NTNU – Trondheim
Norwegian University of
Science and Technology

Improvement of the Positioning Accuracy of Industrial Robots

Vegard Johnsrud

Mechanical Engineering	
Submission date:	June 2014
Supervisor:	Knut Sørby, IPK
Co-supervisor:	Trygve Thomessen, PPM AS

Norwegian University of Science and Technology
Department of Production and Quality Engineering

MASTEROPPGAVE
Våren 2014
for stud. techn. Vegard Johnsrud

Forbedring av posisjoneringsnøyaktigheten til industriroboter
Improvement of the positioning accuray of industrials robots

Industriroboter benyttes i økende grad til oppgaver som krever høy posisjoneringsnøyaktighet, for eksempel finbearbeiding med freseverktøy eller dimensjonsmåling i ulike typer produksjonsprosesser. Oppgaven går ut på å analysere forhold som har betydning for industriroboters nøyaktighet, og studenten skal vurdere metoder for å forbedre nøyaktigheten til eksisterende industriroboter. Arbeidet gjennomføres i samarbeid med PPM. Oppgaven skal omfatte:

1. Kartlegging av forhold som har betydning for posisjoneringsnøyaktigheten
2. Metoder for måling av nøyaktighet
3. Forbedring av nøyaktighet ved justering av kinematikkparametre og andre metoder
4. Verifisering av posisjoneringsnøyaktighet gjennom praktiske forsøk

Oppgaveløsningen skal basere seg på eventuelle standarder og praktiske retningslinjer som foreligger og anbefales. Dette skal skje i nært samarbeid med veiledere og fagansvarlig. For øvrig skal det være et aktivt samspill med veiledere.

Innen tre uker etter at oppgaveteksten er utlevert, skal det leveres en forstudierapport som skal inneholde følgende:

- En analyse av oppgavens problemstillinger.
- En beskrivelse av de arbeidsoppgaver som skal gjennomføres for løsning av oppgaven. Denne beskrivelsen skal munne ut i en klar definisjon av arbeidsoppgavenes innhold og omfang.
- En tidsplan for fremdriften av prosjektet. Planen skal utformes som et Gantt-skjema med angivelse av de enkelte arbeidsoppgavenes terminer, samt med angivelse av milepæler i arbeidet.

Forstudierapporten er en del av oppgavebesvarelsen og skal innarbeides i denne. Det samme skal senere fremdrifts- og avviksrapporter. Ved bedømmelsen av arbeidet legges det vekt på at gjennomføringen er godt dokumentert.

Besvarelsen redigeres mest mulig som en forskningsrapport med et sammendrag både på norsk og engelsk, konklusjon, litteraturliste, innholdsfortegnelse etc. Ved utarbeidelsen av teksten skal kandidaten legge vekt på å gjøre teksten oversiktlig og velskrevet. Med henblikk på lesning av besvarelsen er det viktig at de nødvendige henvisninger for korresponderende steder i tekst, tabeller og figurer anføres på begge steder. Ved bedømmelsen legges det stor vekt på at resultatene er grundig bearbeidet, at de oppstilles tabellarisk og/eller grafisk på en oversiktlig måte og diskuteres utførlig.

Materiell som er utviklet i forbindelse med oppgaven, så som programvare eller fysisk utstyr er en del av besvarelsen. Dokumentasjon for korrekt bruk av dette skal så langt som mulig også vedlegges besvarelsen.
Kandidaten skal rette seg etter arbeidsreglementet ved bedriften samt etter eventuelle andre pålegg fra bedriftsledelsen. Det tillates ikke at kandidaten griper inn i betjeningen av produksjons-maskineriet, idet alle ordrer skal formidles på vanlig måte gjennom fabrikkens bedriftsledelse.

Eventuelle reiseutgifter, kopierings- og telefonutgifter må bære av studenten selv med mindre andre avtaler foreligger.

Hvis kandidaten under arbeidet med oppgaven støter på vanskeligheter, som ikke var forutsett ved oppgavens utforming og som eventuelt vil kunne kreve endringer i eller utelatelse av enkelte spørsmål fra oppgaven, skal dette straks tas opp med instituttet.

Oppgaveteksten skal vedlegges besvarelsen og plasseres umiddelbart etter tittelsiden.

Innleveringsfrist: 10. juni 2014.

Besvarelsen skal innleveres i 1 elektronisk eksemplar (pdf-format) og 2 eksemplar (innbundet), ref. rutinebeskrivelse i DAIM. Det vises til http://www.ntnu.no/ivt/master-siv-ing for ytterligere informasjon om DAIM, uttak, kontrakt, gjennomføring og innlevering.

Ansvarlig faglærer/veileder: Knut Sørby
Telefon: 73 59 03 74 / 918 97 328
E-post: knut.sorby@ntnu.no

Kontaktperson ved PPM: Trygve Thomessen
Telefon: 73 96 50 50
Mobiltelefon: 922 42 189
E-post: tth@ppm.no

INSTITUTT FOR PRODUKSJONS-
OG KVALITETSTEKNIKK

Per Schjølberg

Per Schjølberg
Førsteamanuensis / instituttstyrer

Knut Sørby

Knut Sørby
Ansvarlig faglærer

Summary

Industrial robots are well known for their high repeatability, as reflected in the main industrial applications, such as simple pick-and-place operations and spot welding. However, with the increasing industry demand of automation, robots represent an unique possibility for automation without sacrificing flexibility. As this report points out, the main challenge for industrial robots is to improve their positioning accuracy. This challenge can be solved either by changing the mechanical design of the robot or the more feasible alternative is to improve or alter the software in the robot controller. The last alternative is known as robot calibration, and is the method used in this master thesis. In order to improve the positioning accuracy of an industrial robot, the target for calibration should be to reduce be the most contributing error source, which is the joint angle offset error. This is more commonly known as the joint offset error.

As of this, the practical implementations performed in this thesis center around performing a joint offset calibration using a tool mounted laser distance sensor and a plane as calibration object. The master thesis is based on a mathematical model consisting of a kinematic model of the robot, a joint encoder model, a laser sensor model and a plane model. This mathematical model is used to find a computational model of the laser distance measurement as a function of the joint angles and the model parameters, where the interesting model parameters are the joint-offsets. In order to find the best-fit model between the actual setup and the mathematical model, a sampling operation is conducted to gather samples in the actual setup, where the laser distance measurement is stored with the accompanying joint encoder position. Finally a least square optimization method is applied to calculate the updated model parameters that give the best fit with the sampled data.

The implementation was realized by creating two LabVIEW applications; one for performing the sampling operation and one for performing the optimization operation. The sampling application uses remote control of the robot to create random sample poses with laser measurements against the calibration plane, and storing the distance measurement and encoder data to a data file. In the optimization application, the data set from the sampling operation is used to optimize the system model by updating the model parameters, thus resulting in the updated joint-offsets.

The master thesis presents how the computational model is obtained and how it is applied to an actual setup in the research facility of PPM AS. Tests are performed to verify the computational model and the calibration principle. The tests show that the computational model is adaptable to different robot models, but small changes to the robot model apply. The tests also show that the computational model gives a better fit to the sampled data when using the updated model parameters, and also how the parameter correction fluctuates between the various sample sets. This highlights the need for a statistical analysis of the calibration results. Due to time limitations, statistical analysis of the results and final validation experiments of the calibrated robot was not achieved. Thus, the report has a separate section describing how future work should be carried out.

Sammendrag

Industriroboter er godt kjent for sin høye repeterbarhet, noe som reflekteres i deres hovedbruksområde i industrien, deriblant ”pick and place”-operasjoner og punktsveising. Med det økende behovet for automatisering i industrien, representerer roboter en unik mulighet for nettopp dette uten å redusere fleksibiliteten. Som det imidlertid påpekes i denne masteroppgaven, ligger hovedutfordringen for industriroboter i å forbedre dårlig posisjoneringsnøyaktighet. For å løse denne utfordringen kan man endre på robotens mekaniske utforming, men et mer gjennomførbart alternativ er å forbedre eller endre programvaren i robotens styreenhet. Det siste alternativet er bedre kjent som robotkalibrering, og er metoden som benyttes i denne oppgaven. For at posisjoneringsnøyaktigheten til en industrirobot skal forbedres, bør formålet med kalibreringen være å redusere hovedårsaken til dårlig posisjoneringsnøyaktighet. Dette handler i de fleste tilfeller om robotleddenes vinkelforskyvningsfeil, også kalt offset-feil. I de praktiske implementeringene i denne masteroppgaven, utføres offset-kalibrering ved hjelp av en flensmontert laser avstandssensor og en plan flate som kalibreringsobjekt.

Masteroppgaven tar utgangspunkt i en matematisk modell basert på robotens kinematiske modell, en enkoder-leddmodell, en lasersensormodell og en plan modell. Den matematiske modellen benyttes for å sette opp en beregningsmodell for laseravstandsmålingen som en funksjon av leddvinkler og de øvrige modellparameterne. Der de interessante modellparameterne er leddforskyvninger (offsets), som skal korrigere for leddenes vinkelforskyvningsfeil. For å utvikle en optimal modell som best beskriver det virkelige oppsettet, er det utført flere datainnsamlinger bestående av ulike avstandsmålinger mellom laser og plan med tilhørende vinkelkonfigurasjoner for roboten. Avslutningsvis er det brukt en minstekvadratersmetode optimeringsalgoritme for å identifisere hvilke modellparametere som gir minst avvik mellom beregningsmodellen og faktiske målinger fra datasettet. Fra dette får man leddforskyvningskorreksjonen man er ute etter.

Implementeringen er realisert ved å utvikle to LabVIEW-applikasjoner; én for å utføre datainnsamlingen og én for utførelse av optimalisering. Datainnsamlingsapplikasjonen bruker ”fjernkontroll” for styring og kommunikasjon av roboten, der den lager et tilfeldig utvalg av robotpositurer med tilhørende lasermåling mot et plan. For hver positur lagres robotleddenes posisjon i enkoderverdier og laserens avstandsmåling til et datasett. I optimaliseringsapplikasjonen brukes datasettet fra innsamlingen til å optimalisere beregningsmodellen, noe som resulterer i nye modellparametere og dermed også leddforskyvingskorreksjon.

Masteroppgaven presenterer beregningsmodellens prinsipp og hvordan den er tilegnet bruk på det faktiske oppsettet i forskningsanlegget til PPM AS, samt inneholder tester for å verifisere beregningsmodellen og kalibreringsprinsippet. Testene viser at beregningsmodellen er overførbar til andre robotmodeller, og at den gir en bedre tilpasning til innsamlede data ved bruk av oppdaterte modellparametere funnet i kalibreringen. Testene viser også hvordan parameterkorreksjon i kalibreringen svinger mellom de forskjellige datasettene, noe som belyser behovet for en statistisk analyse av kalibreringsresultatene. På grunn av tidsbegrensninger for gjennomføring av oppgaven, ble den statistiske analysen og endelig valideringsforsøk av den kalibrerte roboten ikke oppnådd. Oppgaven har imidlertid en egen seksjon som beskriver hvordan videre arbeid bør foregå.

Preface

This is a master thesis in production systems, which is a part of an integrated master's degree in mechanical engineering at NTNU, Trondheim.

The thesis is carried out by stud. techn Vegard Johnsrud, on behalf of NTNU and PPM AS.

The problem description is developed by Prof. Knut Sørby at IPK (NTNU) and Prof. Ph.D. Trygve Thomessen, managing director of PPM AS, in close collaboration with the candidate. Due to the amount of work and challenges with the implementation of the robot calibration, as well as the time limitations of a master thesis, the verification experiments of the final result are not performed. Instead verification tests are presented as suggestions for future work. Also theoretical studies concerning accuracy measurement and calibration of industrial robots are given lower priorities, and should therefore be evaluated as such.

The thesis is to be evaluated on the basis of a written report, as well as other material pertaining to it.

Trondheim, 2014-06-08

Vegard Johnsrud

Vegard Johnsrud

Acknowledgement

First and foremost I would like to thank Ph.D. Morten Lind for his great help and support during the thesis project.

I would also like to thank my supervisors; Prof. Ph.D. Trygve Thomessen for his support and for giving me access to PPM AS' research facility and robots, and Prof. Knut Sørby for his support and help in the workshop throughout the semester.

I would also like to thank my colleagues at PPM AS, PhDc Balázs Daniel, PhDc Audun Sanderud and MSc Laszlo Nagy for both technical help and help through discussions.

Finally I would like to thank Nina Helen Aas Røkkum for her moral support, patience and help during the work with my thesis.

VJ

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The foundation of the problem formulation in this thesis was developed during the candidate's specialization project (TPK4150) carried out in the spring of 2013. In this project, titled "Robotized Milling of Wooden Products", a robotized milling cell was built around the NACHI MC70 industrial robot in the PPM lab, which is capable of milling various geometries in soft materials. This was accomplished by creating computer models and cutting trajectories with CAD/CAM software and a custom made post-processor software translating the cutting trajectories to robot trajectories. The project was carried out in close collaboration with a Hungarian B.Sc, Nyirő Péter, who developed the post-processor software as a part of his bachelor thesis.

As a final step of the project, one of the finished products (see figure 1) where taken to the metrology lab at IPK where its dimensions were measured with a CMM, and compared to the CAD model. The CMM machine revealed a deviation from the CAD model in the range of 0.5 – 1.1 mm for the length and width of the milled squares. An interesting point was the repeatability of the error, as each measured width of the squares showed the 1 mm error and the lengths showed the 0.5 mm error. Due to time limitations, the errors were not investigated in the specialization project. Instead the errors laid the basis for the problem formulation in this master thesis.

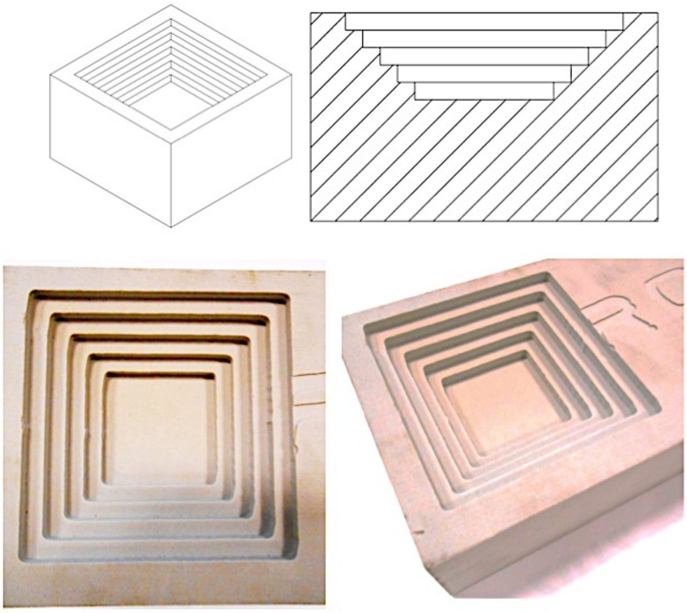


Figure 1 CAD Model (top) and finished product (bottom)

1.2 Problem formulation

The focus of this master thesis is on investigating the different factors affecting the robots' positioning accuracy and exploring the possibilities for improving the positioning accuracy of industrial robots through calibration. The main focus will be on implementing a calibration procedure previously implemented by Ph.D. Morten Lind, Senior Researcher at SINTEF Raufoss Manufacturing, for experimental use on a Universal Robot UR5. His experiments on the UR5 proved an unsatisfactory calibration result. As a natural second step, the candidate is to implement the procedure on a robot dissimilar to the UR5 at the PPM lab in order to verify generality of the mathematical model of the calibration principle, and to test if more satisfying results can be obtained. Hence, this is a research oriented master thesis and should be recognized as such.

Objectives:

In order to meet the requirements in the problem description, the following objectives had to be accomplished:

1. Conduct a study on accuracy assessment of industrial robots.
2. Conduct a study of methods for measuring robot accuracy.
3. Study the ISO standards on accuracy assessment of industrial robots.

4. Study the various error sources contributing to reducing the positioning accuracy of industrial robots.
6. Conduct introductory study on methods to calibrate industrial robots.
7. Obtain an overall understanding of the principle and procedure of performing joint offset calibration on a plane.
8. Implement the calibration procedure at the PPM lab.
9. Perform experiment verifying the accuracy improvement from calibration.
10. Documentation of experimental setup, hardware and software.
11. Documentation of experimental results.

1.3 Available Hardware

This section presents the available hardware during the work of the thesis. Datasheets can be found in appendix H.

Industrial robot

The industrial robot available in this project was the NACHI MC70, a six axis serial manipulator running on the NACHI FD11 robot controller.

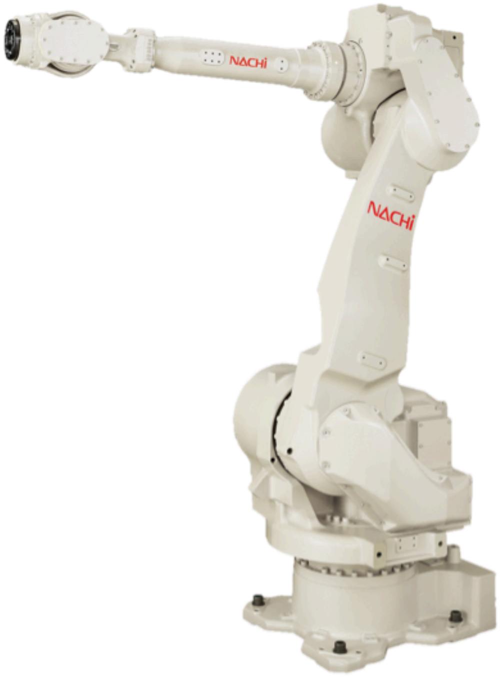


Figure 2 NACHI MC70 industrial robot

Laser distance sensor

The available laser sensor is an Omron ZS-LD80 with a ZS-HLDC controller (figure 3). The ZS-LD80 produces a spot beam with a diameter of 50 μm. The measurement centre distance is located 80 mm from the sensor head and has a measurement range of ± 15 mm. An adapter is used to mount the laser sensor to the robot flange.

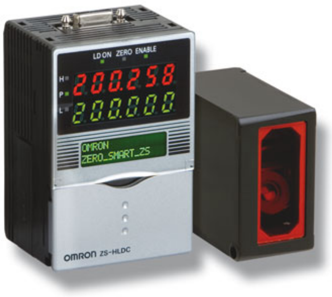


Figure 3 Omron ZS-HLDC controller and ZS-LD80 laser sensor

Calibration plane

The calibration plane (figure 4) was borrowed from the metrology lab at the university and has the dimensions 250x350 mm.

Figure 4 Calibration plane

Real-time system

The real-time (RT) system consists of two computers, both running the LabVIEW Real-Time operating system, communicating between the robot controller's CPU board and servo unit. The system allows the user to read the robot's encoder positions, and send joint movement commands to the servo board from a local computer. The communication principle is described in figure 42 and the communication protocol is described in appendix C.

Computer

The computer was a PC running the program communicating with the RT-interface;- Therefore, the PC must therefore have National Instrument's LabVIEW installed.

Total system layout

The total layout of the above mentioned hardware is schematically illustrated in figure 5. OBS! The calibration plane is not shown in the figure.

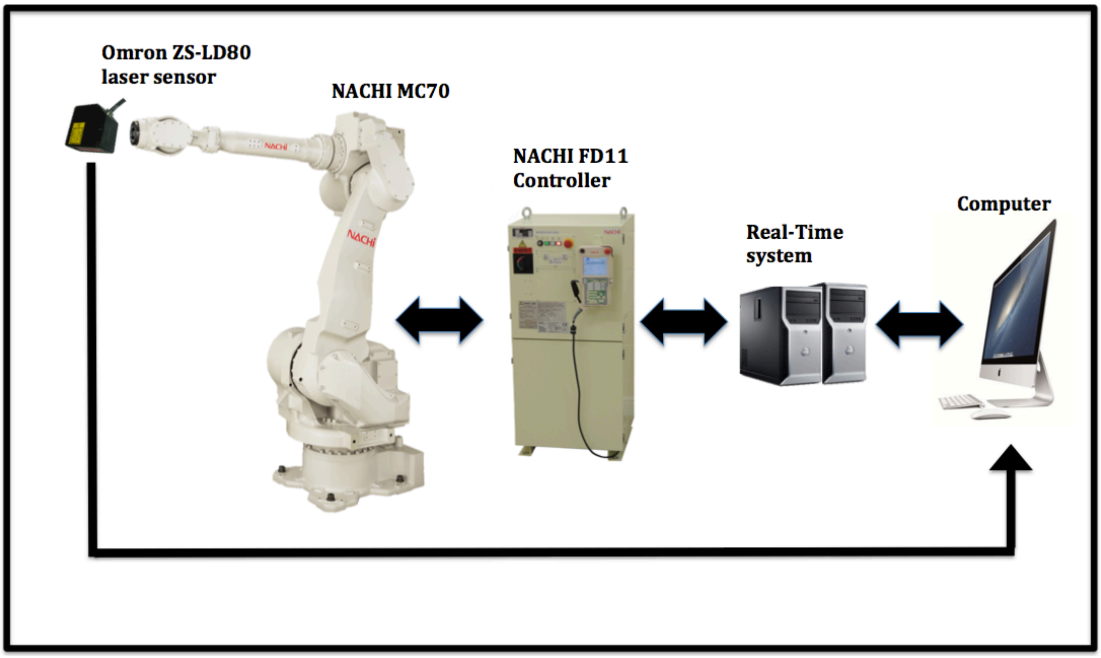


Figure 5 Schematically illustration of the total hardware layout

1.4 Available Software

There were no limitations in the available software. However, a LabVIEW framework for communicating with the FD11 controller, by PhDc Balázs Daniel, was available at the PPM lab. LabVIEW also offers a Robotics Module for kinematic calculations, based on Peter Corke's robotics toolkit for MATLAB (Peter-corke.com 2012). National Instruments LabVIEW 2013 is available free of charge to the students at NTNU. Due to this and in addition to the existing framework at PPM, LabVIEW was chosen as the main software for implementing the calibration procedure.

1.5 Limitations

The problem description was handed out January 15th and the master thesis was due June 10th, which gives a total of 105 working days.

Considering that this is a research oriented master thesis where implementation of the calibration proved to be very demanding within the time frame, it is important to note that the main focus is on the work that has been done and the knowledge that has been gained, and not the lack of final validation results. Also due to the workload of the calibration implementation research, theoretical studies are less extensive than they could have been under other circumstances; this specifically applies to accuracy assessments and different calibration procedures of industrial robots.

1.6 Approach

The master thesis is based on an experimental environment and considers a heavy-duty industrial robot that is installed in PPM AS' laboratory in Trondheim.

After conducting a study of accuracy assessment of industrial robots and the error sources affecting the positioning accuracy, the focus shifted to calibration. First a study of the existing calibration systems and previous work in the field was conducted, before starting to work with the implementation of the joint offset calibration on plane method. First an overall understanding of the calibration principle and the underlying mathematics was obtained, through studies and regularly meetings with Morten Lind. The next step was to start working on the LabVIEW implementations, with communication and control of MC70 using the real-time system and finally the implementation of the sampling application. With the initial sampling and optimization application sorted out, the next step was testing and debugging of both the applications. As the testing and debugging stage was more demanding than anticipated, no time was left for analysis and validation of the final calibration result. Instead an effort was made to give detailed recommendations for the future work of the calibration method.

1.7 Structure of the report

The report is divided into ten chapters and the contents of these are structured in the following manner:

Chapter 2 gives a brief introduction to industrial robotics, where the main focus is on describing the theory of kinematic modelling, as this is an important part of the implementation of the joint offset calibration method.
Chapter 3 presents how to assess the positioning accuracy of industrial robots, following the guidelines presented in ISO9283. It also describes different measuring methods and equipment.
Chapter 4 describes the different errors sources contributing to the total positioning accuracy of serial manipulators.
Chapter 5 gives an introduction to robot calibration and describes some of the important work and studies that have been carried out in the field.
Chapter 6 describes the mathematical principle of performing joint offset calibration on a plane and how the initial model parameters are found from the hardware setup at the PPM lab.
Chapter 7 describes how the calibration method was implemented in LabVIEW and gives a user guide of the final applications.
Chapter 8 presents the tests performed in the process of implementing the calibration method, and shows some final application tests.
Chapter 9 holds the discussion of the test, gives suggestions for future work and discusses the implemented calibration method.
Chapter 10 contains the final conclusion of the report.

Digital Attachment contains the LabVIEW applications developed during the thesis work, the sample data gathered in the experiments and a video of a sampling operation.

Chapter 2

Introduction to Industrial robotics

This chapter aims to give readers who are new or unfamiliar with the subject, a brief introduction to industrial robotics. Furthermore, key concepts in the field of modelling and control of industrial robots are presented, as it is important for the implementation of the calibration procedure later in the report.

2.1 General

Since its birth in the late 1950s, the industrial robot has had a huge impact in modern automation by improving productivity, quality and safety. The International Federation of Robotics (IFR) has crafted the following definition of an industrial robot (ISO 8373):

> *A robot is an automatically controlled, re-programmable, multi-purpose, manipulative machine with three or more re-programmable axes, which may be either fixed in place or mobile for use in industrial automation applications.*

As the definition imply, a variety of different robot configurations exist. The most important are listed below, and can be seen in figure 7:

- Articulated robot arm
- SCARA
- Cartesian manipulators
- Parallel robots

However, the most popular industrial robot is the articulated robot arm, normally consisting of 6-7 revolute joints arranged in series. Robot arms with prismatic joints also exist. The robot used in the thesis is the NACHI MC70, which represents this type of robot. The popularity of these robots comes from their high degree of dexterity, flexibility and versatility. Depending on the size of the robots, they can reach a large workspace, lifting a payload of more than 100 kg while leaving a small footprint on the shop floor.

Applications of industrial robots

In modern day industry, automation is of key to retaining and expanding manufacturing capability and winning orders. In this high demand of automation, industrial robots represent an integral part alongside CNC machines and automated control systems. Figure 7 shows the diversity of the industrial robot applications. On the other hand there is one thing most of these applications have in common, which is that they are very repeatable tasks. This is obviously due to the fact that robots are known for their high repeatable capabilities and less attention is therefore given to the robots positioning accuracy. This is not very good compared to CNC machines and other specialized machinery. However, there is a huge potential in increasing the positioning accuracy of industrial robots, as it will not only broaden their field of expertise, but also allow new tasks in fields where they are already are well established, tasks that are currently performed by humans or expensive and un-flexible specialized machinery. The next section will present two important fields of high accuracy robotics.

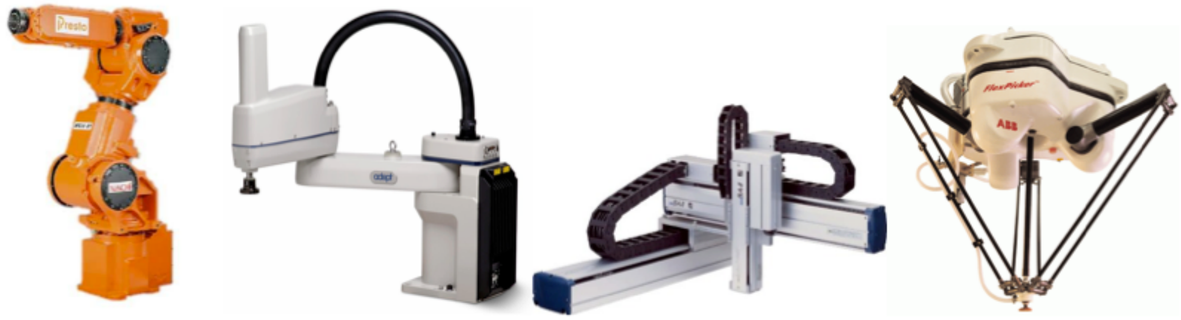


Figure 6 NACHI robot arm, Adept SCARA, Toshiba Cartesian manipulator and ABB parallel robot

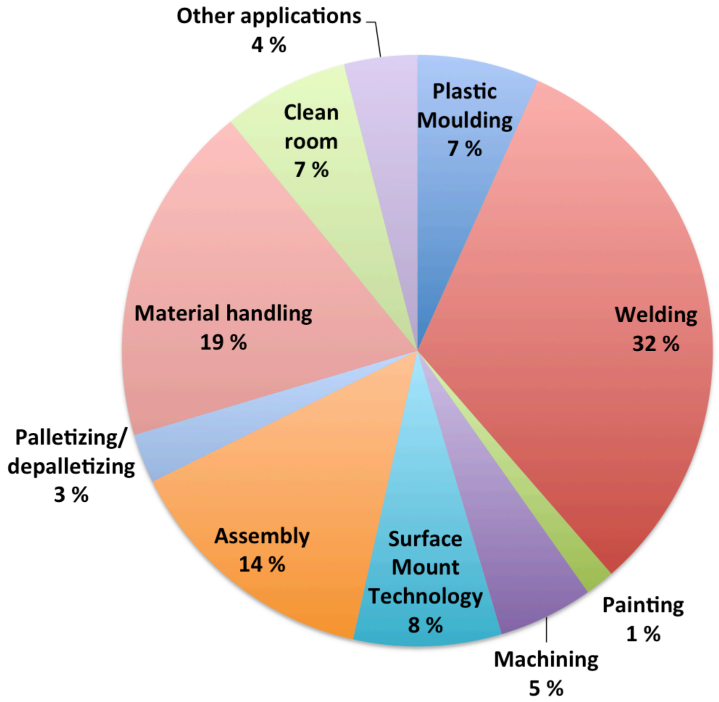


Figure 7 Applications areas of industrial robots sold in Japan 2013 (JARA, 2013)

2.1.1 High accuracy robotics

Important fields for high accuracy robotics are:

- Operations requiring/using external pose data
  - Vision based control
  - Offline programming
- Metrology

External pose data

As long as the operator manually teaches the robot its positions, so-called "teach-in-programming" accuracy is not an issue as the stored points is not defined in the Cartesian space, but as a set of joint angles, thus only relying on the robots repeatability to ensure satisfactory results. The counterpart to teach-in-programming is offline programming where robot paths and positions are generated through simulation software and downloaded to the controller. The benefit of offline programming is that applying changes to a program do not require any downtime of the robot, and generation of new paths can be performed concurrently to normal robot operation, hence improving productivity. Offline programming and simulation is an important tool in robot cell design and process optimization, but in order for the robot simulation to coincide with the actual application, accuracy is of major importance.

Another example of external pose data is the use of vision systems in robot applications. An example of vision systems can be the use of cameras to locate parts and thereby calculate the robot pose necessary to correctly grasp the object. Again robot accuracy ensures that the actual grasping position is as close to the actual calculated grasping position, in order to grasp smaller and more complex parts the demand for high accuracy increases.

Metrology

High accuracy robotics is the key factor in realising industrial robots' potential to perform metrology on manufactured parts. In the automotive and aerospace industry there is a huge potential in utilizing the flexibility of the robot to perform regular and scrupulously comparisons of large manufactured parts, such as body panels, as shown in figure 8.

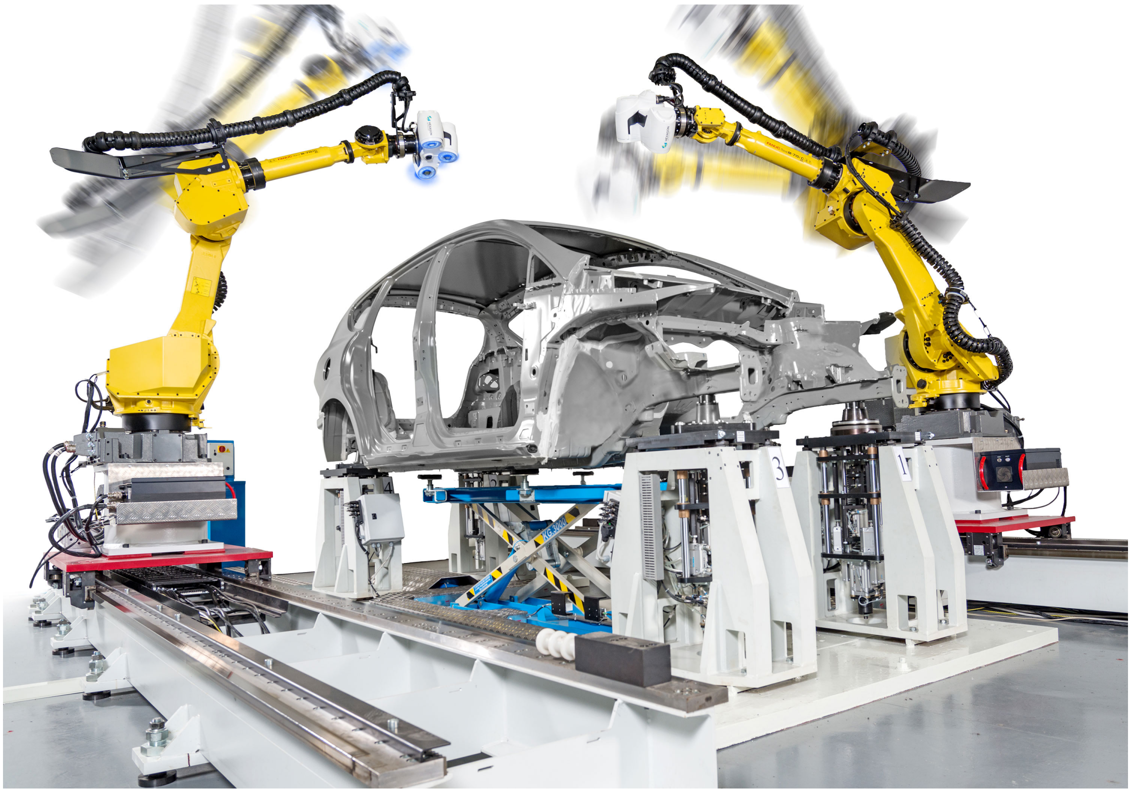


Figure 8 Industrial robots performing measurements on a car body (Courtesy: Hexagon metrology)

2.2 Kinematics of a serial manipulators

This section consists of several excerpts from previous work by the candidate in the specialization project (Johnsrud 2013).

Essentially industrial robots are motion devices, and in order to control this device it need to be mathematically described, hence the importance of the robots kinematic model. The robot's kinematic model describes the relationship between the joint angles and the *pose*[1] of the tool centre point (TCP) in the robots base coordinate frame (see figure 9). The base coordinate frame is a fixed reference frame located where the first joints axis intersects the base plate and the TCP sits in the origin of the flange/tool coordinate frame. Robots have a default TCP located at the centre of the wrist flange (last joint) (as seen in figure 9), but to accommodate for the use of multiple end-effector tools, an initialization process can be performed to position the TCP at the desired location. Good examples of TCP's are in the centre of a mechanical gripper and the tip of a welding nozzle or a cutting tool.

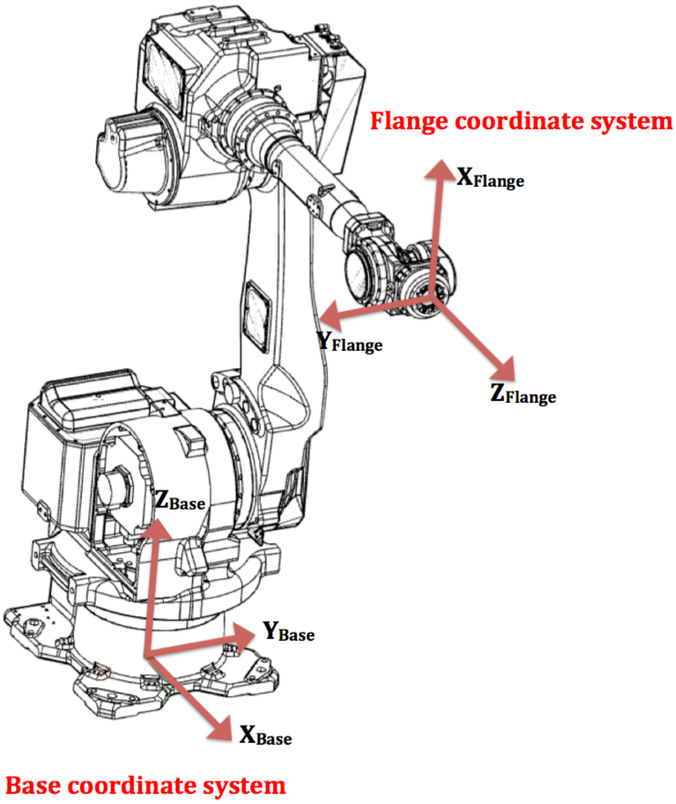


Figure 9 Flange and base coordinate system of industrial robots

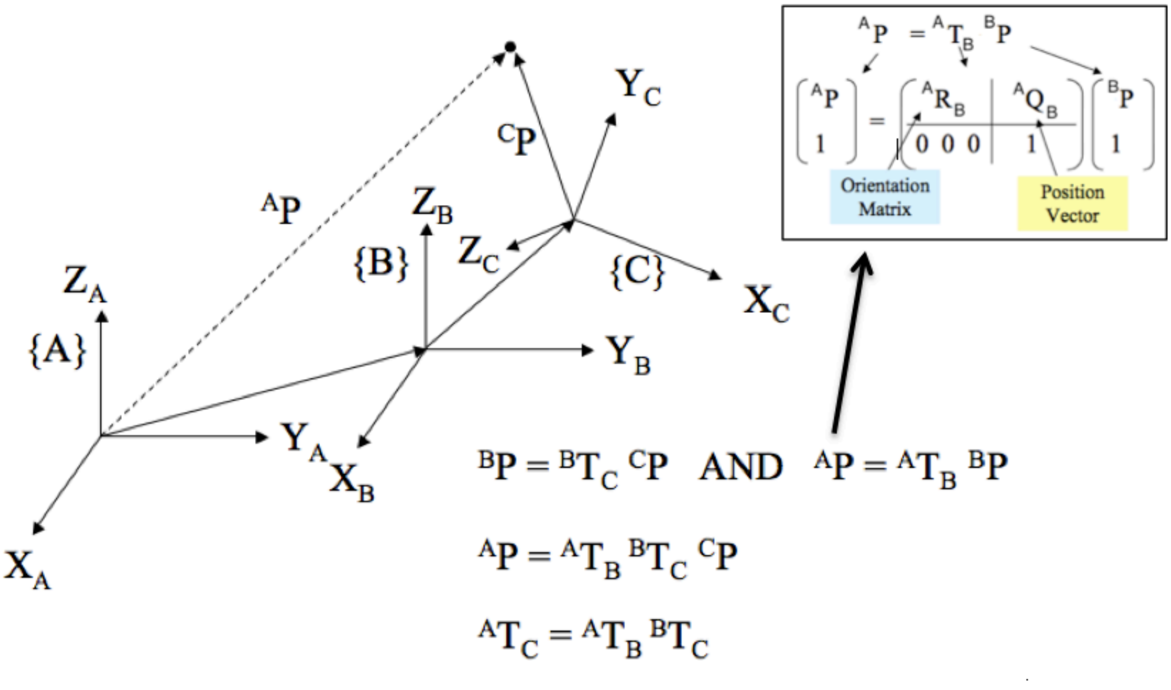
[1] *x, y, z, roll, pitch, yaw* as defined in ISO9283.

Coordinate transformations

As the above-mentioned coordinate systems have different positions and orientations, a tool is needed for describing a point in the different systems. This is achieved by using a 4x4 homogenous transformation operator matrix **T**. The operator is a combination of a set of rotational (orientation) matrices **R** describing rotation around the X,Y,Z direction of the coordinate frame, and a translational (position) vectors **Q**. This gives us a set of 6 parameters, being X,Y,Z distance and the rotation around these axis (Euler angles), hence the *pose.*

By following a set of mathematical rules, the homogenous transformation matrix can be obtained and used as the "connection" between two different coordinate frames. All we need to know is the 6 parameters that separate the two coordinate systems and the position of the point in one of the frames. Below (figure 10) is an example showing how a known Point, P, is translated between the coordinate systems A, B and C, with the transformation operator matrix, **T**. The following notation is used when mapping frames:

$$\text{Reference frame}\rightarrow {}^{A}T_{B} \leftarrow \text{Source frame}$$



${}^{A}P$
${}^{C}P$
Y_C
Z_B
Z_C
$\{B\}$
$\{C\}$
X_C
Z_A
$\{A\}$
Y_B
Y_A
X_B
X_A
${}^{A}P = {}^{A}T_{B}\,{}^{B}P$
${}^{A}P$ 1 = ${}^{A}R_{B}$ ${}^{A}Q_{B}$ 0 0 0 1 ${}^{B}P$ 1
Orientation Matrix
Position Vector
${}^{B}P = {}^{B}T_{C}\,{}^{C}P$ AND ${}^{A}P = {}^{A}T_{B}\,{}^{B}P$
${}^{A}P = {}^{A}T_{B}\,{}^{B}T_{C}\,{}^{C}P$
${}^{A}T_{C} = {}^{A}T_{B}\,{}^{B}T_{C}$

Figure 10 Representing a point in different coordinate frames (Voorthuysen 2013)

Forward and inverse kinematics

When examining a robot schematically, it can be simplified to a set of links and joints (revolute and prismatic). Therefore the TCP pose can be systematically calculated as a function of the robots links and joints parameters, with respect to its base frame (Sicliano 2009). The calculations between the base and end-effector are known as forward and inverse kinematics.

In forward kinematics the *pose* of the TCP is calculated based on the known joint angles and the link lengths of the robot arm. This method is a straightforward calculation, and is mainly used for design and simulations of robotic kinematic chains. In inverse kinematics the joint angles are calculated based on a desired TCP pose (See figure 11). Inverse kinematics are performed on the robots controller when the desired position is given in Cartesian coordinates and thus need to be converted to corresponding joint positions for the servo loop input. A typical example is when performing path-based motion the robot controller uses inverse kinematics to generate intermediate trajectory points (VIA points) between the start and end-point. This is the most computationally intensive task in the robot controller (Greenway 2000). What complicates this operation is that there usually are several solutions to a pose, and the occurrence of singularities and nonlinearities further complicates the calculations.

