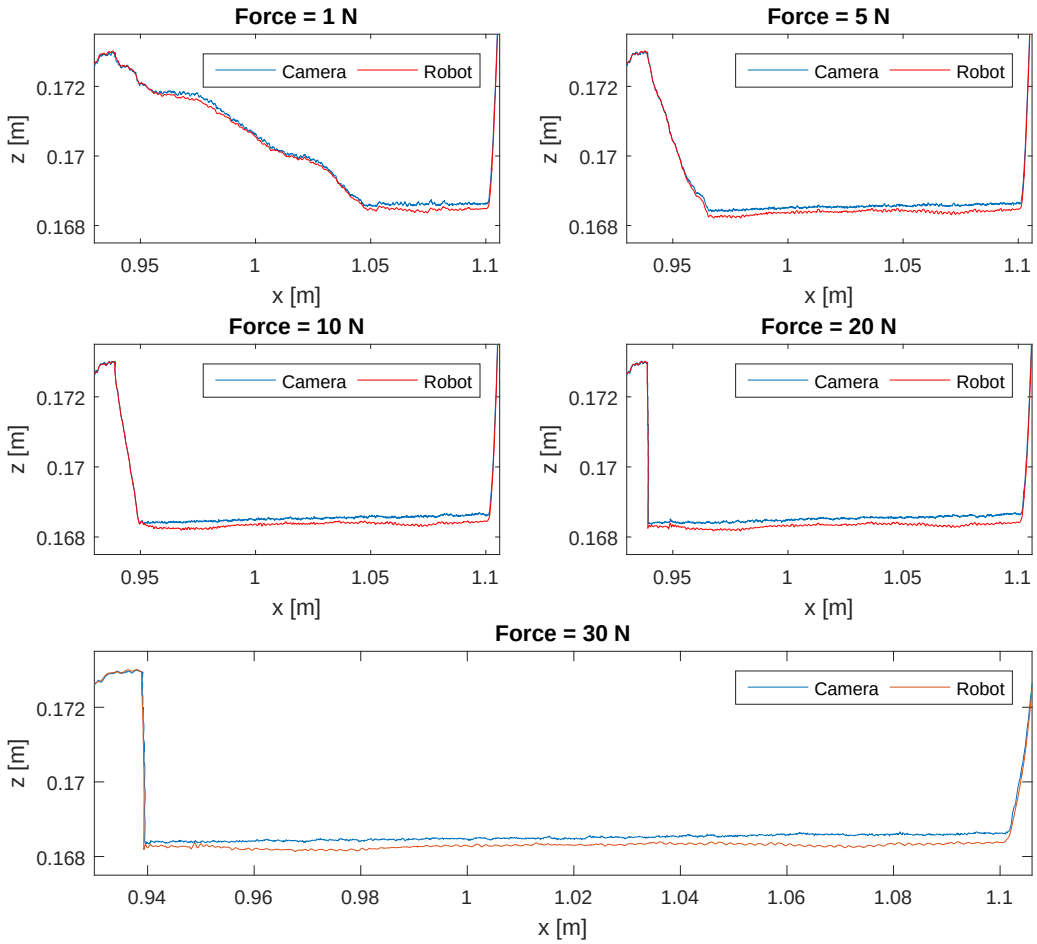


Figure 5.17: Comparison of position measurements from camera and robot encoders for additional calibration

This experiment follows path 2 in table 5.1 that was concluded to be the most approriate path when it came to measuring forces. The result of this experiment is seen in figure 5.17, where the camera measurments are compared to the robot encoder measurements.

Analyzing the new measurements shows several interesting aspects for the connection between the camera and the robot encoder, and the result is better comparable with Petters experiments. First noticed from the figure is that for 1 N the robot acquires a stable path at the workobject, which will be more looked into in the next section 5.2.5. Looking at figure 5.12 in addition with 5.17 one can see that for an increase in force used, there is also an decrease in the angle at which the tcp touches the path. This

can be better understood from figure 5.18. Here it also shows that the robot does not reach the work object at the point that was chosen for most of the forces applied. Only for the case when 10 N is applied does it approach and withdraw at the correct positions.

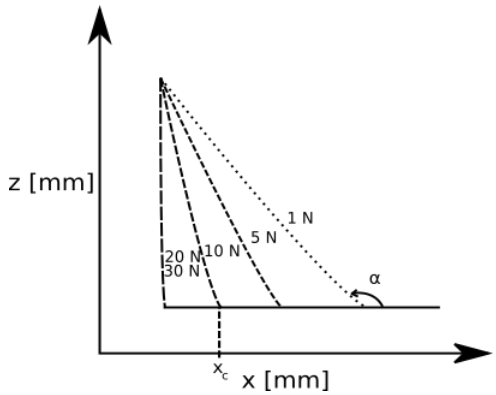


Figure 5.18: Shows the relationship between an increase in force used, with x_c being the chosen point of contact with work object

Looking at the time period when the tcp is in contact with the work object, it can be seen that a marginal increase in displacement error occurs between camera and robot encoder. This is deviation can be seen more clearly in figure 5.19, where each dot marks the difference between camera and robot encoders at a given time sample. This result validates the idea that the robot produces an error in length between tcp and workobject, which increases with an increase in force used. This was stated by Petter [17] where it was concluded that a linear increase could be seen, which though is not totally present in this experiment.

While there is a clear increase in deviation between 1, 5, 10 N, the deviation appears to be almost the same between 20 and 30 N. An explanation of this could come from the fact that the increase would be more noticeable between 25 to 50 N than from 20 to 30 N.

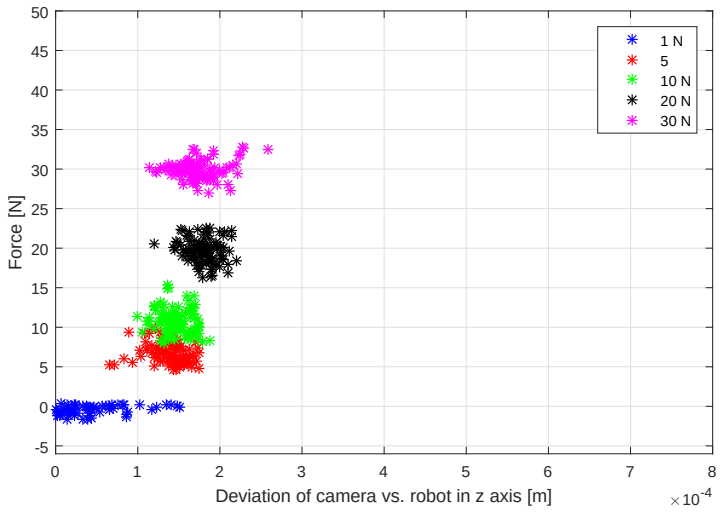


Figure 5.19: Shows the deviation in camera and robot encoder measurements for a set of samples

Still the calibration applied with an additional LED placed on the tool seems to have improved the displacement error between robot and camera in x and z direction, seen in figure 5.20. Compared to figure 5.13 the displacement in x-axis and z-axis has been significantly reduced. While the mean error was approximately $x_d = 0.55$ mm and $z_d = 0.80$ mm before the new calibration, it is now approximately $x_d = 0.25$ mm and $z_d = 0.092$ mm.

The issue still remains with the deviation along the y-axis, which has increased from $y_d = 1.4$ mm to around $y_d = 1.7$ mm. It is not known whether the deviation in y-axis is mainly contributed from displacement of the robot, or that the frames are not perfectly oriented toward the robot base.

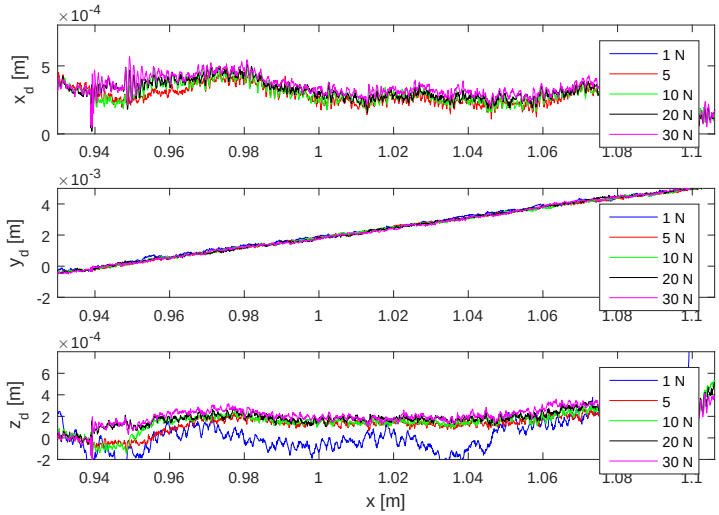


Figure 5.20: Shows the deviation between in the camera measurements along the path

5.2.5 Velocity observations

In Petter's thesis it is discussed that not only displacement is seen for the error in z, but an occurence of oscillations are also observed. In the force measurement plots in figure 5.6 it is also possible to observe these oscillations. The possibillity of achieving a better result on the force measurements was the main reason for conducting an additional experiment using a different velocity.

The new experiment uses the second path shown in table 5.1, and the velocity is decreased for the time that the force control is used. Figure 5.21 includes the speed of the robot for both experiments. Note there are two moments that the velocity is almost reduced to zero, and this is at the time the robot either comes in contact with the work object or widthdraws from it.