In order to fully constrain a rigid body in the open space, it must be given a position (*x, y, z*) and an orientation (*roll, pitch, yaw*). Thus, when performing homogenous transformations on a robot with six joints, the kinematic model will comprise of thirty-six parameters. A different, simpler and more computational effective approach to model a robot is by using the Denavit-Hartenberg convention.

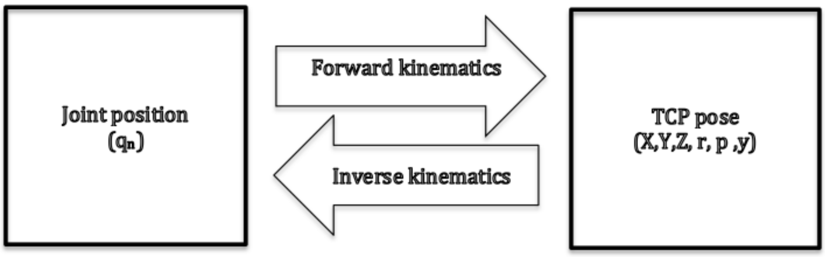


Figure 11 shows the difference between forward and inverse kinematics

2.2.1 Denavit-Hartenberg convention

The Denavit-Hartenberg (DH) is the most common method for describing the kinematics of an industrial robot, and as mentioned earlier the method only requires four parameters per link instead of six. The idea behind the method is to define the links between each joint, rather than the joints itself. This simplification also has a drawback as it introduces an error source; this is discussed in sub-chapter 5.3. The DH parameters are described in the table 1 and their geometric relationship is shown in figure 12. The rotation of each joint is always around the *z*-axis and the *x*-axis is also always normal to its current and previous z-axis. In numbering of the joints, the first is always the base/first joint and the last is always the end/tool-joint.

Table 1 describing the four DH parameters

Name	**Symbol**	**Description**
Joint angle	θ_n	The rotation along Z_n of X_{n-1} into X_n
Joint offset	d_n	The distance between X_{n-1} and X_n along Z_{n-1}
Link twist	α_n	Required rotation of Z_{n-1} about X_n to become parallel with Z_n
Link length	r_n	The distance between Z_{n-1} and Z_n along the X_n

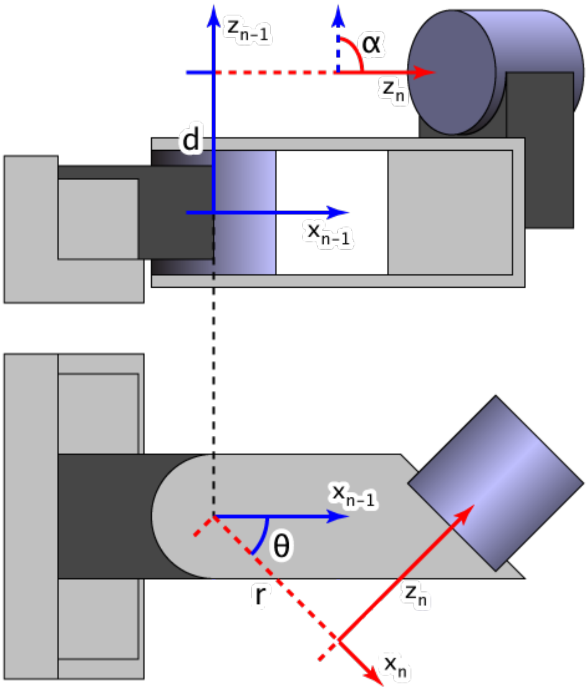


Figure 12 Geometric relationship of the DH parameters (Courtesy: Wikipedia)

When the DH parameters for an industrial robot is established, the transformation matrices for each of the parameters are established using the rotation matrices (see appendix B) for rotation about *x* and *z* together and the translational matrix for *x* and z direction, as follows:

$$\mathbf{T}_{d_n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

$$\mathbf{T}_{\theta_n} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n & 0 & 0 \\ \sin\theta_n & \cos\theta_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$$\mathbf{T}_{r_n} = \begin{bmatrix} 1 & 0 & 0 & r_n \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

$$\mathbf{T}_{\alpha_n} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_n & -\sin\alpha_n & 0 \\ 0 & \sin\alpha_n & \cos\alpha_n & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

The product of multiplying these four transformation matrices produces a new transformation matrix, ${}^{n-1}\mathbf{T}_n$, better known as the D-H transformation matrix for adjacent coordinate frames (Gonzalez et Al. 1987):

$${}^{n-1}\mathbf{T}_n = \mathbf{T}_{d_n}\mathbf{T}_{\theta_n}\mathbf{T}_{r_n}\mathbf{T}_{\alpha_n} \tag{2.5}$$

$${}^{n-1}\mathbf{T}_n = \begin{bmatrix} \cos\theta_n & -\cos\alpha_n \sin\theta_n & \sin\alpha_n \sin\theta_n & r_n \cos\theta_n \\ \sin\theta_n & \cos\alpha_n \cos\theta_n & -\sin\alpha_n \cos\theta_n & r_n \sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

As for an industrial robot with *n*- links, the total transformation matrix will be given as:

$${}^{0}\mathbf{T}_n = {}^{0}\mathbf{T}_1\, {}^{1}\mathbf{T}_2\, {}^{2}\mathbf{T}_3 \ldots {}^{n-1}\mathbf{T}_n \tag{2.7}$$

In section 6.4.1 the DH convention is applied to the NACHI MC70.

Chapter 3

Accuracy assessment of industrial robots

The following chapter presents the ISO definition of the pose accuracy criteria for industrial robots and how it should be assessed. The definitions are obtained from ISO 9283:1998. Finally, selected test equipment and metrology methods recommended in accordance to ISO 9283:1998 is described, the methods are obtained from ISO/TR 13309:1995. Firstly, an overview of ISO 9283 and ISO/TR 13309:1995 is given in the following sub-chapter.

3.1 ISO standards

ISO is an abbreviation for The International Organization for Standardization, this world wide organization is developing international standards that give state of the art specification for products, services and good practice, helping to make industry more efficient and effective (ISO no date).

3.1.1 Importance of ISO standards in robotics

When looking at datasheets of new robot models, numerous of specifications are listed such as speeds, accelerations, loads and repeatability. The problem, especially in the case of repeatability is that the customer has no way of knowing how the repeatability has been assessed and can therefor not accurately compare this to competing robot manufacturers. Large automotive and aerospace companies usually solve this by buying one of each robot brand and performing own tests (Summers 2005). However this is not feasible for small to medium sized companies.

One robot manufacturer that has adapted the ISO 9283 to assess its repeatability is KUKA, NACHI on the other hand follows the Japanese Industrial Standard (JIS), to assess their repeatability and is therefore not correctly comparable to KUKA.

Another problem is that the robot manufacturers do not disclose the accuracy of their robots, and there are no tendencies in the industry that this will change in the nearest future. One of the reasons is obviously that the main operations of robots are of the highly repeatable sort (figure 7), therefore the main focus of the robot manufacturers. However as the demand of high accuracy robots are increasing, the robot manufacturers will in the coming future have to focus on higher accuracy and when that time comes ISO standard will become an important tool for assessing the accuracy.

3.1.2 ISO 9283

ISO 9283: *Manipulating industrial robots – Performance and related test methods*, is a part of a series of standards covering the manipulation of industrial robots. The first edition of ISO 9283 was published in 1990, amended in 1991 and replaced by the current edition, in 1998. The standard describes how the performance characteristics of the industrial robot shall be specified and how they should be tested. An example of how the test results should be reported is also included. ISO9283 defines the following performance criteria's:

1. Pose accuracy and pose repeatability
2. Multi-directional pose accuracy variation
3. Distance accuracy and repeatability
4. Position and stabilization time
5. Position overshoot
6. Drift of pose characteristics
7. Exchangeability
8. Path accuracy and repeatability
9. Path accuracy on re-orientation
10. Cornering deviations
11. Path velocity accuracy
12. Path velocity fluctuation
13. Minimum posing time
14. Static compliance
15. Weaving deviations

Only the Pose accuracy is studied in this thesis.

3.1.3 ISO/TR 13309

ISO/TR13309: *Manipulating industrial robots – Informative guide on test equipment and metrology methods for robot performance evaluation in accordance with ISO 9283,* is a document published in the form of a Technical report (hence: ISO/TR) for the sole purpose of providing an overview of the current state-of-the-art test equipment and metrology methods for evaluating the performance of industrial robots in accordance with ISO 9283. ISO/TR 13309 was published in 1995.

This technical report classifies 8 categories of test methods, from these 16 individual methods are presented. The following list presents the various categories:

1. Positioning test probes methods
2. Path comparison methods
3. Trilateration methods
4. Polar coordinates methods
5. Triangulation methods
6. Inertial measuring methods
7. Coordinate measuring methods
8. Path drawing method

3.2 Pose accuracy

The pose accuracy expresses the deviation between a commanded pose and the mean of the attained pose when approaching the command pose from the same direction several times. ISO9283 divides pose accuracy into:

a) Positioning accuracy
b) Orientation accuracy

3.2.1 Positioning accuracy

The positioning accuracy is the difference between a commanded position and the barycentre of the attained positions. Where the barycentre is the coordinates mean values $\bar{x}$, $\bar{y}$ and $\bar{z}$ calculated from the attained positions by formula 3.2, after *n* number of repetitions.

Positioning accuracy is calculated from the following formula

$$AP_P = \sqrt{(\bar{x} - x_c)^2 + (\bar{y} - y_c)^2 + (\bar{z} - z_c)^2} \qquad (3.1)$$

This is illustrated in figure 13, where

$AP_x = (\bar{x} - x_c),\ \ AP_y = (\bar{y} - y_c),\ \ AP_z = (\bar{z} - z_c),$

and

$$\bar{x} = \frac{1}{n}\sum_{j-1}^{n} x_j, \quad \bar{y} = \frac{1}{n}\sum_{j-1}^{n} y_j, \quad \bar{z} = \frac{1}{n}\sum_{j-1}^{n} z_j \qquad (3.2)$$

where

$\bar{x}, \bar{y}, \bar{z}$ – is the coordinates of the barycentre.

x_j, y_j, z_j – is the attained coordinates of each repetition.

x_c, y_c, z_c – is the coordinates of the programmed point.

n – is the number of repetitions

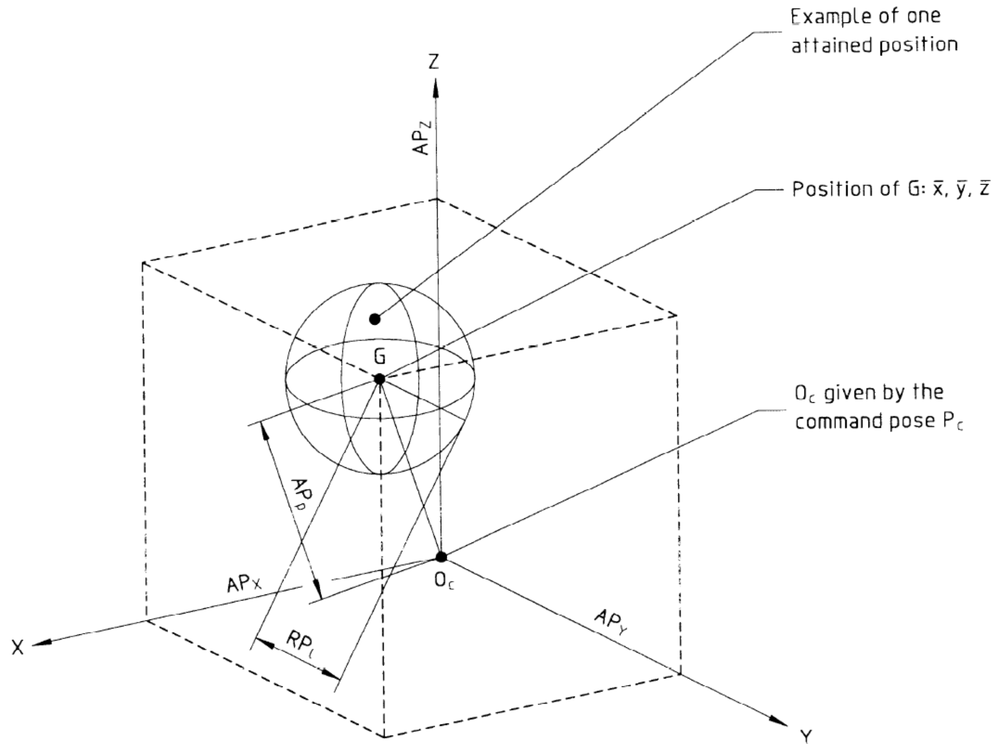

Example of one attained position
Z
AP_Z
Position of G: x̄, ȳ, z̄
G
O_c given by the command pose P_c
AP_P
AP_X
O_c
AP_Y
X
RP_l
Y

Figure 13 Positioning accuracy (Courtesy: ISO 9283)

3.2.2 Orientation accuracy

The orientation accuracy is the difference between the orientation of the commanded (programmed) pose and the average of the attained positions, illustrated in figure 14.

$$AP_a = (\bar{a} - a_c)$$
$$AP_b = (\bar{b} - b_c) \qquad (3.3)$$
$$AP_c = (\bar{c} - c_c)$$

Where

$$\bar{a} = \frac{1}{n}\sum_{j-1}^{n} a_j, \;\; \bar{b} = \frac{1}{n}\sum_{j-1}^{n} b_j, \;\; \bar{c} = \frac{1}{n}\sum_{j-1}^{n} c_j$$

and

a_j, b_j, c_j – is the angles of each attained pose.

a_c, b_c, c_c – is the angles of the programmed pose.

n – is the number of repetitions

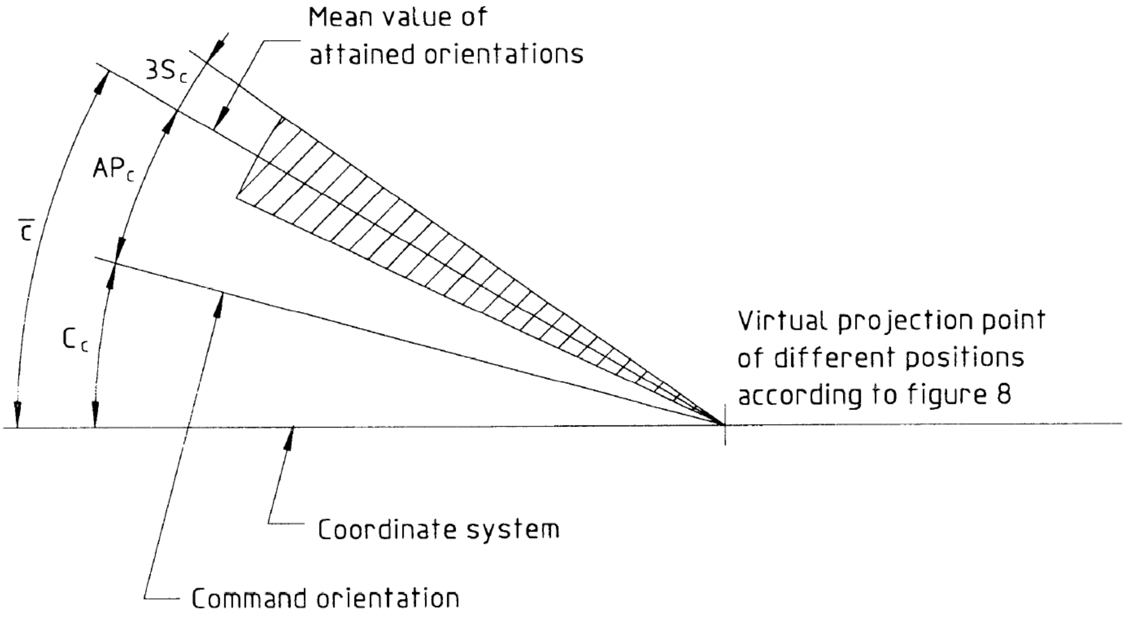

Mean value of attained orientations
3S_c
AP_c
c̄
C_c
Virtual projection point of different positions according to figure 8
Coordinate system
Command orientation

Figure 14 Orientation accuracy (Courtesy: ISO 9283)

3.3 Test procedure

Upon performing the assessment of the pose accuracy, ISO 9283 lists a set of conditions that need to be fulfilled for the test result to be valid. They include operating and environment condition such as a specified warm-up operation and ambient temperature in the testing area etc. All the conditions of the test procedure need to be specified in the final test report.

When assessing pose characteristics five suitable points must be located in a plane, located inside a cube within the robots workspace (see figure 15). The location of the cube is based on the workspace area that the robot will utilize the most. The cube shall also have the maximum possible volume and have edges parallel to the base coordinate system of the robot. The five measurement points shall lie on the diagonals of the measuring plane and shall represent the position of the robot flange in either base (preferred) or joint coordinates. It is important that the 5 poses are chosen so all joints are moved when moving between poses.

The number of cycles required when assessing pose characteristics is 30. The test needs to be executed with its maximum load and maximum speed as defined by the manufacturer. Additional test can be performed with rated velocities and loads, but this is optional.

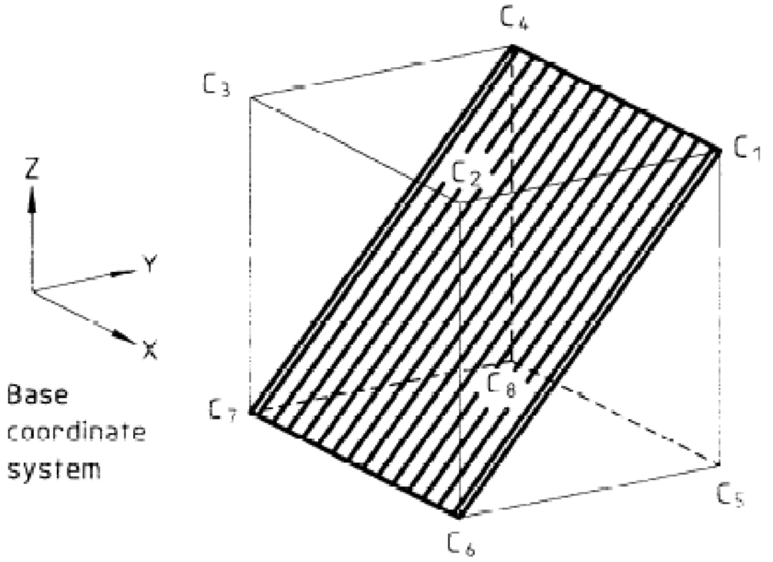


Figure 15 Example of test cube (Courtesy: ISO 9283)

3.4 Test methods

The following section describes some of the, main test equipment and metrology methods recommended in accordance to ISO 9283:1998. The methods are obtained from ISO/TR 13309:1995.

3.4.1 Positioning test probe methods

The robot is fitted with an end-effector with sufficient number of proximity or displacement sensors and a precision artefact (cube or sphere) are positioned in the robots working space. The robot is then programmed to slowly approach the artefact and touch or stop at a certain distance (depending on the sensor, contact or non-contact).

Different types of test artefacts can be combined depending on the number of pose parameters to be measured. (see figure 16). The benefit of non-contact measurements is that it allows you the measure the complete pose of the robot, but it is also relatively expensive compared to contact measurements probes. Due to its simplicity this method is widespread, but when many measurements is to be taken, the method becomes tedious and cumbersome (Brussel 1990). Other drawback for the method is that path characteristics cannot be measured and when performing pose accuracy measurements, only relative accuracy is measured. In order to measure absolute accuracy the pose of the artefact need to be determined by independently means, which will extensively complicate the method.

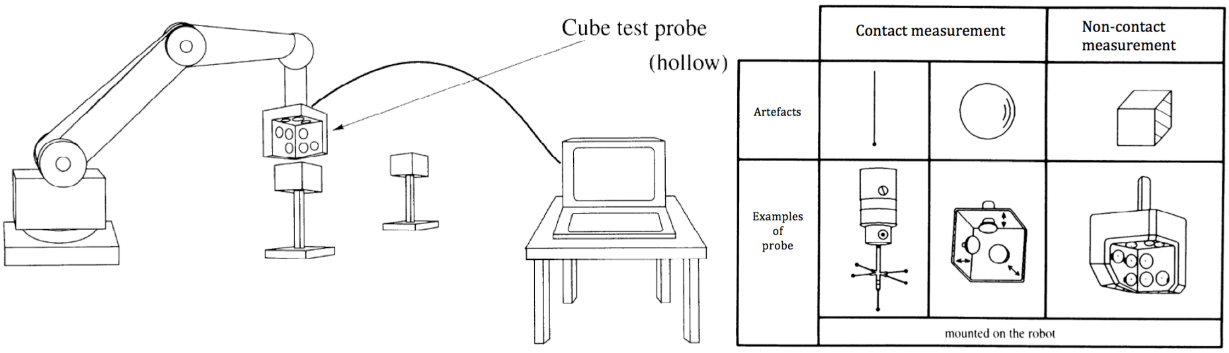


Figure 16 Positioning test probe method (Courtesy: ISO/TR 13309)

3.4.2 Trilateration methods

Trilateration is a method for determining the absolute or relative location of a point in three-dimensional space, by using three distance measures between a point *P* and three observation points (see figure 17). In order to achieve absolute location of *P,* the absolute positions of the observation points must be known. ISO/TR 13309 suggest three different equipment for distance measurement; Laser interferometry, ultrasonic sensors and mechanical cables.

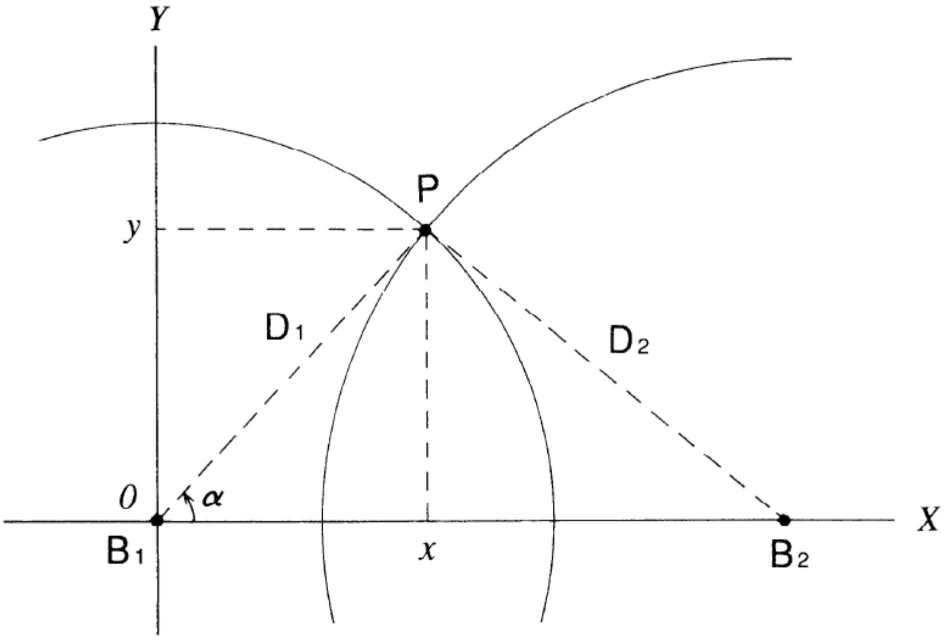


Figure 17 Trilateration principle in two dimensions (Courtesy: ISO/TR 13309)

Multi laser tracking interferometry

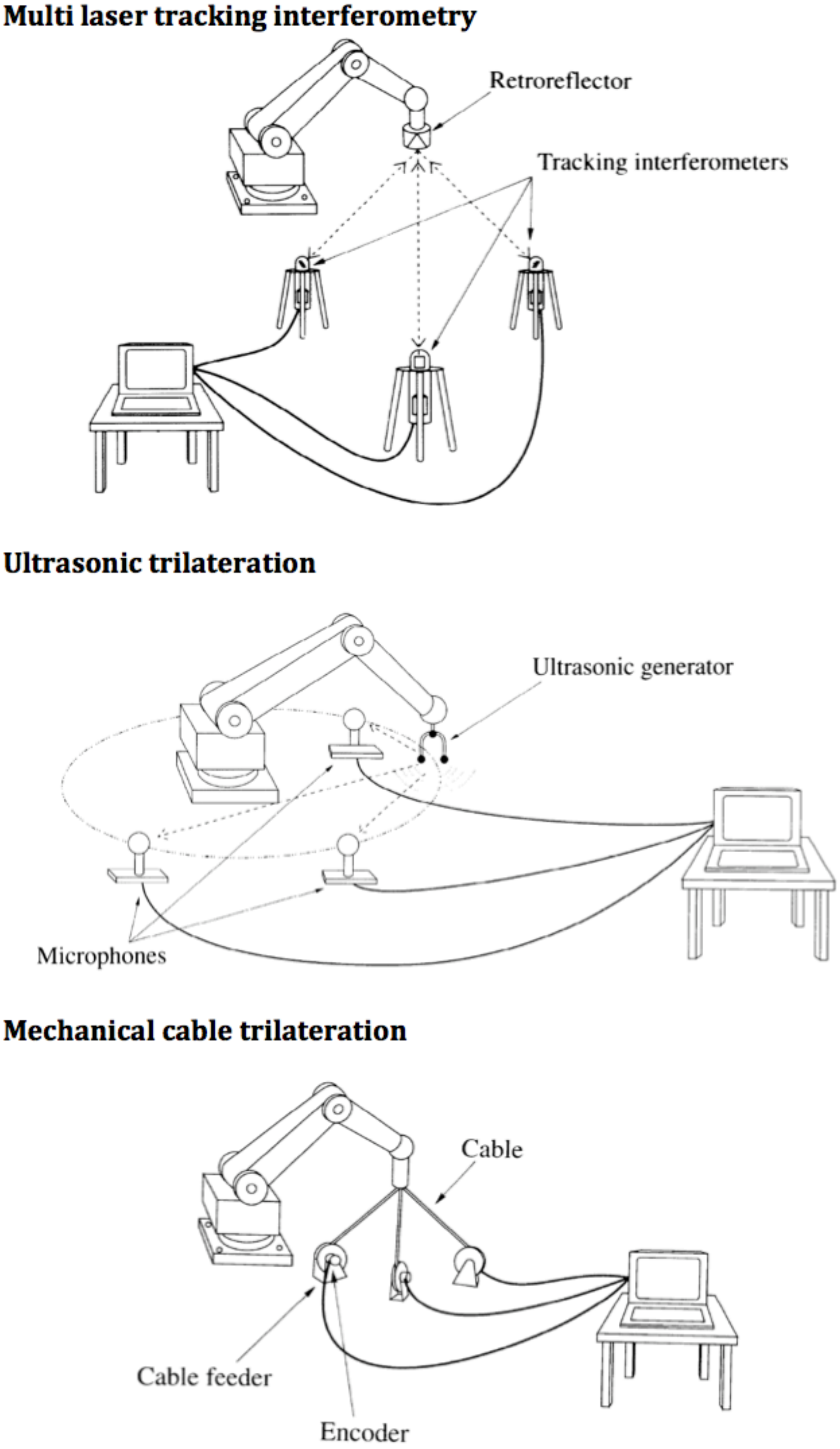
This method consist of three laser interferometers, each sitting on a two-axis servomotor allowing the laser to track a target located on the flange of the robot (see figure 18). This method however can only measure the position of the target. If the orientation is to be tested six laser interferometers are needed, with two beams aimed at three independent targets on the robots flange. The accuracy when using laser interferometry is very high, the disadvantage is the price of the equipment, especially when six is required, as of this laser interferometers are less common in trilateration methods.

Ultrasonic trilateration

This method works by attaching an ultrasonic sound source on the robot flange and by having three ultrasonic sound sensors receiving the pulse train from the sound source the robot end-effector location can be calculated. In order to measure the robot orientation, three independent sound sources must be mounted on the robot flange and the microphones need to be able to distinguish between the sound sources (see figure 18). Compared to the laser equipment, this is much more affordable, but also less accurate.

Mechanical cable trilateration

The final method is based on attaching three cables to a specified end point on the robot flange (see figure 18). The cables are originated in three separate cable feeders each equipped with an encoder or potentiometer that is able to measure the length of each cable under tension and from this the position of the flange is can be computed. The main disadvantage with this method is that it cannot measure the pose orientation.



Multi laser tracking interferometry

Ultrasonic trilateration

Mechanical cable trilateration

Figure 18 Trilateration measurement methods (Courtesy: ISO/TR 13309)

3.4.3 Polar coordinate methods

In this measurement techniques the Cartesian coordinates are determined by measuring the distance *D,* azimuth α and elevation β, between the measurement point and a point on the robot as shown in figure 19.

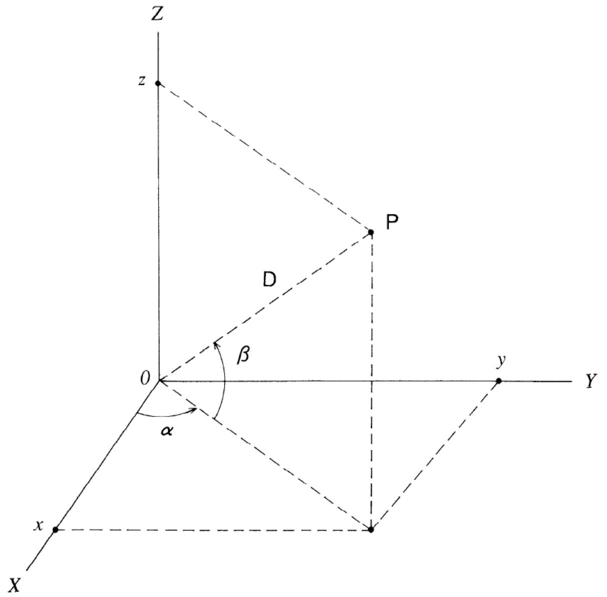


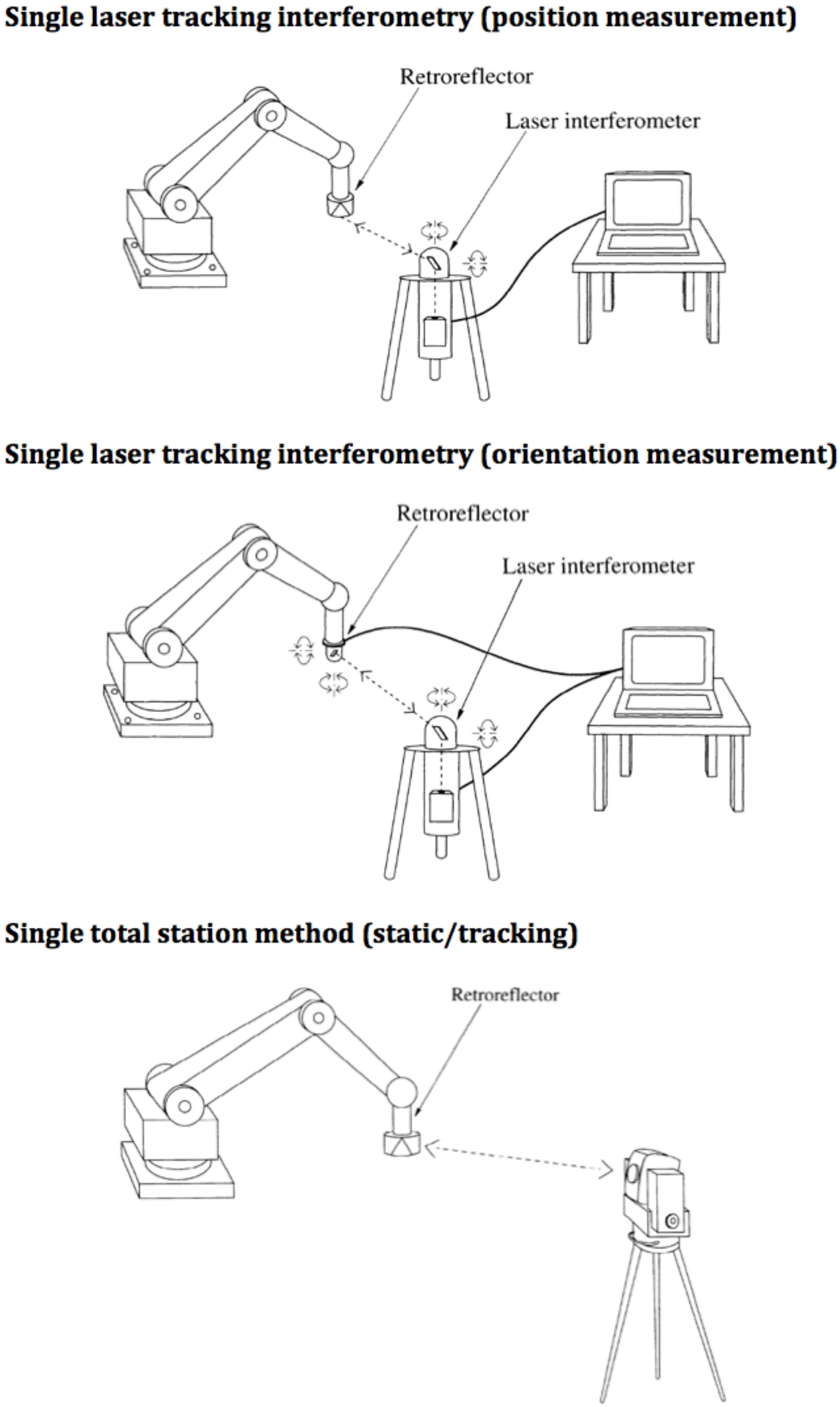
Figure 19 Polar coordinate principle (Courtesy: ISO/TR 13309)

Single laser tracking interferometry

The single laser tracking interferometry method can be used to measure either the position of the robot or the orientation. To measure the position of the robot, the end-effector is fitted with a retro-reflector mirror. By measuring the distance from the laser to the mirror, and reading the azimuth/elevation of the tracking system, the position is obtained (see figure 20). To measure the orientation, the retro-reflector mirror needs the capability to keep its optical axis pointed to the stationary tracking device or by having a tracking device able to analyse the diffracted images reflected by the mirror.

Single total station method (static/tracking)

This system, commonly referred to as a laser tracker, is able to calculate the robots attained pose by measuring the distance, azimuth and elevation point by point (see figure 20). The single total station method can also measure attained path as well as attained position of the robot by mounting a moving retro-reflector on the robots end effector. The laser tracker is the most common metrology equipment for accuracy assessment and calibration of industrial robots (Slamani et Al. 2012), on the other hand they are also very expensive.



Single laser tracking interferometry (position measurement)

Single laser tracking interferometry (orientation measurement)

Single total station method (static/tracking)

Figure 20 Polar coordinate measurement methods (Courtesy: ISO/TR 13309)

3.4.4 Triangulation methods

Another method to define the location of a point, *P,* is by having two or more observation points with a known distance in-between that measures the azimuth and/or elevation from the observation point to *P* (see figure 21). In order to measure absolute position the position of the observation points need to be known.

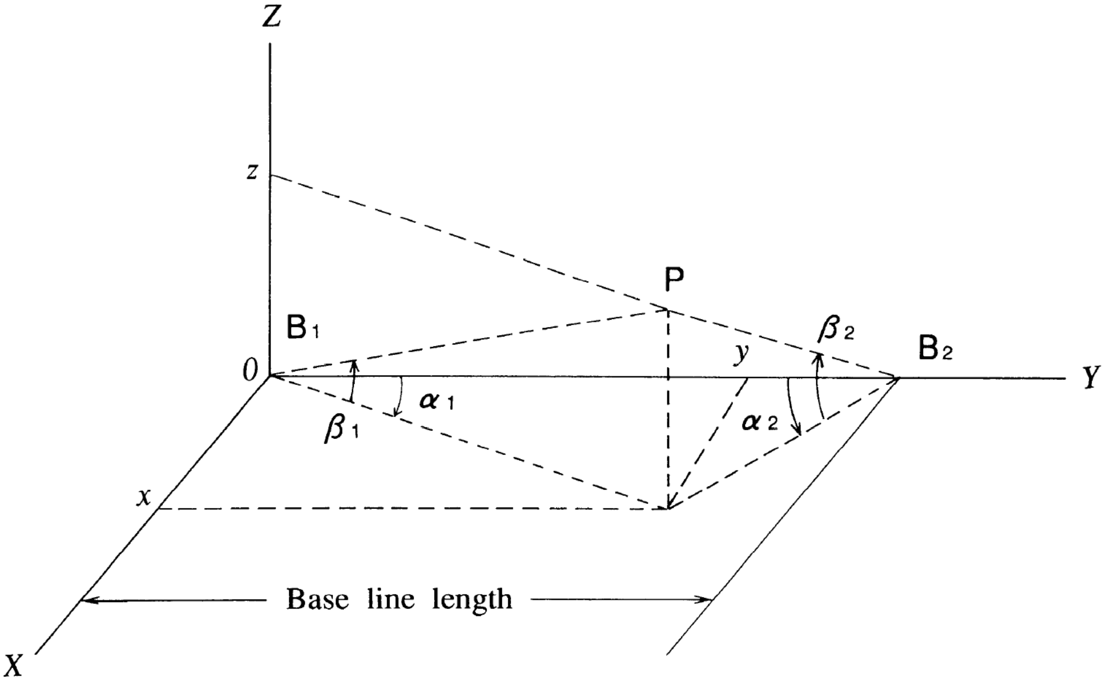


Figure 21 Triangulation principle (ISO/TR 13309)

Optical tracking triangulation methods

In optical tracking the robot position is obtained as a function of two sets of azimuth/elevation data. All the systems described below are mounted on a two-axis tracking system, allowing for both static and dynamic measurements.

Laser tracking triangulation system

Position is measured from two tracking laser beams continuously following a reflector mounted on the robot flange.

Laser scanning triangulation system

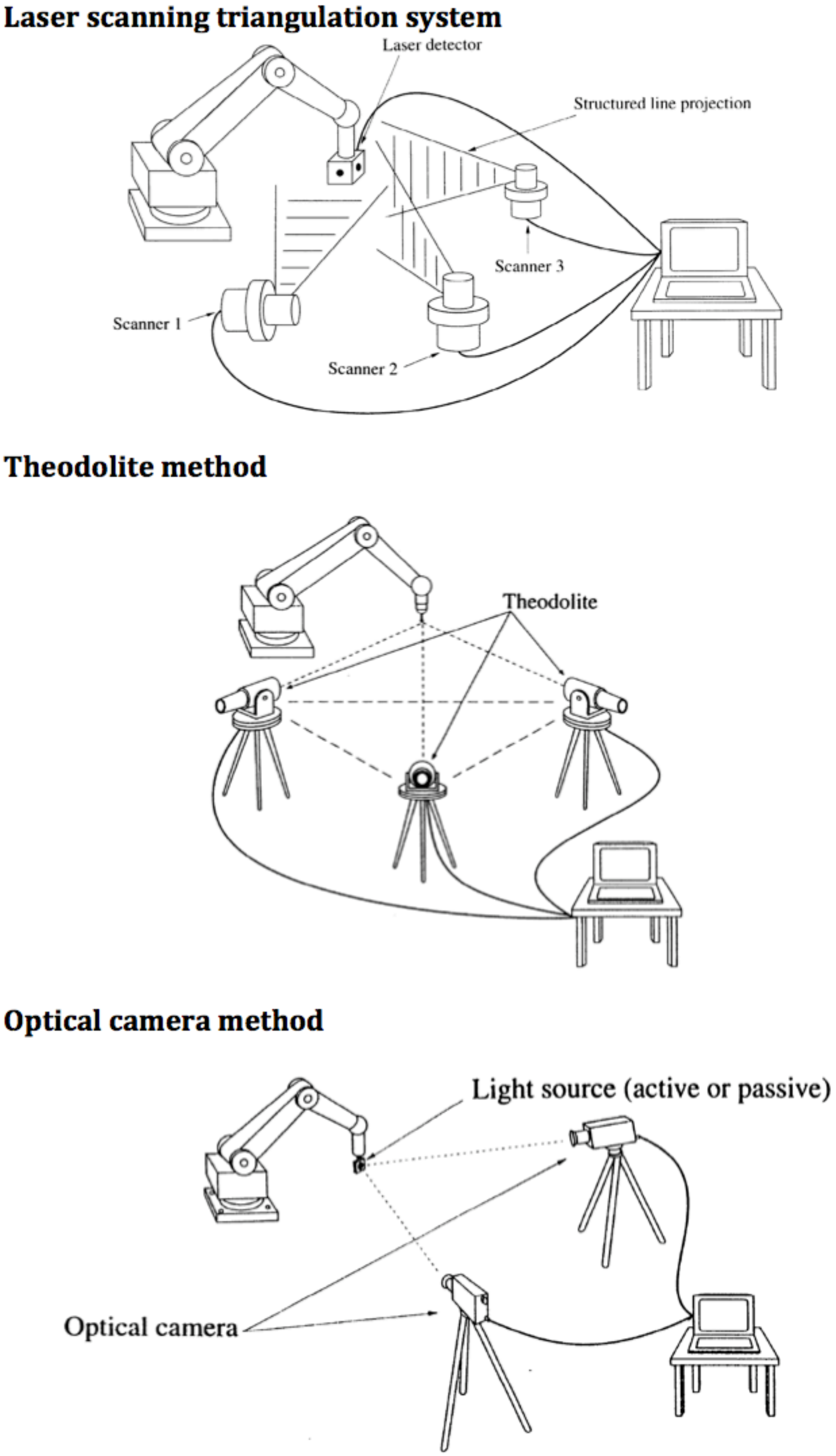
In this setup, three laser scanners detect incident light on the robot flange. Each scanner is emitting a line-projected light where two of the scanner emits a vertical line and the final scanner emits a horizontal line (see figure 22).

Theodolite method

When the research on robotic accuracy began, the most common measurement tool was theodolites. By using two or more of these devices the operator can aim the theodolite on a target point mounted on the robot, by knowing the distance between the theodolites and reading the elevation on the theodolite the position of the robot target point can be calculated (see figure 22). If the robot has multiple target point the orientation can be calculated. Theodolites are manually operated, thus they can only be used for static measurements and as for all manual measurement methods. As of this, performing several measurements is time demanding and tedious and therefore not common in modern applications.