Position	1	2	3	4	5	6	7	8
V_1	20 mm/s	20 mm/s	20 mm/s	20 mm/s	20 mm/s	20 mm/s	50 mm/s	50 mm/s
V_2	20 mm/s	10 mm/s	10 mm/s	10 mm/s	10 mm/s	20 mm/s	50 mm/s	50 mm/s

Table 5.2: Velocity of the robot for different two experiments

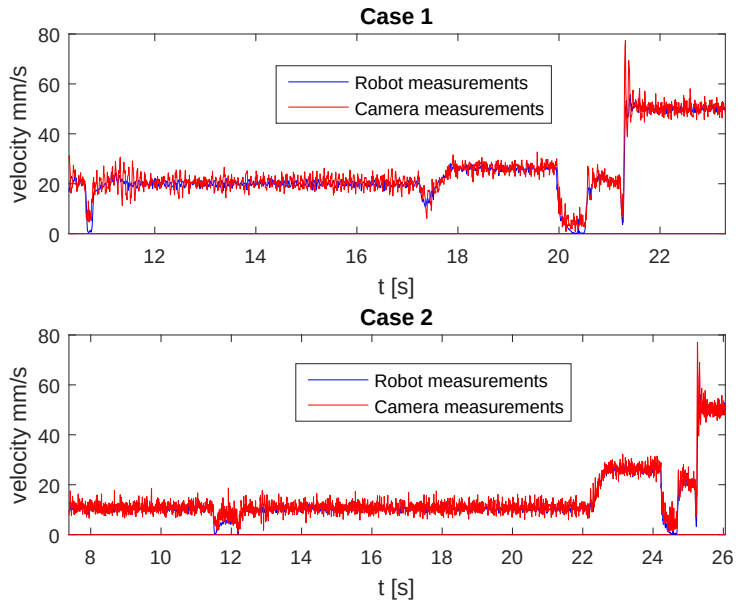


Figure 5.21: Comparison of the speed of the two experiments using path 2

Table 5.2 shows the velocity of the original experiment compared with the new experiment for path 2 specified by table 5.1. Other than the change in velocity the new experiment follows much of the same procedure as the others. Collecting and processing the measurement data gave the plots seen in figure 5.7 and 5.22. This shows the result of the force and robot position measurements for the new experiment. It takes a lot more time to run the procedure for each pressure force, therefore the plot is created to fit exactly with the time the force control is used. An interesting observation that can be made is that for 1 N the tool touches the metal plate at around x = 1.05 m. For the first experiment this was discussed to be a cause of the force sensors resolution, which seems to be correct as the the force control tires to compensate by using a lot more force. The operation

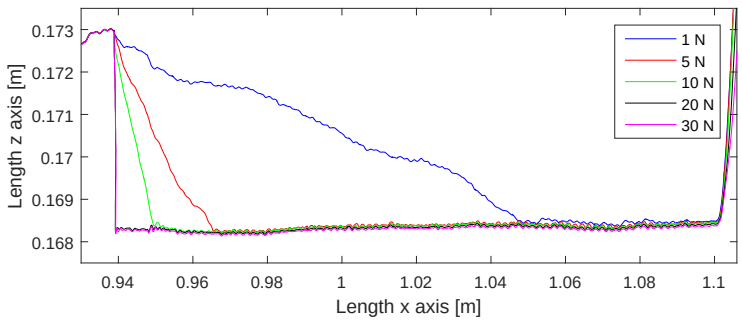


Figure 5.22: Position measurements from robot encoder for experiment with reduced velocity

takes approximately 15 s from approach point to the point where the tool withdraws. This can be seen in figure 5.23, where the force measurements has been fitted to contain the most important part of the procedure. For this experiment the raw data of the force measurements are especially noisy. This is the reason for including a Butterworth filter to the plot, which shows that the force only fluctuates ±1.5 N from desired force.

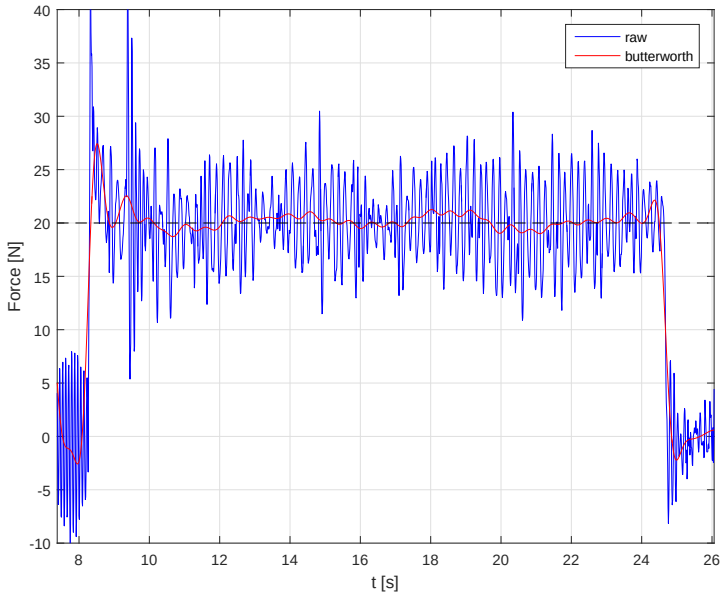


Figure 5.23: Force measurements for experiment with reduced velocity, with time sampled from measurements

Comparing the oscillations for z-axis in figure 5.13 and 5.20, it can be easily seen that the oscillations didn't decrease with a reduction in robots speed. The best bet would therefore be to research more on alternating velocities, which is to be briefly looked at in the next section.

5.3 Discussion

This section will look into the result of the force control experiments discussing their main aspects, and look at what can be done for future experiments. Starting with a more thorough discussion about impacts and collision, and its effect on the robot. This is followed by looking at differences between rod with wheel tool and the drilling spindle. The difference in position measurements and filter design will also be explained on a more detailed level, where the analytical toolbox **Filter Design and Analysis** in Matlab is used for filter design. After this the force measurement offset error observed in the results and alternating velocity components during the process will be discussed.

5.3.1 Impacts and collisions

One of the first things that can be noticed from looking at the plot of the force measurements for 30 N in figure 5.6 is that it has a large overshoot in the signal. This is largely due to the impact between the tool and the metal plate. An impact or collision between two bodies is characterized by large reaction forces and changes in velocities of the bodies, see [14]. From this the bodies are subject to elastic or plastic deformation with dissipation of energy. For impacts with sufficient high velocities this deformation will be permanent, resulting in loss of energy. The impulse of the impact P is calculated by the integral of the contact force from the following equation:

$$P = \lim_{\delta t \to 0} \int F_c(t) dt \tag{5.1}$$

For the plot in the experiment using a slower velocity reduced this overshoot by some degree, as seen in figure 5.7. Still this is of too high value and it would be very noticable when the drilling spindle is to be used. The effect of the impact force is also felt on the each link of the robot, and is probably a cause for collapsing links and joints, which again increases the position error such as seen in figure 5.17. That's why it's important to look at different ways to reduce the effect of the impact force by as much as possible.

From the definition of impact it was described to be the result of two bodies exerting reaction forces on one another. The impact force is determined by the velocities of the two bodies that are colliding. In this thesis only the velocity of the tool can be minimized sufficiently, such that the impact force is reduced while still being able to come in contact. This could be done by having alternating velocities for the robot motion, where it would decrease closer to impact.

Another way to minimize the impact force is by introducing correcting torques. Applying damping torque at the time of impact have the possibility to minimize the force of impact between end effector and environment. This has ealier been done for kinematically redundant manipulators, Gertz et al. [18], and if it is possible to solve the problem of underactuation then this could be useful.

The main cause of collapsing links of the robot is from a lack of high stiffness. The effect of the impact force could be reduced with controllers compensating for this lack of stiffness. Before that it would be necessary to create a model of the stiffness for the IRB 4600, which indentifies the stiffness parameters. In a study by G. Alici and B. Shirinzadeh [6] it is worked on a methodology for stiffness indentification and characterization, and how enhanced stiffness models can be made.

5.3.2 Tool comparison

In this chapter the focus has been on the use of a wheel-rod tool for experimentation with force control. This tool was designed by L. Paramonov and serves as a way to test force control without deforming the material. Using this kind of tool in industrial applications would serve no purpose, since the whole point in any grinding, drilling or deburring operation is to deform the material in some way. In earlier experimentation with force control conducted by Stepan Pchelkin a drilling spindle with drill bit at the tcp has been used. The result of such an operation can be seen in figure 5.24, where the drilling spindle works by having it revolve around its shaft at high velocities.

The spindle motor can be seen in figure 4.4 in chapter 4 with the wheel-tool attached. Using the spindle motor compared to the wheel-rod tool introduces a new set of challenges that have to be dealt with to produce a satisfying result. In the following text some of these issues will be discussed, and possible ways to solve them will be given.

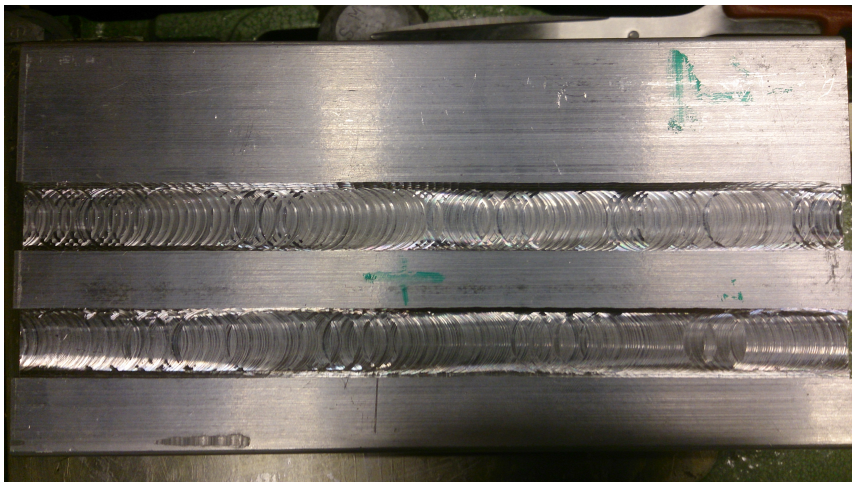


Figure 5.24: Result of a grinding procedure using the spindle motor with drill bits

A major issue is the vibrations introduced from the high angular velocity of the drilling spindle. If left unattended the result of a grinding operation would be non-smooth, which is critical when precision is of utmost concern. The way to counter vibrations is to implement a notch filter on-line, which would then reduce the noise produced by the vibrations. Looking at the result of a grinding experiment in figure 5.24 an uneven surface can be seen. This could be caused from small vibrations acting from the tool, that were not compensated after filtering. These vibrations also appears as noise on the measurement data gathered, and it is therefore important to use the Butterworth filter for additional filtering.

Another challenge that rises with the use of drilling spindle is the increased risk of hazards occuring. Especially during the start up phase does the safety concern increase. A scenario that may appear at the start up phase is that the drill bit may explode,caused by the rapid increase in pressure. Several safety warnings can be found in drilling spindle manual [9], and the need for additional safety equipment is necessary. A good way to reduce the risk of damage is by always wearing the safety equipment recommended in the safety manuals, and be sure to keep a distance from the robot when operating. Normally the manipulator has to be secured in a cell without any way to cause damage outside the cell. The operator also has to be sufficiently trained in using the cnc framework, and know how to handle any situation. When changing tool the first thing to check is that the tool is aligned and secured, and then it is important to test it for low speed before increasing the speed towards its desired value.

For the experiments using wheel-rod tool it was found during the impact with the metal bar a high overshoot in the use of force, but as the wheel-rod tool doesn't deform the surface it had little impact. This is however not the case when using the drilling spindle, and it would cause unaccurate grinding in the beginning of the process. The cause for this high overshoot in use of force is mainly related to the distance from the tcp and the force/torque sensor, such that it takes some of time before it can compensate for the force used.

Another issue that occurs when grinding is additional deformation of the material, which is caused by the high velocity of the drill spindle acting on the metal. These two challenges are both hard to correct, but one way to do it is by conducting several experiments

and then creating a controller which compensate this behaviour based on patterns from the experiments.

In Petter's thesis [17] and for experiments in this thesis it was confirmed that collapsing links and arms of the robot causes displacement to the operation. The fact that the robot links and the base are not fully rigid causes the reaction force acting on the robot from the environment to have an influence on the placement of the robot, which again causes displacement in the procedure. This positioning error can be compensated using some kind of trajectory compensator and an observer. An example of an on-line compensation for the positioning error that exists for the case of robotic friction stir welding (FSW) [16], where trajectory compensation and discrete-time observer was used. This could prove useful if developed for grinding operations as it reduces the positioning error caused by the robot's lack of stiffness.

5.3.3 Difference error in position measurements

In section 5.2.2 the result of the position measurements showed that there was a rather high difference in the position of the tcp from the camera and the robot encoders. Here it was concluded that the main error was caused by a not so good calibration of the camera, because of good compliance in the measurements from ealier experiments. Especially the error in the y-axis was of concern, and was probably caused by using the same line for both base and tool when assigning frames. At the time this justified from the fact that creating a line with the Space Probe along the tool was even more inaccurate.

Another probable cause could come from the placement of the camera and the LEDs at the tool, which is seen in figure 5.3 and 5.4. It was only possible for the camera to be placed at one location, and the reason for this was that it was not possible to calibrate the camera at any other location as the calibration area was too narrow. At this specific location the camera is set in an orientation such that it cannot fully detect the whole spindle tool. To correct this it was tried to use a slightly different placement of the LEDs, where one of the LEDs at the tool was placed further back in the y-direction of the tool frame. This change did not improve anything, and only increased the error in the camera position. What has not been tried yet is changing the orientation of the tool in a way that better shows the shape of the tool, and then place the LEDs more appropriately to accomodate this.

In the last experiment conducted it was placed an additional LED on the tool frame, which reduced the difference error in x-axis and z-axis in the position. At this level the main fault for difference error are contributed from the robot encoders and not the camera, which is more reliable than the robot encoders when calibrated correctly. The difference could now be explained from the fact that the robot experiences a collapse in links and arms when applying pressure force. Controllers that compensate for collapse in links and arms will be therefore be needed to reduce the error even further.

5.3.4 Filter design analysis

In section 5.2.3 it was decided to primarily use Butterworth filter for post processing the force signals. Two other filters, namely Chebyshev Type II and Elliptic, were also examined as possible filter candidates. To get a better understanding as to why the Butterworth filter was chosen and not the other two, one would have to go deeper in the analysis of the filters responses.

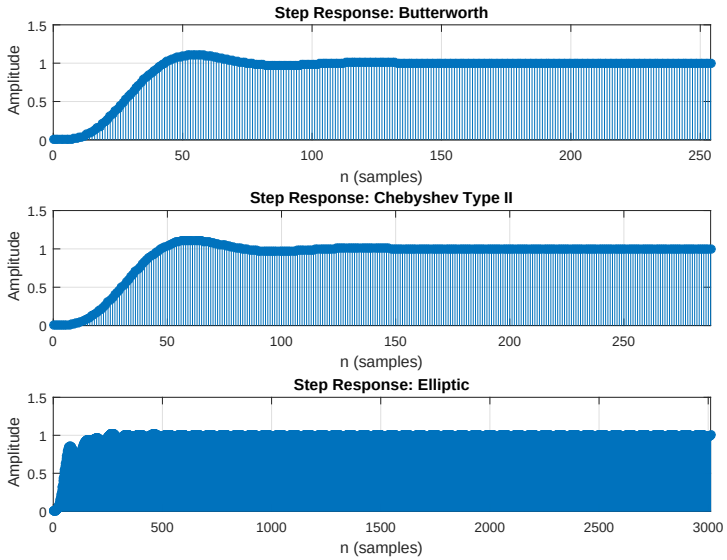


Figure 5.25: The step response of the three filters examined with chosen specifications

From earlier the PSD was studied, as can be viewed in figure 4, and it was relatively easy to see that all the three filters removed the high amplitude noise from the raw data relatively equally within their given specifications. This still does not mean that the filters work equally well, without looking at the fact that Butterworth have no equiripple while the other two have. Using signal processing tools in Matlab it is possible to analyse the filters response more closely.

Their differences can be immediately seen by looking at the plot of the different step responses, which is seen in figure 5.25. The step response uses the sampling frequency of the force signals, which is $f_s = 248.016$ Hz. The Elliptic filter stands out in the step response with a several times higher settling time than the other two. The differences between Butterworth and Chebyshev type II are not that great, and none of them are ruled out yet.

Plots for the group delay and phase delay can be seen in figure 5.26. Both group delay and phase delay are defined in equation 5.2 and 5.3, and can be best described as being the rate of change of phase at a certain frequency and phase angle at a point in frequency respectively.

$$D(\omega) \triangleq -\frac{d}{d\omega}\Theta(\omega) \tag{5.2}$$

$$P(\omega) \triangleq -\frac{\Theta(\omega)}{\omega} \tag{5.3}$$

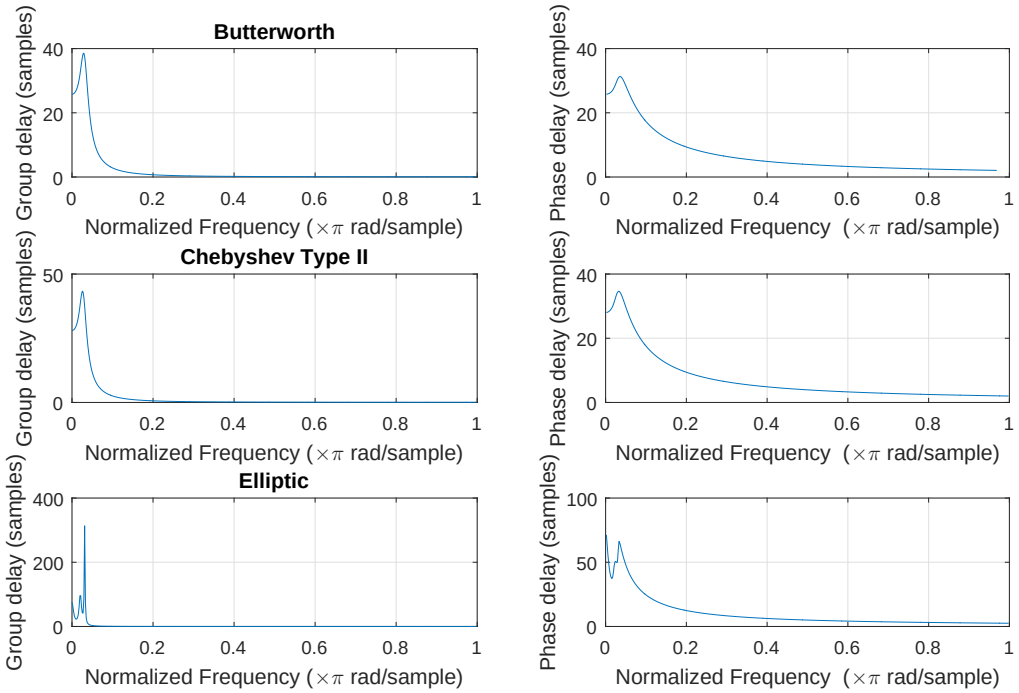


Figure 5.26: Group and phase delay of the three different filters

For linear phase responses group delay and phase are equal, and can be interpreted as time delay. Focusing on the plots for phase delay in the different filters, it is clear that the Butterworth filter has the least phase delay along all frequencies. While the Butterworth filter acquires a peak of almost 24 samples in the group delay, it is found to be over twice as big for Chebyshev filter and ten times as big for the elliptic filter at the lower frequencies. The phase delay shows the same pattern with Chebyshev and Elliptic being atleast twice as big as the Butterworth. This positions the Butterworth filter more clearly as the optimal choice.

Analyzing the zero pole plot could also prove to be valuable, and is a way to show how stable the filters are. A filter is stable if the poles of the system are well defined within the unit circle. The zero pole plot in figure 5.27 shows this for the three filters, and it can be seen that while everyone are within the limit of stability, both the Chebyshev type II filter and Elliptic filter are closer to the unit circle than the Butterworth.