Optical camera method

With optical cameras the position of the robot can be measured as a function of time by images from two cameras. The cameras need to have a known fixed location and the robot end point will be equipped with a light source, either active or passive (see figure 22). The cameras are used to monitor the position of the light source in the cameras coordinate system. By combining the information from the two cameras and the known distance between the cameras, the position of the robot mounted light source can be calculated. By mounting several light sources on the robot, the orientation accuracy can be calculated. This method can be used for both static and dynamic measurements. When performing dynamic measurements, the data processing speed need to be taken into account and the robot speed should be adjusted accordingly.



Laser scanning triangulation system

Theodolite method

Optical camera method

Figure 22 Triangulation measurement methods (Courtesy: ISO/TR 13309)

3.4.5 Cartesian coordinate measuring methods

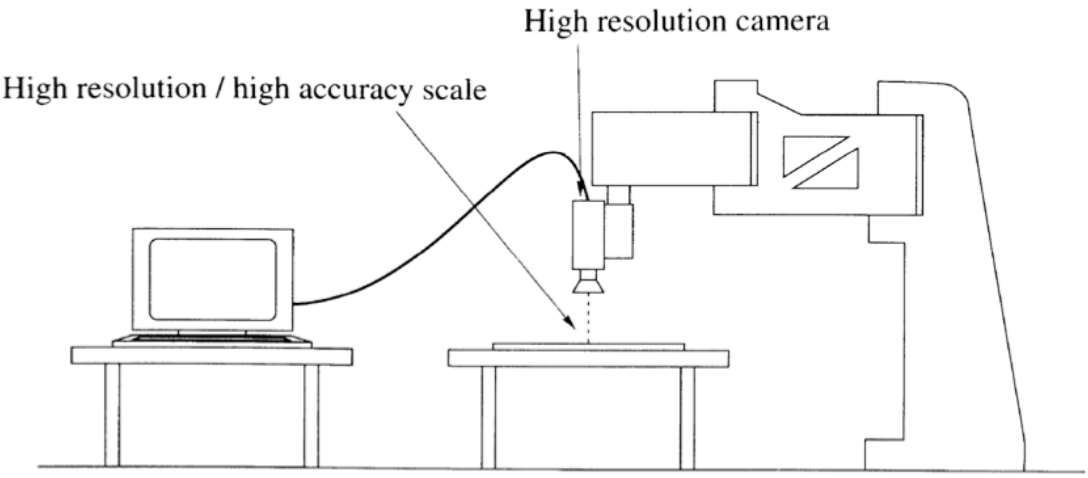
Two-dimensional digitizing method

I this method the robots planar motion can be measured in two dimensions (Y-X, Y-Z or X-Z), by a high-resolution camera and high accuracy-scale (see figure 23). The scale consists of a precision scale line that the camera is counting. This is a simple and normally not too expensive method, however this can only be used to measure relative accuracy in a limited workspace area.

Coordinate measuring machine method

The position of the robots end-effector can be measured by getting its coordinate values from a coordinate measuring machine (CMM). By attaching a cubical artefact to the end-effector and making the CMM measure three point on the artefact, the orientation can be determined (see figure 23). This is one of the most accurate measuring methods; however industrial robots are rarely situated next to a CMM as they are situated in a metrology lab and not on the factory floor. For small robots it is however, possible to bring down to a metrology lab, to perform the test before moving it back to factory floor.

Two-dimensional digitizing method



Coordinate measuring machine method

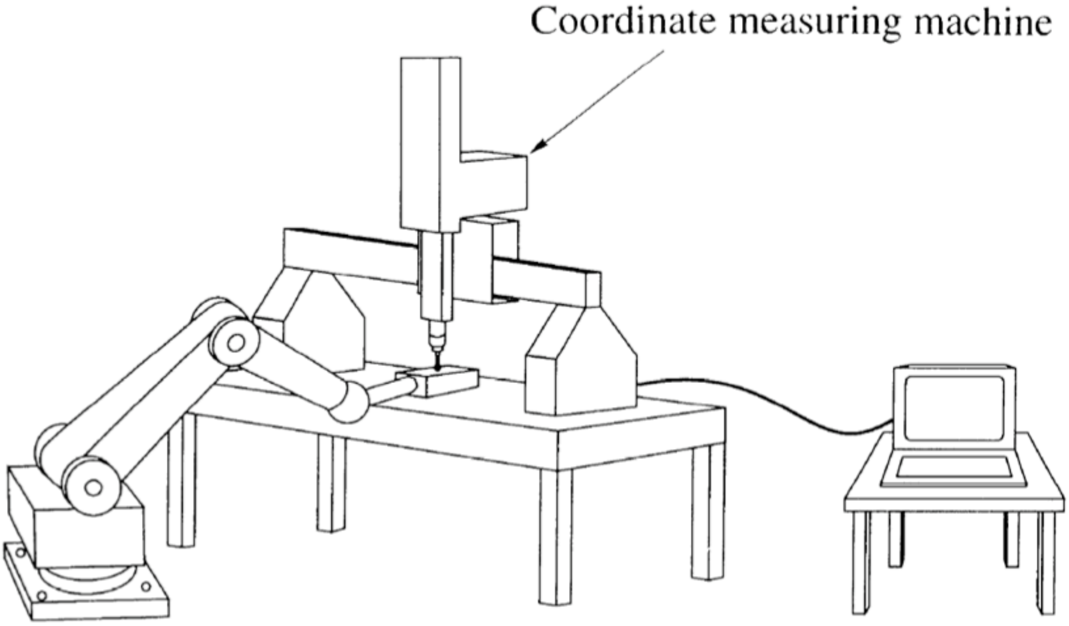


Figure 23 Cartesian coordinate measuring methods (Courtesy: ISO/TR 13309)

Chapter 4

Error sources

4.1 General

In order to improve the positioning accuracy of industrial robots, the sources contributing to creating the deviation between actual position and commanded position need to be identified. As this chapter shows, there exist many different contributors to the total TCP positioning error, all with varying influence. The errors sources in the kinematic chain of industrial robot have been studied the last three decades, and several ways of identifying the errors has been used. In a paper from 1985, Ackerson and Harry grouped the errors into four categories: Digitization error, calibration error, deterministic error and stochastic error. In this chapter the errors are grouped into two main categories, Geometric and non-geometric errors, each with several sub categories (See figure 24), as this categorization appears to be the most widespread (Renders et al. 1991, Chen and Chao 1987 and Wang et al. 1984).

An important contributor to this field is Judd and Knasinski, who in their paper from 1990 present their results from testing various error sources independently. By doing so, Judd and Knasinski were able to identify the relative magnitude of each error.

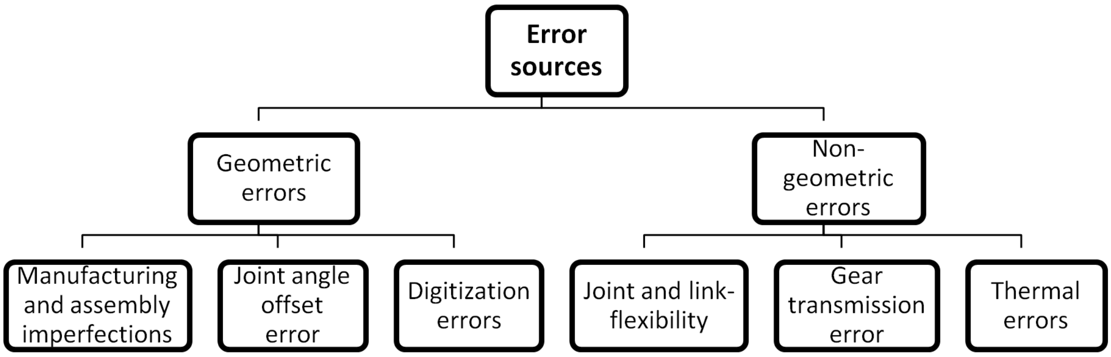


Figure 24 Categorization of error sources

4.2 Geometric errors

Geometric errors are considered to have the greatest contribution to the robots positioning accuracy. These errors are also the simplest to model, identify and correct (Judd and Knasinski 1990).

4.2.1 Manufacturing and assembly imperfections

The kinematic model of the robot is based on robot components manufactured and assemblies with zero tolerances, assuming accurate link lengths, perpendicularity and perfect bearings through the entire robot structure. However it is well known that some kind of tolerance exist either it is in the ± 1 mm range or $\pm 0{,}001$ mm. As the robot in question is a serial manipulator, the manufacturing error in each link will accumulate from the base to the end effector, thus the accumulated error will vary with the different poses of the robot. Judd and Knasinski claimed that the error due to link variations is 5% of the total root mean square error (RMS)

One approach to eliminate this error is to tighten the tolerances specified in the design drawings, the problem with this approach that it will significantly increase the expense of the manufacturing process. A different approach, which is discussed further in sub-chapter 5.3, is to analyse each individual robot, in order to create a more accurate kinematic model for each individual robot.

4.2.2 Joint angle offset error

The Joint angle offsets, also known as the Calibration error (Ackerson and Harry 1985), is the difference between the actual joint position and the zero reference position when the robot is programmed to be in the zero reference position. This will serve as a constant error for each significant joint and the error will manifest throughout the kinematic chain of the robot and thereby vary the end-effector position error wildly throughout robot's working space. As an example, the positioning error due to a joint angle offset in the base joint (q_1) will increase as the distance between the TCP and base increases. Figure 25 shows the end effector error as a result of joint angle offsets.

The main contribution to this error is from the accuracy of the previous calibration method, robot kinematic structure, pose and loss of encoder bits. Situations that causes loss of encoder bits is collisions or an encoder bit jump may occur if the robot is programmed to move very fast between positions, this is especially a risk in remote robot operations. Judd and Knasinski paper from 1990 claims that the joint angle offsets is responsible for 90% of the total RMS error, making the joint angle offset error the biggest contributor to the positioning accuracy. As of this, the calibration method described and implemented in chapter 6 and 7, focuses on reducing/eliminating this error.

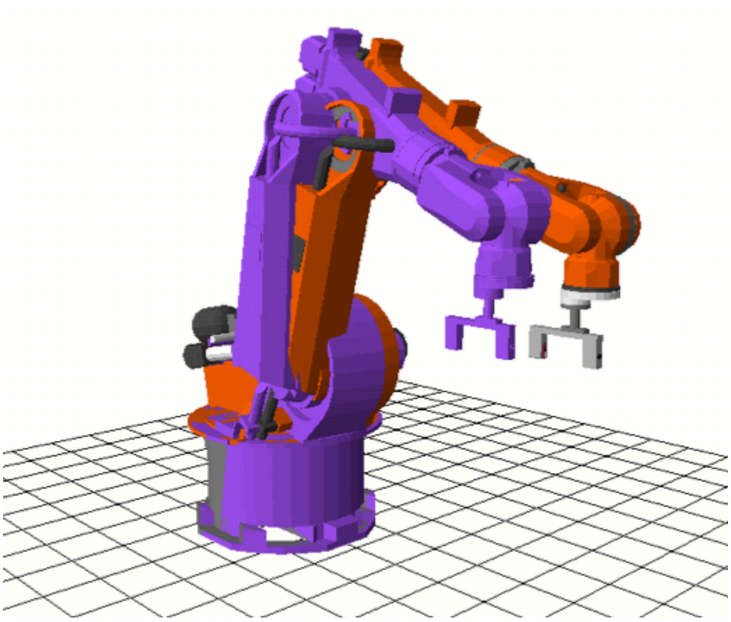


Figure 25 Example of joint offset error (Courtecy: Techonsult)

4.2.3 Digitization error

Digitization errors can be grouped into the three following categories:

Encoder resolution

Digital errors due to the resolution of the encoders, as the servo link only can position itself in a discrete position to correspond to a given position.

Controller round-off error

Digitization error may be caused by digital round-off errors, especially if a robot controller uses scaled integers to represent the angular and Cartesian coordinates. Modern controllers, such as the NACHI FD11 controller uses a floating-point representation instead, in these cases round-off error will be insignificant. (Judd and Knasinski 1990).

Steady-state control error

Steady state error can affect the accuracy and repeatability of the errors. Tests results reported in the Judd and Knasinski paper from 1990, revealed that the steady-state error never exceeded more than two encoder pulses (0.002-0.007 degrees, depending on the resolution of the encoder), regardless of the loading.

4.3 Non-geometric errors

All non-geometric errors are responsible for about 10% of the total positioning error, and that the error is induced by: Joint flexibility, link flexibility, gear transmission error, backlash in gear transmission, and temperature effect (Renders et al. 1991). However, non-geometric errors play a substantial role when high accuracy, (e.g., 0.1 mm) is desired (Gong et Al. 2000).

4.3.1 Joint and link- flexibility

Experiments by Gong et al. in 2000, show that the compliance error due to joint deflection is much greater than errors due to link deflection. The joint deflection arise from two situations:

- Deformations due to high load
- Deformations due to links supporting their own weight

The tests performed by Judd and Knasinski in 1990 revealed a considerate deformation in some of the joints when the robot was fully stretched with a high load. With no load the deformation errors were negligible. It is important to note that structural deformation varies between the different robot designs. These errors can obviously be avoided by performing heavy duty, high accuracy tasks close to the robot base and thereby avoid situation where the robot is fully stretched with a high load.

4.3.2 Gear transmission error

Judd and Knasinski reported that gearing errors accounts for 1% of the total RMS error and that an important factor of this error is how the encoder is connected to the main drive gear. If the main drive is connected through a single anti-backlash gear, only encoder pinion gear and main gear required modeling, but if additional gearing is placed between the encoder and the link the modeling becomes more complex.

Renders et Al claims in 1991 that gear backlash alone contributes to 0.5-1.0% of the total error, but also that gear transmission errors are mainly due to runout and/or orientation error.

Gear runout

Runout occurs when the gears centre of rotation does not coincide with the pitch centre of the gear. This is causing the contact point between the gears to move radially during rotation, generating a non-uniform rotary motion.

Orientation error

Orientation error occurs when the shafts of a ring gear and a pinion gear is not parallel (see figure 26). As of this, the distance between a point on the contact line and the centre of the pitch circle will vary, again inducing a non-uniform rotary motion. Hence, the line of contact projected on an A-A plane will be ecliptic.

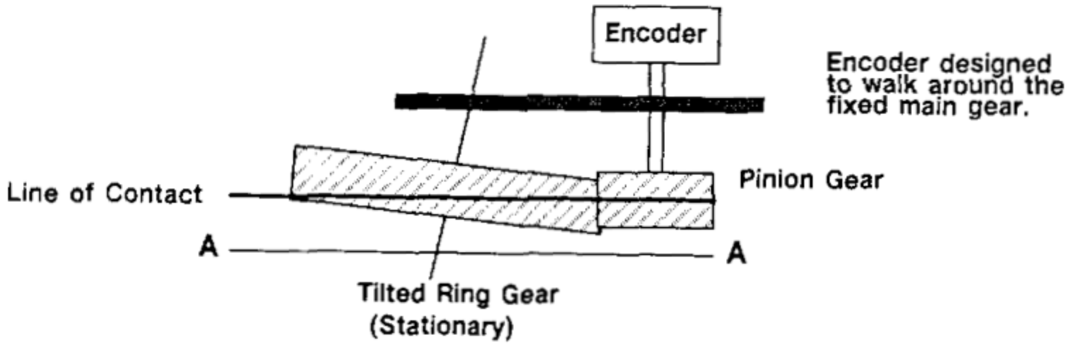

Encoder
Encoder designed to walk around the fixed main gear.
Pinion Gear
Line of Contact
A
A
Tilted Ring Gear (Stationary)

Figure 26 Gear orientation error (Judd and Knasinski 1990)

Dead zone and backlash

All electric industrial robots will have a “dead zone” in each link. The dead zone is a small range the robot is able to move, without developing any error signal. More loosely spoken this means that when the robot stops at a programmed position each link will stop somewhere within this deadzone. The free motion allowed inside this deadzone is known as the gear backlash (See figure 27). The measure for this error is more commonly known as the repeatability of the robot. The following generates the deadzone: resolution of digital encoder, steady state error in the servo loop and gear backlash. Gear backlash is probably one of the most difficult errors to identify (Renders et al. 1991). A common assumption first used by Whitney et al. in 1986 is that the gears are always on one extreme inside the dead zone, however Renders et al. (1991) argues that dry friction may be important and that a static equilibrium can be found inside the dead zone and not only on one of the extremes.

Tests on the ABB IRB 1600 by Nubiola and Bonev in 2013 showed the maximum backlash transpires when Joint 2 = 0° and that the reason it occurs in this position is because the torque on joint 2 is close to zero, making the gear train less stiff.

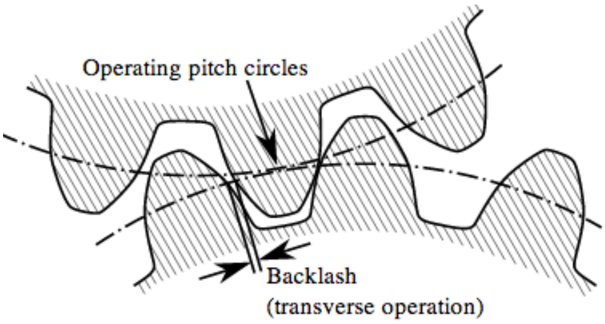


Figure 27 Backlash example

4.3.3 Thermal error

Thermal errors refer to the deviation between the robots kinematic model and actual parameters due to expansion and distortion of the robot structure as a result of thermal expansion. The expansion propagates from two heat sources, internal and external. The motors and bearings generate internal heat during operations and the external is caused by ambient temperature changes in the robots operating environments. As the internal temperature is increased the heat will be dissipated through the mechanical structure of the robot, thus resulting in thermal expansion. In their paper from 1990, Judd and Knasinki reported that the error could be more or less avoided by controlling the ambient temperature in the robots operating area and they were followed up by the paper by Renders et al. in 1991 stating that the error due to temperature effects is responsible for only 0.1% of the total positioning error.

However the capability test of several robots from different manufacturers by Mark summers, on behalf of Airbus UK in 2005 concluded that temperature changes seriously affects the positional accuracy of all the test robots. In a performance assessment of an ABB IRB 1600 by Slamani et Al. in 2012, also showed obvious effects of thermal expansion. In their assessment the robot performed a 30-cycles repeatability test, in accordance with ISO 9283, several times where each test cycle took approx. 2,5 hours. The result (see figure 28) showed that the thermal expansion was very small during the first two hours and that the thermal expansion was clearly over after 24 hours.

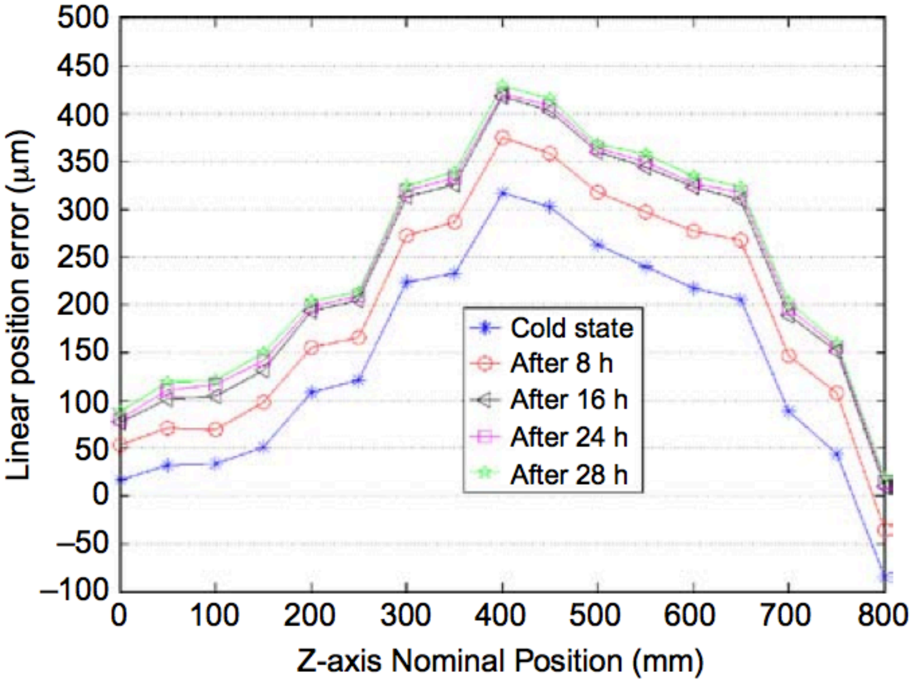


Figure 28 Time variation of linear position error due to internal heat sources (Slamani et al. 2012)

Chapter 5

Introduction to robot calibration

This chapter aims to give a brief introduction to the field of robot calibration, three different levels of calibration are described and an overview of their realization is presented. Finally some of the benefits of robot calibration are discussed.

5.1 General

In 1987 Roth et Al. published a paper giving an overview of Robot calibration, where they describes the term *"Robot calibration"* as a process to improve the positioning accuracy of a robot by improving/altering the software in the robot controller, rather than changing the mechanical design of the robot. To do so, a nominal relationship between the robots joint transducer (encoder) readings and end-effector needed be known. Thus, we can simplify the definition to: Identifying a more accurate relationship between the joint encoders and the actual position of the end-effector. Robot calibration must not be confused with adaptive control, whereas calibration is a discrete event and adaptive control is a continuous process.

There are different approaches to calibrating an industrial robot with varying complexity and results. In the previously mentioned paper by Roth et al. (1987), different calibration approaches were classified into three different levels:

Level 1	Joint calibration
Level 2	Kinematic model calibration
Level 3	“Non-kinematic” (non-geometric) calibration

The first level describes the most straightforward calibration procedure and level 3 the most complex. Furthermore each calibration procedure is described by a four-step sequence:

Step 1	Modelling step: Find the functional relationship
Step 2	Measurement step: Relate model input with output
Step 3	Identification step: Identify the coefficients of the model
Step 4	Correction step: Implement the new model in the controller

This classification will be used in the following sub-chapters to describe the different robot calibration procedures.

5.2 Level 1: Joint calibration

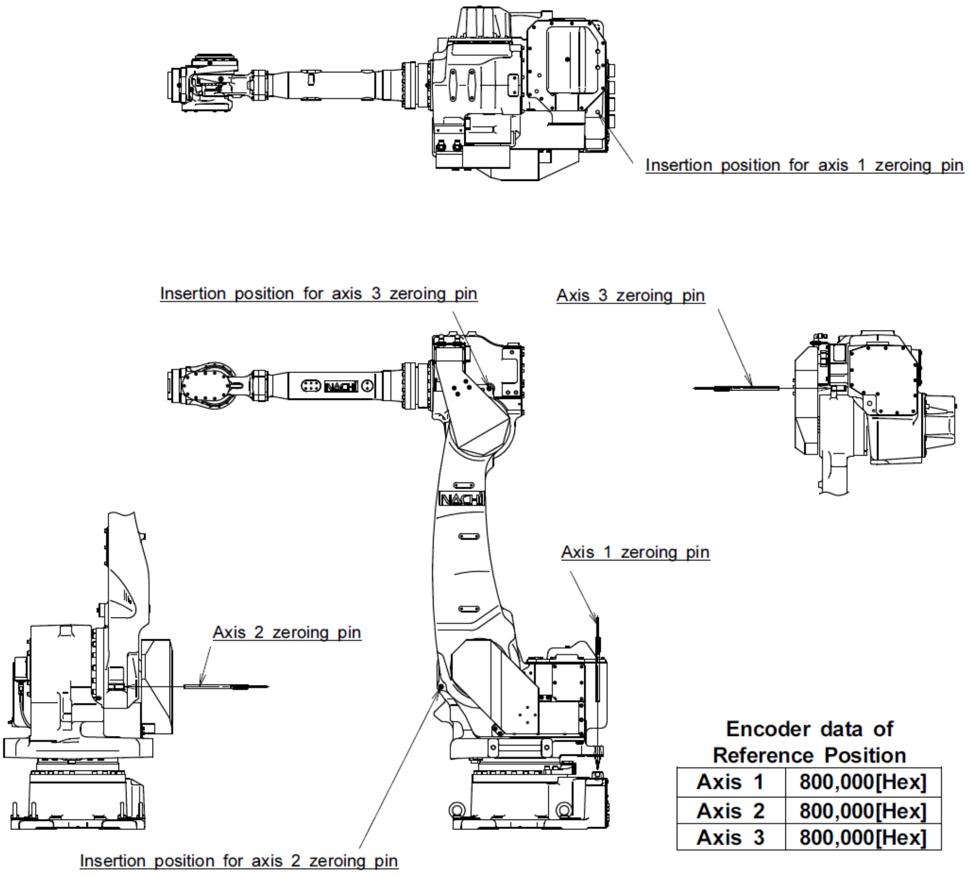
As mentioned this is the most straightforward procedure, where the purpose of the calibration is to ensure that the signal from the joint encoders correspond to the actual displacement of the mechanical joint, thus corresponding to the joint angle offset error described in section 4.4.2. This calibration is typically performed in the final stages of assembling the robot. The procedure should also be repeated after collisions and in the case of disassembly due to maintenance purposes. The technique of performing this method varies from just using visual inspection by moving the robot to a known position that is easily verified by the geometry of the robot, such as a known angle or “straight up”.

A more accurate calibration approach is to use a tolerance peg. Most modern robots are equipped with alignment holes that shall be perfectly aligned when the joint is in a certain position (usually the zero reference position), for some joints (wrist joints in the case of the MC70) external fixture is needed to create the alignment holes. Figure 29 shows a figure from the NACHI manual on calibration of the first three joints of the MC70.

By, following the four-step sequence there is no need for modelling, but the fitting of the tolerance peg will be the *measurement step*. When the holes are perfectly aligned and the tolerance peg easily fits in the two holes, the encoder value is compared to the, known, actual joint-offset value (*identification step)*. If the actual encoder readings doesn't match the zero-reference position values, the encoders are reset to match the joint offset, correcting the joint-offset values in the encoder model, thus the *correction step*. Obviously the procedure is performed for each of the joints.

There are several other approaches to perform this procedure; in 1986 Whitney et Al. presented a method using polygonal mirrors and theodolites. The mirrors were attached to the joint and the operator was looking through the theodolite and rotated the joint until the mirror reflection was seen centred in the theodolite. At this point the mirror is almost perpendicular to the axis of the theodolite and the signal from the joint encoder is read. The joint is then, similarly, rotated to the next mirror. As the angle between the mirrors is known, this is used to identify and correct the error. Compared to the tolerance-peg method, this procedure relies on a mathematical model of the entire set-up. A different approach similar to the tolerance peg method is the use of matchmarks, to indicate the zero-reference position (see figure 30). This is more effective than the tolerance peg method, however as it relies on the visual estimation accuracy of the operator it is less accurate.

The calibration procedure described and implemented in the next following chapters, aims to perform a joint-offset calibration automatically, to reduce the joint angle offset error in order to achieve higher accuracy in a limited space in the robots workspace.


Insertion position for axis 1 zeroing pin
Insertion position for axis 3 zeroing pin
Axis 3 zeroing pin
Axis 1 zeroing pin
Axis 2 zeroing pin
Insertion position for axis 2 zeroing pin

Encoder data of Reference Position

Axis 1	800,000[Hex]
Axis 2	800,000[Hex]
Axis 3	800,000[Hex]

Figure 29 Tolerance peg calibration (Courtesy: NACHI)

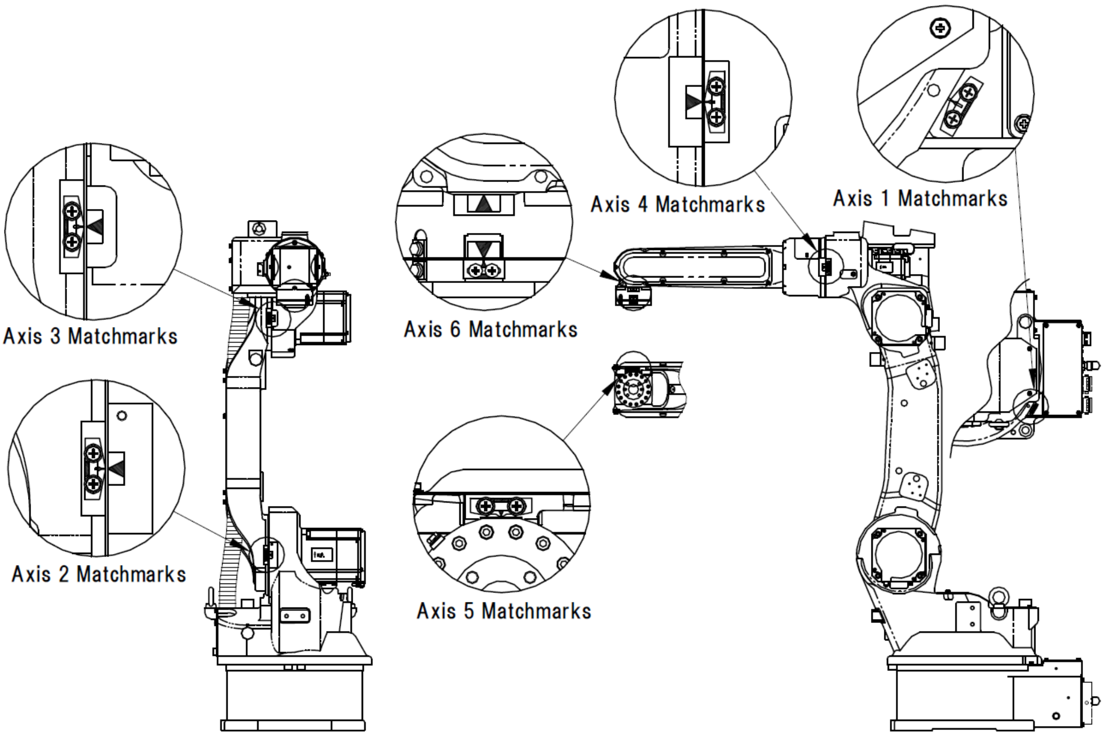


Figure 30 Matchmark calibration (Courtesy: NACHI)

5.3 Level 2: Kinematic model calibration

The purpose of this level of calibration is to improve the accuracy of the robots kinematic model, as well as the correct joint angle relationship. During this level of calibration it is assumed that the links of the robot is rigid and the joints are assumed to be perfect, meaning that they will have no undesired motion about their axes.

Modelling step

For kinematic robot modelling the most popular method is the Denavit-Hartenberg convention (described in section 2.2.1). But, as many researchers have pointed out, this method has a weakness, in the form of a singularity problem for two consecutive parallel joints. If the two joints are not actually parallel this induce an error that cannot be modelled by the D-H convention (see figure 31). In 1983 Hayati proposed a solution to this problem, by adding an extra rotation parameter for parallel revolute axes. Other methods for modelling the kinematics of industrial robots that is used in robot calibration is the S-model (Stone 1986), which uses six parameters to represent each link, and the Zero-reference model (Mooring and Tang 1984) which does not use any common normal as link parameters, thus eliminating the singularity problem in D-H convention. Nonetheless, The D-H convention is still the preferred method and is used in most of the existing robot controllers.

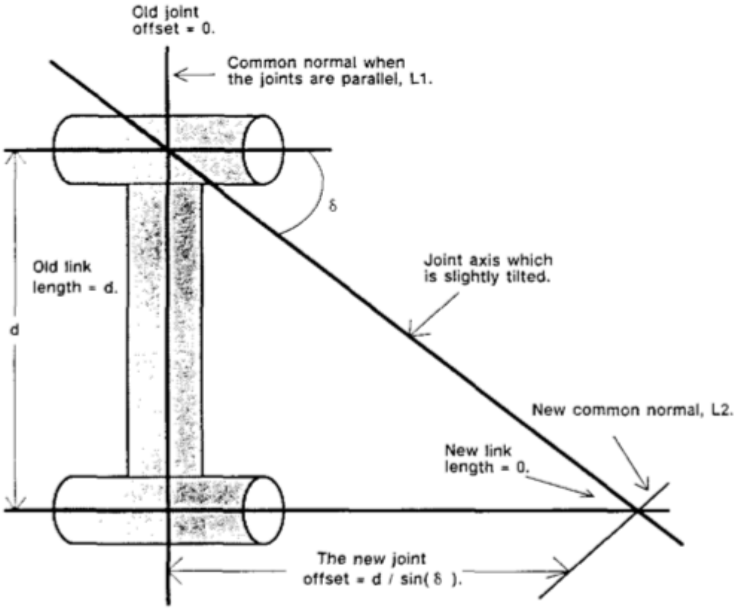

Old joint offset = 0.
Common normal when the joints are parallel, L1.
δ
Old link length = d.
d
Joint axis which is slightly tilted.
New common normal, L2.
New link length = 0.
The new joint offset = d / sin(δ).

Figure 31 Singularity problem using the DH convention (Judd and Knasinski 1990)

Measurement step

The measurement step involves workspace sensing of the positions of the robot end-effector or tool. The actual positions are measured for later being compared to the mathematical model (identification step). The measurement phase is the most time consuming and difficult step of robot calibration. The accuracy measurement equipment, described in sub-chapter 3.4, is used when performing the measurement step and Similar to accuracy measurement there is no "ideal" measurement equipment; "*In most cases, the selection of measurement scheme is a trade-off between the accuracy desired and the cost of the system*" (Mooring et al. 1991:104)

Other important factor to take into account when selecting measurement equipment is mentioned below (Elatta et al. 2004):

- To operate the measuring equipment properly, trained personnel are needed.
- The data collecting is time-consuming, fatiguing and difficult to automate
- The measurement techniques are mainly designed for calibration in a laboratory environment and in many cases not suitable for industry sites.
- Set-up and measurements procedures require a lot of operator intervention, which make it unsuitable for on-site calibration in many industrial operations.

Identification step

When the measurements are collected the error between the actual data and the mathematical model is identified. From this the kinematic parameter errors of the model are identified. Finally the model parameters that gives the "best fit" with the actual model is estimated, normally, by a linear or non-linear least square algorithms, as these are the most straightforward methods (Mooring et al. 1991),

Correction step

Finally the new model is implemented in the robot control software. In some cases the robot manufacturers will not allow direct modification of the controllers parameters. A typical solution for this issue is to create joint compensations to the encoder readings, by solving inverse kinematics of the calibrated robot externally.

5.4 Level 3: Non-kinematic calibration

In this final level of calibration, the underlying assumption in level 2 calibrations; that the pose of the robot could be defined as a function of links and joints only, is no longer valid. The errors that are calibrated in level 3 correspond to the Non-geometric errors described in section 4.3, in addition to including the level 1 and 2 calibration. The challenge with level 3 calibrations is to obtain an accurate model of these error sources. Due to the complexity of this type of calibration, very few researchers have attempted this type of calibration.

As of the lack of research in the field and since it has minor significance for the calibration implementation described in the next chapter Non-kinematic calibration is not given further attention in this report.

5.5 Benefits of robot calibration

Robot calibration plays an important role in robot production, implementation and operation. Calibration leads to significant improvements in robot accuracy and/or cost savings. The benefits can be divided into four groups (Elatta et al. 2004), explained below:

Implementing off-line planned and simulated robot tasks

Improved robot accuracy allows for implementation of large offline simulated systems. Rather than having to test and program on the actual system, programming and simulations of new (or improvements of existing) operations can be performed offline, thus saving time, minimizing production downtime and costly mistakes can be avoided. Offline simulation is relying on the similarity between the model and the dimensions of the actual set-up; a small difference will show up as a positioning error on the actual robot.

Improving control and simulation of robot motion

As robot calibration gives a precise identification of the robots internal parameters, such as friction and structural stiffness this can be modelled to create a more accurate control and simulation of the robot.

Evaluation of robot production

As mentioned in section 4.2.1, the robot accuracy is highly dependant of minimum tolerances in the robot manufacturing. Robot calibration gives a way to assess the production procedure and to identify if the specifications are met after the robot is assembled. Calibration also allows improving of the robot accuracy without having to make any mechanical changes, thus minimal additional cost to the robot's total production cost.

Monitoring robot components wear

A robot will, during normal operation, eventually suffer from wear and tear that will affect the positioning accuracy. This error can be eliminated by a periodic re-calibration of the robot and if the calibration parameter correction exceeds a pre-defined value it can notify the operator that a repair is needed, if not the parameters are adjusted and the operation continues. By automating the calibration procedure, this can be programmed to run each time the robot is waiting for a new task.

Chapter 6

Principle for joint offset calibration on plane

This chapter describes the mathematic principle for performing a joint offset calibration on a 6-axis industrial robot, using a flange-mounted high precision laser distance sensor and a plane as calibration object. Figure 32, shows an overall flowchart of the calibration principle. The method is previously implemented by Ph.D. Morten Lind, Senior Researcher at SINTEF Raufoss Manufacturing, for experimental use on a Universal Robot UR5 with an application written in Python. As mentioned, his experiments showed an unsatisfactory calibration result, and stated the hypothesis that the result is either due to errors in the UR5 kinematic model and/or the fact that the UR5 cannot be modelled as a stiff robot. As a natural second step the candidate will implement this calibration method on a NACHI MC70 robot, a stiff heavy load industrial robot, where the kinematic parameters are assumed to be in compliance with the actual robot. An important part of this research is to verify that the mathematical principle is adaptable to a different kind of robot with a dissimilar hardware setup than for the UR5 experiments. The chapter will involve step 1 and 3, modelling and identification step, of the previously mentioned four-step calibration sequence defined by Roth et al. in 1987. The measurement step (step 2) and correction step (step 4) is described in chapter 5, that describes the LabVIEW implementation, where the contents of this chapter forms the basis.

Next the derivation of the mathematical model is presented, followed by the optimization procedure; both based on a note by Morten Lind and finally the model parameters are identified with respect to the NACHI MC70 and the actual hardware set-up at the PPM lab.

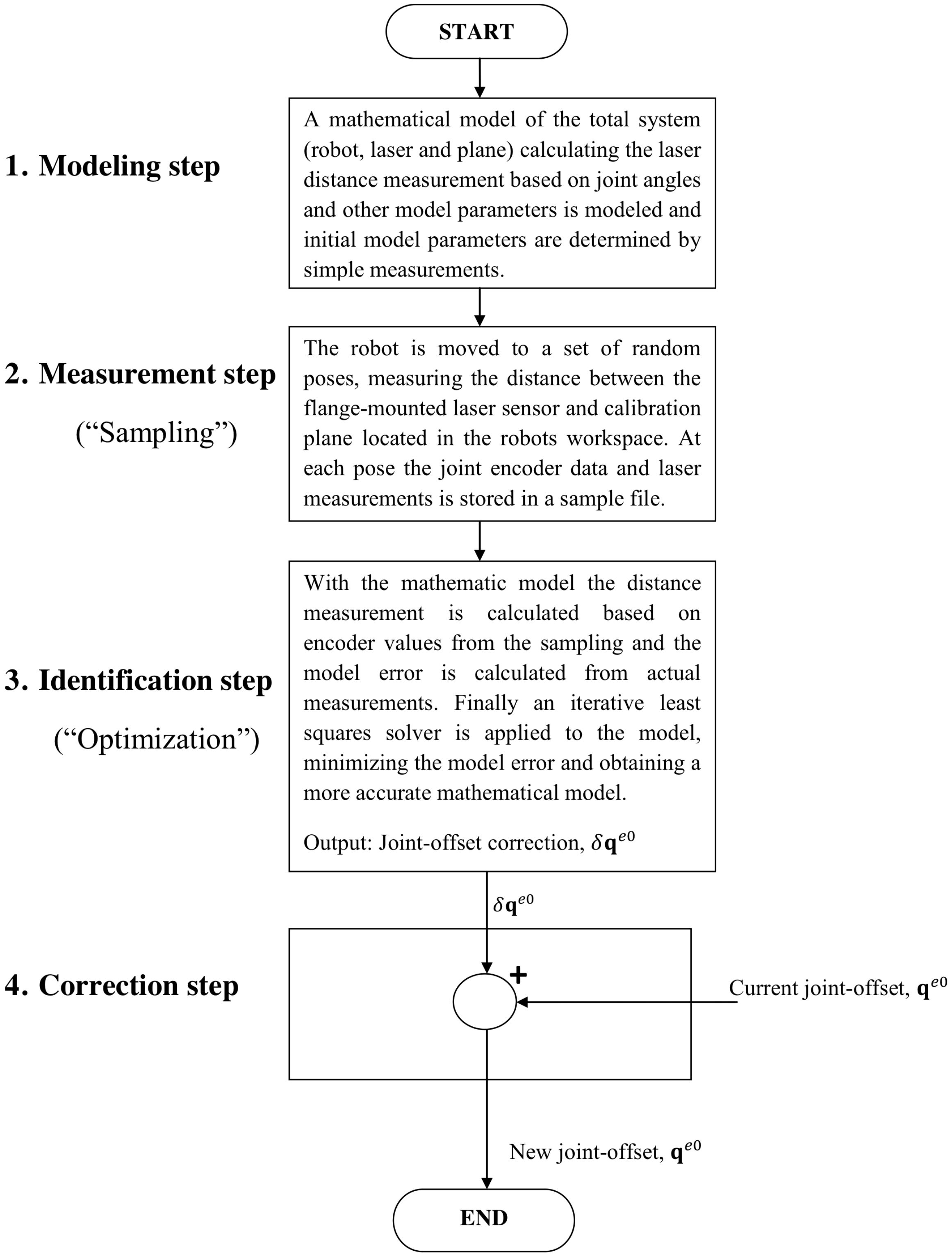

START
1. Modeling step
A mathematical model of the total system (robot, laser and plane) calculating the laser distance measurement based on joint angles and other model parameters is modeled and initial model parameters are determined by simple measurements.
2. Measurement step
("Sampling")
The robot is moved to a set of random poses, measuring the distance between the flange-mounted laser sensor and calibration plane located in the robots workspace. At each pose the joint encoder data and laser measurements is stored in a sample file.
3. Identification step
("Optimization")
With the mathematic model the distance measurement is calculated based on encoder values from the sampling and the model error is calculated from actual measurements. Finally an iterative least squares solver is applied to the model, minimizing the model error and obtaining a more accurate mathematical model.
Output: Joint-offset correction, $\delta\mathbf{q}^{e0}$
$\delta\mathbf{q}^{e0}$
4. Correction step
+
Current joint-offset, $\mathbf{q}^{e0}$
New joint-offset, $\mathbf{q}^{e0}$
END

Figure 32 Flowchart of the principle of the joint offset calibration on plane method

6.1 Total system modelling

The purpose of the total system model is to be able to obtain a computational model for the laser distance measurement on the calibration plane, for a given set of joint angles and the model parameters. In order to achieve this, all the system elements need to be modelled. To give an overview of the total system model it can be broken down into the sub models shown in figure 33. The subsequent sections will present each of the sub-models and finally combine them to the total system model.