Analysis of the filter design is especially important to look at when choosing filter parameters. There are also several trade offs involved that one needs to be beware off. For example while the Chebyshev filter can have a higher cross rate than the other two, it is more limited on the attenuation used in the passband and stopband.

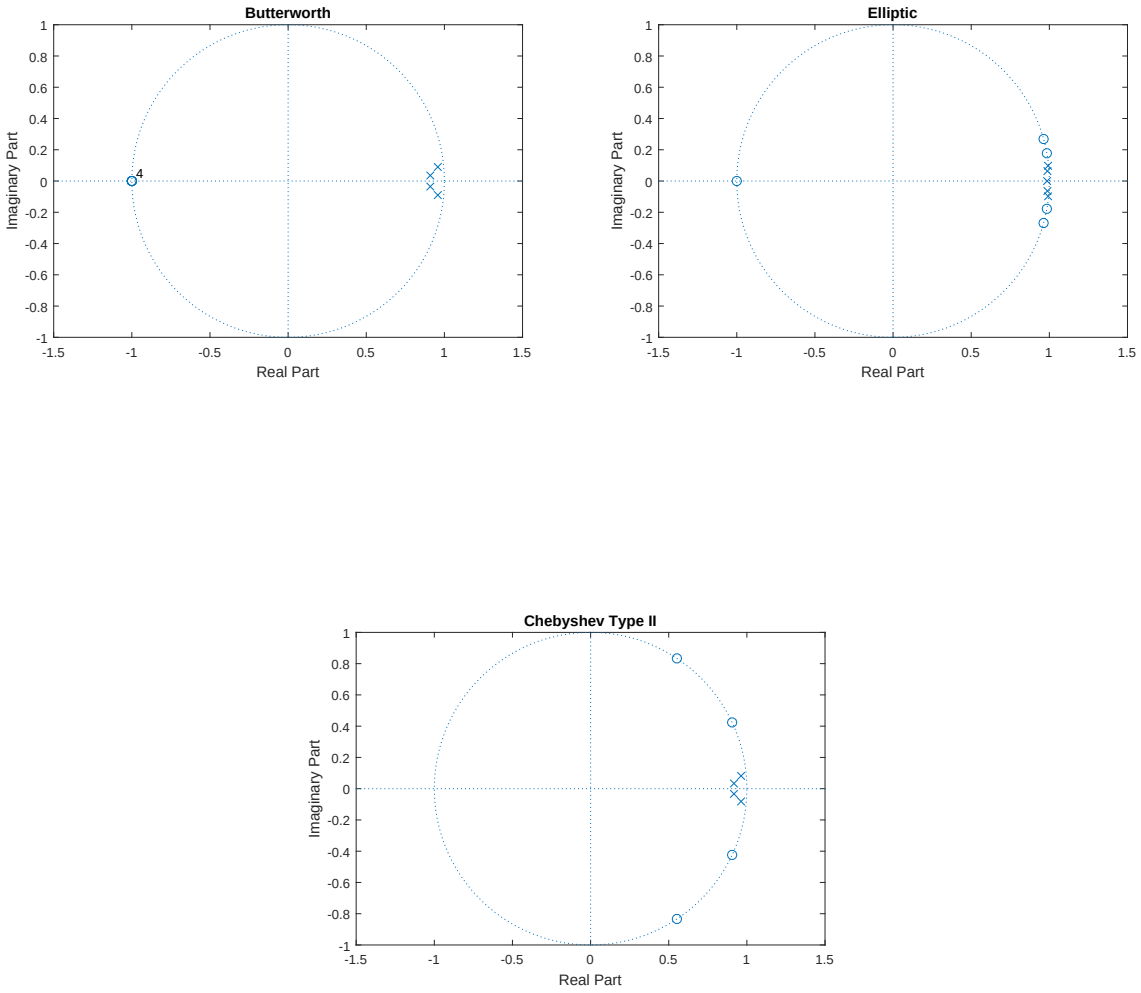


Figure 5.27: Zero-pole plot of the Butterworth, Elliptic and Chebyshev Type II

5.3.5 Force measurement error

By looking at the plots for force measurments in figure 5.6, it can be noticed that the force signals have a slight offset error during its non critical phase. This was especially noticeable for some of the first runs, and it seemed to change offset value every time force control was run. Figure 5.28 shows the plot from one of the earliest runs that were processed.

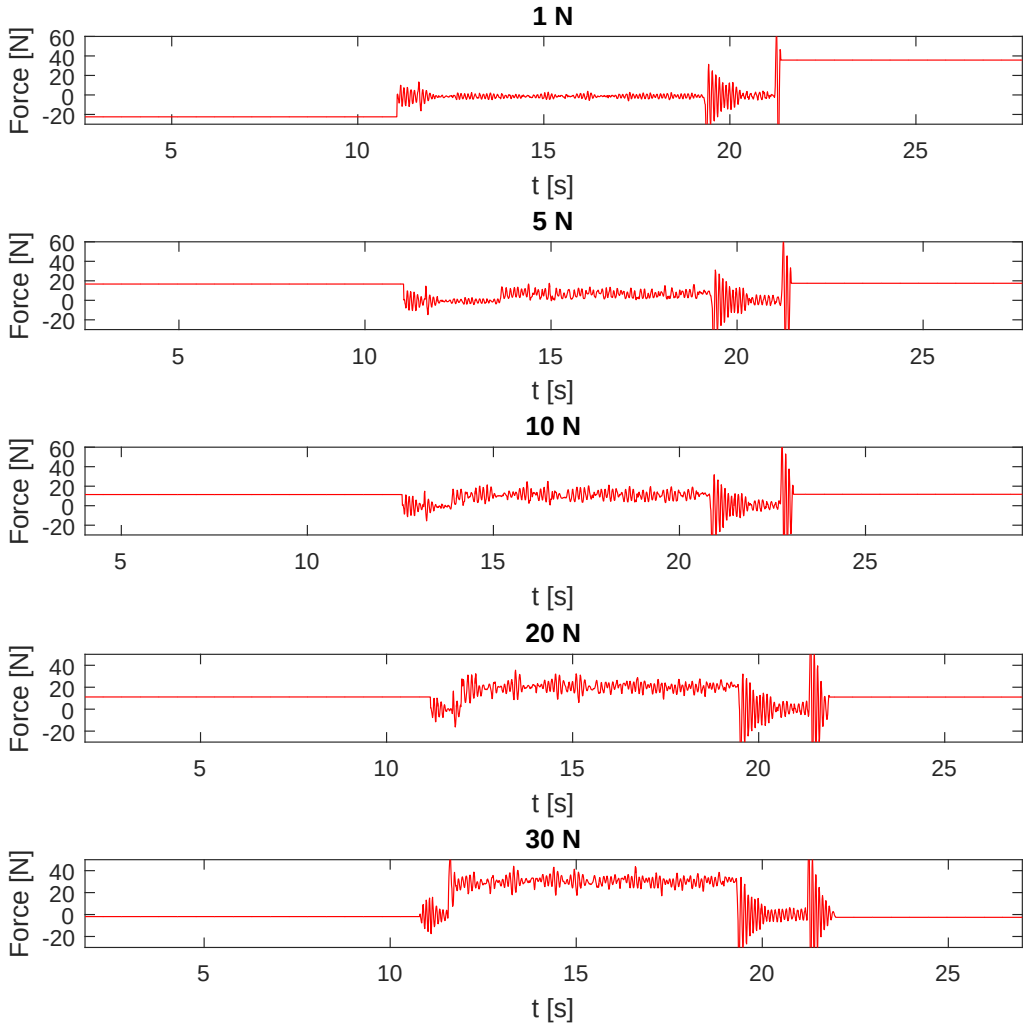


Figure 5.28: The force measurement for the raw data plotted with time sampled

Quite fascinating discovery in these plots were that the offset got smaller with how much pressure force was used. There are several possible reasons as to why this error emerged, and it could be everything from long downtime of the industrial manipulator to wrongly defined load data. Especially the downtime of the robot was of special concern, as the robot had not been used since Petter [17] performed the force control experiments. However knowing this gave no obvious solution to fix the problem, so the other ones have to be analysed more thoroughly.

Updating both the tool data and the load data did not seem to work either, and it was not until path 2 from table 5.1 was used that a reduction was observed. This could be explained from looking at the direction the robot moves as force control is turned off. In path 1 the robot changes its direction almost 180 degrees at its withdrawal phase, while in path 2 it continues forward in the same direction for the withdraw phase.

5.3.6 Frequency components

The topic of frequency components for the measured position and force will be examined further for a run using 20 N. This is done to get a better understanding of oscillatory behaviour that occurs during a run, and to look at ways to compensate for this unwanted behaviour. Figure 5.29 and 5.30 shows the plot of the force and z position of the robot during the process phase of the experiment.

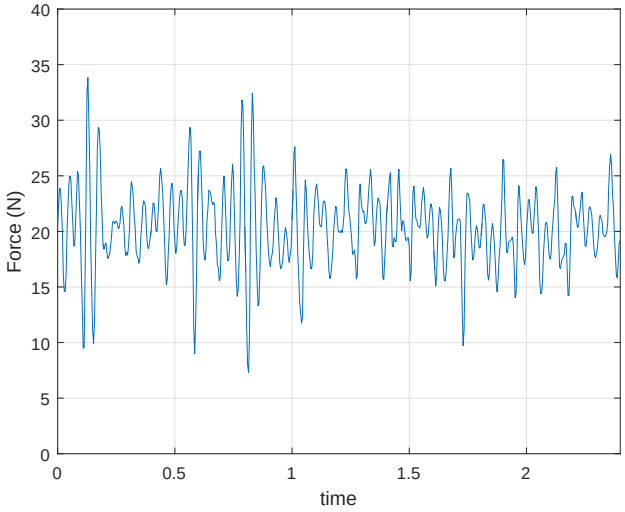


Figure 5.29: Oscillatory behaviour of the force measurement while processing for 20 N

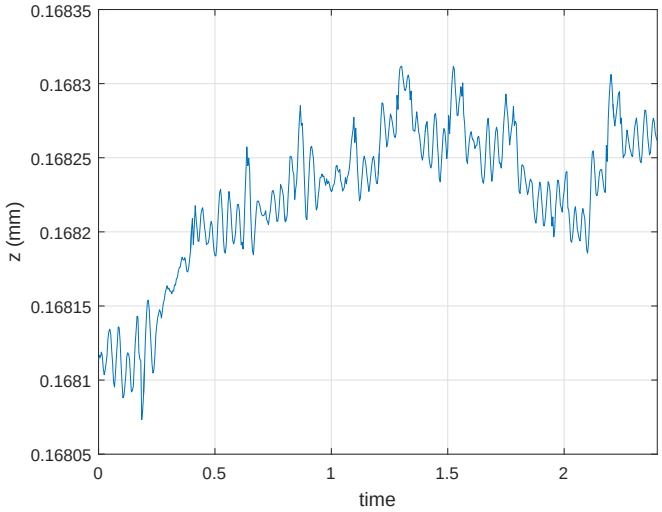


Figure 5.30: Oscillatory behaviour for movement in the z-direction while processing

Looking at the oscillations it is difficult to determine if they have similar behaviour. It is therefore needed to look at the power spectrum of these signals to get a better view. This can be seen in figure 6 in the appendix, and it shows that the dominant frequencies of the force measurments have the same form as the frequencies from encoder measurements.

It could also be valuable to look at the oscillations between two different forces from

the experiments done in this chapter, and to examine if the increase in force used increases oscillatory behavior. Comparing force plot in figure 5.29 and figure 5.31, it is seen that they have slightly different oscillations. It can also be seen that the oscillation for 10 N yield a higher peak value than for the oscillations for 20 N.

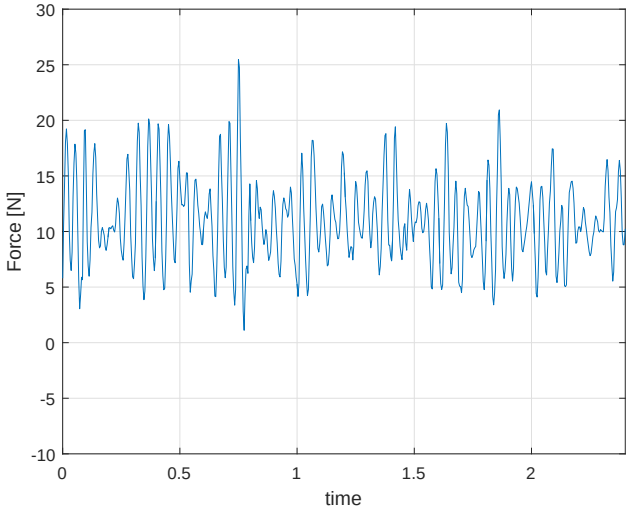


Figure 5.31: Shows the oscillations of the force for 10 N at a small time sample during the processing

This can be especially noticed by looking at the power spectrum in figure 5 in the appendix. A higher amplitude of noise is seen for 10 N at a higher frequency, while for 20 N a high amplitude of noise is seen at low frequencies.

5.3.7 Researching velocity components

In section 5.2.5 the result of an experiment with a different velocity was presented. The result in figure 5.7 showed that using less velocity along the path of the robot helped increase the quality of the force control, for example with lowering amplitude peak of the force and giving the robot enough time to achieve contact with the metal plate on all forces applied. With the reduced velocity experiment it took the robot approximately 15 s compared to 8 s for the earlier experiment.

Theoretically this reduces the oscillations of the robot from 0.13 Hz to 0.07 Hz, meaning

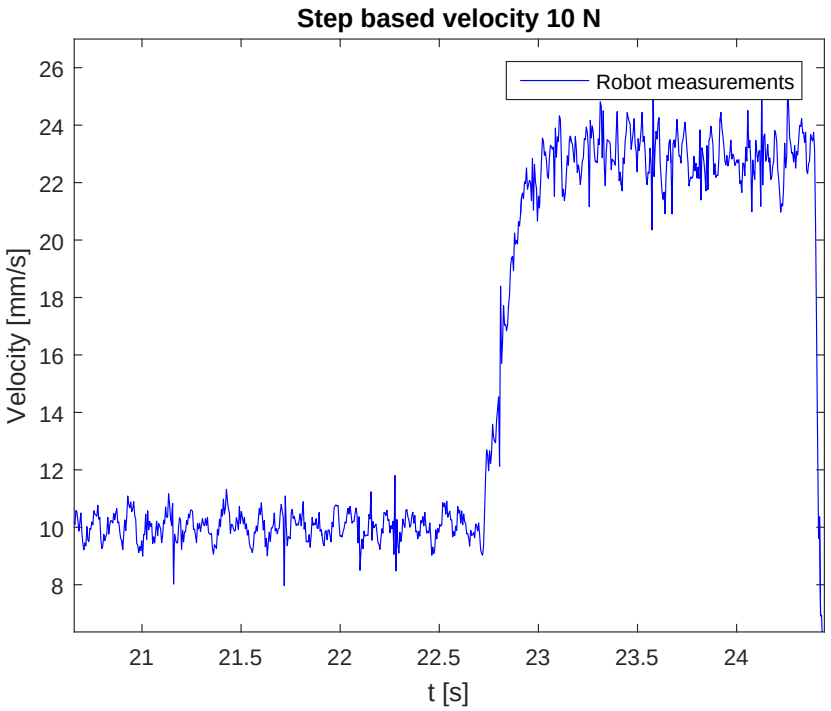


Figure 5.32: Shows the velocity of the tcp during the robot's path, where step based method is used during the processing

that it is basically halved. From figure 5.8 and 5.23 it is difficult to observe the reduction in oscillations, and it is therefore concluded that only reducing the velocity is not enough to get a good result. Also by reducing the velocity makes the procedure operate at a much too slow rate, which is not preferable in the industry in terms of efficiency of procedure.

It would therefore be better to do experiments with alternating velocities, which has the capability to gradually reduce the velocity as the robot approaches the point of impact. Just as suggested by Petter [17] polynomial and step based velocity may increase the performance during the process phase of the path. A pretty simple kind of step based velocity is possible to be tested using FC Machining. This can be done by setting an additional point which on a straight line between point 5 and 6 for path 2 in table 5.1, and then altering the velocity at this particular point. Figure 5.32 shows how the velocity changes using a step based velocity along the process phase from 10 mm/s to 20 mm/s. Using this method the oscillations on the process are changed to approximately 0.09 Hz.

Using the step based velocity it is still difficult to detect differences in oscillations, which

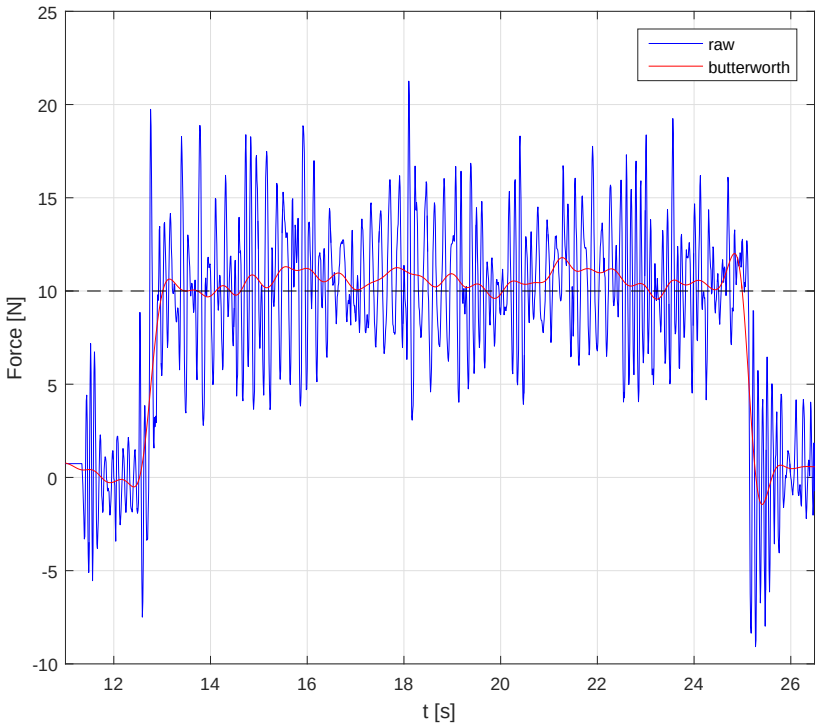


Figure 5.33: Shows the force measurements for 10 N using a step based velocity method

is seen in figure 5.33. Thus creating a controller to compensate for the oscillations when step based method is used would not be suitable. It could therefore be more interesting to investigate the polynomial velocity method, which is on the form

$$\dot{x} = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n$$

where n is the chosen number of polynomials. The velocity can be optimized by a controller to change as a polynomial, which depends on how far along the path the robot is.

In this chapter all force control experiments have been looked at and several topics related to the result have been discussed. Some of the plots discussed here that were not present can be found in the appendices.

Chapter 6

Conclusion

In this thesis the work has consisted of conducting several force control experiments using an IRB 4600 manipulator. The experiments can be divided into planning, executing and processing measurements gathered. K-610 series Optical CMM was used to capture additional measurements, which could be compared to the measurements from the robot encoders.