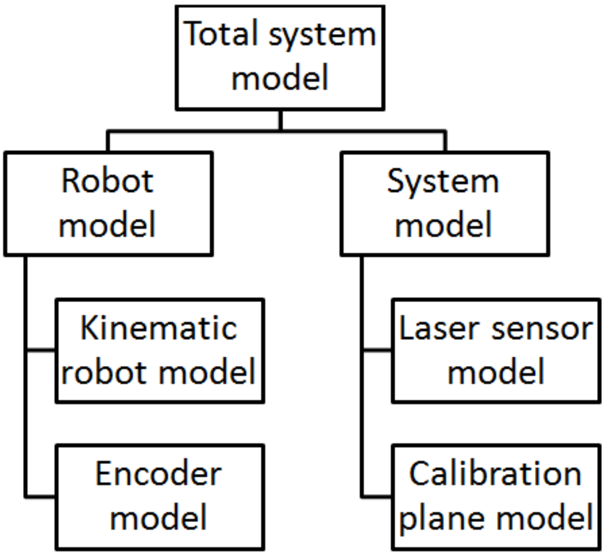


Figure 33 Sub-models of the total system model

6.1.1 Robot model

The robot model consists of a Kinematic robot model and an encoder model. By combining the two models we obtain the *encoder tool flange model.*

Kinematic robot model

As mentioned in sub-chapter 2.3, the purpose of modelling the robot is to be able to describe the pose of the tool flange coordinate system, in the base coordinate system, given a set of joint angles. As described in section 2.2.1, this can be calculated using the Denavit-Hartenberg (DH) convention to create the homogenous transformation matrix;

$$ {}^{Base}\mathbf{T}_{Flange}(\overline{\boldsymbol{q}}) = \begin{bmatrix} \begin{bmatrix} \cdot & & \\ & {}^{Base}\mathbf{R}_{Flange} & \\ & & \cdot \end{bmatrix} & \begin{bmatrix} \cdot \\ {}^{Base}\mathbf{Q}_{Flange} \\ \cdot \end{bmatrix} \\ 0 \quad\quad 0 \quad\quad 0 & 1 \end{bmatrix} \tag{6.1} $$

Encoder model

The encoder model describes the relationship between the joint angles and the encoder values by an encoder gain vector and an encoder-offset vector. The encoder-offset vector describes the encoder value of the robot when it is in the zero-reference position.

In layman's term the encoder model can be described as follows

$$ Joint\ positions\ [rad] = \frac{Current\ encoder\ value - Encoder\ offset\ value}{Pulse\ constant} $$

This can be written as

$$ \overline{\boldsymbol{q}} = \overline{\boldsymbol{g}}^{-1}(\overline{\boldsymbol{q}}^{e} - \overline{\boldsymbol{q}}^{e0}) \tag{6.2} $$

Thus

$$ \overline{\boldsymbol{q}}^{e} = \overline{\boldsymbol{g}}\overline{\boldsymbol{q}} + \overline{\boldsymbol{q}}^{e0} \tag{6.3} $$

where,

$\overline{\boldsymbol{q}}$ = Joint position vector [Radians]

$\overline{\boldsymbol{g}}$ = Encoder pulse constant vector

$\overline{\boldsymbol{q}}^{\boldsymbol{e}}$ = Current encoder position vector [Long unsigned integer]

$\overline{\boldsymbol{q}}^{\boldsymbol{e0}}$ = Encoder offset vector [Long unsigned integer]

By combining the encoder model (6.2) and the robots kinematic model (6.1) we obtain the *encoder tool flange model:*

$$ {}^{Base}\mathbf{T}_{Flange}^{\ \ Enc}(\overline{\boldsymbol{q}}^{e};\ \overline{\boldsymbol{g}}, \overline{\boldsymbol{q}}^{\boldsymbol{e0}}) = {}^{Base}\mathbf{T}_{Flange}(\overline{\boldsymbol{q}}(\overline{\boldsymbol{q}}^{e};\ \overline{\boldsymbol{g}}, \overline{\boldsymbol{q}}^{\boldsymbol{e0}})) \tag{6.4} $$

The main objective with the calibration procedure is to find the correct value for the encoder offset; $\overline{\boldsymbol{q}}^{\boldsymbol{e0}}$, herby referred to as the *Joint offset.* Another value that can be subject for calibration is the encoder pulse constant; $\overline{\boldsymbol{g}}$, however this is assumed to be a fixed value and therefore not subject to calibration.

6.1.2 System model

In this section the two final components of the system; the laser sensor and the calibration plane, is modelled.

Laser sensor model

The laser model simply describes the relationship between the measured point and the origin of the measurement, offset by the laser distance measurement, see figure 34.

$$\mathrm{p_m} = \mathrm{o}_{Sensor} + d_m \,\hat{\boldsymbol{z}}_{Sensor} \tag{6.5}$$

where,

d_m = The distance measured by the laser

o_{Sensor} = The origin in space from which the laser distance is measured

$\hat{\boldsymbol{z}}_{Sensor}$ = The direction of the laser beam

$\mathrm{p_m}$ = The point in space which is the object of measurement by laser beam; i.e. the point hit by the laser.

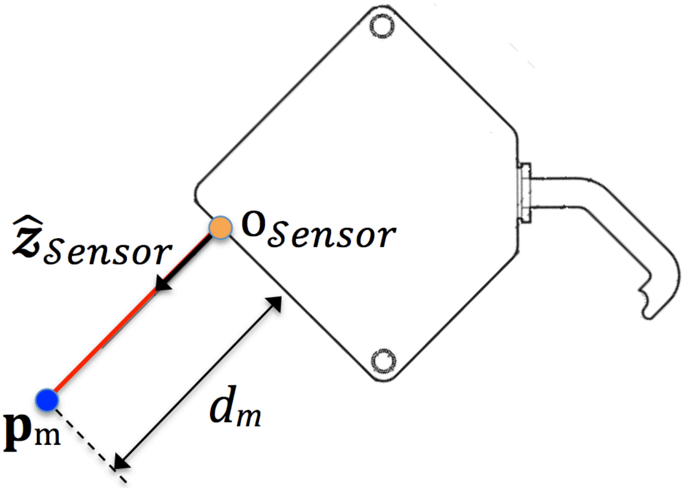


Figure 34 Laser sensor model

A partial coordinate system, $Sensor$, is created for the laser sensor, as it only consist of a point (o_{Sensor}) and a z-direction ($\hat{\boldsymbol{z}}_{Sensor}$). Generally, we are free to chose the $\hat{\boldsymbol{x}}_S$ and $\hat{\boldsymbol{y}}_S$ direction, thus creating a complete coordinate system; however they are of no significance for the total system model and therefore unnecessary. In the partial sensor coordinate system, o_{Sensor} will be the point (0,0,0) and the $\hat{\boldsymbol{z}}_{Sensor}$, the unit-vector $[0\ 0\ 1]^{\mathrm{T}}$. To give more meaning to these parameters they are defined in the Flange coordinate system. For the origin this generates a translational vector:

$$^{Flange}o_{Sensor} = \left[^{Flange}x_{Sensor}\ \ ^{Flange}y_{Sensor}\ \ ^{Flange}z_{Sensor}\right]^{\mathrm{T}} \tag{6.6}$$

and $^{Flange}\hat{\boldsymbol{z}}_{Sensor}$ will be described by two proper Euler rotations (as a third rotation only will be around the laser beam and therefore not off importance) in the Flange coordinate system, by a pitch Φ_y around flange y-axis and a roll Φ_z around the flange y-axis. Thus, we can describe the laser orientation by

$$^{Flange}\hat{\boldsymbol{z}}_{Sensor} = Rot(\hat{\boldsymbol{z}}_{Flange,}\Phi_z)\ Rot(\hat{\boldsymbol{y}}_{Flange,}\Phi_y)\ ^{Flange}\hat{\boldsymbol{z}}_{Flange}$$

By using the rotational matrices found in appendix B, the following expression is obtained:

$$^{Flange}\mathbf{R}_{Sensor} = \begin{bmatrix} cos\Phi_z cos\Phi_y & -sin\Phi_z & cos\Phi_z sin\Phi_y \\ sin\Phi_z cos\Phi_y & cos\Phi_z & sin\Phi_z sin\Phi_y \\ -sin\Phi_y & 0 & cos\Phi_y \end{bmatrix}$$

where

$$^{Flange}\hat{\boldsymbol{z}}_{Sensor} = \begin{bmatrix} cos\Phi_z sin\Phi_y \\ sin\Phi_z sin\Phi_y \\ cos\Phi_y \end{bmatrix} \tag{6.7}$$

Combining the translational vector and rotation matrix of the laser we obtain the homogenous transformation matrix

$$^{Flange}\mathbf{T}_{Sensor} = \begin{bmatrix} \begin{bmatrix} \cdot & & \\ & ^{Flange}\mathbf{R}_{Sensor} & \\ & & \cdot \end{bmatrix} & \begin{bmatrix} \cdot \\ ^{Flange}o_{Sensor} \\ \cdot \end{bmatrix} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{6.8}$$

Finally, the sensor direction parameters is organized into a vector $\vec{\boldsymbol{\phi}} = \left[\Phi_y \; \Phi_z\right]$

Since we, in a flexible system setup, are not able to accurately specify the origin and orientation of the measurement with respect to the tool flange, they have to be included in the calibration procedure.

Calibration plane model

This calibration method can be used with different calibration objects such as a cube or a sphere. However the simplest and most attainable object is a plane. The standard plane equation is commonly written as

$$ax + by + cz + d = 0 \tag{6.9}$$

For an arbitrarily, but fixed coordinate system $\mathcal{P}$ a plane can therefor be described by

$${}^{\mathcal{P}}\hat{\mathbf{n}}^{\mathrm{T}\,\mathcal{P}}\mathrm{p} + d_{\mathcal{P}} = 0 \Leftrightarrow {}^{\mathcal{P}}\hat{\mathbf{n}}^{\mathrm{T}\,\mathcal{P}}\mathrm{p} = -d_{\mathcal{P}} \tag{6.10}$$

Where ${}^{\mathcal{P}}\hat{\boldsymbol{n}}$ is the unit plane normal and $d_{\mathcal{P}}$ is the signed shortest distance between the origin of the coordinate system $\mathcal{P}$ and the plane (see figure 35).

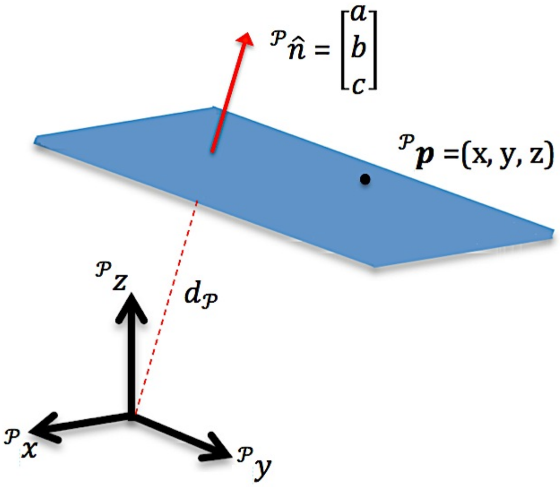

${}^{\mathcal{P}}\hat{n} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$
${}^{\mathcal{P}}\boldsymbol{p}$ =(x, y, z)
${}^{\mathcal{P}}z$
$d_{\mathcal{P}}$
${}^{\mathcal{P}}x$
${}^{\mathcal{P}}y$

Figure 35 Point and normal-representation of plane

We can normalize Equation 6.10 by substituting the unit normal vector with a non-unit length normal vector, ${}^{\mathcal{P}}\tilde{\boldsymbol{n}}$, hereby referred to as the *normalized plane vector*. This is obtained by dividing both sides of the equation with -$d_{\mathcal{P}}$, giving the following formula

$$ {}^{\mathcal{P}}\tilde{\boldsymbol{n}}^{\mathrm{T}\,\mathcal{P}}\,\mathrm{p} = 1 \tag{6.11} $$

where

$$ {}^{\mathcal{P}}\tilde{\boldsymbol{n}} = \frac{1}{-d_{\mathcal{P}}}\,{}^{\mathcal{P}}\hat{\boldsymbol{n}}\,. $$

The equality in equation 6.11 is true for any given point located on the plane defined by ${}^{\mathcal{P}}\tilde{\boldsymbol{n}}$. The sign of $d_{\mathcal{P}}$ determines the direction of the vector, obviously by having a positive sign, the *normalized plane vector* will be footed in the origin of the coordinate system, pointing towards the plane with a magnitude inversely proportional with the distance between the origin and the plane, see figure 36. It is important that the normalized plane vector is always pointing from origin of the coordinate frame and towards the plane.

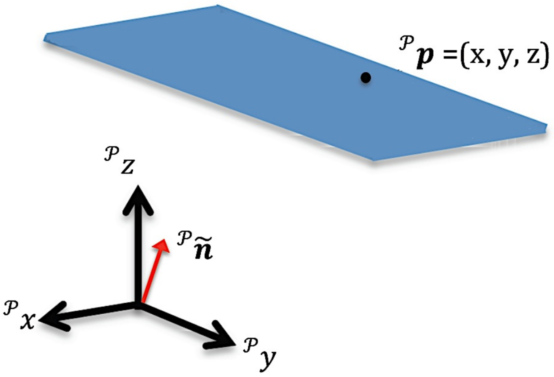

${}^{\mathcal{P}}\boldsymbol{p}$ =(x, y, z)
${}^{\mathcal{P}}z$
${}^{\mathcal{P}}\tilde{\boldsymbol{n}}$
${}^{\mathcal{P}}x$
${}^{\mathcal{P}}y$

Figure 36 Normalized plane vector representation of plane

The normalized plane vector will in the final model be represented in the robot's base coordinate system, and since we are not able to accurately specify the normalized plane vector it also has to be included in the calibration.

6.1.3 Total system model

The complete system model is obtained by combining the robot model with the system model. As previously mentioned, the objective of the mathematical model is to compute the laser distance measurement as a function of the joint positions and other parameters in the model, thus $d_m = d_m(\overline{\boldsymbol{q}};\overline{\boldsymbol{\alpha}})$, where $\overline{\boldsymbol{\alpha}}$ represents an array of all the model parameters.

Equation 6.5 describes the laser distance measurement in the sensor partial coordinate system. By combining the transformation matrices 6.2 and 6.8 we get a transformation between the base coordinate system and the sensor reference system, based on joint angles in encoder values

$$ {}^{Base}\mathbf{T}_{Sensor}{}^{Enc}(\overline{\boldsymbol{q}}^{e};\ \overline{\boldsymbol{g}}, \overline{\boldsymbol{q}}^{e\mathbf{0}}) = {}^{Base}\mathbf{T}_{Flange}{}^{Enc}(\overline{\boldsymbol{q}}^{e};\ \overline{\boldsymbol{g}}, \overline{\boldsymbol{q}}^{e\mathbf{0}})\,{}^{Flange}\mathbf{T}_{Sensor} \tag{6.12} $$

Thus allowing us the describe equation 6.5 in the base coordinate frame

$$ {}^{Base}p_{m} = {}^{Base}o_{Sensor} + d_{m}\,{}^{Base}\hat{\boldsymbol{z}}_{Sensor} \tag{6.13} $$

By substituting equation 6.13 into the *normalized plane equation* (6.11) we obtain the following model for the measured distance

$$ d_{m} = \frac{1 - {}^{Base}\tilde{\boldsymbol{n}}^{\mathrm{T}}\,{}^{Base}o_{Sensor}}{{}^{Base}\tilde{\boldsymbol{n}}^{\mathrm{T}}\,{}^{Base}\hat{\boldsymbol{z}}_{Sensor}} \tag{6.14} $$

where

$$ {}^{Plane}o_{Sensor} = {}^{Base}\mathbf{T}_{Sensor}{}^{Enc}(\overline{\boldsymbol{q}}^{e};\ \overline{\boldsymbol{g}}, \overline{\boldsymbol{q}}^{e\mathbf{0}})\,{}^{Flange}o_{Sensor} \tag{6.15} $$

$$ {}^{Plane}\hat{\boldsymbol{z}}_{Sensor} = {}^{Base}\mathbf{R}_{Sensor}{}^{Enc}(\overline{\boldsymbol{q}}^{e};\ \overline{\boldsymbol{g}}, \overline{\boldsymbol{q}}^{e\mathbf{0}})\,{}^{Flange}\hat{\boldsymbol{z}}_{Sensor} \tag{6.16} $$

The model parameters targeted for calibration can be organized into a vector

$$ \overline{\boldsymbol{\alpha}} = \left[\, \overline{\boldsymbol{q}}^{e0\,\mathrm{T}}\ \ \tilde{\boldsymbol{n}}^{\mathrm{T}}\ \ {}^{Flange}o_{Sensor}{}^{\mathrm{T}}\ \ \boldsymbol{\phi}^{\mathrm{T}} \right]^{\mathrm{T}} \tag{6.17} $$

6.2 Calibration principle

As mentioned, the calibration principle is based on a sample set of laser distance measurements on a planar surface, with a flange-mounted laser sensor. The goal of the procedure is to calibrate the joint-offset values of the robot; however both the plane and sensor system are assumed to have parameters for calibration.

For a given number of samples, a set of poses (defined in encoder values) and corresponding distance measurements from the laser sensor are given, $\{\bar{q}_i^e, \tilde{d}_{mi}\}$. Based on the joint angles and the system model (formula 6.14) the computed measurement is obtained $\{d_{mi} = d_m(\bar{q}_i^e, \bar{\alpha})\}$ from this, the error related to the model is found by: $e_i = \widetilde{d_{mi}} - d_{mi}$.

We now assume that a minor correction in the model parameters, $\delta\bar{\alpha}$ will adjust the model to match the actual measurements, thus

$$\widetilde{d_{mi}} \approx d_m\left(\bar{q}_i^e, \bar{\alpha} + \delta\bar{\alpha}\right) \Leftrightarrow \Delta d_m(\delta\bar{\alpha}) = e_i$$

From this we can formulate the expression

$$\begin{bmatrix} \vdots \\ \frac{\partial d_m}{\partial \bar{\alpha}} \\ \vdots \end{bmatrix} \delta\bar{\alpha} = \begin{bmatrix} \vdots \\ e_i \\ \vdots \end{bmatrix} \quad \Leftrightarrow \quad \mathbf{A}\vec{\boldsymbol{x}} = \vec{\boldsymbol{b}} \tag{6.18}$$

Where $\begin{bmatrix} \vdots \\ \frac{\partial d_m}{\partial \bar{\alpha}} \\ \vdots \end{bmatrix}$ is the parameter Jacobian matrix, its computation, is shown in section 6.2.2.

This stacked error equation (6.18) is to be solved using an iterative least squares method. In more detail, it can be written as follows:

$$\begin{bmatrix} \frac{\partial d_m}{\partial \bar{\boldsymbol{q}}^{e0}}_1 & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}}_1 & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_1 & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_1 \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial d_m}{\partial \bar{\boldsymbol{q}}^{e0}}_i & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}}_i & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_i & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_i \end{bmatrix} \begin{bmatrix} \delta\bar{\boldsymbol{q}}^{e0} \\ \delta\tilde{\boldsymbol{n}} \\ \delta\, {}^{Flange}\boldsymbol{o}_{Sensor} \\ \delta\boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} e_1 \\ \vdots \\ \vdots \\ e_i \end{bmatrix}$$

It is important to note that this calibration is not suitable for calibrating joint offsets for joint q_1 and q_6. An example to illustrate this is when the calibration plane is parallel to the base joint, a change in joint offset at the base will not have any effect on the calculated distance measurement, and therefore it cannot be accounted for in the calibration. By setting the calibration plane not parallel to the base, the calibration of q_1 will be indistinguishable from the rotation of the plane around the base and therefore also not identifiable. Similarly, q_6, the end joint can't be a part of the calibration as it is indistinguishable from the calibration of sensor rotation; Φ_z.

As of this the sensor and plane parameters are kept in the model parameters, while q$_1$ and q$_6$ is eliminated, finally the array of model parameters, subjected for calibration, is arranged as follows:

$$\bar{\boldsymbol{\alpha}} = \left[\, q_2^{e0}\; q_3^{e0}\; q_4^{e0}\; q_5^{e0}\;\; \tilde{n}_x\; \tilde{n}_y\; \tilde{n}_z\; {}^{Flange}x_{Sensor}\;\; {}^{Flange}y_{Sensor}\;\; {}^{Flange}z_{Sensor}\;\; \Phi_y\;\; \Phi_z \right]^{\mathrm{T}}$$

6.2.1 Model optimization

In the model optimization the goal is to find the set $\delta\bar{\alpha}$ that gives

$$min \left\| \begin{bmatrix} \vdots \\ \frac{\partial d_m}{\partial \bar{\alpha}} \\ \vdots \end{bmatrix} \delta\bar{\alpha} - \begin{bmatrix} \vdots \\ e_i \\ \vdots \end{bmatrix} \right\|$$

As this is a non-linear least squares problem, where to goal is minimization of a function, the iterative Gauss-Newton algorithm is applied to equation 6.18. In this algorithm, the Correction is calculated from the following formula;

$$\delta\bar{\alpha} = \begin{bmatrix} \vdots \\ e_i \\ \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ \frac{\partial d_m}{\partial \bar{\alpha}} \\ \vdots \end{bmatrix}^{+} \qquad (6.19)$$

Where $\begin{bmatrix} \vdots \\ \frac{\partial d_m}{\partial \bar{\alpha}} \\ \vdots \end{bmatrix}^{+}$ is the pseudo inverse of the parameter Jacobian. In this case, a pseudo inverse is used because the inverse of the matrix does not exist as the parameter Jacobian is not a square matrix, unless the sample set only consist of 12 samples, this however is too few samples. In LabVIEW the VI that handles pseudo inverse, uses the Singular value decomposition (SVD) algorithm to compute the pseudo inverse of a matrix.

6.2.2 Calculating the parameter Jacobian

This section will mainly focus on the final formulas for the parameter Jacobian and how they are to be used in the software implementation, and not the deriving of the formulas itself. As already emphasized this is the work of Morten Lind, and should be credited as such.

Each row in the parameter Jacobian consists of the following elements:

$$\frac{\partial d_m}{\partial \bar{\alpha}} = \left[\frac{\partial d_m}{\partial \bar{\boldsymbol{q}}^{e0}} \; \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}} \; \frac{\partial d_m}{\partial {}^{Flange}o_{Sensor}} \; \frac{\partial d_m}{\partial \boldsymbol{\phi}} \right]$$

The linearization and iterative method allows for treating the various parameters separately.

Derivative of measurement with respect to normalized plane vector

$$\frac{\partial d_m}{\partial {}^{\mathcal{P}lane}\tilde{\boldsymbol{n}}} = \frac{{}^{Base}\tilde{\boldsymbol{n}} \times \left({}^{Base}\hat{\boldsymbol{z}}_{Sensor} \times {}^{Base}o_{Sensor}\right) - {}^{Base}\hat{\boldsymbol{z}}_{Sensor}}{\left({}^{Base}\tilde{\boldsymbol{n}}^{\mathrm{T}}\,{}^{Base}\hat{\boldsymbol{z}}_{Sensor}\right)^2} \tag{6.20}$$

All parameters   need to be defined in the same coordinate system, typically the robots base or a world coordinate system.

Derivative of measurement with respect to sensor coordinate system

$$\frac{\partial d_m}{\partial {}^{Flange}\boldsymbol{o}_{Sensor}} = -\frac{{}^{Flange}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}}{{}^{Flange}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}\;{}^{Flange}\hat{\boldsymbol{z}}_{Sensor}}$$

Important: All parameters need to be defined in the flange coordinate system. Note that this formula uses the unit plane vector and not the normalized plane vector.

Similarly the derivative with respect to the sensor direction can be calculated as

$$\frac{\partial d_m}{\partial\, {}^{Base}\hat{\boldsymbol{z}}_{Sensor}} = -d_m \frac{{}^{Base}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}}{{}^{Base}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}\, {}^{Base}\hat{\boldsymbol{z}}_{Sensor}}$$

This furthermore can be concatenated into an array, as follows

$$\frac{\partial d_m}{\partial Sensor} = \begin{bmatrix} -\frac{{}^{Base}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}}{{}^{Base}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}\, {}^{Base}\hat{\boldsymbol{z}}_{Sensor}} & -\, d_m \frac{{}^{Base}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}}{{}^{Base}\hat{\boldsymbol{n}}_{Sensor}{}^{\mathrm{T}}\, {}^{Base}\hat{\boldsymbol{z}}_{Sensor}} \end{bmatrix} \quad (6.21)$$

In equation 6.21, all parameters need to be evaluated in the base coordinate frame. This formula can be utilized for any parameter variation that can be expressed in a displacement of the sensor system.

The differential relationship between the flange and sensor coordinate is described in the following equation

$$\frac{\partial Flange}{\partial Sensor} = \begin{bmatrix} \mathbf{I}_3 & -\left[{}^{Base}o_{Sensor} - {}^{Base}o_{Flange}\right]_\times \\ \mathbf{O}_3 & \left[{}^{Base}\hat{\boldsymbol{z}}_{Sensor}\right]_\times \end{bmatrix}$$

Where $\mathbf{I}_3$ is a 3x3 identity matrix and $\mathbf{O}_3$ is a 3x3 zero-matrix, the equation uses the $[\cdot]_\times$ which casts a vector into its cross product operator, defined by:

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

The derivative of the laser measurement with respect to the sensor orientation vector $\vec{\boldsymbol{\phi}}$ can be described, using the chain rule as

$$\frac{\partial d_m}{\partial \boldsymbol{\phi}} = \frac{\partial d_m}{\partial\, {}^{Flange}\hat{\boldsymbol{z}}_{Sensor}} \frac{\partial\, {}^{Flange}\hat{\boldsymbol{z}}_{Sensor}}{\partial \boldsymbol{\phi}} \quad (6.22)$$

Where

$$\frac{\partial^{Flange}\hat{\boldsymbol{z}}_{Sensor}}{\partial\boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial^{Flange}\hat{\boldsymbol{z}}_{Sensor}}{\Phi_y} & \frac{\partial^{Flange}\hat{\boldsymbol{z}}_{Sensor}}{\Phi_z} \end{bmatrix} = \begin{bmatrix} cos\Phi_z cos\Phi_y & -sin\Phi_z sin\Phi_y \\ sin\Phi_z cos\Phi_y & cos\Phi_z sin\Phi_y \\ -sin\Phi_y & 0 \end{bmatrix}$$

Derivative of measurement with respect to sensor coordinate system

Using the chain rule, this can be expressed as

$$\frac{\partial d_m}{\partial\overline{\boldsymbol{q}}^{e0}} = \frac{\partial d_m}{\partial Sensor}\frac{\partial Flange}{\partial Sensor}\frac{\partial Flange}{\partial\overline{\boldsymbol{q}}}\overline{\boldsymbol{g}}^{-1} \tag{6.23}$$

Where the manipulator Jacobian is defined by

$$\frac{\partial Flange}{\partial\overline{\boldsymbol{q}}} = J(\overline{\boldsymbol{q}}) = \begin{bmatrix} \dots & z_i \times (o_{Flange} - o_i) & \dots \\ \dots & z_i & \dots \end{bmatrix}$$

and

$$\frac{\partial Flange}{\overline{\boldsymbol{q}}^{e0}} = \frac{\partial Flange}{\partial\overline{\boldsymbol{q}}}\overline{\boldsymbol{g}}^{-1}$$

6.3 Calibration considerations

Several planes calibration planes

To increase the identifiability of the joint offsets, a solution may be to use several calibration planes inside the workspace that the robot is to be calibrated for. When using several calibration planes, this needs to be incorporated in the model and demands a certain arrangement of the total stacked error equation (equation 7.18). Assuming a hardware setup with 3 planes located in the robots workspace with different orientations, firstly we rearrange the model parameters to

$$\overline{\boldsymbol{\alpha}} = [\, \overline{\boldsymbol{q}}^{e0\,\mathrm{T}} \quad {}^{Flange}o_{Sensor}{}^{\mathrm{T}} \;\; \boldsymbol{\phi}^{\mathrm{T}}\, \tilde{\boldsymbol{n}}_1{}^{\mathrm{T}} \;\; \tilde{\boldsymbol{n}}_2{}^{\mathrm{T}} \;\; \tilde{\boldsymbol{n}}_3{}^{\mathrm{T}} \,]^{\mathrm{T}}$$

Finally the stacked error equation is set up as follows for three different planes will be:

$$\begin{bmatrix} \begin{bmatrix} \frac{\partial d_m}{\partial \overline{\boldsymbol{q}}^{e0}}_1 & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_1 & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_1 & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}_{\boldsymbol{1}_1}} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial d_m}{\partial \overline{\boldsymbol{q}}^{e0}}_i & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_i & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_i & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}_{\boldsymbol{1}_i}} & 0 & 0 \end{bmatrix}_{\tilde{\boldsymbol{n}}_{\boldsymbol{1}}} \\ \begin{bmatrix} \frac{\partial d_m}{\partial \overline{\boldsymbol{q}}^{e0}}_1 & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_1 & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_1 & 0 & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}_{\boldsymbol{2}_1}} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial d_m}{\partial \overline{\boldsymbol{q}}^{e0}}_i & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_i & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_i & 0 & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}_{\boldsymbol{2}_i}} & 0 \end{bmatrix}_{\tilde{\boldsymbol{n}}_{\boldsymbol{2}}} \\ \begin{bmatrix} \frac{\partial d_m}{\partial \overline{\boldsymbol{q}}^{e0}}_1 & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_1 & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_1 & 0 & 0 & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}_{\boldsymbol{3}_1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial d_m}{\partial \overline{\boldsymbol{q}}^{e0}}_i & \frac{\partial d_m}{\partial\, {}^{Flange}\boldsymbol{o}_{Sensor}}_i & \frac{\partial d_m}{\partial \boldsymbol{\phi}}_i & 0 & 0 & \frac{\partial d_m}{\partial \tilde{\boldsymbol{n}}_{\boldsymbol{3}_1}} \end{bmatrix}_{\tilde{\boldsymbol{n}}_{\boldsymbol{3}}} \end{bmatrix} \begin{bmatrix} \delta \overline{\boldsymbol{q}}^{e0} \\ \delta\, {}^{Flange}\boldsymbol{o}_{Sensor} \\ \delta \boldsymbol{\phi} \\ \delta \tilde{\boldsymbol{n}}_{\boldsymbol{1}} \\ \delta \tilde{\boldsymbol{n}}_{\boldsymbol{2}} \\ \delta \tilde{\boldsymbol{n}}_{\boldsymbol{3}} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_i \end{bmatrix}_{\tilde{\boldsymbol{n}}_{\boldsymbol{1}}} \\ \begin{bmatrix} e_1 \\ \vdots \\ e_i \end{bmatrix}_{\tilde{\boldsymbol{n}}_{\boldsymbol{2}}} \\ \begin{bmatrix} e_1 \\ \vdots \\ e_i \end{bmatrix}_{\tilde{\boldsymbol{n}}_{\boldsymbol{3}}} \end{bmatrix}$$

Own coordinate frame for plane to reduce sensitivity

As the length of the normalized plane vector is inversely proportional to the distance between the plane and the coordinate frame it is defined in, an idea is to define the normalized plane vector in a local plane coordinate system, as this will increase the sensitivity in the correction of the plane. Since we still want to define the normalized plane vector in the base coordinates a special transformation is needed to describe the normalized plane vector in a different coordinate systems, this is described in the initial system setup in appendix B.

Scaling joint offsets to radians

In the parameter correction all values except the joint offsets are treated in geometric units, meters and radians. This generates a big difference in correction values for the joint offsets versus the other parameters and can make the system unstable. To avoid this, the joint offsets can be scaled to radians, which is done by setting the scaling factor/pulse constant:

$$\overline{\boldsymbol{g}}^{-\boldsymbol{1}} = 1$$

Obviously, the joint offsets are scaled back to encoder values, using the actual encoder gains after the optimization procedure. The reason this scaling is possible is because we assume fixed gains, and because of the linear relationship between joint angles in encoder values and radians. Finally, after the parameter correction is completed, the joint offset can be calculated in encoder values. In the final application, the candidate implemented this scaling.

6.4 Obtaining initial model parameters from hardware setup

In section 6.1 the total system model was derived. As this is a general model it needs to be adapted to the actual lab setup, used by the candidate at the PPM lab. Thus, this section presents how the actual computational model was obtained. Note that during the testing stage of the thesis, a different calibration setup was initially used, however this did not produce any good results, still the setup is described in the same manner in appendix B, as the difference between the two are subject for the discussion in sub-chapter 9.3.

The idea behind this calibration procedure is that the initial model parameters do not need to be accurately known in order to get a good calibration result, however the tolerances should be within ∓0,01 m for length measurements and ∓0,05 rad. for angular measurements. All the measurements in the following sections are performed using either a ruler or a Vernier caliper.

6.4.1 Robot model

Robot kinematic model

Figure 37 shows a wireframe model of the NACHI MC70 together with the DH-parameters. The DH-parameters is shown in table 2. The length parameters in the DH matrix are based on the datasheet of the NACHI MC70 (See appendix) and the constant file on the FD11 controller.

By using the data in table 2 and the forward kinematics function in *LabVIEW 2013 Robotics Module*, we are given the four-by-four transformation matrix between the base coordinate system and the tool flange based from a set of joint coordinates; ${}^{Base}\mathbf{T}_{Flange}(\overline{\boldsymbol{q}})$.

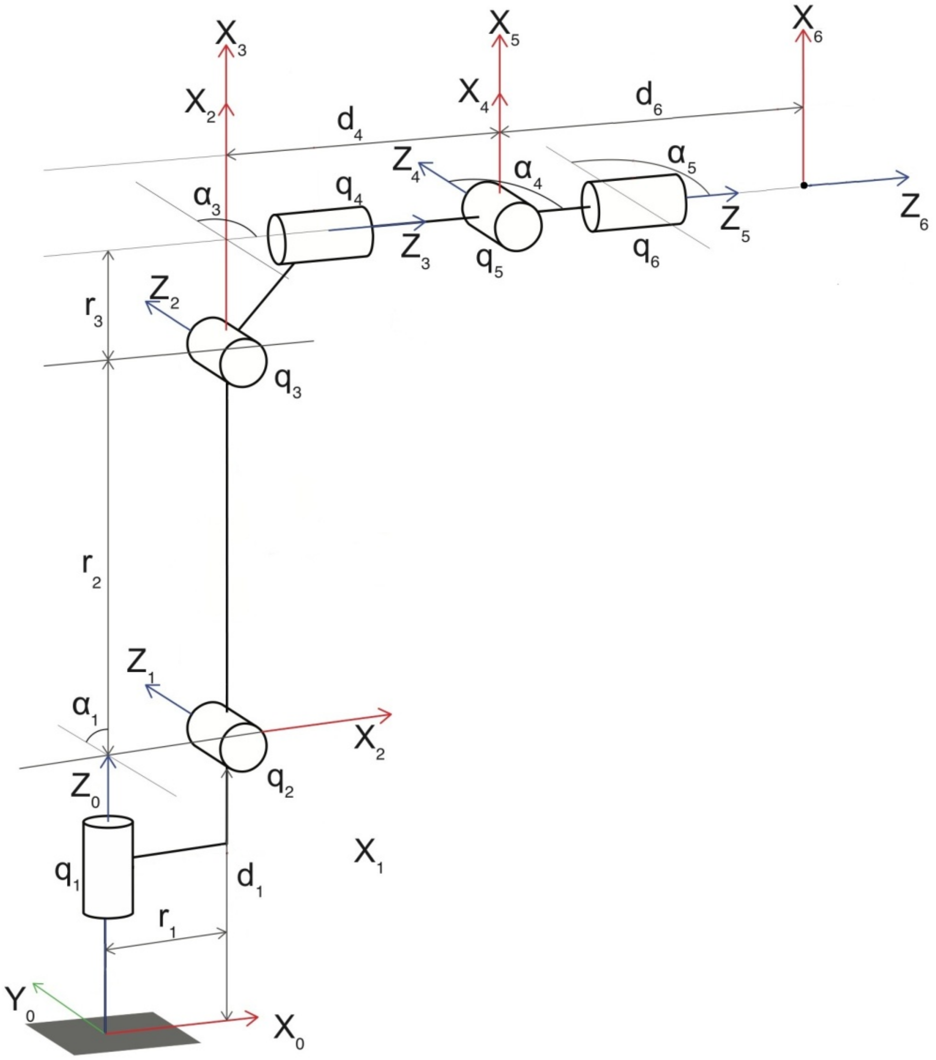


Figure 37 Wireframe model of the NACHI MC70's links

Table 2 Denavit-Hartenberg parameters for NACHI MC70

Link	**θ - Joint angle [rad]**	**d – Joint distance [m]**	**r – Link length [m]**	**α - Link twist [rad]**
1	0	0,6	0	$\frac{\pi}{2}$
2	$\theta_1 + \frac{\pi}{2}$	0	0,15	0
3	θ_2	0	0,9	$\frac{\pi}{2}$
4	θ_3	0,99	0,15	$-\frac{\pi}{2}$
5	θ_4	0	0	$\frac{\pi}{2}$
6	θ_5	0,175	0	$\frac{\pi}{2}$

Encoder model

On the constant file from the robot controller the encoder pulse constants and encoder offsets value (Zero-reference position) for each joint is extracted, the data for the NACHI MC70 is shown in table 3 and 4.

Table 3 Encoder model parameters

Joint	Encoder pulse constant $\bar{g}$	Scaling factor $\bar{g}^{-1}$
q_1	42744,10000	2,33950E-05
q_2	55511,67709	1,80142E-05
q_3	-53651,30000	-1,86389E-05
q_4	37012,10000	2,70182E-05
q_5	-37428,00000	-2,6718E-05
q_6	24536,10000	4,07563E-05

Table 4 Zero reference position for the NACHI MC70

Joint	Joint offset position ($\bar{q}^{e0}$)		
	Hex value	*Long value*	*Angle*
q_1	80000	8388608	0°
q_2	80000	8388608	90°
q_3	80000	8388608	0°
q_4	80000	8388608	0°
q_5	80000	8388608	0°
q_6	80000	8388608	0°

By inserting the encoder gain and joint offset position data into the transformation matrix between the base and flange, this is known as the *Home or Zero-reference-position* and has the following transformation matrix:

$$^{Base}\mathbf{T}_{Flange}{}^{Enc}(\bar{\boldsymbol{q}}(\bar{\boldsymbol{q}}^{e};\ \bar{\boldsymbol{g}}, \bar{\boldsymbol{q}}^{e\mathbf{0}})) = \begin{bmatrix} 0 & 0 & 1 & 1{,}315 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1{,}65 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It is easy to see the compliance between the translation vector and the dimensions of the robot in the datasheet in appendix H1. (OBS! Datasheet displays lengths in mm).

Actuator to joint coupling matrix

When implementing the calibration method using only the encoder model described in section 6.1.1, the calibration results where much worse than Morten Lind's tests results on the Universal UR5. In the last stages of testing the error was found to be caused by insufficient modelling of joint coupling in the wrist of the MC70. The test and results is described in sub-chapter 8.1. The test showed that due to the mechanical structure of the MC70 wrist ($q_{4,}$ q_5 and q_6), a change in q_4^e will produce a small change in q_5^e and q_6^e, also a change in q_5 will produce a change in q_6^e, however these changes will not effect the joint angle positions q_5 and q_6.

As of this, the encoder model has to be expanded to incorporate this effect and give a better fit to the MC70. From this the actuator space was introduced (see figure 38), and thus a tool is needed to transfer between actuator space and joint space, which will be called the *coupling matrix*. In the case of the UR5, the coupling matrix would only be a diagonal matrix of 1, as there is no coupling effect between the joints. On the MC70 on the other hand, the mechanism in the wrist joints creates off-diagonal elements.

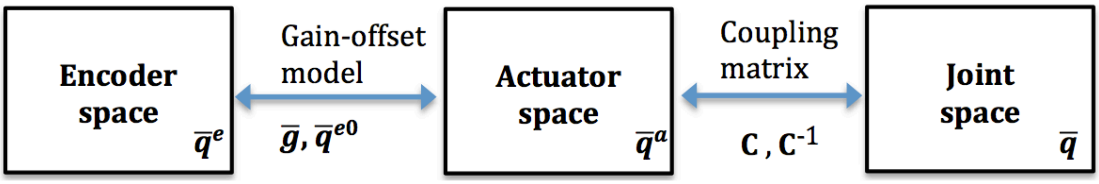


Figure 38 Encoder to joint space model

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & a & 1 & 0 \\ 0 & 0 & 0 & b & c & 1 \end{bmatrix}$$

Where the values of a, b and c was found on the FD11 robot controller to be
a= -0.01111111, b= 0.017137476 and c= -0.016949152

This gives the updated encoder model:

$$\bar{q} = \bar{g}^{-1}(\bar{q}^e - \bar{q}^{e0})\mathbf{C}$$

$$\bar{q}^e = (\bar{g}\bar{q} + \bar{q}^{e0})\mathbf{C}^{-1}$$

6.4.2 System model

Calibration model

The calibration plane is slightly tilted with respect to the robot base frame and the normalized plane vector is calculated from a point and normal representation in the base coordinate system. It is important to avoid a plane orientation where the plane intersects the origin of the base coordinate system, as the normalized plane vector and the point-and-normal representation cannot describe a plane intersecting origin of the coordinate system it is defined in.