This was done foremost to validate the experiments conducted by Petter Kvernberg [17] using the same method and calibration procedure. During the experiment several issues with the equipment arised delaying the process, and it made it especially hard to compare some parts of the measurements. Especially an offset error in the force measurements was difficult to handle, as it changed for every run and got lower with the increase in force. It was believed that the error was caused by a incorrect load data, however as the load data was indentified several times with minimal changes this was concluded not to be the case. The offset error was reduced significantly from being around $\pm 20 - 40N$ to becoming around $\pm 1 to 2N$ using a different trajectory. The new trajectory prolonged the movement in the positive x-axis of the robot when withdrawing, such that the transition between process and withdraw became smoother.

The force measurements using the new path is quite comparable to the measurements from earlier experiments done using the same setup. Force control works quite well when the robot is in contact with the object, where for all the different forces applied the actual force stays between $\pm 0.5N$ to the desired force. Interesting to note is that the impact force increased tremendously with the force used, for example when using a force of 30 N a peak of almost 80 N in impact force for the raw signal was observed. With on-line low pass filter can the noise in the impact force be minizimed to approximately 46 N, which is still a relatively high value. In the case of using a small force of 1 N it was not possible for the manipulator to even come in contact with the workobject, as can be seen in the plots for the position in figure 5.12. This can be explained by the fact that the force sensor has a resolution at 1-2 N, which makes it hard for the robot to determine if a force of 1 N is applied or not.

Getting well defined frames to be used by the camera system was quite difficult to accomplish, as the error increased with the distance from the calibration area. It was attempted to use an additional LED for the tool to increase the accuracy of the camera system and additional calibration of the frames. The result was that the deviation error between

camera and robot encoder was reduced to approximately $x_d = 0.25$ mm and $z_d = 0.092$ mm. At the same time the deviation in the y-axis increased to $y_d = 1.8$ mm, which could be contributed from calibration error in the orientation of the frames. It could also be contributed from displacement error in the robot encoders. Due to an error with the camera system, where the camera was not being able to see the LEDs at different time steps when measuring, it was not possible to do additional experiments with a different calibration.

It was decided to investigate several filters in post processing of the raw force signals with the different filters being Butterworth, Chebyshev Type II and Elliptic. This was done to determine which filter removed noise best without altering the behaviour of the signal, and to get a better clue as to how close the force measurements stayed to its desired force value. The different filters relied on different specifications, which made it necessary to first tune the different filters to a good response and then analyse the parameters used. While the Butterworth filter would require a rather low cross rate at around 5 Hz to reduce all noise, the Chebyshev filter could have a cross rate of 16 Hz. At the same time both the Chebyshev and Elliptic filter depended on choosing a reasonable passband and stopband attuenation, which is not the case for the Butterworth filter. Looking at the group and phase delay it was seen that the Butterworth filter outperformed the other two filters with a very low delay in both magnitude and phase. It was therefore concluded that the best filter to be used for post processing of the force signals was the Butterworth filter.

An experiment using a different velocity along the robots trajectory was conducted to discover how manipulator behaved differently with change in velocity. By finding unwanted behaviour using different velocities, controllers can be used for compensation. This was also suggested by Petter [17] were he also proposed to use two alternative velocity methods, namely step based method and polynomial. It was tested using a velocity of 10 mm/s rather than 20 mm/s along the contact part of the trajectory. The result of this experiment showed that by moving slower made the robot be able to come in contact with the metal bar for a force of 1 N. Still the force applied for 1 N is a lot higher than desired when in contact with the workobject. It was also discovered that the impact force was lowered somewhat for the raw force signals, while also increasing the duration of which the impact force was acting. The oscillations for the force signals decreased slightly from 0.14 Hz to 0.06 Hz in this experiment, which was not a very noticable change in the force plots. It was also experimented with using a method that changed velocity during the contact phase with the metal just as the step based method. The result showed little to no change in behaviour of the robot manipulator, which makes it more interesting to experiment with the polynomial method.

It was also planned to do experiments with the active use of drilling spindle, however this was not possible as the necessary training and security measurments were not provided for in the time. It will therefore be interesting to investigate the effects of using a drilling spindle in the future. This thesis provides a clear understanding of how to conduct experiments and process measurements, and also discusses several aspects related to force control used in machining operations.

Bibliography

[1] ABB. Test signal viewer manual. 2005.

[2] ABB. Product manual irb 4600. 2009.

[3] ABB. Application manual, force control. 2015.

[4] ABB. Flexpendant. 2016. http://www.posr.us/store/pc/viewPrd.asp?idproduct=1508.

[5] ABB. Irb 4600 cad schematics. 2016. http://new.abb.com/products/robotics/industrial-robots/irb-4600/irb-4600-cad.

[6] Gürsel Alici and Bijan Shirinzadeh. Enhanced stiffness modeling, identification and characterization for robot manipulators. 2005.

[7] ATI Industrial Automation. Installation and operation manual, force/torque sensor system. 2005.

[8] Reginald L. Gott Bruce M. Kramer, John J. Bausch and David M. Dombrowski. Robotic deburring. 1984.

[9] Giordano Colombo. Ra rc 110 sn manual. 2010.

[10] Comet. Comet project. 2013. http://www.cometproject.eu/.

[11] George C. Devol. Patent of the reprogrammable manipulator unimate. http://cyberneticzoo.com/early-industrial-robots/1954-programmed-article-transfer-patent-george-c-devol-jr-american/.

[12] Isolde Dressler. Force control interface for abb s4/irc5. 2009.

[13] Vulcan Engineering. Grinding example. 2013. http://castingssa.com/vulcan-engineering-and-acetarc-engineering-partner-with-globen-engine

[14] I. Sharf G. Gilardi. Literature survey of contact dynamics modelling. 2002.

[15] ISO. Iso standard 8373:2012 for robotic device. 2012. https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en.

[16] Fracois Leonard Jinna Qin and Gabriel Abba. Real-time trajectory compensation in robotic friction stir welding using state estimators. 2016.

[17] Petter Kvernberg. Developing force control scenarios on abb irb 4600 with camera capture of dynamic motions. 2015.

[18] Jin-Oh Kim Matthew Wayne Gertz and Pradeep K. Khosla. Exploiting redundancy to reduce impact force. 1991.

[19] Nikon Metrology. K - series training manual. 2015.

[20] International Federation of Robotics. Industrial robot statistics. 2015. http://www.ifr.org/industrial-robots/statistics/.

[21] Alan V Oppenheim, Ronald W Schafer, John R Buck, et al. *Discrete-time signal processing*, volume 3. Prentice hall Englewood Cliffs, NJ, 2010.

[22] John G Proakis and Dimitris G Manolakis. *Digital Signal Processing*, volume 4. Pearson Education, 2014.

[23] Researchgate. Advanced quaternion forward kinematics algorithm including overview of different methods for robot kinematics - scientific figure. 2014. https://www.researchgate.net/262561339_fig1_Figure-6-Frames-and-parameters-in-D-H-convention.

[24] Eric Roberts. History of robotics. 1998. https://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/history.html.

[25] Mark W Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[26] Leslie D Thede. *Practical analog and digital filter design*. Artech House New Jersey, 2005.

[27] Matteo Ansaloni Ulrich Schneider, Mahdi Momeni-K and Alexander Verl. Stiffness modeling of industrial robots for deformation compensation in machining. 2014.

[28] Steven Xie. Practical filter design challenges and considerations for precision adcs. 2016. http://www.analog.com/library/analogdialogue/archives/50-04/practical-filter.html.

[29] Fenghua Dong Yonghua Chen. Robot machining, recent development and future research issues. 2012.

Appendices

Abbreviation

DH - Denavit- Hartenberg
DOF - Degrees of Freedom
FC - Force control
HMI - Human Machine Interface
IRB 4600 - Industrial robot 4600 by ABB
LED - Light Emitting Diode
PSD - Power Spectral Density
TSV - Test Signal Viewer
TCP - Tool Center Point