Point and normal representation of plane

The parameters that need to be identified in order to calculate the normalized plane vector is shown in figure 39. Firstly **a**, **b** and φ is defined by a simple measurement and some calculations, the results is shown in table 5

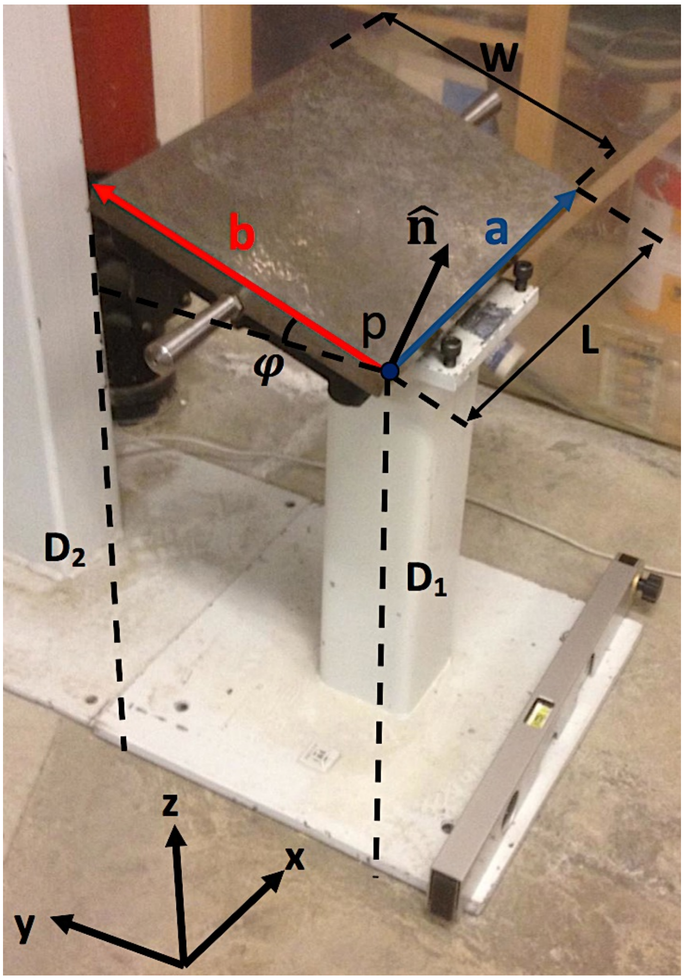


Figure 39 Plane calculations

Table 5 Results from plane calculations

L	0,350 m
W	0,250 m
D₁	0,484 m
D₂	0,552 m
$\boldsymbol{\varphi}$	0,2755 rad.
a	[0,350 0 0]
b	[0 0,2406 0,068]

Whereas

$$\boldsymbol{\varphi} = \ sin^{-\mathbf{1}}(\frac{\mathbf{D_2-D_1}}{\mathbf{W}})$$

Then the unit normal is calculated as the vector perpendicular to **a** and **b**

$$^{Base}\hat{\boldsymbol{n}} = \frac{\mathbf{a}\times\mathbf{b}}{\|\mathbf{a}\times\mathbf{b}\|} = \begin{bmatrix} \mathbf{0} \\ \mathbf{-0,27} \\ \mathbf{0,96} \end{bmatrix}$$

The closest corner of the plane in figure x.x, was estimated to robot base coordinates

$$^{Base}\mathrm{p} = (0{,}729 \quad 0{,}737 \quad 0{,}484)$$

Calculating the normalized plane vector

Finally the normalized plane vector is calculated in the base coordinates by the following formula:

$$^{Base}\tilde{\boldsymbol{n}} = \frac{^{Base}\mathrm{p}^{\mathrm{T}}\,{}^{Base}\hat{\boldsymbol{n}}}{\left|^{Base}\mathrm{p}^{\mathrm{T}}\,{}^{Base}\hat{\boldsymbol{n}}\right|^2}\,{}^{Base}\hat{\boldsymbol{n}}$$

Giving:

$$^{Base}\tilde{\boldsymbol{n}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{-1,02511} \\ \mathbf{3,62708} \end{bmatrix}$$

Laser sensor model

Figure 40 shows an exploded view of the sensor to flange connection. Based on measurements with a Vernier caliper, of the flange adapter and laser sensor the following translation is obtained:

$$^{Flange}o_{Sensor} = \begin{bmatrix} 0{,}0257 \\ -0{,}0725 \\ 0{,}1769 \end{bmatrix} \mathrm{m}$$

The orientation is defined by a 71.2-degree pitch about the flange y-axis followed by a -60 degree roll about flange z-direction. This gives:

$$^{Flange}\hat{\mathbf{z}}_{Sensor} = Rot(\hat{\mathbf{z}}_{Flange,} - 1{,}0472)\, Rot(\hat{\mathbf{y}}_{Flange,} 1{,}2427)\ ^{Flange}\hat{\mathbf{z}}_{Flange}$$

Thus,

$$^{Flange}\hat{\mathbf{z}}_{Sensor} = \begin{bmatrix} \cos(-1{,}0472)\sin(1{,}2427) \\ \sin(-1{,}0472)\sin(1{,}2427) \\ \cos(1{,}2427) \end{bmatrix} = \begin{bmatrix} 0{,}4733 \\ -0{,}8198 \\ 0{,}3222 \end{bmatrix}$$

Thus,

$$\vec{\boldsymbol{\phi}} = [\mathbf{1,2427} \quad \mathbf{-1,0472}]$$

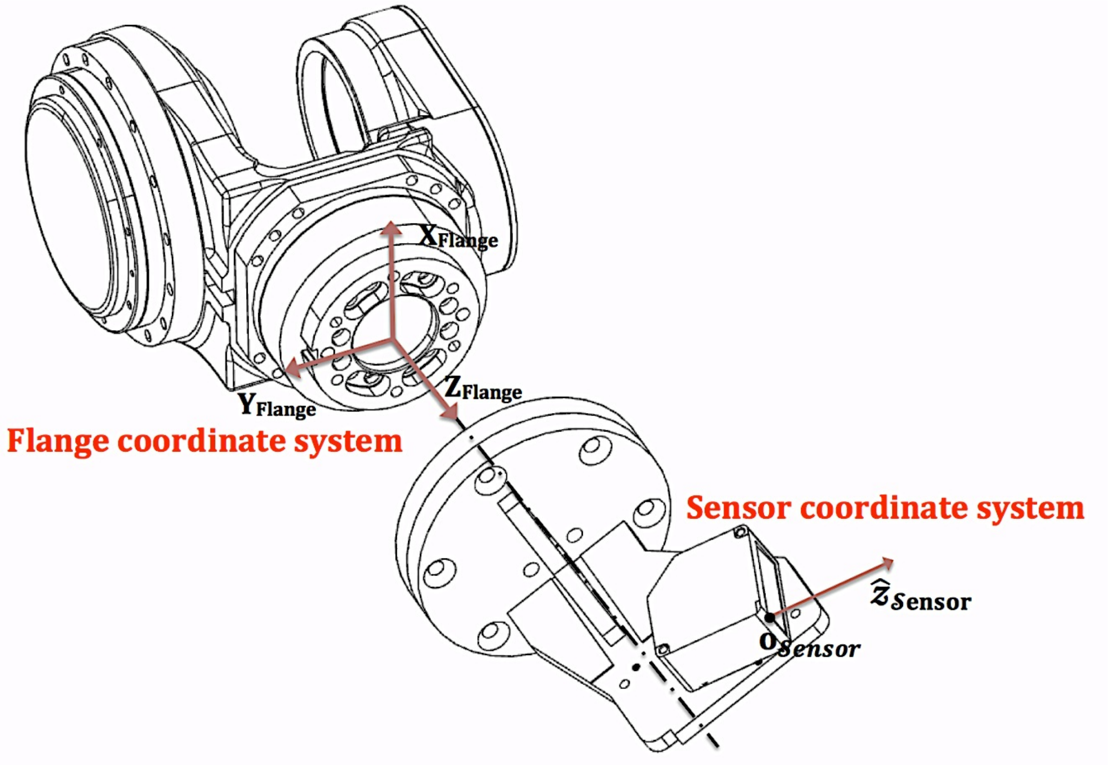


Figure 40 Exploded view of the flange and sensor relationship

6.4.3 Total system model

Figure 41 shows an schematically illustration of the different coordinate system of the hardware setup, together with important model parameters. Based on the previous calculations, initial model parameter array will be:

$$\overline{\boldsymbol{\alpha}} = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ -1{,}02511\ \ 3{,}62708\ \ 0{,}4733\ \ -0{,}8198\ \ 0{,}3222\ \ 1{,}2427\ \ -1{,}0472]^{T}$$

The joint offsets are set equal to zero is because of scaling, discussed in section 6.3.

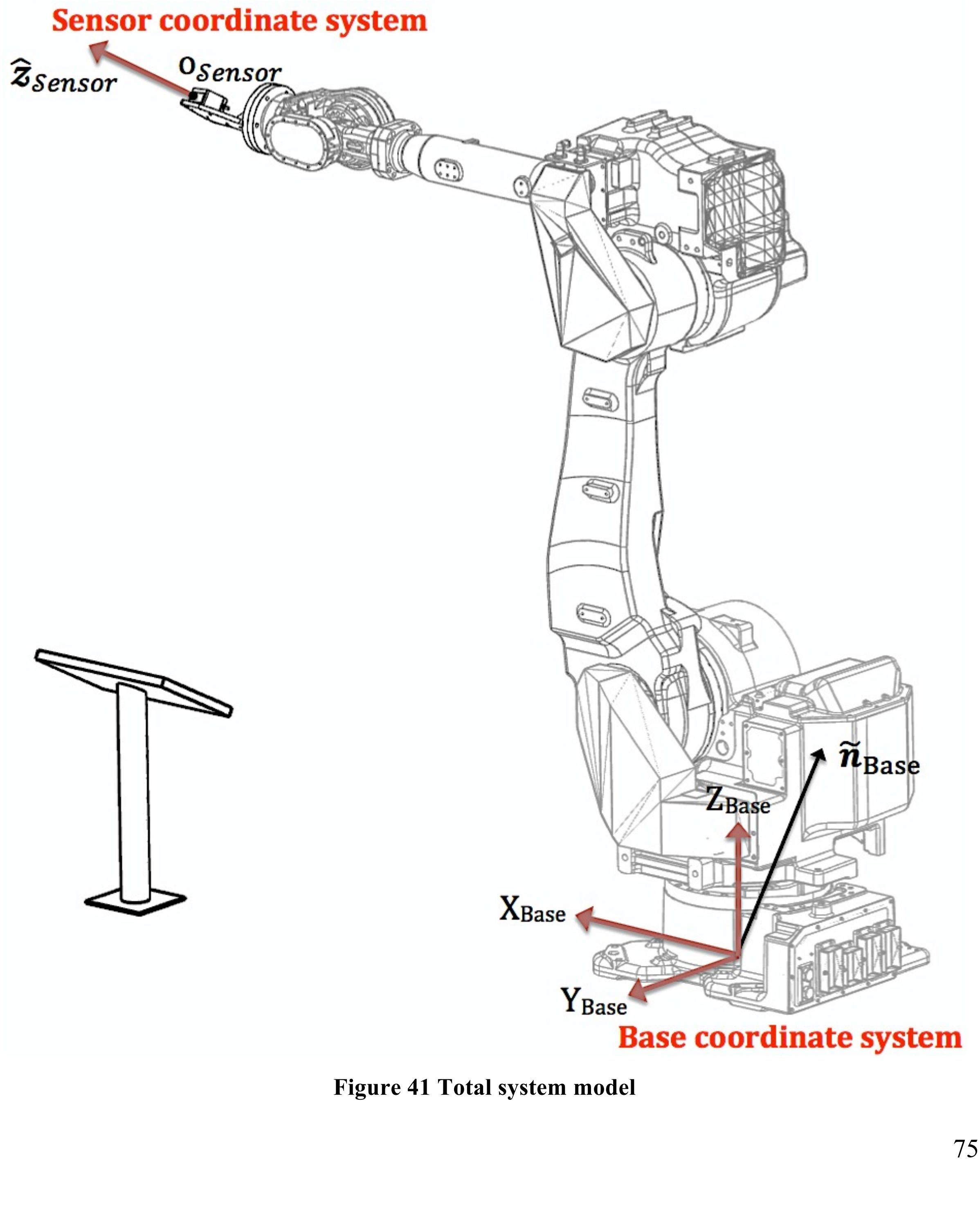


Figure 41 Total system model

Chapter 7

Implementation of joint offset calibration on plane

This chapter will describe how the calibration principle described in the previous chapter was implemented in LabVIEW. As mentioned, Morten Lind originally implemented the calibration in a Python application in order to perform the experiments on the Universal Robot. The reason the candidate has chosen to do the implementation using LabVIEW, this is because of the existing Real-Time framework, for remotely controlling and communicating with the MC70, at the PPM lab. The LabVIEW implementation consists of two LabVIEW applications commonly referred to as VI's, where VI is an acronym for Virtual Instrument.

The first main VI is handling the sampling of laser measurement using the RT-system to control the robot. This VI stores the encoder position data and its corresponding laser measurement in a text-document, which will be opened in the second main VI; the optimization VI. The optimization VI handles the procedure of calculating the model parameters that will give the smallest error between actual measurements and computed measurements, whereas some of these parameters are the calibration parameters.

In the final section, a short user guide of both the applications are presented.

7.1 Sampling application

Obviously the sampling could be done manually by jogging the robot to different poses, and manually record the laser distance and encoder data. However, typical sample sizes will typical lie in the top range of 30-150 samples thus is making a manual operation tedious and cumbersome. Therefore this application was made for automatically generate a specified number of random poses, and move the robot to each pose and perform a laser measurement.

7.1.1 RT system and communication

RT system principle

The sampling application is built upon the existing LabVIEW RT-system framework, developed by PhDc Balázs Daniel, at PPM. The principle of the RT system is described in figure 42

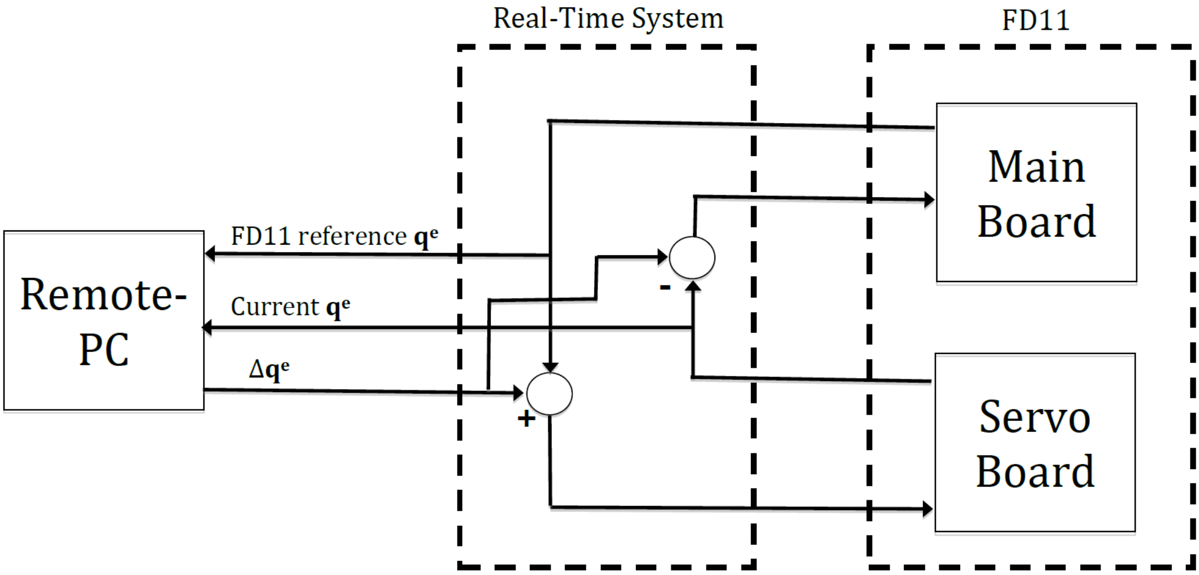


Figure 42 Principle of the Real-time system

When the sampling application is booted it starts off by sending an array of delta joints movements in long integers (hereby referred to as $\Delta\mathbf{q}^e$) to the RT-system. This $\Delta\mathbf{q}^e$ is added to the FD11 reference $\mathbf{q}^e$ to get the commanded encoder position to the servo board:

$$\text{Commanded }\ \mathbf{q}^e = \text{FD11 reference }\mathbf{q}^e + \Delta\mathbf{q}^e$$

Concurrently the $\Delta\mathbf{q}^e$ is subtracted from the Current $\mathbf{q}^e$ and because of this the FD11 reference $\mathbf{q}^e$ will represent the robots initial encoder position when application is booted up and will remain constant while the applications is running. Thus, as the robot, initially, is not programmed to move the initial $\Delta\mathbf{q}^e$ array sent will be [0,0,0,0,0,0], meaning

$$\text{Commanded } \mathbf{q}^e = \text{FD11 reference } \mathbf{q}^e$$

It is important to distinguish $\Delta\mathbf{q}^e$ from the commanded encoder positions.
The reason why it starts off by sending a message is because the RT system is set up to only broadcast messages if it knows that someone is listening, thus avoiding excessive traffic on the network. Thus, when a $\Delta\mathbf{q}^e$ is received by the RT-system (joint change or not) it will respond with the current encoder position (Current $\mathbf{q}^e$).

Robot communication in sampling application

As mentioned, the main structure of the sampling application is built upon the RT system principle and if $\Delta\mathbf{q}^e$ is not altered, it will keep sending the same zero $\Delta\mathbf{q}^e$, or the last sent $\Delta\mathbf{q}^e$ in the case of a completed path, with a delay of 10 ms (100 Hz). This can be seen in the top section of the flowchart in, figure 43. The encoding and decoding of the data is described in the communication protocol in appendix B. The alteration of the $\Delta\mathbf{q}^e$ occurs when pushing the move button; given that a path matrix is generated. See the user guide in section 7.3.1 to see the user interface.

Assuming that a path is generated, thus the path matrix is not empty, the robot will start moving when the operator press the move button, thus setting the Boolean move variable to TRUE and the movement commences. In each iteration, the first row of the path matrix will be extracted and sent to the robot as $\Delta\mathbf{q}^{e,}$ this will continue until the path matrix is empty or the application is stopped (Run is set to FALSE). This is shown in the bottom section of the flowchart in figure 43.

In the next section the path matrix generation will de described.

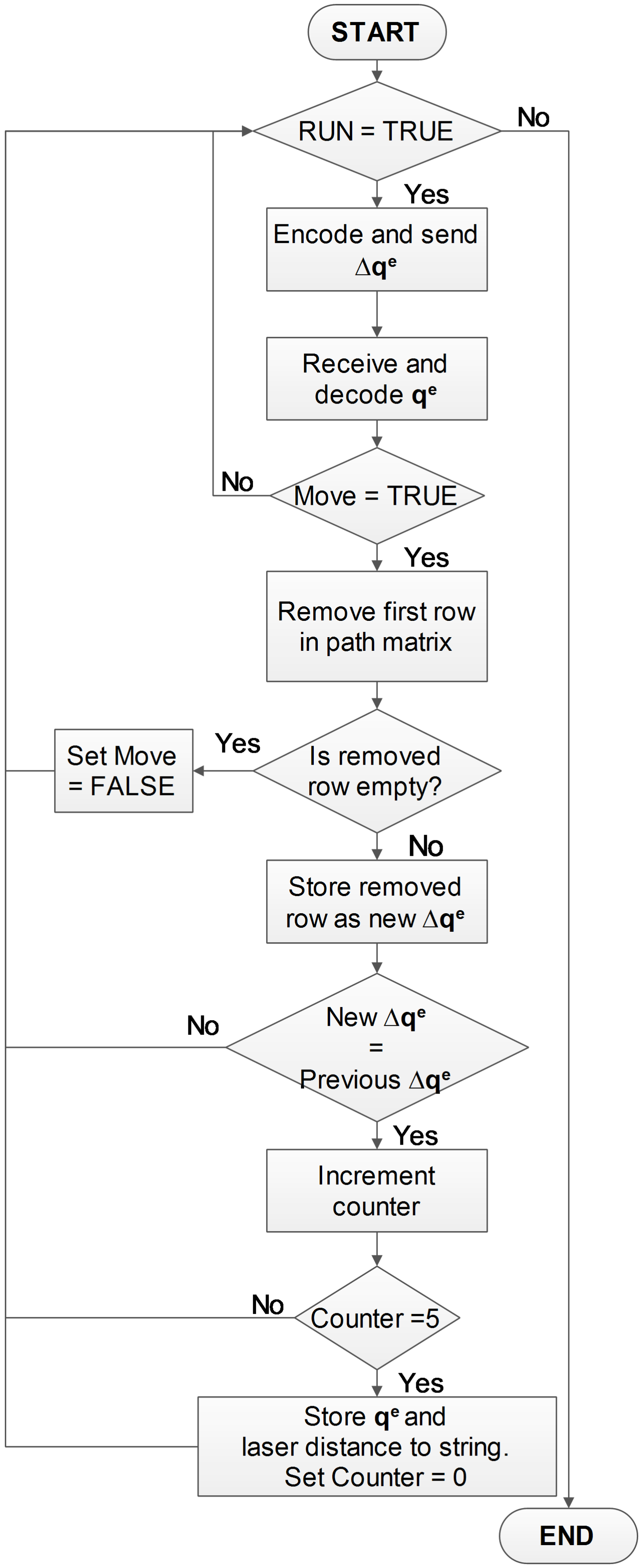

START
RUN = TRUE
No
Yes
Encode and send $\Delta \mathbf{q}^e$
Receive and decode $\mathbf{q}^e$
No
Move = TRUE
Yes
Remove first row in path matrix
Set Move = FALSE
Yes
Is removed row empty?
No
Store removed row as new $\Delta \mathbf{q}^e$
No
New $\Delta \mathbf{q}^e$ = Previous $\Delta \mathbf{q}^e$
Yes
Increment counter
No
Counter =5
Yes
Store $\mathbf{q}^e$ and laser distance to string. Set Counter = 0
END

Figure 43 Flowchart showing the main structure of the Sampling application

7.1.2 Path generation

In path generation the path matrix is created, based upon the desired movement chosen by the operator and from the path matrix the $\Delta\mathbf{q}^{\mathrm{e}}$ is extracted. In the sampling application the operator is given three different path alternatives:

- Move robot to home position
- Move robot to calibration position
- Perform a specified number of samples

The path generated is stored in a path matrix where each row represents new, small joint change ($\Delta\overline{\boldsymbol{q}}^{\boldsymbol{e}}$) and each column represents a joint. The matrix is generated in the step generator VI.

Step generator

The step generator is a simple trajectory interpolator in joint space. It takes in the current joint positions and the commanded joint positions in encoder values and interpolates a set of VIA points in joint space. The logic behind this is that the RT system can only handle a maximum joint change ($\Delta\overline{\boldsymbol{q}}^{\boldsymbol{e}}$) of maximum 60 encoder steps every 10 ms (100 Hz). If this limit is exceeded the robot will get an encoder bit jump error where it applies the breaks, thus stopping the robot motion and displaying an error message on the teach pendant. To avoid this problem the interpolation creates a set of VIA-points that controls that each $\Delta\overline{\boldsymbol{q}}^{\boldsymbol{e}}$ does not exceed a change larger than 51 steps (51 in case of round-ups), thus having a safety margin (as the system can handle 60 steps).

<u>A simple example:</u>

Assuming that the current $\Delta\overline{\boldsymbol{q}}^{\boldsymbol{e}}$ = [0,0,0,0,0,0] and that the current joint position is the *Home* position

$$\text{Current position: } \begin{bmatrix} 8388608 \\ 8388608 \\ 8388608 \\ 8388608 \\ 8388608 \\ 8388608 \end{bmatrix} \quad \text{Commanded position: } \begin{bmatrix} 8388908 \\ 8388908 \\ 8388908 \\ 8388308 \\ 8388308 \\ 8388608 \end{bmatrix}$$

This will result is the path matrix: $\begin{bmatrix} 50 & 50 & 50 & -50 & -50 & 0 \\ 100 & 100 & 100 & -100 & -100 & 0 \\ 150 & 150 & 150 & -150 & -150 & 0 \\ 200 & 200 & 200 & -200 & -200 & 0 \\ 250 & 250 & 250 & -250 & -250 & 0 \\ 300 & 300 & 300 & -300 & -300 & 0 \end{bmatrix}$

The path matrix show how the joint changes are accumulating with a maximum step size of 50 encoder steps per joint change. This is a simple example where all the joints, except $q_{6,}$ changes with the same value in each step, this rarely occur in normal operations.

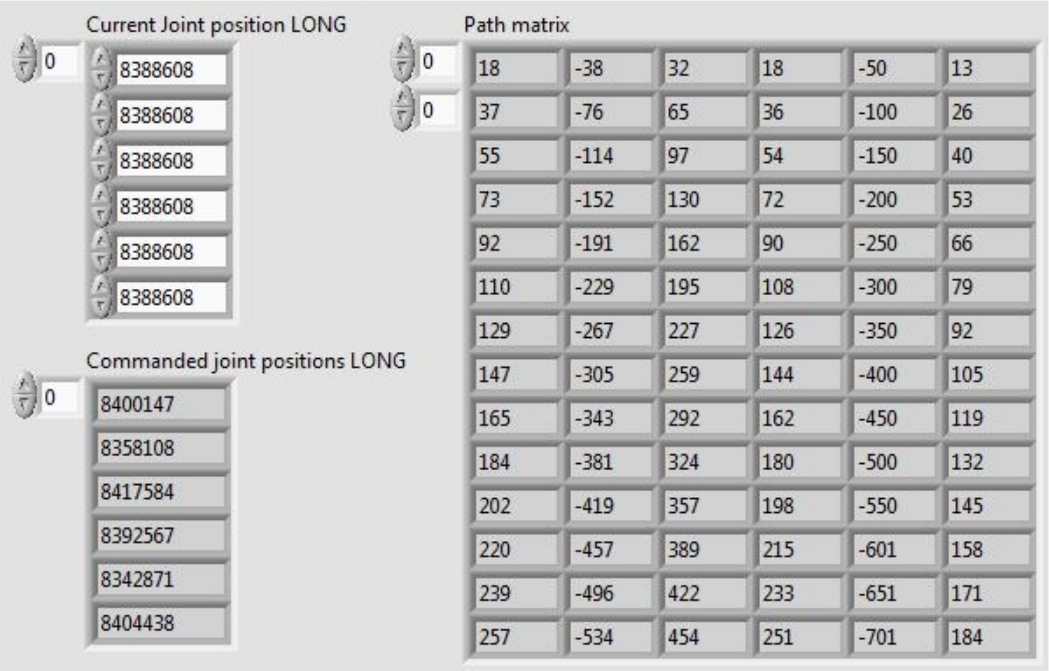
Normally, the joint movement required by each joint when generating a path between two points varies extensively, therefore some joints will have minimal change, while others will produce a large change. As of this, the step generator will make sure the joint with the largest change will govern the number of via-points for a path. The number of via-points is calculated by the following formula (rounded down to closest integer):

$$No.\ of\ via\text{-}points = max\left\{\frac{Commanded\ joint\ position\ [Long]-\ Current\ joint\ position\ \ [Long]}{50\ \ [Long]}\right\} \quad (7.1)$$

Further the size of each step, per joint, is calculated by:

$$Interpolation\ step\ size = \frac{Commanded\ joint\ position\ (Long)-\ Current\ joint\ position\ (Long)}{No.\ of\ via-points} \quad (7.2)$$

A for loop is running a number of iterations given by the number via-points calculated by formula 7.1, in each iteration a counter is multiplied with the interpolation step size (formula 7.2) and added to the accumulated delta values array and finally stored to the Path matrix. When the for-loop is finished the final movement of the sample is copied and stored to the Accumulated delta movement array. Figure 44, shows a more complex example of a path generated from the robots home position, in this case the accumulated $\Delta\bar{\boldsymbol{q}}^{\boldsymbol{e}}$ is zero, otherwise its values would have been had added to the path matrix. The path matrix also shows that q_5 performs the most movement in the path and therefore has the biggest change in every movement.



Current Joint position LONG

0

8388608
8388608
8388608
8388608
8388608
8388608

Commanded joint positions LONG

0

8400147
8358108
8417584
8392567
8342871
8404438

Path matrix

0
0

18	-38	32	18	-50	13
37	-76	65	36	-100	26
55	-114	97	54	-150	40
73	-152	130	72	-200	53
92	-191	162	90	-250	66
110	-229	195	108	-300	79
129	-267	227	126	-350	92
147	-305	259	144	-400	105
165	-343	292	162	-450	119
184	-381	324	180	-500	132
202	-419	357	198	-550	145
220	-457	389	215	-601	158
239	-496	422	233	-651	171
257	-534	454	251	-701	184

Figure 44 LabVIEW screenshot showing the path matrix

As mentioned the operator is given three path creation options, home, calibration position and sampling. Whereas the home and calibration position uses hardcoded encoder values that is fed to the step generator as the commanded position. For the sampling operation the creation of commanded positions is more complicated.

Sample generator

The sample generator consists of a for-loop performing a number of iterations based on the specified number of samples. In every iteration, a random pose is generated within a threshold specified in the software. As the generated pose is defined in x, y, z, roll, pitch, yaw, inverse kinematics is used to calculate the joint angles from each pose and feeding it into the step generator, where the path matrix for each sample is outputted and concatenated into a total path matrix. In between each sample path matrix the last movement in each sample path matrix is placed 5 times, see figure 45.

This means that after each sample the robot will send exactly the same $\Delta\overline{\boldsymbol{q}}^{\boldsymbol{e}}$ five times subsequently. As can be seen in the flowchart, figure 43, this is used as a sign to perform a laser measurement for the sample and save it to the sample file with the corresponding encoder data. With a loop running at 100 Hz this correspond to a movement pause of 0.5

seconds, enough to get a stable encoder position. Next the random pose generation is described.

$$[\text{Path matrix}]_{\text{Sampling}} = \begin{bmatrix} \begin{bmatrix} \text{Path matrix: Sample } 1 \end{bmatrix} \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } 1] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } 1] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } 1] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } 1] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } 1] \\ \vdots \\ \vdots \\ \begin{bmatrix} \text{Path matrix: Sample } n \end{bmatrix} \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } n] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } n] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } n] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } n] \\ [\text{Final } \Delta\bar{\boldsymbol{q}}^{e} \text{ of sample } n] \end{bmatrix}$$

Figure 45 Path matrix structure in a sampling operation

Random sensor pose generation

The purpose of the random pose generation is to create a random sensor pose, resulting in a random laser measurement against the calibration plane, within the sensor's measurement range. An important concern when creating random poses is to try to achieve maximum joint change between the random poses, in order to get good identifiability of the joints for the optimization procedure.

The random sensor pose is calculated from a random point (X_{P}, Y_{P}) on the calibration plane, a random sensor orientation ($\hat{\mathbf{z}}_{\boldsymbol{Random}}$), based on a known sensor orientation perpendicular to the plane ($\hat{\mathbf{z}}_{\boldsymbol{Perpendicular}}$) and a random distance measurement (d_m). In the sampling application the operator is also able to set a fixed distance measurement (see section 7.3.1). The geometric relationships between these parameters are shown in figure 46.

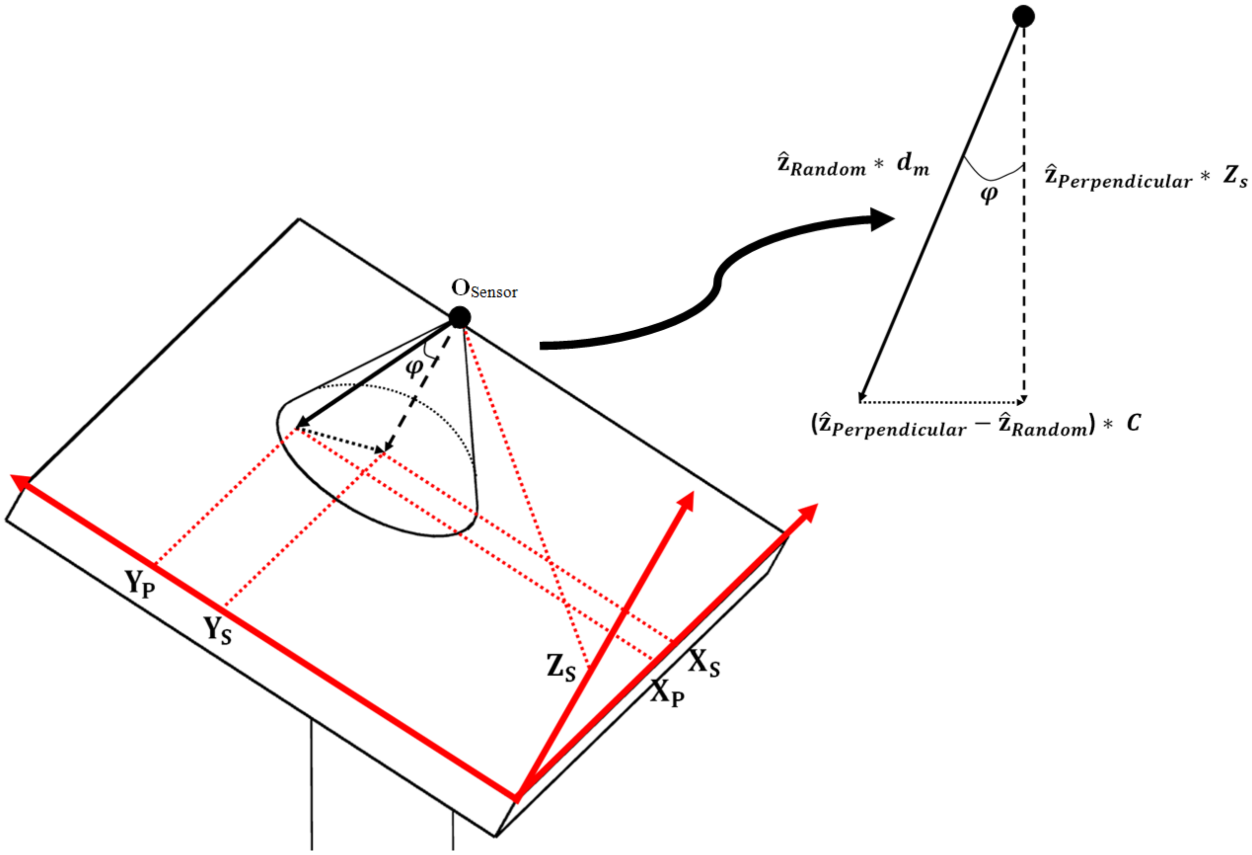
Note that the following calculations are performed in a coordinate frame on the plane (indicated in figure 46), thus $\hat{\mathbf{z}}_{\boldsymbol{Random}}$ and $\hat{\mathbf{z}}_{\boldsymbol{Perpendicular}}$ need to be defined in the plane coordinate system, as they are unit vectors it is easily achieved by a homogenous transform.

$\hat{\mathbf{z}}_{\boldsymbol{Perpendicular}}$ is determined by jogging the robot until the sensor housing is parallel with the calibration plane, and reading out the roll, pitch and yaw of the sensor defined in the base coordinate system. From $(Roll, Pitch, Yaw)_{Perpendicular,}$ $\hat{\mathbf{z}}_{\boldsymbol{Perpendicular}}$ can easily be defined in the plane coordinate system, using the rotational matrices described in appendix B.

When creating random orientations, random values between $\pm\ 30\,°$ are added to the perpendicular roll, pitch, yaw independently;

$$(Roll, Pitch, Yaw)_{Random} = \begin{pmatrix} Roll_{Perp.} + \ Rand(\pm\ 30°) \\ Pitch_{Perp.} + \ Rand(\pm\ 30°) \\ Yaw_{Perp.} + \ Rand(\pm\ 30°) \end{pmatrix}$$

From this, $\hat{\mathbf{z}}_{\boldsymbol{Random}}$ is similarly using the rotational matrices, defined in the plane coordinate system. A 60-degree cone, perpendicular to the calibration plane, as shown in figure 46, can be used to illustrate the various random sensor orientations. Obviously the bigger orientation threshold the better, but due to the short range of the laser sensor a bigger threshold increases the chance for collision., between the robot and the calibration plane.



$\hat{\mathbf{z}}_{\boldsymbol{Random}} * \boldsymbol{d_m}$
φ
$\hat{\mathbf{z}}_{\boldsymbol{Perpendicular}} * \boldsymbol{Z_s}$
$(\hat{\mathbf{z}}_{\boldsymbol{Perpendicular}} - \hat{\mathbf{z}}_{\boldsymbol{Random}}) * \boldsymbol{C}$
O_{Sensor}
φ
Y_P
Y_S
Z_S
X_S
X_P

Figure 46 Geometric relations in calculating the random poses for sampling

After generating the random values of the laser point (X_P, Y_P), distance measure (d_m) and orientation $\hat{\mathbf{z}}_{\boldsymbol{Random}}$ the sensor origin (O_S) can be calculated. First the angle between random orientation and the perpendicular orientation is calculated by the following formula:

$$\boldsymbol{cos(\varphi)} = \frac{{}^{Random}\hat{\mathbf{z}}_{Sensor} \cdot {}^{Perpendicular}\hat{\mathbf{z}}_{Sensor}}{\left\|{}^{Random}\hat{\mathbf{z}}_{Sensor}\right\| \times \left\|{}^{Perpendicular}\hat{\mathbf{z}}_{Sensor}\right\|}$$

Thus,

$$Z_S = d_m * \cos(\varphi)$$

Similarly

$$C = d_m * \sin(\varphi)$$

From figure 46 it can be seen that:

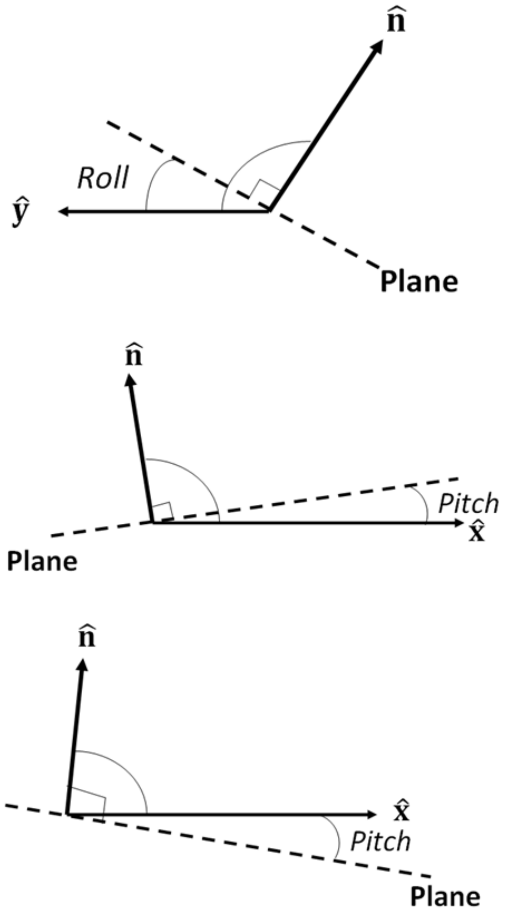
$$\boldsymbol{(\hat{z}_{Perpendicular} - \hat{z}_{Random}) * C} = \begin{bmatrix} \mathbf{X_P - X_S} \\ \mathbf{Y_P - Y_S} \\ \mathbf{0} \end{bmatrix}$$

And from this, X_S and Y_S are calculated.

Finally the sensor origin is transformed to base coordinates and defined together with $(Roll, Pitch, Yaw)_{Random}$ and the random pose is generated. The next section describes how the plane and base transformation are calculated.

Calculating the plane to base transformation

The plane to base transformation, needed to express the sensor origin position defined in the plane coordinate frame to the base coordinate frame. The homogenous transformation matrix is calculated from the plane normal and origin of the plane coordinate system. The origin of the coordinate frame must be determined through measurement, therefore using the same position that was measured during the initial definition of the plane in section 6.4.2. The orientation of the plane is calculated based on the normalized plane vector and the idea behind this solution is to be able to apply changes in the model parameter to the sampling operation to perform “live” validation samples. The orientation of the plane is calculated by the formulas shown in figure 47. Since the plane is oriented with its length and width parallel to the base X- and Y-axis we can set the Yaw = to zero.



$$\text{Roll} = Angle\ (\hat{\mathbf{n}}\,\hat{\mathbf{y}}_{\boldsymbol{Base}}) - \frac{\pi}{2}$$

$$\text{If } Angle\ (\hat{\mathbf{n}}\,\hat{\mathbf{x}}_{\boldsymbol{Base}}) > \frac{\pi}{2}$$

$$\text{Pitch} = Angle\ (\hat{\mathbf{n}}\,\hat{\mathbf{x}}_{\boldsymbol{Base}}) - \frac{\pi}{2}$$

$$\text{If } Angle\ (\hat{\mathbf{n}}\,\hat{\mathbf{x}}_{\boldsymbol{Base}}) < \frac{\pi}{2}$$

$$\text{Pitch} = -(\frac{\pi}{2} - Angle\ (\hat{\mathbf{n}}\,\hat{\mathbf{x}}_{\boldsymbol{Base}}))$$

Figure 47 Calculating plane orientation from unit normal

7.1.3 Laser read

The laser sensor is connected with to a COM-port in the computer running the sampling application. The laser measurement is sent after every time the computer sends the text string *"M",* thus the computer receives a package with a 12 byte count, which is decoded to the laser measurement in mm by the formula:

$$Laser\ ditance\ [mm] = (Laser\ output \times 10^{-6}) +\ 80$$

The LabVIEW program code is shown in figure 49.

If the laser measurement is out of range (either too close or too far) the returned laser distance is 0. If the laser distance is not zero it is stored in a 100 element circular buffer following the FIFO (First In, First Out) queue system.

The Read laser function is placed inside a while loop that is terminated if the spread of the circular buffer becomes less than 0,005 mm or the laser gets 400 out of range reads in a row, see flow chart in figure 48. The loop reading the laser sensor is running with a speed of 100 Hz. Thus, the distance measurement is assumed to be stable when a maximum sensor movement of 0,005 mm per second is registered; this is down one the sensor noise level and therefore a valid assumption.

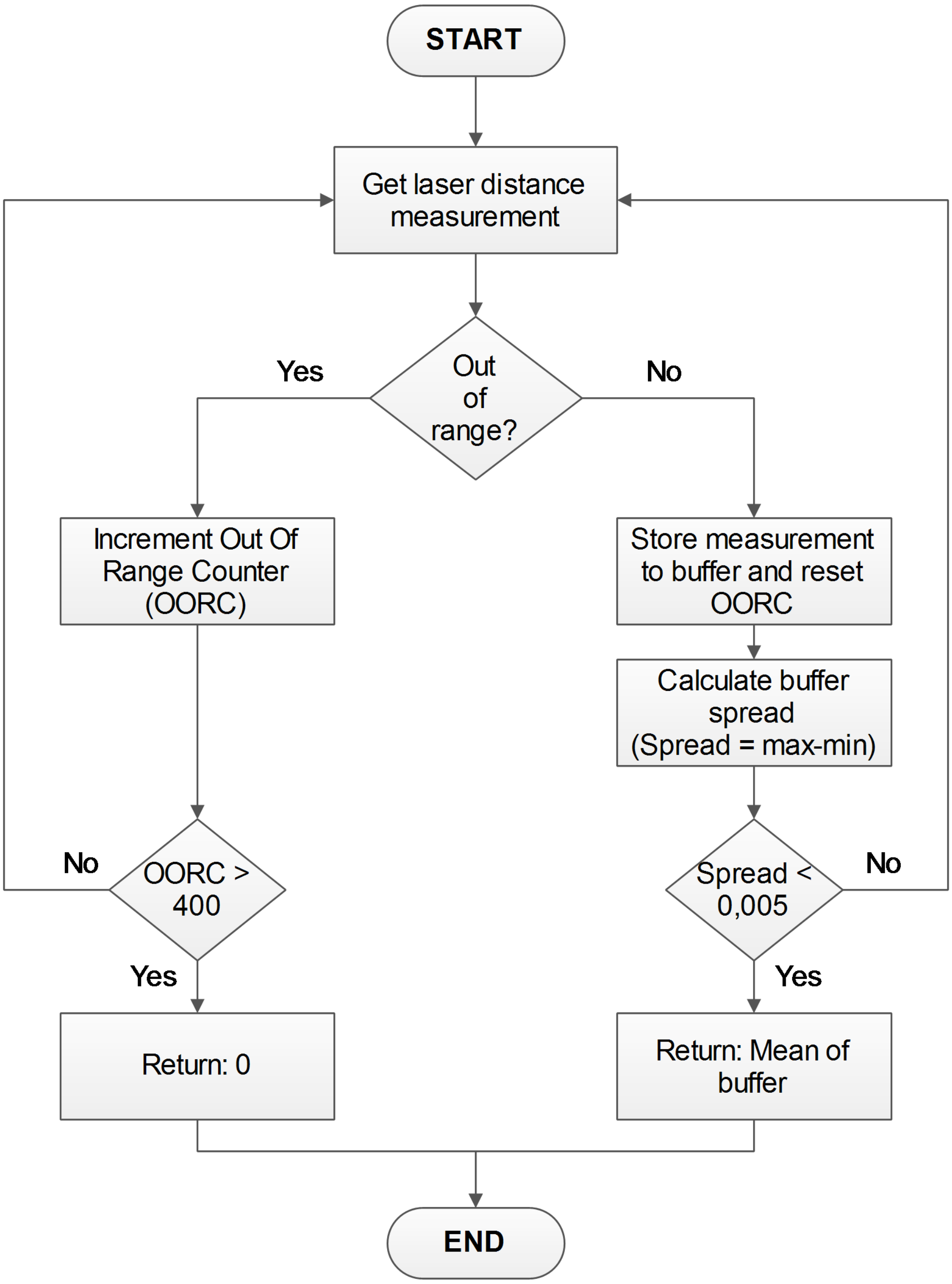

START
Get laser distance measurement
Out of range?
Yes
No
Increment Out Of Range Counter (OORC)
Store measurement to buffer and reset OORC
Calculate buffer spread (Spread = max-min)
No
OORC > 400
Yes
Spread < 0,005
No
Yes
Return: 0
Return: Mean of buffer
END

Figure 48 Flowchart describing the laser measurement of the sampling operation

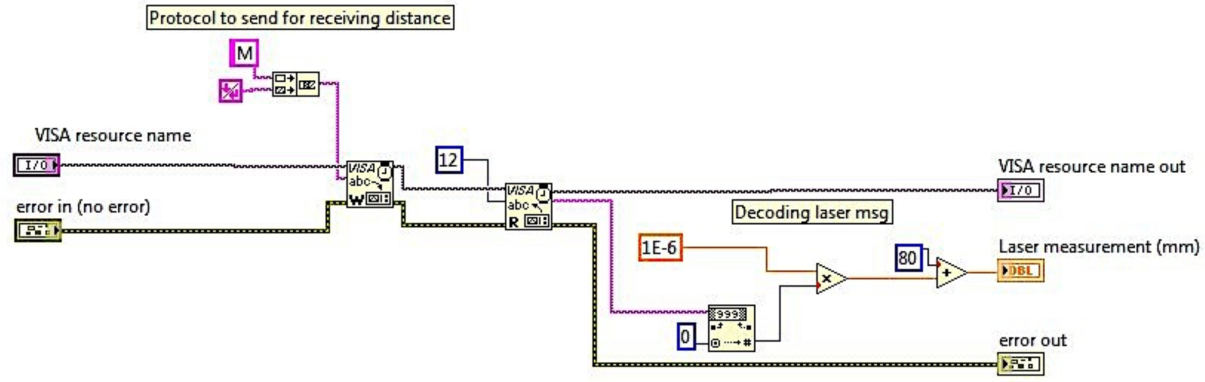
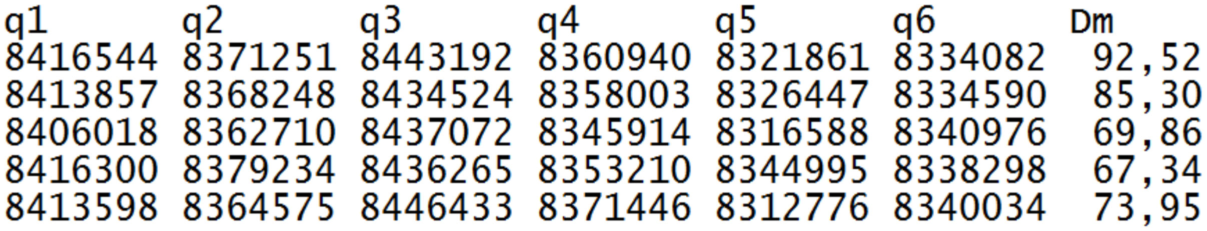


Figure 49 LabVIEW code for reading the laser sensor

7.1.4 Sample data file

As mentioned the output of the sampling application is a text file with the encoder positions and the laser distance measurement in mm, below is an example of a sample data set of 5 samples:



```
q1      q2      q3      q4      q5      q6      Dm
8416544 8371251 8443192 8360940 8321861 8334082  92,52
8413857 8368248 8434524 8358003 8326447 8334590  85,30
8406018 8362710 8437072 8345914 8316588 8340976  69,86
8416300 8379234 8436265 8353210 8344995 8338298  67,34
8413598 8364575 8446433 8371446 8312776 8340034  73,95
```

7.2 Optimization application

In the optimization application, a new set of model parameters ($\overline{\boldsymbol{\alpha}}$) is calculated from a data set, as shown in the previous section, gathered in the sampling application. The following presentation of the Optimization VI will be much briefer than for the Sampling VI as this basically is a straightforward implementation of the calibration principle presented in chapter 6, therefore the rest of this sub-chapter is used to briefly describe the sub-processes of the flowchart, in figure 49.

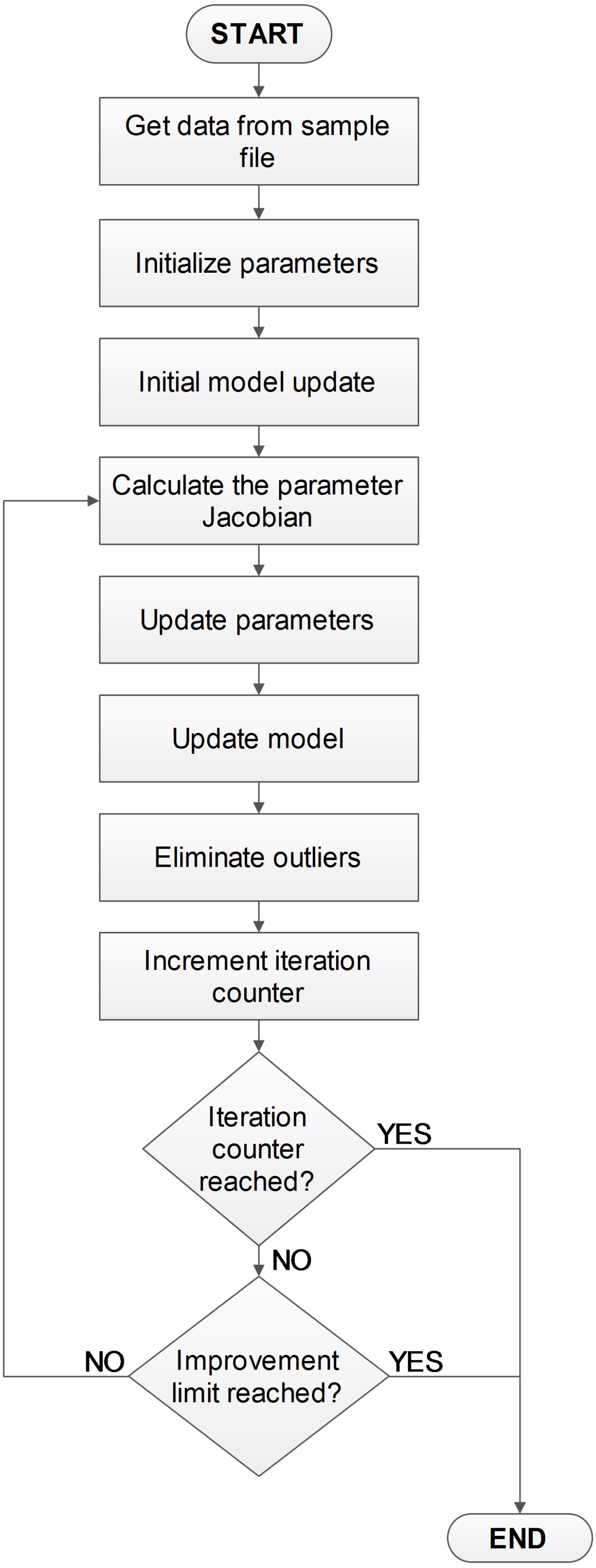

START
Get data from sample file
Initialize parameters
Initial model update
Calculate the parameter Jacobian
Update parameters
Update model
Eliminate outliers
Increment iteration counter
Iteration counter reached?
YES
NO
NO
Improvement limit reached?
YES
END

Figure 50 Flowchart covering the Optimization application

Get data from sample file

A sample file is opened and its contents a stored to one matrix containing all the encoder values (Herby referred to as the *encoder matrix*) and an array containing all the samples distance measurements (herby referred to as the *measurement array*).

Initialize parameters

The initial parameters array is initialized based on the values found in section 6.3

Initial model update

The computed distance is calculated from the values in the *initial* parameter array and the encoder matrix, using total system model, formula 6.14 in section 6.1.3. Finally the model error is calculated as the difference between actual measurements and computed measurements, and stored in the model error array.

Calculate the parameter Jacobian

Based on the formulas presented in section 6.2.2, the parameter array, the encoder matrix and computed distance array, the parameter Jacobian is computed.

Update parameters

Using the parameter Jacobian calculated in the previous step, the model error array and formula 6.17 in section 6.2.1, the parameter correction is calculated and added to the current parameter array.

Model update

The computed distance is calculated from the values in the *updated* parameter array and the encoder matrix, formula 6.14 in section 6.1.3. Finally the model error is calculated as the difference between actual measurements and computed measurements, and stored in the model error array.

Eliminate outliers

This function allows the user to eliminate those measurements from the sample file that generates a model error that is outside a predefined number of standard deviation of the mean of the error model array, so-called outliers (see figure 51). Measurements that are found to be outliers are deleted from the encoder matrix and measurement array, hence no longer a part of future iterations.

End condition

The optimization ends when a predefined number of iterations are completed or if the improvement limit is reached. The improvement is calculated as the absolute mean of the parameter correction, in every iteration. With the optimization finished, the new model parameter array is given.

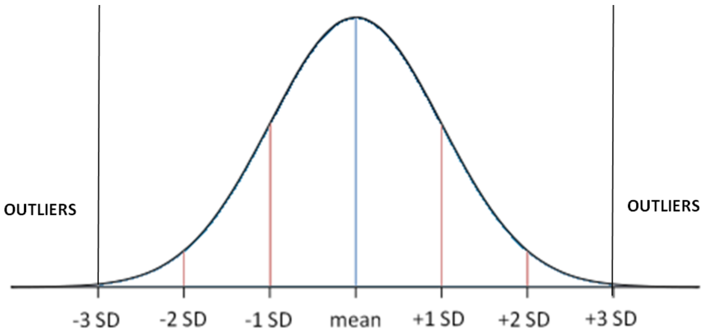


Figure 51 Outliers in a sample set

7.3 User guides

7.3.1 Sampling application

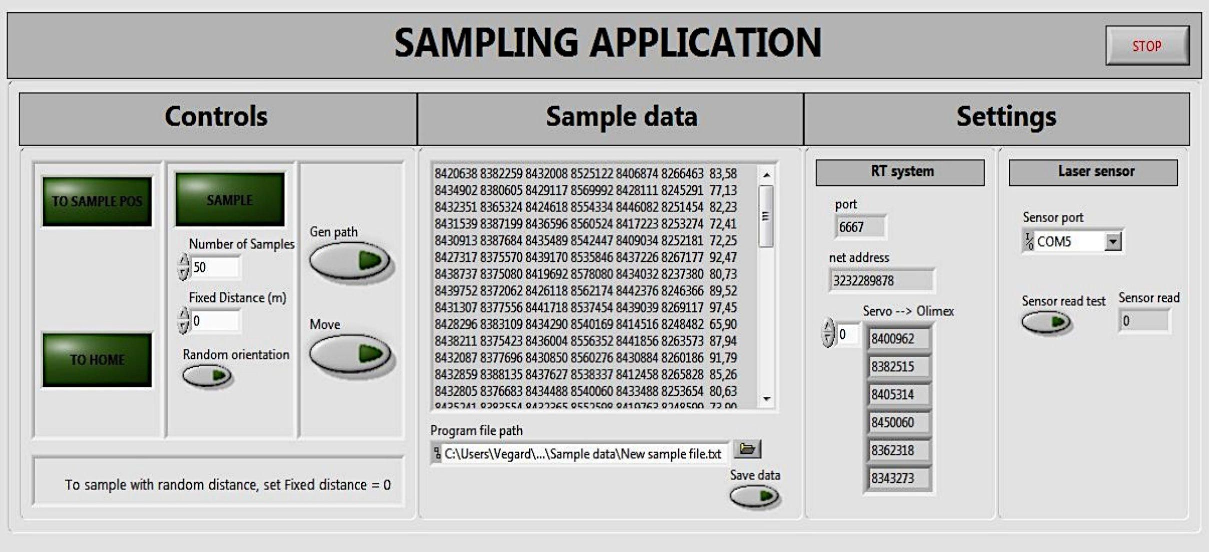
Start up process

1. Turn on both the RT-system computers. Wait approx. 1-2 minutes to boot up.
2. Turn on the FD controller.

If the boot-up is successful, no error message will be displayed on the teach pendant. If a CPU servo communication occurs on the teach pendant, turn of the controller then check the connections between controller and RT-system, before restarting the start-up process.

3. Open the Calibration folder and launch Main – Sampling application.VI

A window similar to figure 52, will appear.



Number of Samples: 50

Fixed Distance (m): 0

To sample with random distance, set Fixed distance = 0

```
8420638 8382259 8432008 8525122 8406874 8266463  83,58
8434902 8380605 8429117 8569992 8428111 8245291  77,13
8432351 8365324 8424618 8554334 8446082 8251454  82,23
8431539 8387199 8436596 8560524 8417223 8253274  72,41
8430913 8387684 8435489 8542447 8409034 8252181  72,25
8427317 8375570 8439170 8535846 8437226 8267177  92,47
8438737 8375080 8419692 8578080 8434032 8237380  80,73
8439752 8372062 8426118 8562174 8442376 8246366  89,52
8431307 8377556 8441718 8537454 8439039 8269117  97,45
8428296 8383109 8434290 8540169 8414516 8248482  65,90
8438211 8375423 8436004 8556352 8441856 8263573  87,94
8432087 8377696 8430850 8560276 8430884 8260186  91,79
8432859 8388135 8437627 8538337 8412458 8265828  85,26
8432805 8376683 8434488 8540060 8433488 8253654  80,63
```

Program file path: C:\Users\Vegard\...\Sample data\New sample file.txt

port: 6667

net address: 3232289878

Servo --> Olimex: 0

8400962
8382515
8405314
8450060
8362318
8343273

Sensor port: COM5

Sensor read: 0

Figure 52 Screenshot of the front panel in the MAIN - Sampling application.VI

Sampling Application

To perform a sampling operation

1. Highlight “TO SAMPLE POS” and click the “Gen. path”-button to create the path matrix.

2. With the path generated, ensure that the robot is in tech-mode, press and hold the enabling device on the teach-pendant and click the “Move”- button. The robot will only move when the dead-man-switch (DMS) is pressed.

3. When the calibration position is reached, highlight “SAMPLING” and set the number of measurement to be performed. Here, the user is given two options:

   Fixed distance: By setting the fixed distance the robot will limit the measurement distance to the fixed value, by setting the fixed distance to zero, samples are generated with random distance.

   Random orientation: By clicking the “Random orientation” button the sampling operation will be with random orientations of the laser, if not clicked the laser will always stay perpendicular to the plane.

4. Click the “Gen. path”, press and hold the DMS on the teach pendant and finally click the “Move”- button.

5. As the samples are taken they appear in the sample data window. When the sampling operation is finished release the DMS, save the sample data file to a desired file location.

   IMPORTANT: During the sampling operation the operator must ALWAYS watch the robot, since the path is interpolated in joint space, there is a risk for collision between the robot and calibration between certain poses. The robot motion will stop as soon as the operator releases the dead-man-switch on the teach pendant.

6. When finished the user can move the robot back to home position, by highlighting “TO HOME”. Click the “Gen. path”-button”, press and hold the DMS and finally click the “Move”- button.

In the settings the user can see the current encoder position of the robot, and the port and IP settings of the RT system. In this section the COM port of the Laser sensor can be set, and a laser read test can be performed, in order to test the communication.

7.3.2 Optimization application

Start by opening the Calibration folder and launch Main – Optimization application.VI and a window similar to figure 53, will appear.

To perform an optimization:

1. Click on the folder icon to open the sample data file

2. Set the optimization controls

   Eliminate threshold: The number of standard deviations that all model errors shall lie within.

   Improvement limit: If the mean correction in an iteration is lower than the improvement limit, the optimization is terminated.

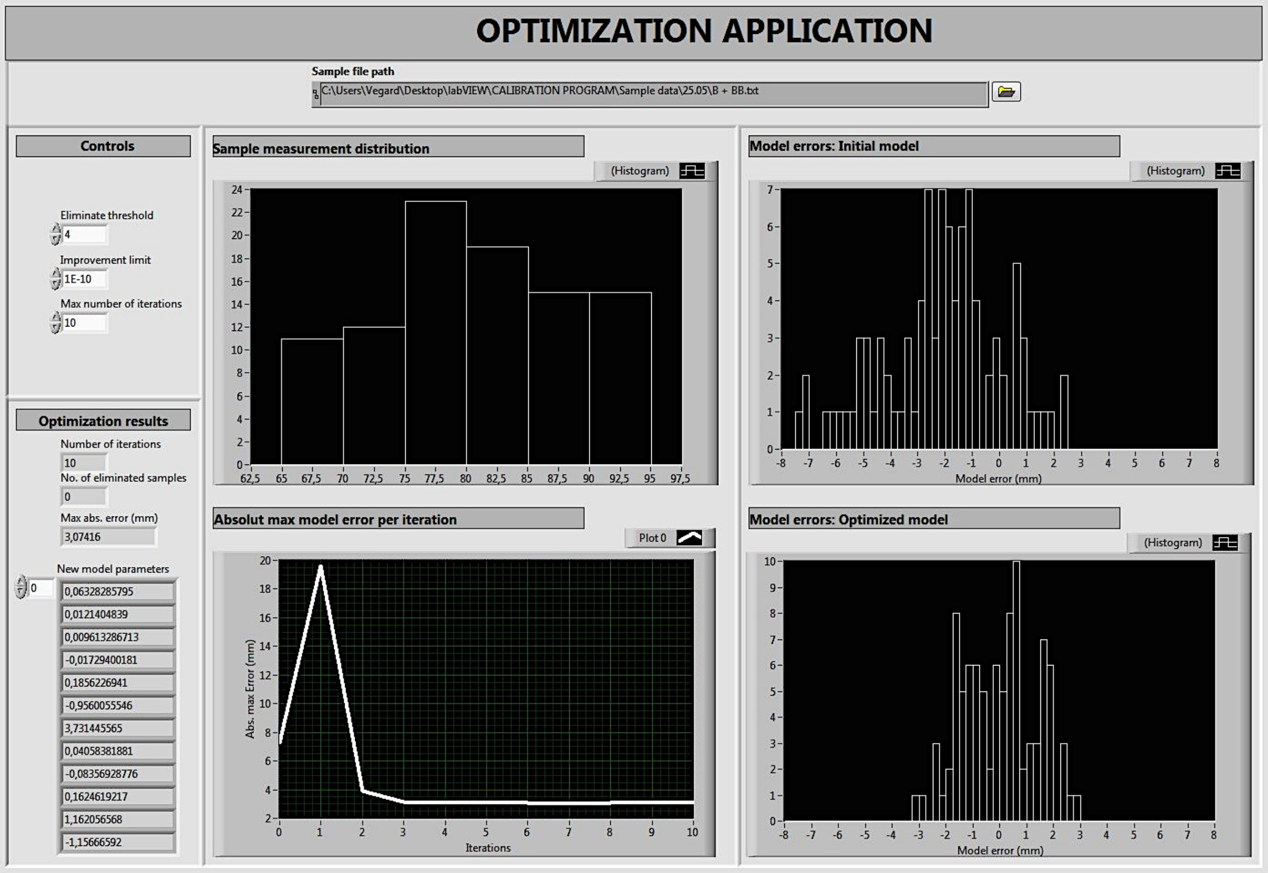
   Max no. of iterations: Sets the maximum number of iterations.

3. Run the application

4. View the results

   Optimization results: Displays the new model parameters and the maximum absolute model error with the new parameters. The number of completed iterations and eliminated outliers are also displayed.

   Graphs:
   - Histogram shown the distance distribution of the dataset
   - Graph showing the convergence of the absolute max. model error per iteration.
   - Histogram showing the model error distribution of the initial model parameters.
   - Histogram showing the model error distribution of the new model parameters.



OPTIMIZATION APPLICATION

Sample file path

C:\Users\Vegard\Desktop\labVIEW\CALIBRATION PROGRAM\Sample data\25.05\B + BB.txt

Controls

Eliminate threshold: 4

Improvement limit: 1E-10

Max number of iterations: 10

Optimization results

Number of iterations: 10

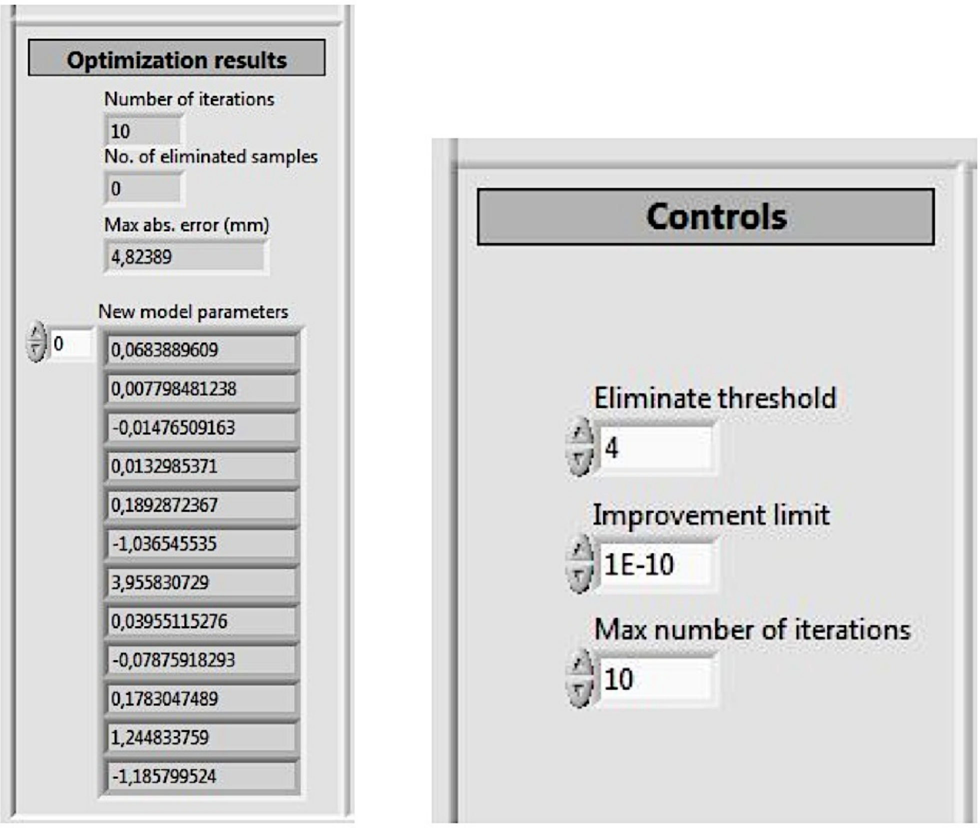
No. of eliminated samples: 0

Max abs. error (mm): 3,07416

New model parameters (index 0)

0,06328285795
0,0121404839
0,009613286713
-0,01729400181
0,1856226941
-0,9560055546
3,731445565
0,04058381881
-0,08356928776
0,1624619217
1,162056568
-1,15666592

Figure 53 Screenshot of the front panel in the MAIN - Optimization application.VI



Optimization results

Number of iterations: 10

No. of eliminated samples: 0

Max abs. error (mm): 4,82389

New model parameters (index 0)

0,0683889609
0,007798481238
-0,01476509163
0,0132985371
0,1892872367
-1,036545535
3,955830729
0,03955115276
-0,07875918293
0,1783047489
1,244833759
-1,185799524

Controls

Eliminate threshold: 4

Improvement limit: 1E-10

Max number of iterations: 10

Figure 54 Details from the front panel screenshot

Chapter 8

Tests and results

This chapter presents some of the tests that have been performed through the course of the implementation of the calibration procedure. They concern tests of the different models, such as the encoder model and the total system model, but also test of the final optimization and sampling application.

8.1 Encoder model verification

In order to verify if the encoder model corresponded to the encoder model used on the NACHI FD11-controller. A simple robot program was generated turning all the joints 10 degrees, keeping all the other joints in the zero-reference position. This was the test disclosing the need of the coupling matrix, described in section 6.4.1.

A simple LabVIEW application utilizing the RT-system was generated to read the encoder position after each joint rotation and finally the encoder model calculated the joint position. Two encoder models were used, one with the coupling matrix and one without. Finally the model readouts were compared to the joint positions displayed on the teach pendant (TP). The result is presented below, where table 6 shows the output on the TP, and table 7 and 8, show the encoder model output without and with the coupling matrix, respectively.

Table 6 Teach pendant readout (Unit: Degrees)

	q1	q2	q3	q4	q5	q6
q1	10	0	0	0	0	0
q2	0	10	0	0	0	0
q3	0	0	10	0	0	0
q4	0	0	0	10	0	0
q5	0	0	0	0	10	0
q6	0	0	0	0	0	10

Table 7 Encoder model without coupling readout (Unit: Degrees)

	q1	q2	q3	q4	q5	q6
q1	9,999661127	0	0	0	0	0
q2	0	10,00142912	0	0	0	0
q3	0	0	9,999019289	0	0	0
q4	0	0	0	10,0026304	0	0
q5	0	0	0	0,1117503448	9,999999604	0
q6	0	0	0	-0,170466859	0,170466859	9,999170161

Table 8 Encoder model with coupling readout (Unit: Degrees))

	q1	q2	q3	q4	q5	q6
q1	9,999661127	0	0	0	0	0
q2	0	10,00142912	0	0	0	0
q3	0	0	9,999019289	0	0	0
q4	0	0	0	10,0026304	0	0
q5	0	0	0	0,000636422119	9,999999604	0
q6	0	0	0	0,00098166495	-0,00097166	9,999170161

8.2 Total system model verification

In order to test the mathematical model two sampling operations were performed. The first sample operation where performed with a fixed orientation where the sensor housing remained parallel to the plane with a fixed distance. This orientation is assumed to produce a laser beam, perpendicular to the calibration plane and the idea is that by placing the sensor direction perpendicular to the plane, the sensor orientation and translation error will be less significant and thus reducing the model error. In the second operation the measuring distance was still fixed, but random orientation was set ON, thus is making the errors in the sensor model more prominent. In both experiments 30 random measurements were taken with a fixed measuring distance of 0.08 m. The laser measurement distribution is showed in figure 55 and figure 56 and the results can be summarized as:

Random orientation OFF	**Random orientation ON**
Mean = 72,49 mm	Mean: 71,14 mm
Spread = 2,94 mm	Spread: 7,41 mm

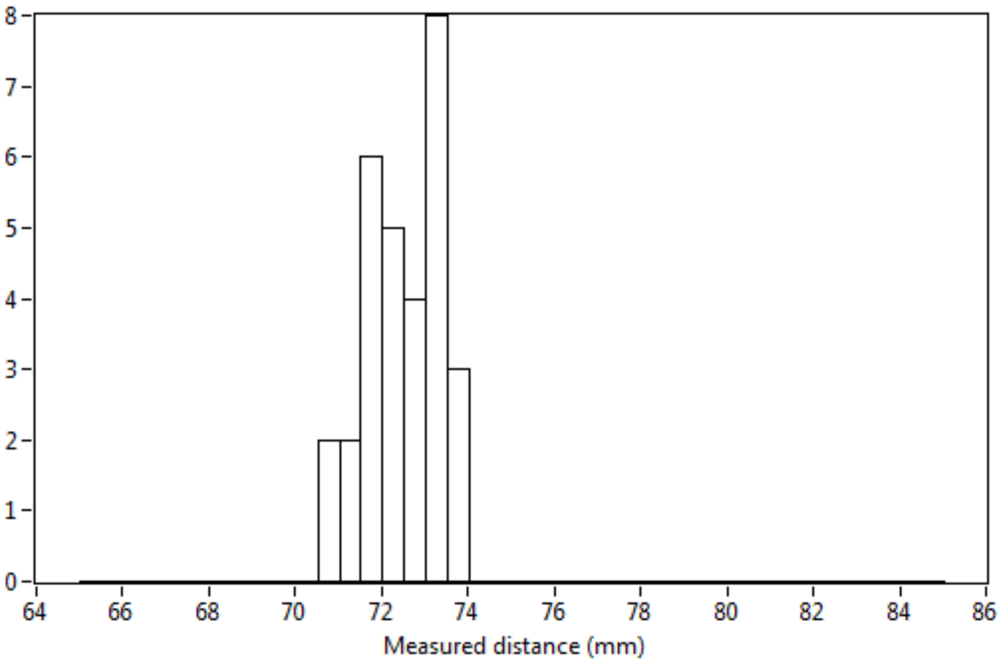


Figure 55 Laser measurement distribution with fixed distance and orientation, with initial model parameters

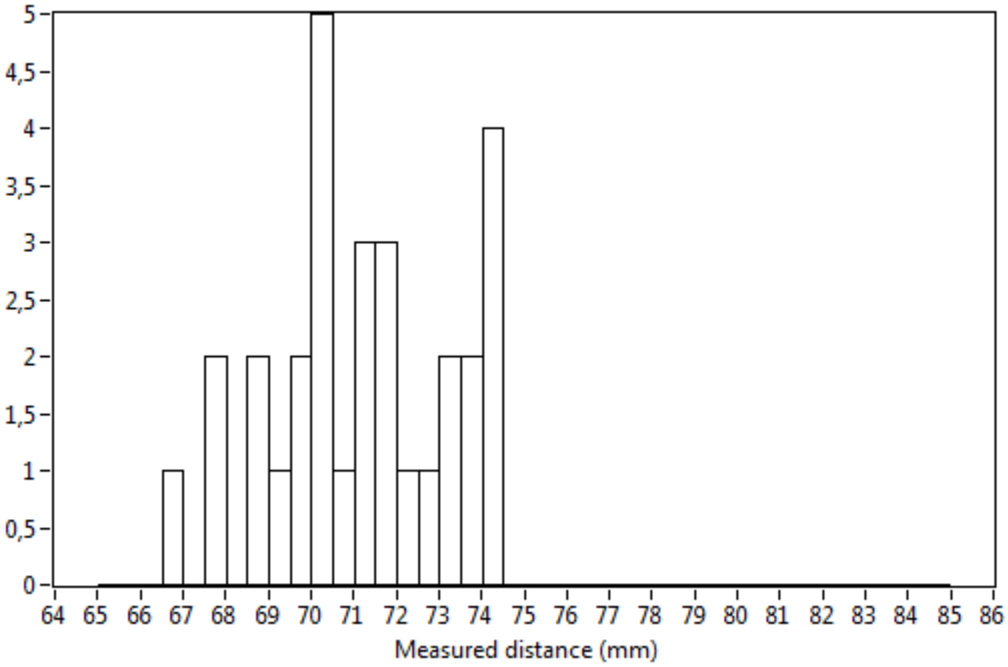


Figure 56 Laser measurement distribution with fixed distance and random orientation with initial model parameters

8.3 Convergence test

In order to check if the optimization application would converge towards similar solutions for different sized sample sets, taken under similar condition, the following data sets where gathered (table 9):

Running the 6 sample sets through the optimization application, with a max number of iterations of 10. Eliminate threshold of 4 standard deviations and an improvement limit of 1E-10, gave the parameter correction of the initial model parameters shown in table 9. No outliers were eliminated and each time 10 iterations where performed, during the optimization of the different sample sets.

Note that the joint offsets parameters are scaled to radians, in all the experiments, as discussed in the calibration consideration sub-chapter 6.3. All the data plots from the different sample sets is found in appendix E.

Table 9 Parameter correction values from different sized sample sets

Model param. correction	**A**	**B**	**BB**	**C**	**CC**	**C+CC**	**A+B+BB+C+ CC**
no. of samples in set	98	49	48	30	32	62	257
q_2	3,2178E-5	0,002158	0,00080	-0,004659	-0,001430	0,000318	0,000375
q_3	-0,001118	-0,000139	-0,001192	0,000422	-0,000225	-0,000604	-0,000826
q_4	0,001187	0,002878	0,002073	-0,001025	0,002593	0,002812	0,002579
q_5	0,000140	0,001323	0,000348	0,000944	-0,000709	-0,000614	7,398883E-5
n_x	0,007798	0,004682	0,011100	-0,008679	0,005414	0,009131	0,009926
n_y	-0,023504	-0,025297	-0,018888	-0,053506	-0,029726	-0,024266	-0,020376
n_z	0,031390	0,025708	0,023987	0,077798	0,037405	0,034130	0,026041
o_x	0,009773	0,009961	0,009308	0,009454	0,009528	0,009697	0,009621
o_y	-0,003985	-0,004341	-0,004127	-0,004390	-0,004144	-0,004034	-0,003973
o_z	0,000342	-0,000733	0,000135	-0,000536	0,000148	-4,4572E-5	-2,77931E-5
z_y	0,013284	0,007146	0,010798	0,002887	0,017734	0,013632	0,011266
z_x	0,001989	0,003159	0,007805	0,008424	0,006931	0,003565	0,003629
Abs. max. model error [mm]	0,147381	0,133476	0,137771	0,102081	0,121123	0,182197	0,188273

Figure 57 shows pictures from the sampling procedure. In the digital attachment, following this thesis, a video of a sampling procedure can be found.

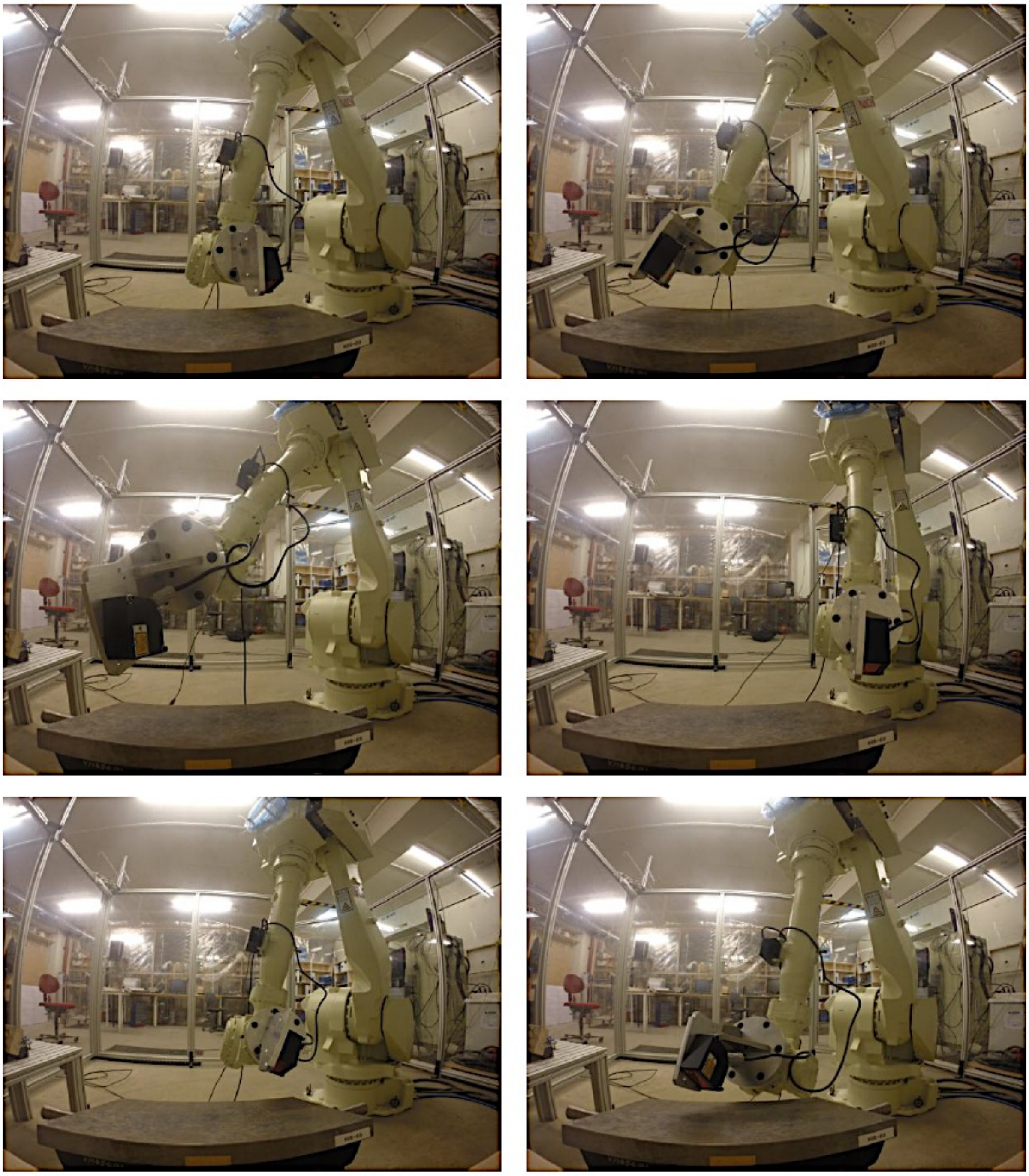


Figure 57 Pictures from a sampling operation at the PPM lab

8.4 Optimization test

In this test, the sample set A, B and C are merged together and from this new dataset, 50 samples are extracted to perform the optimization to obtain the updated model parameters. The updated model parameters are then applied to the remainder of the samples in the data set (herby called the validation set) and from there the model error is calculated. Finally the updated parameters are applied to the sampling application to perform two sampling sets similar to the test described in total system model verification, sub-chapter 8.2.

Figur 58 shows the model errors of the validation set using the initial model parameters.