Additional Plots

Force Measurements

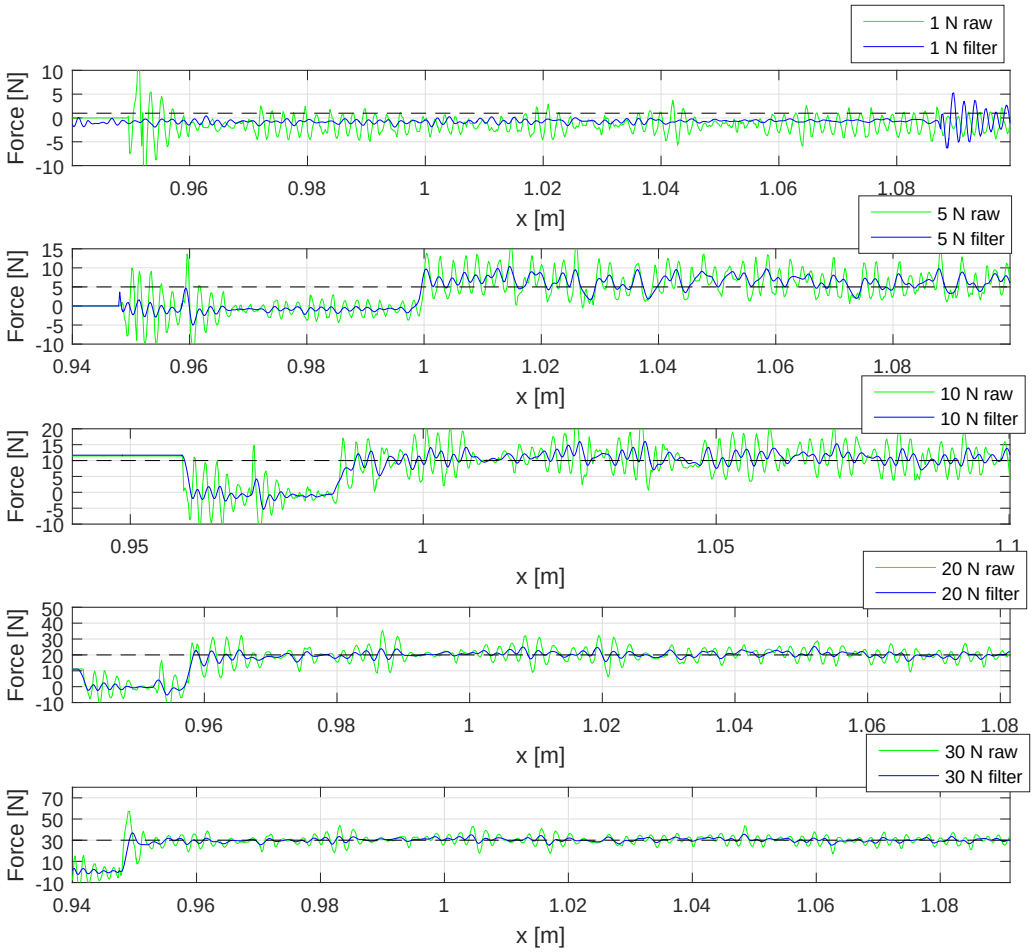


Figure 1: Force measurements from 1-30 N for path 1

Position Measurements

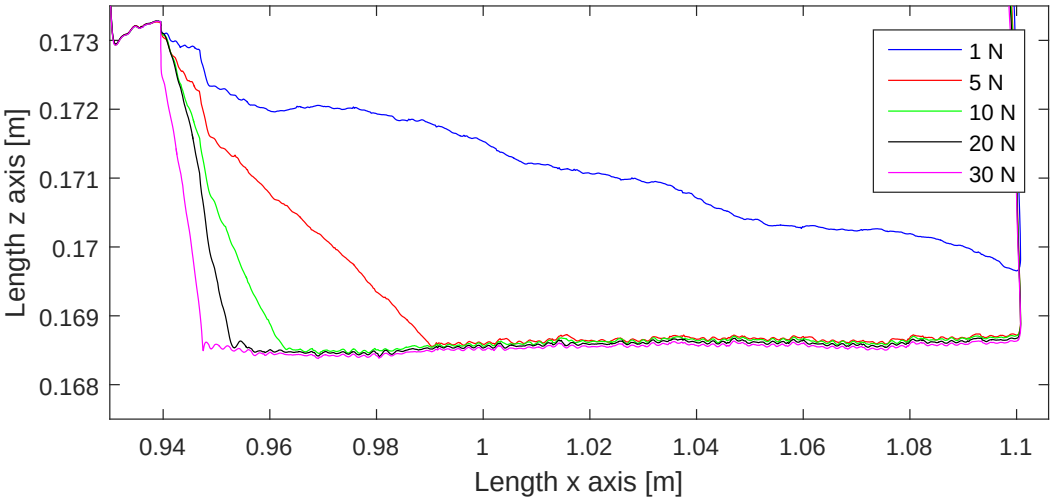


Figure 2: Position measurements from 1-30 N for path 1 measured by robot encoders

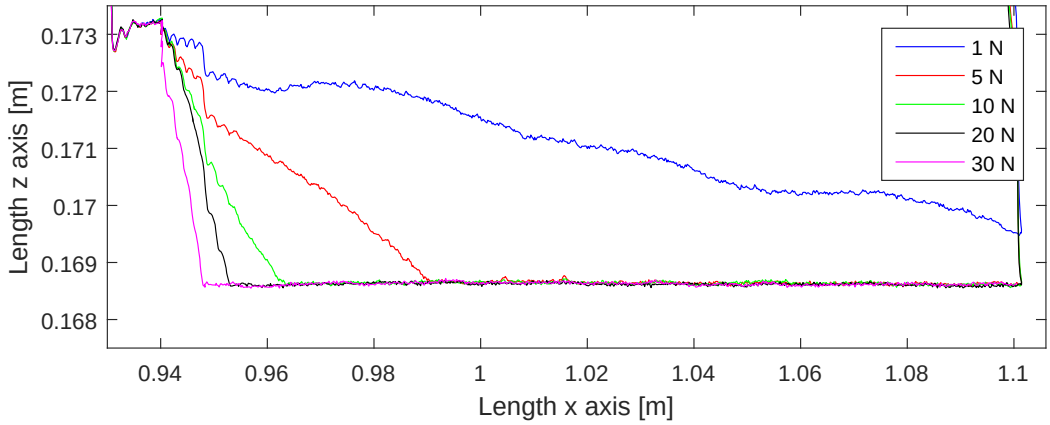


Figure 3: Position measurements from 1-30 N for path 1 by camera system

Frequency Measurements

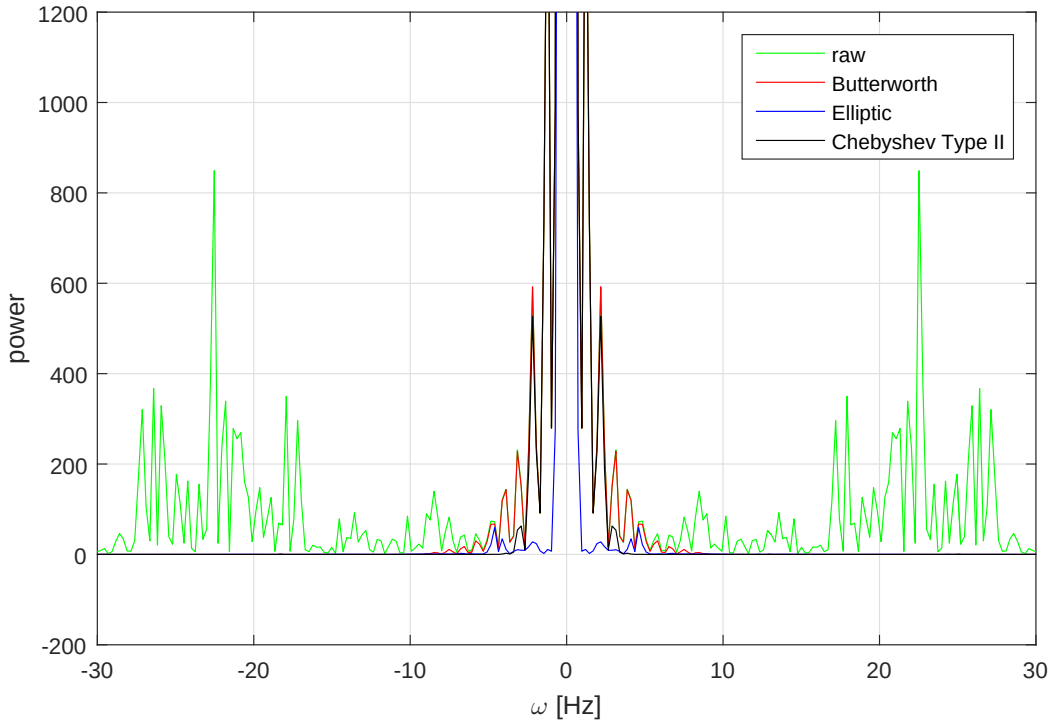


Figure 4: Plot of the power spectrum density of the different filters along with the raw data