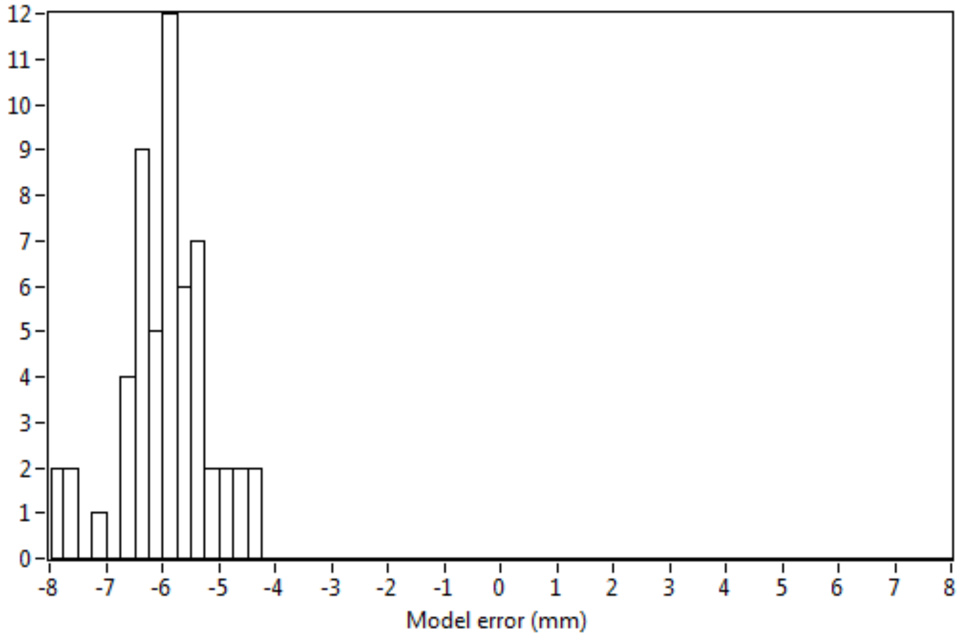


Figure 58 Model error distribution of validation set using initial parameters

The new model parameters calculated from the extracted samples are shown in table 10.

Table 10 Updated parameters obtained from extracted samples

Updated model parameters	
q_2	0,000762
q_3	-0,001237
q_4	0,000731
q_5	4,851769E-5
n_x	-0,017564
n_y	-1,249418
n_z	4,005407
o_x	0,035436
o_y	-0,076901
o_z	0,177291
z_y	1,258098
z_x	-1,041635

The error distribution when applying the updated model parameters to the validation set is shown in figure 59.

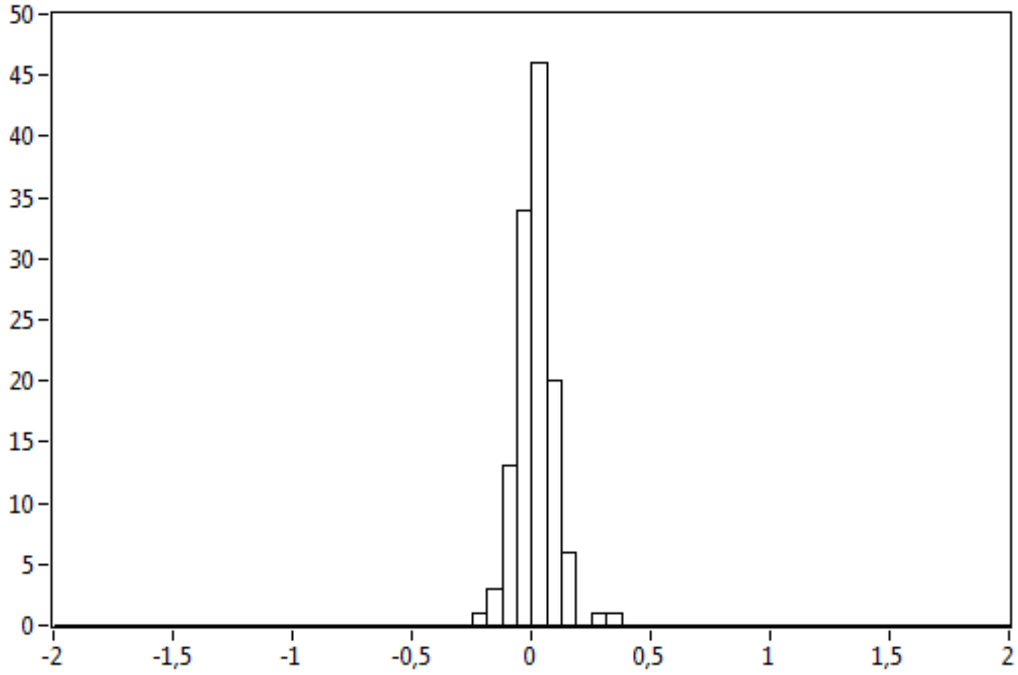


Figure 59 Model error distribution of validation set using updated parameters

Figure 60 and 61 shows the laser measurement distribution when using the updated parameters, with fixed and random orientation respectively. In both experiments 30 random measurements were taken with a fixed measuring distance of 0.08 m. The results can be summarized as:

Random orientation OFF	**Random orientation ON**
Mean = 80,48 mm	Mean: 80,82mm
Spread = 2,45mm	Spread: 3,47mm

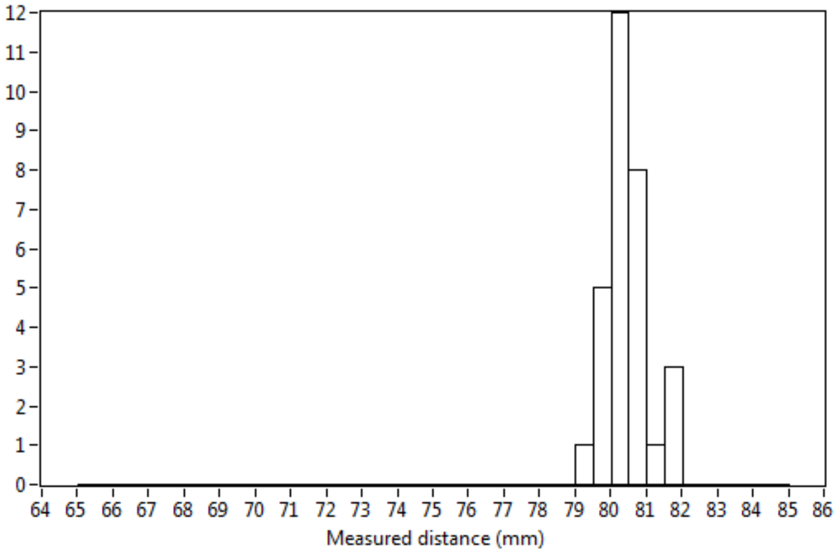


Figure 60 Laser measurement distribution with fixed distance and orientation, with updated model parameters

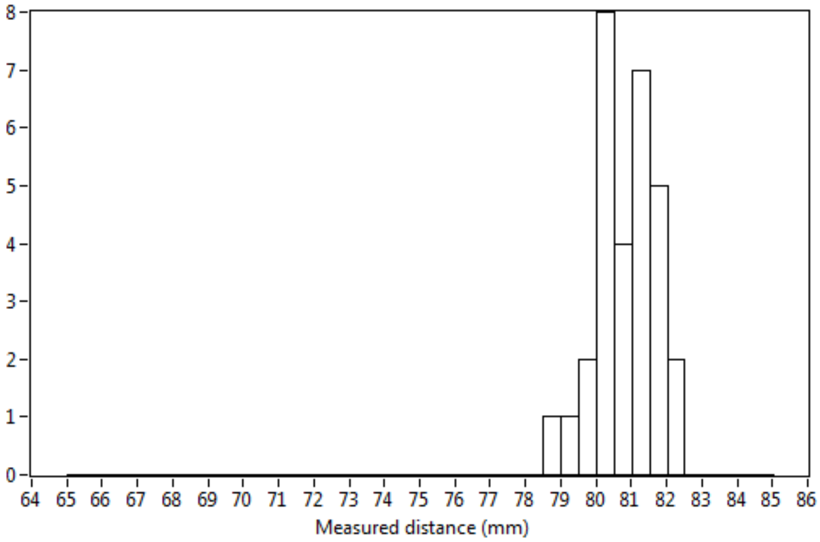


Figure 61 Laser measurement distribution with fixed distance and random orientation with updated model parameters

Chapter 9

Discussion and recommendations for further work

This chapter will first discuss the results seen in chapter 8. Then, the challenges of implementing the calibration procedure will be discussed. Suggestions for the future work with the calibration method will be pointed out. Finally, the calibration method will be discussed.

9.1 Test results

This sub-chapter will systematically discuss the test results obtained in chapter 8.

Encoder model verification

The encoder model verification test shows the difference between using the actuator space definition with the coupling matrix to transform to joint space, and not using it, when changing all the joints 10 degrees. As the coupling matrix is added to the actuator space definition, it is easy to perceive the error magnitude of not using the coupling matrix when the robot is performing the sampling operation, whereas the q_4^e has a much higher value (>70°). As mentioned earlier, this was one of the final bugs corrected in the debugging stage, and figure 62 shows the difference in model error for sample set *A* after correction, with and without the coupling matrix between actuator and joint space. This was also reflected in the updated model parameters that gave odd corrections when not using the coupling matrix. As this improvement is evident for all the sample sets, it points to the importance of investigating the encoder model for the robot to be calibrated.

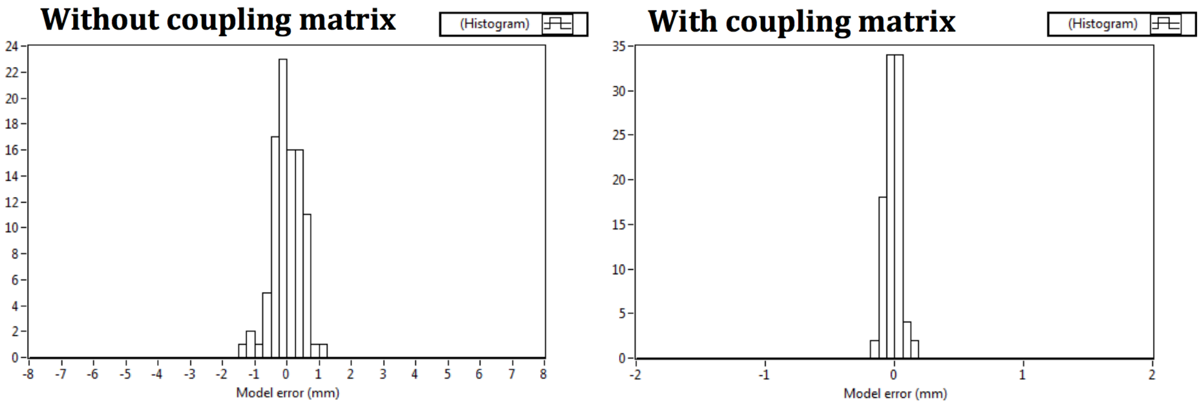


Figure 62 Difference in error model, using the actuator space definition

Total system model verification

When checking the mathematical model of the total system with fixed orientation, the model produces a large constant error but with a low spread. This points to good model, but still with a constant error of approximately 7,5 mm. When performing a similar sampling operation with random orientation it shows the same constant error and that the spread is more than doubled, which points to an error in the initial sensor model.

Convergence test

When comparing the correction based on different sample sets in table 9, we see that the model error of the data set with the updated model is very low. A fair assumption is that the main factor contributing to this final error is mostly un-modeled noise, however no tests have been conducted to verify this assumption. When comparing the parameter corrections of different sample sets in table 9, we see that the correction fluctuates, especially for the joint offsets, which is the only parameter of importance in the calibration procedure. The biggest difference is seen when comparing the two sample sets of approximately 35 samples (C and CC). Of the two we see that the CC-correction especially deviates from the other samples.

By looking at the measurement distribution in appendix E, we see that they are uneven and not as uniformly distributed as for data sets of 50 to 100 samples. But when merging the C and CC sample sets, we see that measurement distribution becomes more uniform, the big deviations in the CC-correction are wiped out and the correction is becoming similar to the other datasets. This indicates that the sample sets need to be off a certain size. Regardless of this, a parameter fluctuation as shown in table 9 points to a bad calibration method. In order to draw a final conclusion, more samples are needed and a statistical analysis of the effect of the calibrating with different sample sizes needs to be performed. As the time limitations of the master thesis did not allow for performing this analysis, a method for conducting the study is proposed in the future work in sub-chapter 9.3

Optimization test

The optimization test was conducted to see how the model error of the sample set was improved when using updated parameters calculated from a different set of samples. The test shows a big improvement in the model error for the validation set from using the initial model parameters. In the final test the updated parameters were used in the sampling operation, and we see that the constant error produced when sampling with the initial parameters were removed. We also see that that the big spread in sampling with random orientation was reduced. Thus the model parameters are more accurate than the previous. On the other hand, the spread should be much less than 2,5 mm with an accurate calibration. Another uncertainty that can contribute to the error is the accuracy of the random pose generation calculations. Obviously, with the fluctuation of the parameter correction, it is impossible to say if the updated parameters used in the sampling operation are the most accurate. This emphasizes the importance of having a statistical analysis of the calibration result.

9.2 Hardware setup

This section discusses some important consideration when creating the calibration setup. Appendix D shows the initial hardware setup. A setup was not successful due two to main factors:

- Sensor orientation w.r.t. the flange
- Plane orientation w.r.t. the base.

Sensor orientation

The sensor orientation problem can be broken down into two problems; practical and mathematical. The practical problem appeared during the sampling operation. Due the orientation, the risk of collisions between plane and the flange adapter was very high. Thus the random orientation was limited to 25 degrees for roll, pitch and yaw. The mathematical problem comes from having a 90-degree angle, whereas the cosine of this angle will be zero, thus introducing several zeros to the parameter Jacobian calculations.

Plane orientation

In the initial model, the plane was parallel to the base and thus having a the normal vector of

$$\hat{\mathbf{n}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Thus, introducing several zeros to the calculations of the parameters Jacobian, especially in formula 6.21, which is included in the joint offset correction by formula 6.23. This shows that there exists some kind of relationship between the plane and q_2, as the correction of this joint would be constantly zero, in all iterations regardless of the initial joint parameters for q_2 were set to extreme values. However, no research was conducted to find the origin of this relationship.

9.3 Recommendations for further work

The thesis holds several elements that need further addressing. Three main topics are suggested for future work:

- Experimentation with different hardware setups.
- Implement a data analysis and validation procedure.
- Further improvement of existing application.

Hardware Setup

The main focus of a future setup should be to use several calibration planes located at different positions, with different orientations in the workspace the robot. How this is realized in the calibration principle, is described in sub-chapter 6.3. As discussed in the previous section, the parameter correction for different sample sets fluctuates. One explanatory theory for this is that the hardware setup only utilizes one plane to gather the sample sets, because the joint movements in the sampling operation are limited to one single calibration plane. This theory argues that small un-modeled linear relationships may create confusion in the optimization procedure, leading to the fluctuations in the parameter correction. Therefore it should be tested if having several calibration planes would remove these linear relationships and by having a statistical analysis, as described below, of the parameter correction it can be used to compare the difference in parameter correction fluctuation when using one and several calibration planes.

A different, more experimental approach is to use a sphere as calibration object instead of several planes. This will require a change in the total system model and also the sphere needs to be of a certain size. On the other hand a more practical setup would be a cube, which basically represents 5 different planes. However the first priority should be to test with several planes as they can be distributed around the robot workspace and therefore require different arm configurations on the robot, which will increase the joint identifiability.

Experimentations with other sensors.

As the range of the laser used in this implementation is very small, a longer distance laser will help avoid collisions and the trade-off can be loss of accuracy. A different approach to implement this calibration method is by using a fixed probe and force sensor mounted on the robot.

A method that is previously been experimented with by Morten Lind, uses a steel rod with hemispherical tip and force sensor to sense the calibration plane. This method is will be easier to implement in robot application already using force sensors, and in the case of a mechanical gripper, as the robots end-effector, this only need to grab a rod and perform the sampling application.

Result analysis and validation tests

Result analysis

In order to validate the calibration result a statistical analysis of the data need to be conducted. In a discussion with with Morten Lind , he suggested the following procedure:

All the sample data should be gathered in on big sample file, with preferably as many samples as possible. From this, an application should be implemented that selects random samples to build a sample set that will be used for calibration. The idea is that the application can be set to perform a calibration using sample sets of for instance 30 random samples from the file, and perform this operation 10 times. This will result in 10 different sample sets, with 10 different sets of updated parameters, and from this the parameter fluctuation can be calculated. Further the same applications can be performed with 50-100-150 (and so on) sized sample sets to see if the parameter fluctuations get stable with the increasing sample size. If the parameter correction for 10 sample sets of a given number of samples converges to a similar solution with minimal fluctuation, it will verify the calibration.

Note that it is important to consider the size of the total sample file and the sample sets used for calibration. If the big sample file contains 500 samples, and the analysis creates 10 different sets of 200 random selected samples, it is obvious that the sample sets will become similar and therefore produce stable correction values. As of this the size of the sample file and the size of the calibration set must be adjusted accordingly.

Validation tests

Assuming that a good and stable calibration result is obtained, the calibrated robot needs to be validated. A validation test proposed by Morten Lind is to measure the length of a gauge block with the calibrated robot. This can be done by measuring four points on each side of the gauge block and from the two sets of four points, two planes can be constructed and finally the distance between the constructed planes is calculated. A good calibrated robot should be able to measure approximately the same length as the known length of the gauge block. The verification must of course be carried out in the same workspace area where the calibration has taken place. If this test provides a good result, the test should be expanded to larger objects with known lengths that will induce a larger movement in the robot joints in order to perform the measurements. Similarly, the length is measured by constructing two planes.

A different approach can be to perform laser measurements on an object similar to the one in figure 63. The object has a “podium-shape” with three different heights. By programming the robot to follow a straight line, as indicated in the figure, and measuring the distance to each level with the laser distance, the height difference of each level can be accurately determined.

Obviously, the actual height needs to be verified in a CMM machine beforehand. The object could only allow one direction of measurement, or two as suggested in the figure. The problem with the method is that is only gives verification in a very small area of the workspace, and in order to expand the test it needs to be manually moved around and re-oriented and therefore making it hard to automate. Also, the four-point measurement to construct a plane and measure the distance between the planes could be applied to this piece.

Application

The existing optimization and sampling application should be merged together and a statistical analysis and verification application should be added. In discussions, Morten Lind has emphasized the importance of a collision checker for the robot for performing random samples. By implementing a collision checker to the random sampling application, we can identify if the robot during any point of the path interpolation steps will crash to either its surroundings or itself, and from this abort the current pose and instead do a different one. This will free the operator from keeping watch of the robot during the sampling operation, freeing up time and making it easier to collect large sample sets. Also it allows for bigger orientation changes in the random pose generation, allowing for larger joint variations. In order to achieve this, a complete emulator needs to be created with all hardware pertaining to the cell. In the original implementation by Morten Lind this was solved by creating a simulation environment using the Blender engine for Python.

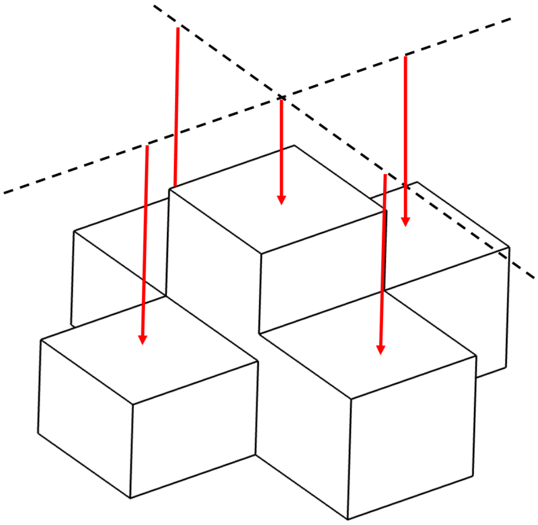


Figure 63 Proposed validation object, with measuring points

9.3 Joint offset calibration on plane

This section discusses the possibilities, advantages and disadvantages of this calibration method.

The joint offset calibration on plane method is very flexible with respect to the initial determination of the model parameters. Since uncertainties, such as sensor and plane location and orientation are calibrated together with the robot, there is no need for accurate initial measurements of these objects. This makes the method easier to implement without overseeing the error sources they introduce. Also, the calibration procedure uses simple and cheaper tools, compared to procedures using laser-trackers and laser interferometers. Another advantage of this method is the possibility for implementing it in the robot's normal routine. For instance, in many applications the robot will spend time waiting for the next operations. This time can be spent to automatically calibrate the robot, thus ensuring good results throughout operations. These advantages covers most of the concerns stated by Elatta et al. (2004), described in sub-chapter 5.3 on selecting measurement equipment for calibration.

The trade-off from having such a flexible setup is, as mentioned in sub-chapter 6.2, reduced identifiability of particular joints. Due to the inaccurate plane location, the joint-offset parameter of q_1 cannot be identified, as its influence is indistinguishable from rotating the calibration plane around the robot. Also, q_6 cannot be identified because it is indistinguishable from the correction of the laser sensor orientation. A way to work around these problems is to eliminate the plane and sensor models from the total system model. This can be done if the pose of the sensor and calibration plane is accurately known with respect to the robot. This may also help increase the accuracy of the calibration and can be achieved using one of the measurement methods described in sub-chapter 3.4. However, as the goal of this calibration method is to prove that good calibration results can be obtained without the need of this kind of expensive equipment and challenging calibration routines, this is not a recommended direction for the future work.

Chapter 10

Conclusion

The vast development in the robot industry demands a higher degree of automation and flexibility without sacrificing the quality and costs of their products. In order to meet the new demands, industrial robots are already common equipment in most industries. However, the operations are limited to those of high tolerances and mainly benefitting from the robots' high repeatability. As presented in this thesis, industrial robots still have a lot of unrealized potential when it comes to high accuracy tasks and this is mainly due to the inaccuracy compared to application-dedicated-equipment. Another challenge is that it is not much information about the accuracy of the individual robots on the market, and manufacturers will not disclose it and rather focus of the repeatability of the robot. Obviously, this is because the repeatability is much better than the accuracy, and therefore manufacturers fear that disclosing the accuracy will create negative publicity and thus loss of customers to other companies withholding information about their accuracy. It is from this kind of scenarios that the importance of ISO standards emerges. Ideally the robot assessment presented in ISO9283, should be mandatory for all robot manufacturers. This would give a fair and honest information of all the robot models on the market, and it would help drive the development of increasing the robots' accuracy.

As mentioned earlier in this thesis, there are two ways of increasing the robots accuracy. The first way is to change the mechanical design of the robot and second is to improve or alter the software in the robot controller. The latter alternative is better known as robot calibration, and represents the most feasible and cost effective alternative. It is important to bear in mind that robot accuracy improvement is limited by the robot’s repeatability and the accuracy of the calibration, where the calibration accuracy depends on the accuracy of the measurements and the inaccuracies due to the effects of un-modelled parameters.

The main work of this thesis centres around the implementations of a calibration method focusing on the joint offset errors in a serial manipulator has been implemented on a 6-axis industrial robot. The following points can summarize the implementation:

- A hardware setup.
- A sampling application.
- An optimization application.

The work with this implementation proved to be beyond the time limits of a master thesis. Thus, a statistical analysis of the results and final validation experiments of the method was not achieved. Instead this thesis has a separate section describing how future work should take place, and this mainly concern a statistical analysis of the calibration data, a change in the calibration setup by using several calibration tests instead of one, and implementation of a final validation experiment of the calibrated robot. Still several test where performed to verify:

- Mathematical model of the calibration principle.
- Error model convergence.
- Model optimization.

The tests points out the importance of validating the encoder model of the robot that is to be calculated as this varies from different robot manufacturers, but more importantly that the computational model is transferable to different types of robots. The tests also show how the parameter correction fluctuates between the various sample sets, arguing the need of a statistical analysis of the calibration results. However, the tests still show a reduction in model errors when using the updated model parameters to different sample sets, and when performing sampling operations. In conclusion, the tests show the mathematics in the calibration principle work, implying the potential in this simple calibration method and emphasize that further work should be done in order to realize it.

Bibliography

Ackerson, D.S. and Harry, D.R. (1985). "Theory, Experimental Results, and Recommended Standards regarding the Static Positioning and Orienting Precision of Industrial Robots". *Robotics & Computer-Integrated Manufacturing,* Vol. 2, pp. 247-259.

Brussel, H.V. (1990). "Evaluation and Testing of Robots". *Anneals of the CIRP,* Vol. 39.

Chen, J. and Chao, L-M. (1987). "Positioning Error Analysis for Robot Manipulators with All Rotary Joints". *IEEE Journal of Robotics and Automation,* Vol. RA-3, No. 6, pp. 539-545.

Elatta, A.Y., Gen, L.P., Fan, L.Z., Daoyuan, Y. and Fei, L. (2004). "An Overview of Robot Calibration". *Information Technology Journal,* Vol. 3, pp. 74-78.

Greenway, B. (2000). "Robot Accuracy". *Industrial Robot: An International Journal,* No. 4, pp. 257-265.

Gong, C., Yuan, J. and Ni, J. (2000). "Nongeometric Error Identification and Compensation for Robotic System by Inverse Calibration". *International Journal of Machine Tools & Manufacture,* Vol. 40, pp. 2119-2137.

Gonzalez, R., Fu, K. and Lee, C. (1987). *Robotics: Control, Sensing, Vision and Intelligence.* Mcgraw-Hill Book Company.

Hayati, S.A. (1983). "Robot Arm and Geometric Link Parameter Estimation". *Preceedings of the 22nd IEEE conference on decision and control,* pp. 1477- 1483.

International Organization for Standardization (no date). *About ISO*. Retrieved 06.02.14 from http://www.iso.org/iso/home/about.htm.

ISO 8373 (2012). *Robots and Robotic Devices – Vocabulary.* Geneva: International Organization for Standardization.

ISO 9283 (1998), *Manipulating Industrial Robots – Performance Criteria and Related Test Methods.* Geneva: International Organization for Standardization.

ISO TR 13309 (1995), *Manipulating Industrial Robots – Informative Guide on Test Equipment and Metrology Methods for Robot Performance Evaluation in Accordance with ISO 9283.* Geneva: International Organization for Standardization.

JARA (2013). *Yearly Results, January – December 2013 Results*. Retrieved 04.03.13 from http://www.jara.jp/e/h/statistics02.html.

Johnsrud, V. (2013). "Robotic Milling of Wooden Products". *Specialization project*. Trondheim: Norwegian University of Science and Technology.

Judd, R.P. and Knasinski, A.B. (1990). “A Technique to Calibrate Industrial Robots with Experimental Verification”. *IEEE Transactions on Robotics and Automation,* Vol 6, pp. 351-357.

Mooring, B.W, Roth, Z.S. and Driels, M.R. (1991). *Fundamentals of Manipulator Calibration.* John Wiley & Sons, Inc.

Mooring, B.W. and Tang, G.R. (1984). “An Improved Method for Identifying the Kinematic Parameters in a Six Axis Robot”. *Preceedings of the 1983 ASME computers in Engineering Conference, Las Vegas, Nevada,* pp. 79-84.

Nubiola, A. and Bonev, I.A. (2013). “Absolute Calibration of an ABB IRB 1600 Robot Using a Laser Tracker”. *Robotics and Computer-Integrated Manufacturing,* Vol. 29, pp. 236-245.

Petercorke.com (2012). *Robotics Toolbox*. Retrieved 10.02.13 from http://www.petercorke.com/Robotics_Toolbox.html.

Renders, J-M., Rossignol, E., Becquet, M. and Hanus, R. (1991). “Kinematic Calibration and Geometrical Parameter Identification for Robots”. *IEEE Transactions on Robotics and Automation,* Vol. 7, pp. 721-731.

Roth, Z.S., Mooring, B.W. and Bahram, R. (1987). “An Overview of Robot Calibration”. *IEEE Journal of Robotics and Automation, Vol. 3, pp. 377-385.*

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

Slamani, M., Nubiola, A. and Bonev, I.A. (2012). “Assessment of the Positioning Performance of an Industrial Robot”. *Industrial Robot: An International Journal,* Vol. 39, pp. 57–68.

Stone, H.W. (1986). *Kinematic Modelling, Identification and Control of Robotic Manipulator.* Ph.D Thesis. Caregie Mellon University: Robotic Institute.

Summers, M. (2005). “Robot Capability Test and Development of Industrial Robot Positioning System for the Aerospace Industry”. *Grapevine (TX), USA: SAE 2005 AeroTech Congress & Exhibition,* No. 2005-01-3336.

Voorthuysen, Dr Erik van (2013). *Lecture notes in MTRN4230: Robotics.* Sydney: University of New South Wales.

Wang, C.B., Chen, J. and Yang, J.C.S. (1984). “Robot Positioning in Accuracy Improvement through Kinematic Parameter Identification”. *Toronto (ON), Canada: Proc. 3rd Canadian CAD/CAM and Robotics Conference.*

Whitney, D.E., Lozinski, C.A. and Rourke, J.M. (1986). “Industrial Robot Forward Calibration Method and Results”. *ASME Journal of Dynamic Systems, Measurement and Control,* Vol. 108, pp. 1-8.

APPENDIX

A – Abbreviations

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CMM	Coordinate Measuring Machine
CNC	Computer Numerical Control
CMM	Coordinate Measuring Machine
DH	Denavit-Hartenberg
DMS	Dead-Man-Switch
IFR	The International Federation of Robotics
IPK	Institutt for Produksjons- og Kvalitetsteknikk (English: Department of Production and Quality Engineering)
ISO	The International Organization for Standardization
JIS	Japanese Industrial Standard
RMS	Root mean square
RT	Real Time
TCP	Tool Centre Point
TP	Teach Pendant
VI	Virtual Instrument

B – Rotation matrices

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} \tag{B.1}$$

$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{B.2}$$

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.3}$$

C – Real-time system communication protocol

Data to send (remote port 5555): (Number of axes) x (Signed Int 32 of delta values in little endian format*) + (Number of axes) x (Signed Int 32 of minus delta values in little-endian format*)

Data to read (local port 5556): (Number of axes) x (Unsigned Int 32 of reference values in little endian format*) + (Number of axes) x (Unsigned Int 32 of current position values in little-endian format*)

* Little-endian:

The 32 bit number: 0x12345678

in little-endian format:
4 * 8 bit number: 0x78 0x56 0x34 0x12

D – Initial hardware setup

This section presents the initial, less successful hardware setup, as it is subject for the discussion in sub-chapter 9.3. The main difference from the final calibration setup is the laser orientation, plane orientation and method for defining the normalized plane vector.

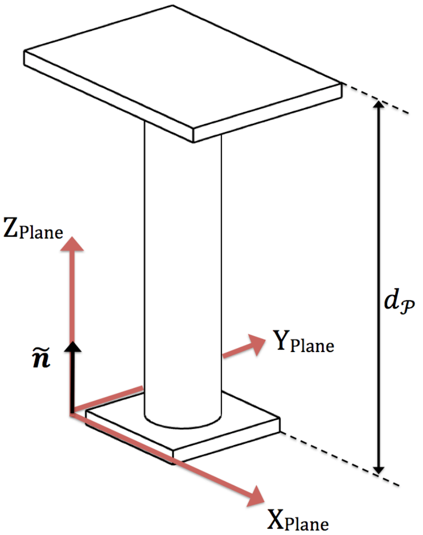
Calibration plane model

The figure below shows a simplified model of the actual calibration plane.
The calibration plane has the following dimensions 250 x 350 mm. By assuming a completely flat plane surface this gives a unit normal vector of: $\hat{\boldsymbol{n}} = [0\ 0\ 1]^{\mathrm{T}}$.
The coordinate frame is arbitrarily chosen on the floor, underneath the calibration plane and is defined with respect to the base coordinate system as this allows for easily determining the normalized plane vector. The shortest distance between the coordinate system origin and plane is roughly measured to

$$d_{\mathcal{P}} = 0{,}510 \text{ m}$$

Thus, the normalized plane vector in the plane coordinate system is calculated as:

$$^{\mathcal{P}}\tilde{\boldsymbol{n}} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{d_{\mathcal{P}}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{0{,}51} \end{bmatrix}$$



The transformation from the plane coordinate system to Base is given by

$$^{Base}\mathbf{Q}_{Plane} = \begin{bmatrix} 0{,}651 \\ -1{,}4974 \\ 0 \end{bmatrix} \text{m}$$

Rotation around Z_{Plane} of 90 degrees

$$^{Base}\mathbf{R}_{Plane} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since the normalized plane vector needs to be defined in the base coordinate system for the parameter Jacobian matrix, a special transformation is need, which is described below

Coordinate transformation of the normalized plane vector

Since the normalized plane vector is not a Euclidean vector, a special transformation is needed to transform the normalized plane vector between coordinate systems. A way to solve this is by converting the normalized plane vector into a *point and normal*-representation, which can be directly transferred between coordinate systems, and finally the point and normal-representation can be used to calculate the normalized plane vector in the new coordinate system.

Given a normalized plane vector

$$^{\mathcal{P}}\widetilde{\boldsymbol{n}}$$

We calculate the unit normal, $^{\mathcal{P}}\widehat{\boldsymbol{n}}$, and a point, $^{\mathcal{P}}$ p, on the plane by

$$^{\mathcal{P}}\{\text{p}, \widetilde{\boldsymbol{n}}\} = \left\{ \frac{^{\mathcal{P}}\widetilde{\boldsymbol{n}}}{\left\| ^{\mathcal{P}}\widetilde{\boldsymbol{n}} \right\|^2}, \frac{^{\mathcal{P}}\widetilde{\boldsymbol{n}}}{\left\| ^{\mathcal{P}}\widetilde{\boldsymbol{n}} \right\|} \right\}$$

Where the point, p fulfils equation (7.11). Since this consist of a position vector and a free vector, it can easily be transformed from plane to base coordinate system using the homogenous transformation matrix, $^{Base}_{Plane}\mathbf{T}$, giving: $^{Base}\{\text{p}, \widetilde{\boldsymbol{n}}\}$, which can be transformed back to a normalized plane vector by

$$ {}^{Base}\widetilde{\boldsymbol{n}} = \frac{{}^{Base}\mathrm{p}^{\mathrm{T}\,Base}\widehat{\boldsymbol{n}}}{\left|{}^{Base}\mathrm{p}^{\mathrm{T}\,Base}\widehat{\boldsymbol{n}}\right|^{2}}\,{}^{Base}\widehat{\boldsymbol{n}} $$

Laser sensor model

The figure below shows an exploded view of the sensor to flange connection.

Based on the CAD model of the flange adapter and laser sensor the following transformation is obtained:

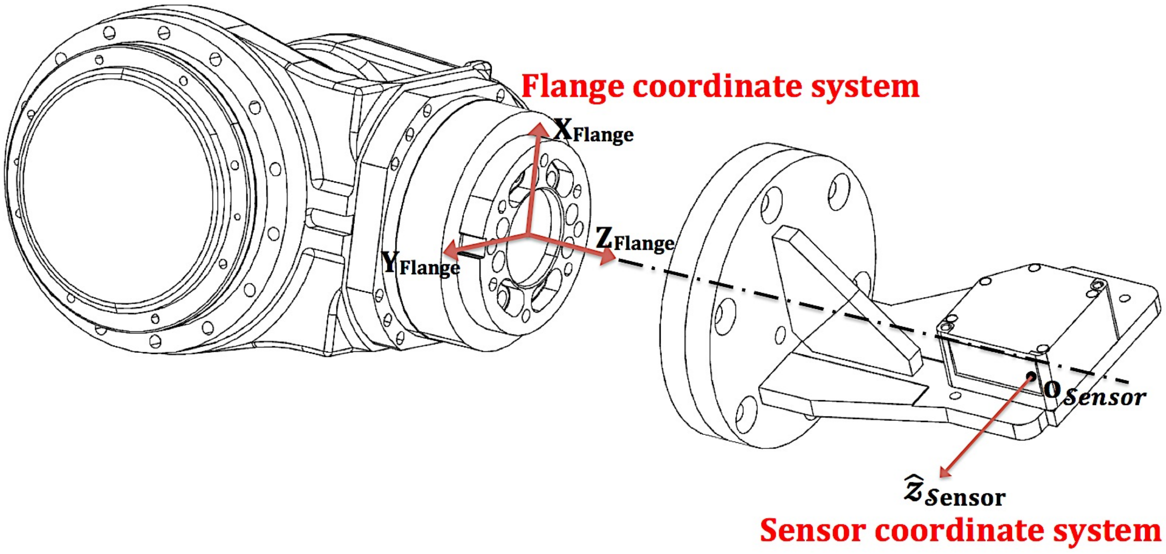
$$ {}^{Flange}o_{Sensor} = \begin{bmatrix} -0{,}0147 \\ -0{,}002 \\ 0{,}1741 \end{bmatrix} \mathrm{m} $$

The orientation is defined by a -90 degree pitch about the flange y-axis followed by a -60 degree roll about flange z-direction. This gives:

$$ {}^{Flange}\widehat{\boldsymbol{z}}_{Sensor} = \begin{bmatrix} \cos(-60)\sin(-90) \\ \sin(-60)\sin(-90) \\ \cos(-90) \end{bmatrix} = \begin{bmatrix} -0{,}5 \\ 0{,}86660 \\ 0 \end{bmatrix} $$

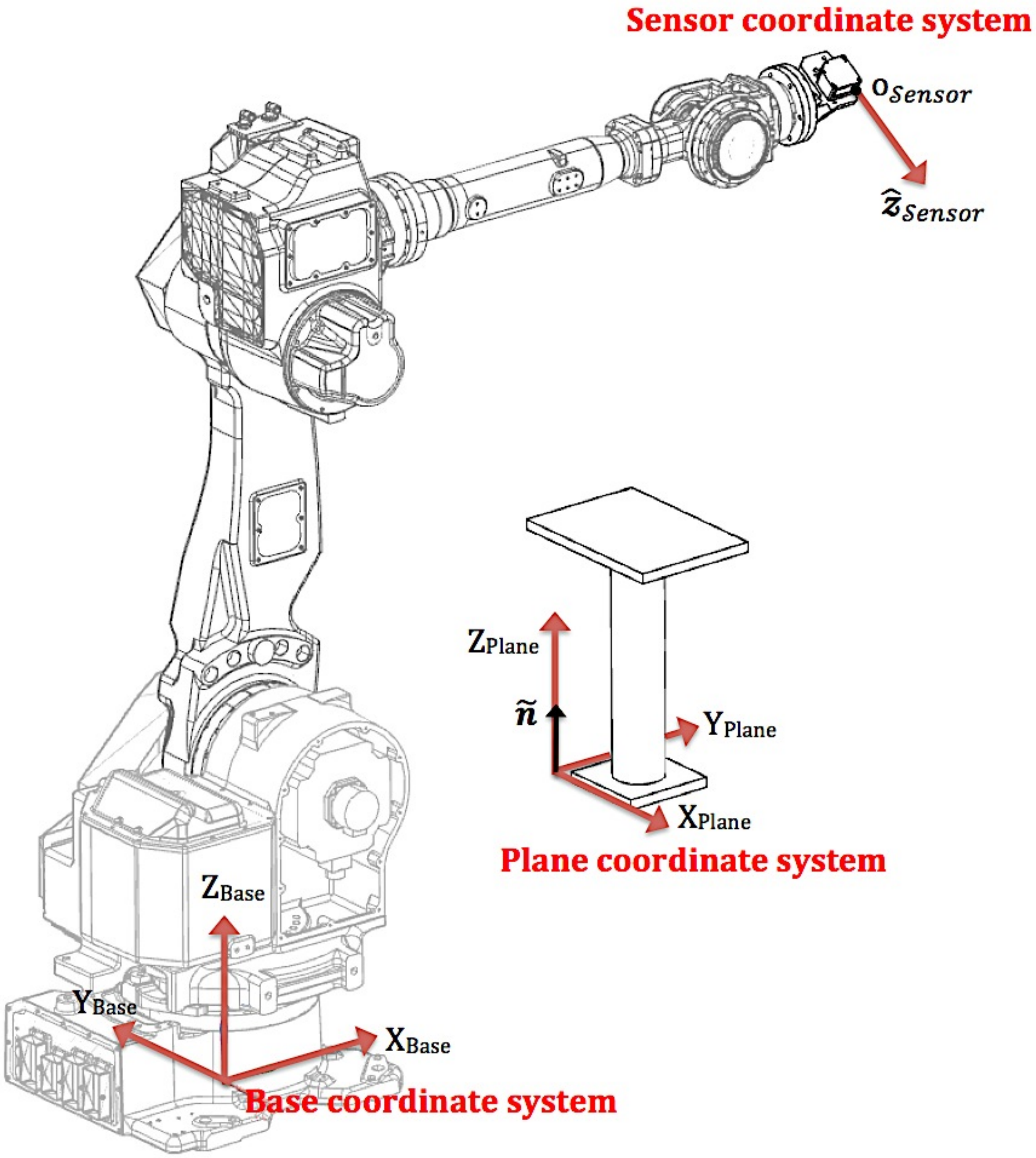
Thus,

$$ \vec{\boldsymbol{\phi}} = [\mathbf{-1,5708} \quad \mathbf{-1,0472}] $$



Total system model

The figure below shows schematically an illustration of the different coordinate system of the hardware setup, together with important model parameters.