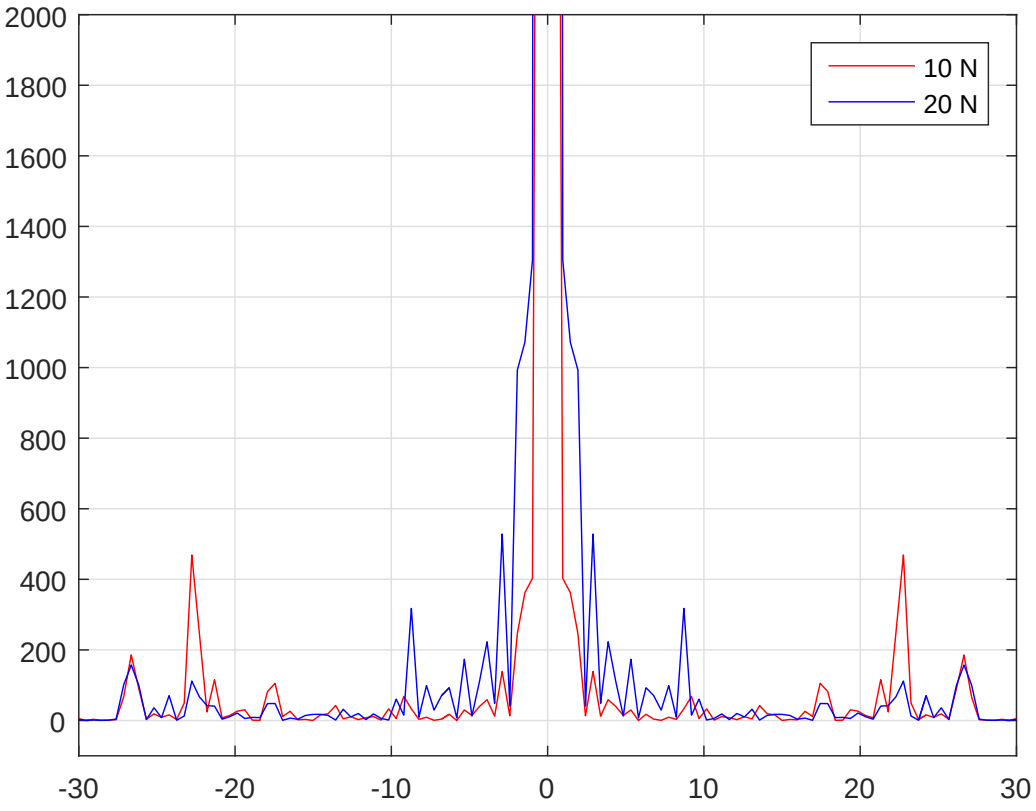


Figure 5: Power spectrum density of the force for 10 and 20 N

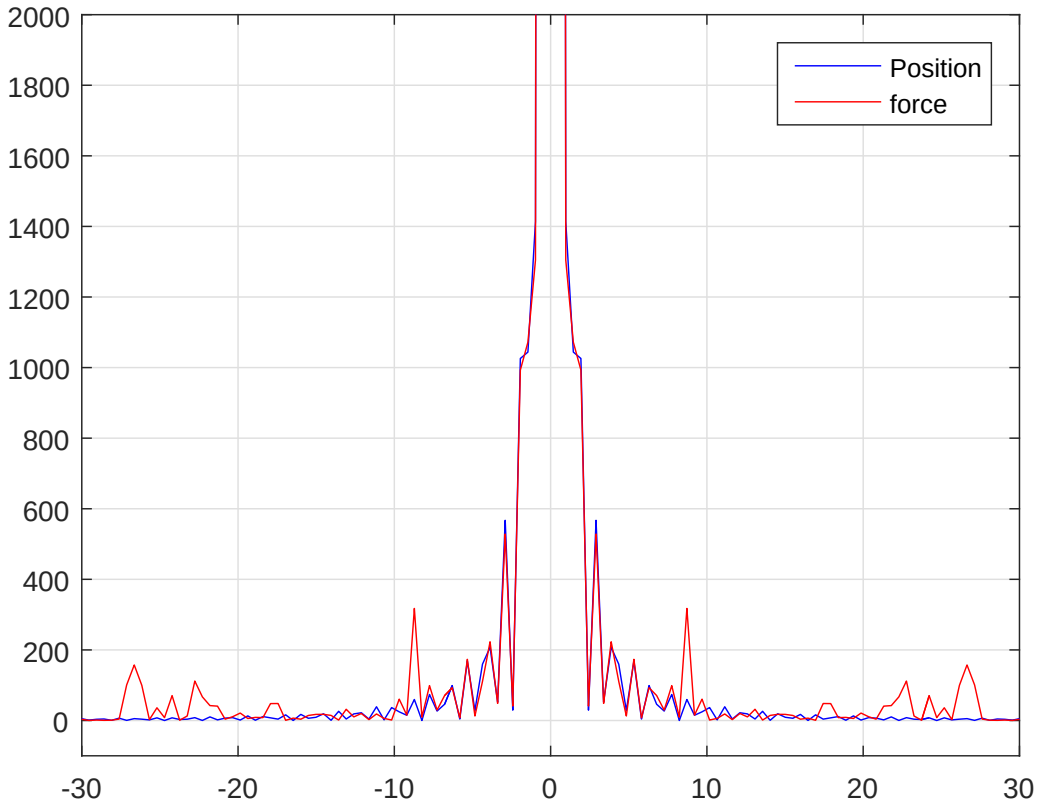


Figure 6: Power spectrum density of the position and force for 20 N

Example of Machining FC code

```
PROC RunPathFCpath2 (bool bRecover, bool bSpindleOn)
        FCDeact;

        IF bRecover = true THEN
                FCCalib  toolMiller_LD \Recovery;
        ELSE
                FCCalib  toolMiller_LD;
        ENDIF

        IF bSpindleOn = true THEN
                SpindleOn;
        ENDIF

        MoveL T2Approach1, v20, z1,toolMiller\wobj :=wobj0;
        MoveL T2Approach2, v20, z1,toolMiller\wobj :=wobj0;
        MoveL T2Approach3, v20, fine,toolMiller\wobj :=wobj0;

        FCPress1LStart T2Process1, v20 \Fx:= n2ForceX \Fy:= n2ForceY \Fz:=

        FCPressL T2Process2, v20, 10, z1, toolMiller, \wobj:=wobj0;
        FCPressL T2Process3, v20, 10, z1, toolMiller, \wobj:=wobj0;

        FCPressLEnd T2Withdraw1,v50,\ForceChange :=50,\ZeroContactValue :=
        MoveL T2Withdraw2, v50, z1,toolMiller\wobj :=wobj0;

        IF bSpindleOn = true THEN
                SpindleOff;
        ENDIF

ENDPROC
```

Robotics

IRB 4600
Industrial Robot

The IRB 4600 is a pioneer of the sharp robot generation; with enhanced features and new capabilities. The design has been optimized to make it superior for the targeted applications. The IRB 4600 enables more compact manufacturing cells with increased production output and higher quality - and that means improved productivity.

Sharpest accuracy

With the best accuracy in its class, the IRB 4600 can help you increase output with higher process speeds and lower scrap rates, resulting in improved productivity. This is particularly useful in materials handling, dispensing, machining, measuring, assembly and welding applications. In addition, the programming time is minimized since what you program is what you get, and that in the shortest possible cycle time. This is useful in all applications to shorten commissioning times and minimize production stops when new programs or work pieces are introduced.

Shortest cycle times

Thanks to the new compact and optimized design resulting in a low weight, the IRB 4600 can cut the cycle times of the industry benchmark by up to 25%. The maximum acceleration achievable is highest in its class, together with high maximum speeds. The high acceleration is possible to use to avoid obstacles or to follow the path. The benefit is increased production capacity and higher productivity.

Ultra-wide working range

You can position the IRB 4600 in the most favourable way with regard to reach, cycle time and auxiliary equipment. Flexible mounting with floor, tilted, semi-shelf or inverted mounting is very useful when you are simulating the best position for your application.

Compactness

The small footprint, the slim swing base radius around axis 1, the fine elbow behind axis 3, the small lower and upper arms, and the compact wrist all contribute to the most compact robot in its class. With the IRB 4600 you can create your production cell with reduced floorspace by placing the robot closer to the served machines, which also increases your output per m^2 and your productivity.

Best protection available

ABB has the most comprehensive protection program on the market and it will be even further enhanced with the IRB 4600. Foundry Plus includes IP 67, resistant paint, rust-protected mounting flange and protection for molten metal spits on non-moving cables on the rear of the robot and extra protection plates over the floor cable connections on the foot.

Optimize and go sharp

To get the IRB 4600 ready for the targeted applications you have access to high performing workpiece positioners, Track motions, and the motor unit range.

To simulate your production cell to find the optimal position for the robot and program it offline, RobotStudio is available on subscription together with PowerPacs for several applications.

If you would like to learn more about how to use the IRB 4600 in your applications and environments, you can watch simulations on several applications at www.abb.com/robotics

IRB 4600

Main applications

Machine tending, Material handling, Arc welding, Cutting, Dispensing, Assembly, Palletizing and packing, Measuring

Specification

Variants:	Reach	Payload	Armload
IRB 4600-60/2.05	2.05 m	60 kg	20 kg
IRB 4600-45/2.05	2.05 m	45 kg	20 kg
IRB 4600-40/2.55	2.55 m	40 kg	20 kg
IRB 4600-20/2.50	2.51 m	20 kg	11 kg
Number of axes:	6+3 external (up to 36 with MultiMove)		
Protection:	Standard IP67, as option Foundry Plus 2		
Mounting:	Floor, shelf, inverted or tilted		
IRC5 Controller variants:	Single cabinet, Dual cabinet		

Physical

Dimensions robot base:	512 x 676 mm
Robot height: IRB 4600-60/2.05 and IRB 4600-45/2.05	1727 mm
Robot height: IRB 4600-40/2.55 and IRB 4600-20/2.50	1922 mm
Robot weight:	412 to 435 kg

Performance

Position repeatability (RP)	0.05 - 0.06 mm
Path repeatability (RT)	0.13 - 0.46 mm (measured at speed 250 mm/s)

Movement

Axis movements:	Working range:	Maximum speed:
Axis 1	+180° to -180°	175°/s
Axis 2	+150° to -90°	175°/s
Axis 3	+75° to -180°	175°/s
Axis 4	+400° to -400°	250° (20/2.50 has 360°)/s
Axis 5*	+120° to -125°	250° (20/2.50 has 360°)/s
Axis 6	+400° to -400°	360° (20/2.50 has 500°)/s

* Axis 5 for IRB 4600-20/2.50 +120°-120°

Electrical connections

Supply voltage:	200-600 V, 50-60 Hz

Environment

Ambient temperature for mechanical unit:

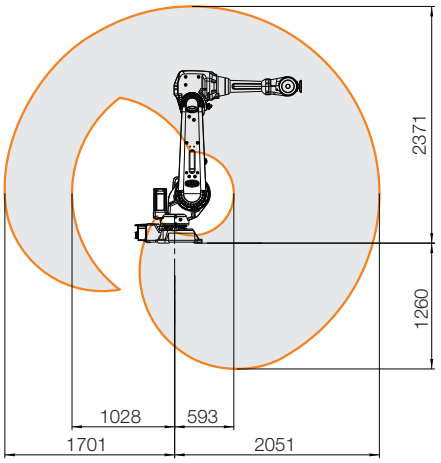
During operation:	+5° C (41° F) to + 45°C (113°F)
During transportation and storage:	-25° C (-13° F) to +55° C (131° F)
For short periods (max 24 h):	up to +70° C (158° F)
Relative humidity:	Max 95%
Safety:	Double circuits with supervisions, emergency stops and safety functions. 3-position enable device
Emission:	EMC/EMI shielded

Data and dimensions may be changed without notice.

Working range

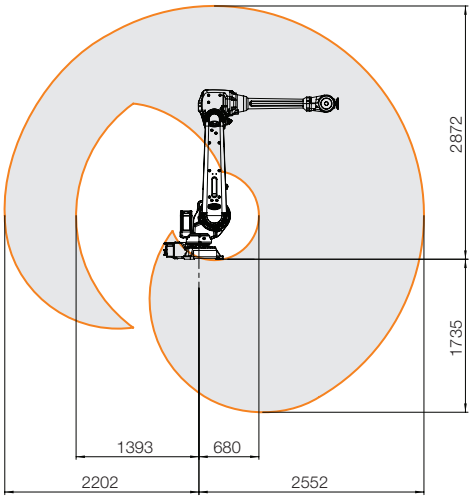
IRB 4600-60/2.05

IRB 4600-45/2.05



IRB 4600-60/2.05
IRB 4600-45/2.05

IRB 4600-40/2.55



IRB 4600-20/2.50