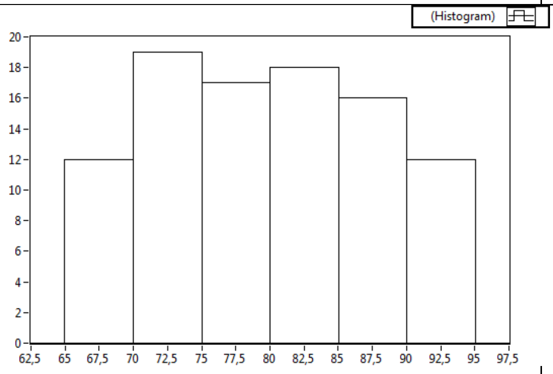
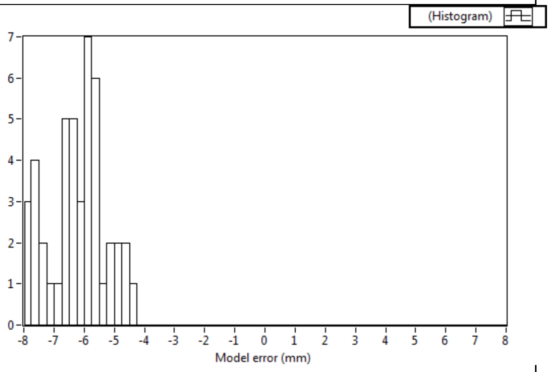
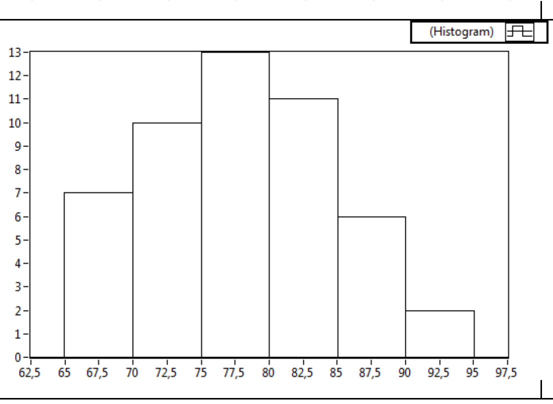
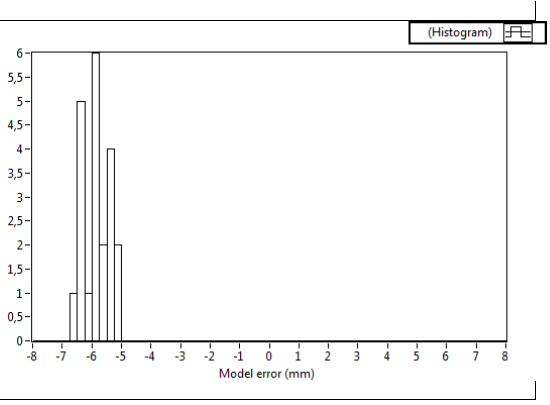


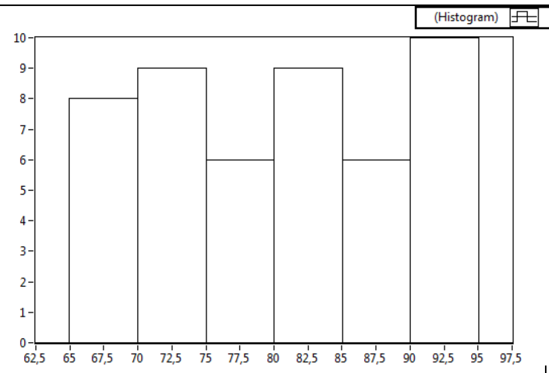
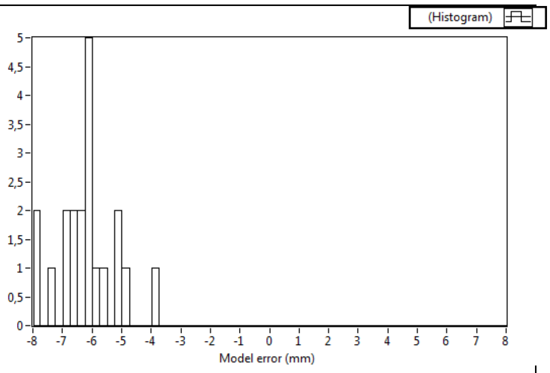
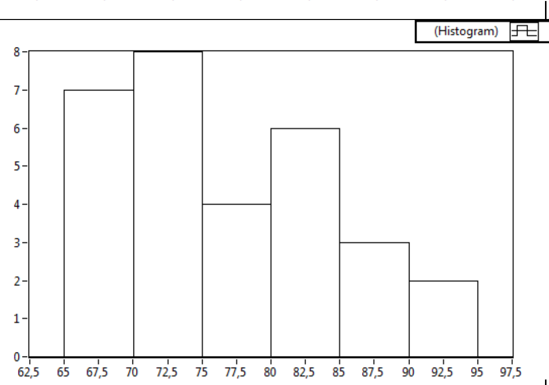
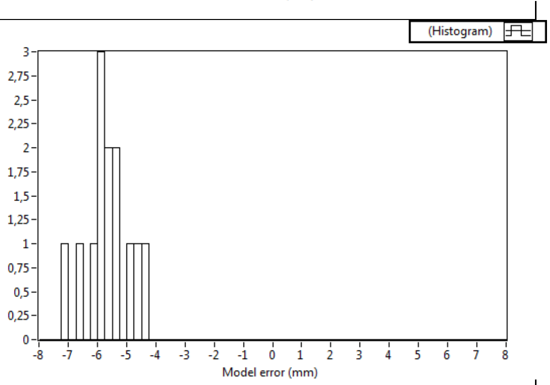
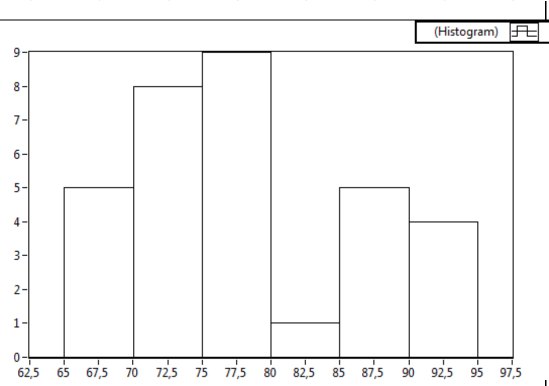
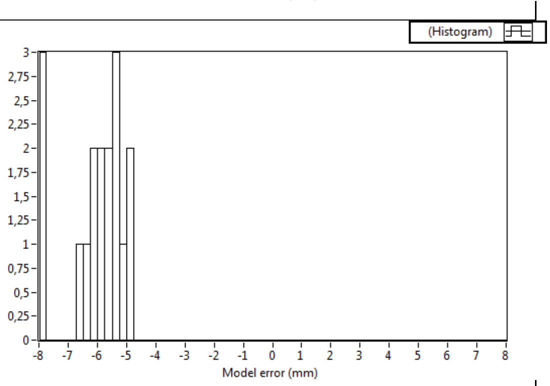
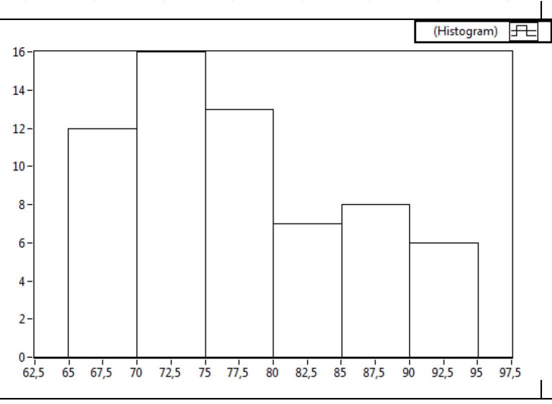
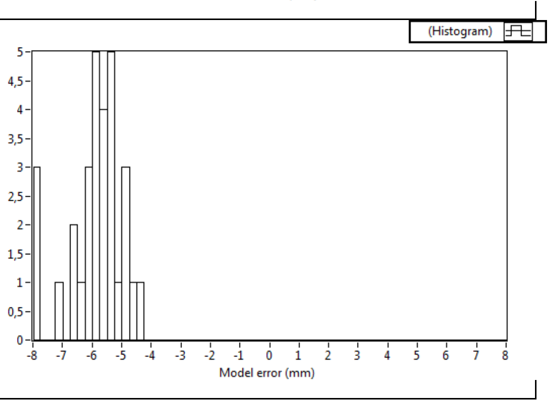
E – Optimization data plots

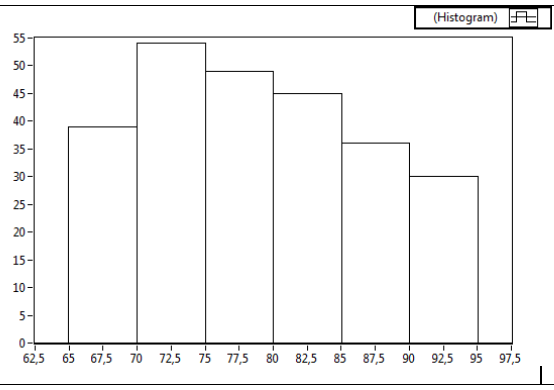
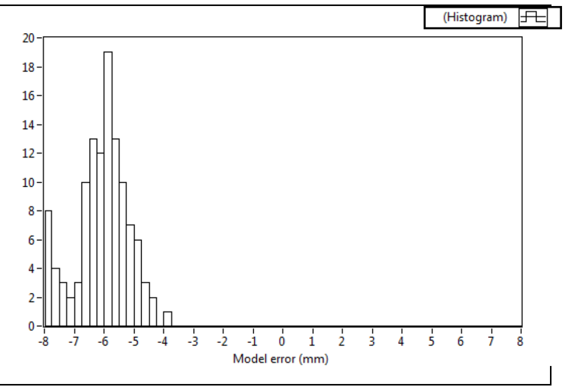
All the data plots from the different sample sets of the convergence test described in sub-chapter 8.2

Name	**Number of samples in data set**	**Bad reads**
A	100	2
B	50	1
BB	50	2
C	35	5
CC	35	3

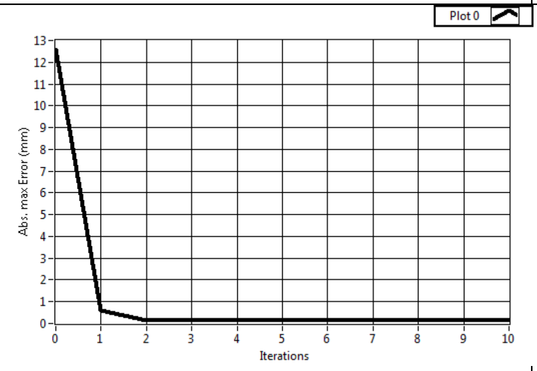
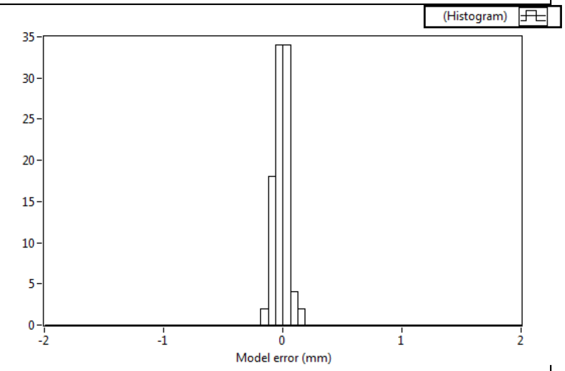
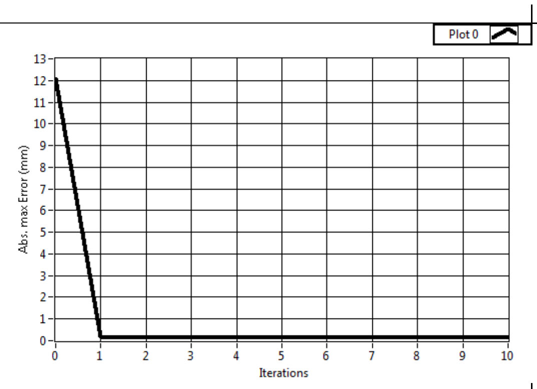
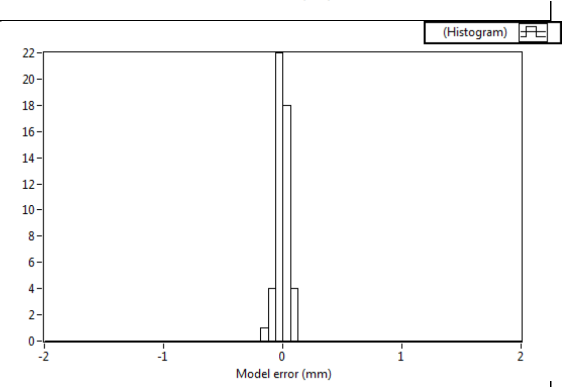
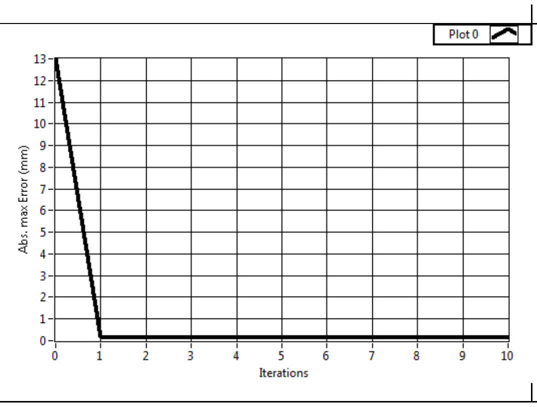
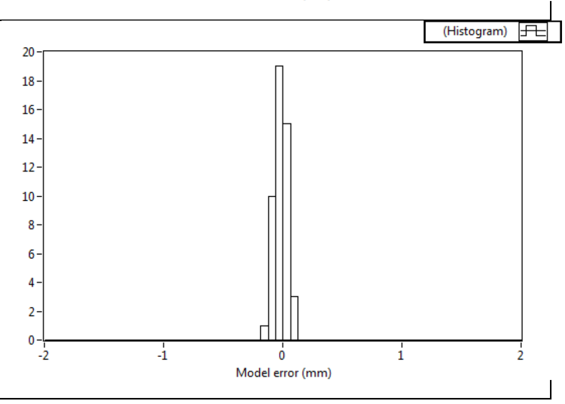
Before optimization

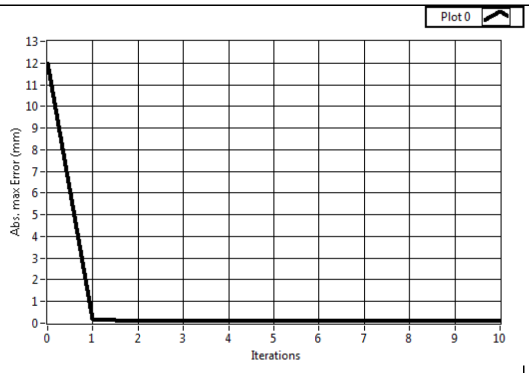
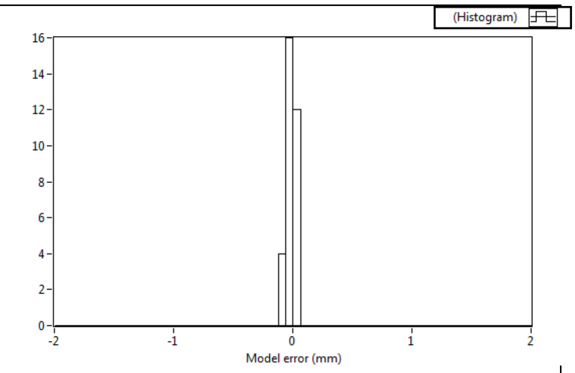
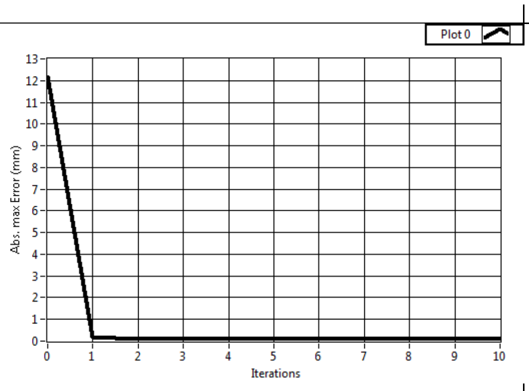
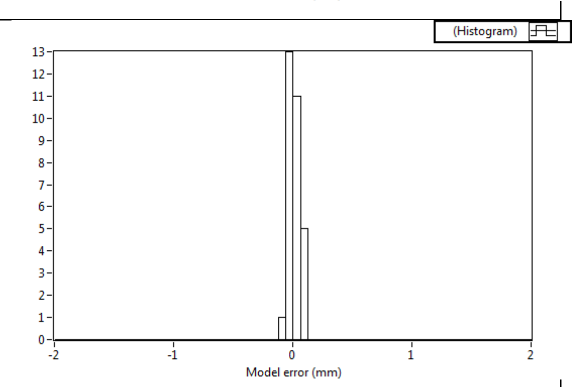
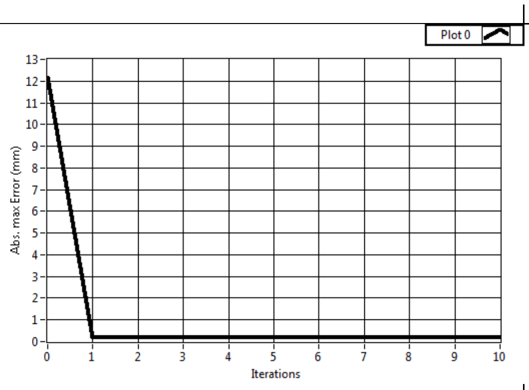
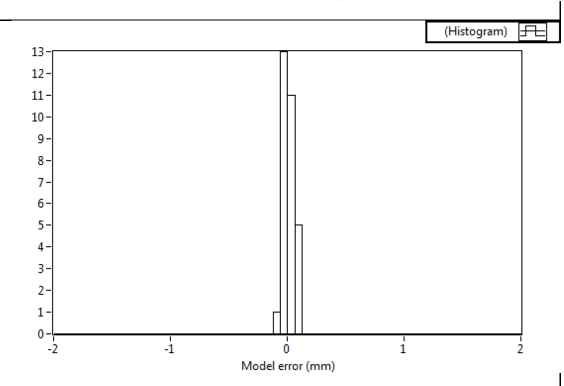
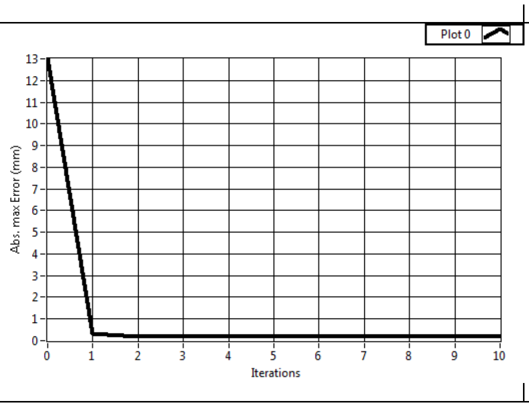
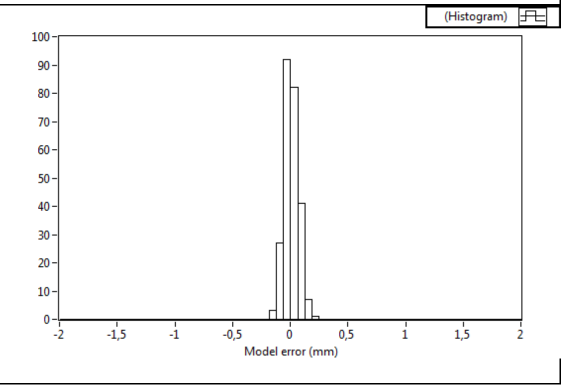
	Measurement distribution	**Error distribution (initial model)**
A		
B		

BB	(Histogram) 0 1 2 3 4 5 6 7 8 9 10 62,5 65 67,5 70 72,5 75 77,5 80 82,5 85 87,5 90 92,5 95 97,5	(Histogram) 0 0,5 1 1,5 2 2,5 3 3,5 4 4,5 5 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 Model error (mm)
C	(Histogram) 0 1 2 3 4 5 6 7 8 62,5 65 67,5 70 72,5 75 77,5 80 82,5 85 87,5 90 92,5 95 97,5	(Histogram) 0 0,25 0,5 0,75 1 1,25 1,5 1,75 2 2,25 2,5 2,75 3 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 Model error (mm)
CC	(Histogram) 0 1 2 3 4 5 6 7 8 9 62,5 65 67,5 70 72,5 75 77,5 80 82,5 85 87,5 90 92,5 95 97,5	(Histogram) 0 0,25 0,5 0,75 1 1,25 1,5 1,75 2 2,25 2,5 2,75 3 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 Model error (mm)
C+ **CC**	(Histogram) 0 2 4 6 8 10 12 14 16 62,5 65 67,5 70 72,5 75 77,5 80 82,5 85 87,5 90 92,5 95 97,5	(Histogram) 0 0,5 1 1,5 2 2,5 3 3,5 4 4,5 5 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 8 Model error (mm)

A + B + BB + C + CC		

After optimization

	Absolute max model error per iteration	Error distribution (updated model)
A		
B		
BB		

C		
CC		
C+ **CC**		
A + **B +** **BB +** **C +** **CC**		

F – VI’s made for this Thesis

This section aims to describe some of the important VIs, other than the MAIN-Sampling and MAIN-Optimization VIs, made in the implementation of the joint offset calibration method.

Eliminate outliers Search through the model error array to find outliers and deletes them from the sample set.

Get laser distance reading Reads the laser sensor distance measurement.

LONG to RAD Transform from Encoder-space ($\overline{\mathbf{q}}^{e}$) to Joint-space ($\overline{\mathbf{q}}^{e}$)**.**

Mathematical model Holds the Computational model for the distance measurement.

NACHI MC70 Robot definition Holds the robot definition, the DH-matrix and the tool definition.

New laser read Governs the laser read operation in the sampling operation, storing the laser reading to the circular buffer.

RAD to LONG Transform from Joint-space ($\overline{\mathbf{q}}$) to Encoder space ($\overline{\mathbf{q}}^{e}$)

Random pose generation Generates a random laser sensor pose for the sampling application.

Regressor-MAIN Calculates the parameter Jacobian.

Sample generator Generates the path matrix for the sampling operation.

Step Generator Performs the joint-space path interpolation

Transform-Plane to base given normal and origin Calculates the plane to base transformation given a normal vector and a coordinate frame origin in base coordinates.

Update-correction and parameters Calculates the parameter correction and adds it to the current parameter array.

Update - Model Calculates the computed distance measurement from a sample set, using the current parameter array and calculates the model error.

G – Contents of digital attachments

The digital attachments contains:

- Calibration application (LabVIEW)
    - Sampling application
    - Optimization application
- Sample data from tests presented in chapter 8
- Video showing the robot sampling.

H – Datasheets

Included datasheets:

- NACHI MC70
- Omron ZS-LD80

H.1 – NACHI MC70

MC35/50/70

◆ 本体仕様
Robot specifications

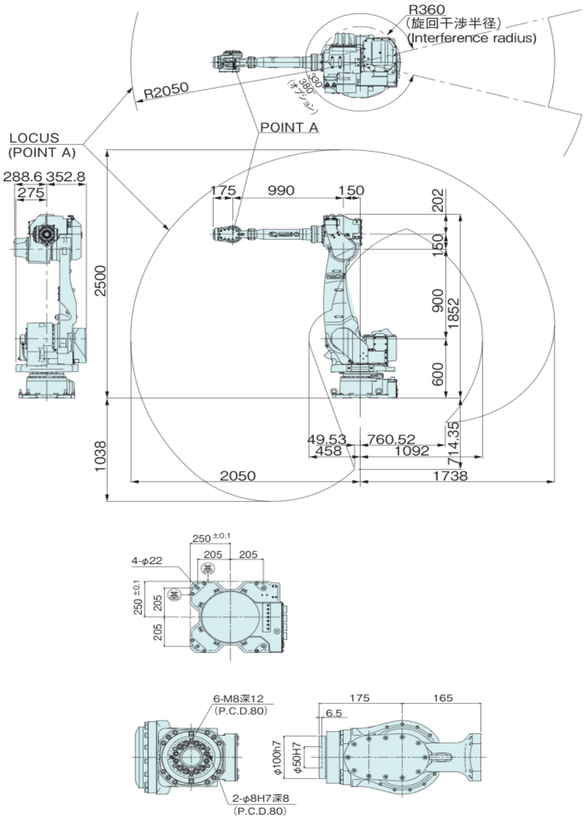
項目　Item		仕様　Specifications		
ロボット型式　Robot model		MC35-01	MC50-01	MC70-01
構造　Construction		関節形　Articulated construction		
自由度　Number of axes		6		
駆動方式　Drive system		AC サーボ方式　AC servo system		
最大動作範囲 Max. operating area	J1	±2.88rad（±165°）		
	J2	+1.39～−2.35rad（+80～−135°）		
	J3	+4.54～−2.55rad（+260～−146°）		
	J4	±6.28rad（±360°）		
	J5	±2.18rad（±125°）		
	J6	±7.84rad（±450°）		
最大速度 Max. speed	J1	3.23rad/s(185°/s)	3.14rad/s(180°/s)	3.05rad/s(175°/s)
	J2	3.14rad/s(180°/s)		2.53rad/s(145°/s)
	J3	3.32rad/s(190°/s)	3.14rad/s(180°/s)	2.88rad/s(165°/s)
	J4	5.32rad/s(305°/s)	4.45rad/s(255°/s)	4.10rad/s(235°/s)
	J5	5.32rad/s(305°/s)	4.45rad/s(255°/s)	4.10rad/s(235°/s)
	J6	7.33rad/s(420°/s)	6.46rad/s(370°/s)	6.11rad/s(350°/s)
可搬質量 Payload	手首部 Wrist	35kg	50kg	70kg
	上腕部 Forearm	15kg		
手首トルク Wrist torque	J4	160N·m	210N·m	300N·m
	J5	160N·m	210N·m	300N·m
	J6	90N·m	130N·m	150N·m
手首慣性モーメント※1 Wrist moment of inertia※1	J4	16kg·m²	30kg·m²	
	J5	16kg·m²	30kg·m²	
	J6	5kg·m²	12kg·m²	
位置繰返し精度※2 Position repeat accuracy※2		± 0.07mm		
最高使用空気圧力 Maximum working air pressure		0.49MPa（5.0kgf/cm²）以下 0.49MPa (5.0kgf/cm²) or less		
周囲温度　Ambient temperature		0～45℃		
設置条件　Installation		床置　Floor mounted		
耐環境性※3 Environmental resistance※3	本体部 Body	IP54 相当（オプションでIP65/67相当） Meets the IP54 standard (optional IP65/67)		
	手首部 Wrist	IP67 相当（防塵防滴） Meets the IP67 standard (for dust and waterproofing)		
本体質量　Robot mass		640kg		

1[rad] = 180/π[°],　1[N·m] = 1/9.8[kgf·m]

※1：手首許容慣性モーメントは、手首負荷条件により異なりますので、ご注意ください。
※2：JIS B 8432に準拠しています。
※3：有機溶剤、酸、アルカリ、塩素系、ガソリン系切削液など、シール部材を劣化させる液体は使用できません。

※1：Note that wrist moment of inertia varies depending on wrist load conditions.
※2：JIS B 8432 compliant.
※3：Fluids that corrode the seal material, such as organic solvents, acids, alkalis, salts and petroleum-based cutting fluids, cannot be used.

◆ 外形寸法及び動作範囲
Exterior dimensions and operating envelope



●製品改良のため、定格、仕様、外寸などの一部を予告なしに変更することがあります。
●本製品の最終使用者が軍事関係、または兵器等の製造用に使用する場合、「外国為替及び外国貿易管理法」の定める輸出規制の対象となることがあります。輸出される際には、十分な審査及び必要な輸出手続きをお取り下さい。

•The specifications are subject to changes without notice.
•In case that an end user uses this product for military purpose or production of weapon, this product may be liable for the subject of export restriction stipulated in the Foreign Exchange and Foreign Trade Control Law. Please go through careful investigation and necessary formalities for export.

NACHI
株式会社 不二越

東京本社　東京都港区東新橋1-9-2 汐留住友ビル17F　〒105-0021
Tel：03-5568-5111　Fax：03-5568-5206
富山本社　富山市不二越本町1-1-1　〒930-8511
Tel：076-423-5111　Fax：076-493-5211
URL：http://www.nachi-fujikoshi.co.jp

東日本支社	Tel：03-5568-5286	北陸支店	Tel：076-425-8013
中日本支社	Tel：052-769-6825	広島支店	Tel：082-568-7460
西日本支社	Tel：06-7178-5105	国際営業本部	Tel：03-5568-5245

株式会社 ナチロボットエンジニアリング
NACHI ROBOTのサービス・メンテナンスは—

本　　社	Tel：03-5568-5180	大阪センター	Tel：06-6748-2532
北関東センター	Tel：0276-33-7888	広島センター	Tel：082-284-5175
東北サービス室	Tel：022-346-0605	岡山サービス室	Tel：0866-90-3407
西関東センター	Tel：0467-71-5115	九州センター	Tel：093-434-9133
名古屋センター	Tel：0565-29-5811	北陸センター	Tel：076-423-6283
東海サービス室	Tel：053-454-4160		

〈Tokyo Head Office〉 Shiodome Sumitomo Bldg.17F 1-9-2 Higashi-shinbashi, Minato-ku, Tokyo105-0021, Japan
Tel: +81-(0)3-5568-5111　Fax: +81-(0)3-5568-5206
〈Toyama Head Office〉 1-1-1 Fujikoshi-Honmachi, Toyama 930-8511, Japan
Tel: +81-(0)76-423-5111　Fax: +81-(0)76-493-5211
〈Oversea Div.〉 Tel: +81-(0)3-5568-5245

CATALOG NO.　7308-3
2011.11.Q-ABE-ABE

H.2 – Omron ZS-LD80

The scalable high-precision laser measurement sensor

ZS-HL series

The ZS laser sensor family provides outstanding measurement performance on all kind of materials. Its huge range of sensor heads and scalable concept makes it a versatile platform for all high precision sensing applications.

- Highest resolution and dynamic sensing range for all surfaces
- Modular and scalable platform concept for up to 9 sensors
- Easy to use, install and maintain for all user levels
- Fast response time of 110 µs
- Multi-tasking capability – manages up to 4 measurement tools in one controller

Ordering information

Sensors

ZS-HL-series sensor heads

Optical system	Sensing distance	Beam shape	Beam diameter	Resolution[*1]	Model
Regular reflective models	20±1 mm	Line beam	1.0 mmx20 µm	0.25 µm	ZS-HLD2ST
	25±2 mm		2.2 mmx45 µm	0.6 µm	ZS-HLDS2VT
Diffuse reflective models	50±5 mm		1.0 mmx30 µm	0.25 µm	ZS-HLDS5T
	100±20 mm		3.5 mmx60 µm	1 µm	ZS-HLDS10
	600±350 mm		16 mmx0.3 mm	8 µm	ZS-HLDS60
	1500±500 mm		40 mmx1.5 mm	500 µm	ZS-HLDS150

[*1] Refer to the table of ratings and specifications for details.

ZS-HL-series sensor heads (for nozzle gaps) also compatible with ZS-L controller

Optical system	Sensing distance	Beam shape	Beam diameter	Resolution[*1]	Model
Regular Reflective	10±0.5 mm	Line beam	900x25 µm	0.25 µm	ZS-LD10GT
	15±0.75 mm				ZS-LD15GT

[*1] Refer to the table of ratings and specifications for details.

ZS-L-series sensor heads

Optical System	Sensing distance	Beam shape	Beam diameter	Resolution[*1]	Model
Regular reflective models	20±1 mm	Line beam	900x25 µm	0.25 µm	ZS-LD20T
		Spot beam	25 µm dia.		ZS-LD20ST
	40±2.5 mm	Line beam	2000x35 µm		ZS-LD40T
Diffuse reflective models	50±5 mm	Line beam	900x60 µm	0.8 µm	ZS-LD50
		Spot beam	50 µm dia.		ZS-LD50S
	80±15 mm	Line beam	900x60 µm	2 µm	ZS-LD80
	130±15 mm	Line beam	600x70 µm	3 µm	ZS-LD130
	200 ±50 mm	Line beam	900x100 µm	5 µm	ZS-LD200
	350 ±135 mm	Spot beam	240 µm dia.	20 µm	ZS-LD350S

[*1] This is the peak-to-peak displacement conversion value in the displacement output at the measuring center distance in high-precision mode when the number of samples to average is set to 128 and the measuring mode is set to the high-resolution mode. The standard workpiece is white aluminum ceramics in diffuse reflection mode and glass in the regular reflection mode.

ZS-L-series sensor heads

Item	Model	ZS-LD20T		ZS-LD20ST		ZS-LD40T		ZS-LD10GT	ZS-LD15GT
Applicable controllers		ZS-HLDC/LDC series							
Optical system		Regular reflection	Diffuse reflection	Regular reflection	Diffuse reflection	Regular reflection	Diffuse reflection	Regular reflection	
Measuring center distance		20 mm	6.3 mm	20 mm	6.3 mm	40 mm	30 mm	10 mm	15 mm
Measuring range		±1 mm	±1 mm	±1 mm	±1 mm	±2.5 mm	±2 mm	±0.5 mm	±0.75 mm
Light source		Visible semiconductor laser (wavelength: 650 nm, 1 mW max., JIS Class 2)							
Beam shape		Line beam		Spot beam		Line beam			
Beam diameter[1]		900x25 µm		25 µm dia.		2,000x35 µm		Approx. 25x900 µm	
Linearity[2]		±0.1%F.S							
Resolution[3]		0.25 µm		0.25 µm		0.4 µm		0.25 µm	0.25 µm
Temperature characteristic[4]		0.04% FS/°C		0.04% FS/°C		0.02% FS/°C		0.04% FS/°C	
Sampling cycle[5]		110 µs (high-speed mode), 500 µs (standard mode), 2.2 ms (high-precision mode), 4.4 ms (high-sensitivity mode)							
Indica-tors	NEAR indicator	Lights near the measuring center distance, and nearer than the measuring center distance inside the measuring range. Flashes when the measurement target is outside of the measuring range or when the received light amount is insufficient.							
	FAR indicator	Lights near the measuring center distance, and further than the measuring center distance inside the measuring range. Flashes when the measurement target is outside of the measuring range or when the received light amount is insufficient.							
Operating ambi-ent illumination		Illumination on received light surface: 3,000 lx or less (incandescent light)							
Ambient temperature		Operating: 0 to 50°C, storage: -15 to 60°C (with no icing or condensation)							
Ambient humidity		Operating and storage: 35% to 85% (with no condensation)							
Degree of protection		Cable length 0.5 m: IP66, cable length 2 m: IP67						IP40	
Materials		Case: Aluminum die-cast, front cover: Glass							
Cable length		0.5 m, 2 m							
Weight		Approx. 350 g						Approx. 400 g	
Accessories		Laser labels (1 each for JIS/EN, 3 for FDA), ferrite cores (2), insure Locks (2), instruction sheet						Laser safety labels (1 each for JIS/EN),ferrite cores (2), insure locks (2)	

[1] Defined as $1/e^2$ (13.5%) of the center optical intensity at the actual measurement center distance (effective value). The beam diameter is sometimes influenced by the ambient conditions of the workpiece, such as leaked light from the main beam.

[2] This is the error in the measured value with respect to an ideal straight line. The standard workpiece is white aluminum ceramics in diffuse reflection mode and glass in the regular reflection mode of the ZS-LD20T/40T/50. Linearity may change according to the workpiece.

[3] This is the peak-to-peak displacement conversion value in the displacement output at the measuring center distance in high-precision mode when the number of samples to average is set to 128 and the measuring mode is set to the high-resolution mode. The standard workpiece is white aluminum ceramics in diffuse reflection mode and glass in the regular reflection mode.

[4] This is the value obtained at the measuring center distance when the Sensor and workpiece are fixed by an aluminum jig.

[5] This value is obtained when the measuring mode is set to the high-speed mode.

Item	Model	ZS-LD50		ZS-LD50S		ZS-LD80		ZS-LD130		ZS-LD200		ZS-LD350S
Applicable controllers		ZS-HLDC/LDC series										
Optical system (reflection)		Diffuse	Regular	Diffuse	Regular	Diffuse	Regular	Diffuse	Regular	Diffuse	Regular	Diffuse
Measuring center distance		50 mm	47 mm	50 mm	47 mm	80 mm	78 mm	130 mm	130 mm	200 mm	200 mm	350 mm
Measuring range		±5 mm	±4 mm	±5 mm	±4 mm	±15 mm	±14 mm	±15 mm	±12 mm	±50 mm	±48 mm	±135 mm
Light source		Visible semiconductor laser (wavelength: 650 nm, 1 mW max., JIS Class 2)										
Beam shape		Line beam		Spot beam		Line beam		Line beam		Line beam		Spot beam
Beam diameter[1]		900x60 µm		50 µm dia.		900x60 µm		600x70 µm		900x100 µm		240 µm dia.
Linearity[2]		±0.1%F.S.							±0.25% F.S.	±0.1%F.S.	±0.25% F.S.	±0.04% F.S.
Resolution[3]		0.8 µm		0.8 µm		2 µm		3 µm		5 µm		20 µm
Temperature characteristic[4]		0.02% FS/°C		0.02% FS/°C		0.01% FS/°C		0.02% FS/°C		0.02% FS/°C		0.04% FS/°C
Sampling cycle[5]		110 µs (high-speed mode), 500 µs (standard mode), 2.2 ms (high-precision mode), 4.4 ms (high-sensitivity mode)										
Indica-tors	NEAR indicator	Lights near the measuring center distance, and nearer than the measuring center distance inside the measuring range. Flashes when the measurement target is outside of the measuring range or when the received light amount is insufficient.										
	FAR indicator	Lights near the measuring center distance, and further than the measuring center distance inside the measuring range. Flashes when the measurement target is outside of the measuring range or when the received light amount is insufficient.										
Operating ambient illumination		Illumination on received light surface: 3,000 lx or less (incandescent light)						Illumination on received light surface: 2,000 lx or less (incandescent light)		Illumination on received light surface: 3,000 lx or less (incandescent light)		
Ambient temperature		Operating: 0 to 50°C, storage: -15 to 60°C (with no icing or condensation)										
Ambient humidity		Operating and storage: 35% to 85% (with no condensation)										

Item	Model	ZS-LD50	ZS-LD50S	ZS-LD80	ZS-LD130	ZS-LD200	ZS-LD350S
Degree of protection		Cable length 0.5 m: IP66, cable length 2 m: IP67					
Materials		Case: Aluminum die-cast, front cover: Glass					
Cable length		0.5 m, 2 m					
Weight		Approx. 350 g					
Accessories		Laser labels (1 each for JIS/EN, 3 for FDA), ferrite cores (2), insure Locks (2), instruction sheet					

[1] Defined as 1/e² (13.5%) of the center optical intensity at the actual measurement center distance (effective value). The beam diameter is sometimes influenced by the ambient conditions of the workpiece, such as leaked light from the main beam.

[2] This is the error in the measured value with respect to an ideal straight line. The standard workpiece is white aluminum ceramics in diffuse reflection mode and glass in the regular reflection mode of the ZS-LD20T/40T/50. Linearity may change according to the workpiece.

[3] This is the peak-to-peak displacement conversion value in the displacement output at the measuring center distance in high-precision mode when the number of samples to average is set to 128 and the measuring mode is set to the high-resolution mode. The standard workpiece is white aluminum ceramics in diffuse reflection mode and glass in the regular reflection mode.

[4] This is the value obtained at the measuring center distance when the sensor and workpiece are fixed by an aluminum jig.

[5] This value is obtained when the measuring mode is set to the high-speed mode.

Sensor controllers

ZS-HLDC11/HLDC41

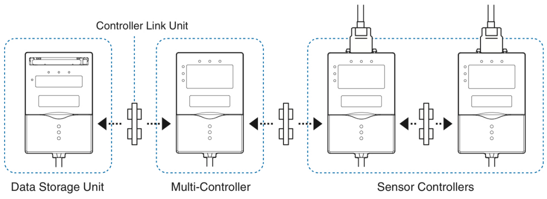
Sensor controllers		Model	ZS-HLDC11	ZS-HLDC41
NPN/PNP			NPN	PNP
No. of samples to average			1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1,024, 2,048, or 4,096	
Number of mounted sensors			1 per sensor controller	
External interface	Connection method		Serial I/O: connector, other: pre-wired (standard cable length: 2 m)	
	Serial I/O	USB 2.0	1 port, full speed (12 Mbps max.), MINI-B	
		RS-232C	1 port, 115,200 bps. max.	
	Output	Judgement output	HIGH/PASS/LOW 3 outputs NPN open collector, 30 VDC, 50 mA max., residual voltage 1.2 V max	HIGH/PASS/LOW: 3 outputs PNP open collector, 50 mA max., residual voltage 1.2 V max
		Linear output	Selectable from 2 types of output, voltage or current (selected by slide switch on bottom). Voltage output: .10 to 10 V, output impedance: 40 Ω Current output: 4 to 20 mA	
	Inputs	Laser OFF, ZERO reset timing, RESET	ON: Short-circuited with 0 V terminal or 1.5 V or less OFF: Open (leakage current: 0.1 mA max.)	ON: Short-circuited to supply voltage or within 1.5 V of supply voltage. OFF: Open (leakage current: 0.1 mA max.)
Functions			Display: Measured value, threshold value, voltage/current, received light amount, and resolution/terminal block output Sensing: Mode, gain, measurement object, head installation Measurement point: Average, peak, bottom, thickness, step, and calculations Filter: Smooth, average, and differentiation Outputs: Scaling, various hold values, and zero reset I/O settings: Linear (focus/correction), judgments (hysteresis and timer), non-measurement, and bank (switching and clear) System: Save, initialization, measurement information display, communications settings, key lock, language, and data load Task: Single task or multitask (up to 4)	
Status indicators			HIGH (orange), PASS (green), LOW (orange), LDON (green), ZERO (green), and ENABLE (green)	
Segment display		Main digital	8-segment red LED, 6 digits	
		Sub-digital	8-segment green LEDs, 6 digits	
LCD			16 digitsx2 rows, colour of characters: green, resolution per character: 5x8 pixel matrix	
Setting inputs		Setting keys	Direction keys (UP, DOWN, LEFT, and RIGHT), SET key, ESC key, MENU key, and function keys (1 to 4)	
		Slide switch	Threshold switch (2 states: High/Low), mode switch (3 states: FUN, TEACH, and RUN)	
Power supply voltage			21.6 V to 26.4 VDC (including ripple)	
Current consumption			0.5 A max. (when sensor head is connected)	
Ambient temperature			Operating: 0 to 50°C, storage: -15 to +60°C (with no icing or condensation)	
Ambient humidity			Operating and storage: 35% to 85% (with no condensation)	
Degree of protection			IP20	
Materials			Case: Polycarbonate (PC)	
Weight			Approx. 280 g (excluding packing materials and accessories)	
Accessories			Ferrite core (1), instruction sheet	

ZS-MDC11/MDC41 multi-controllers

Basic specifications are the same as those for the sensor controllers.
The following points, however, are different.
(1) Sensor heads cannot be connected.
(2) A maximum 9 of controllers can be connected. Control link units are required to connect controllers.
(3) Processing functions between controllers: Math functions

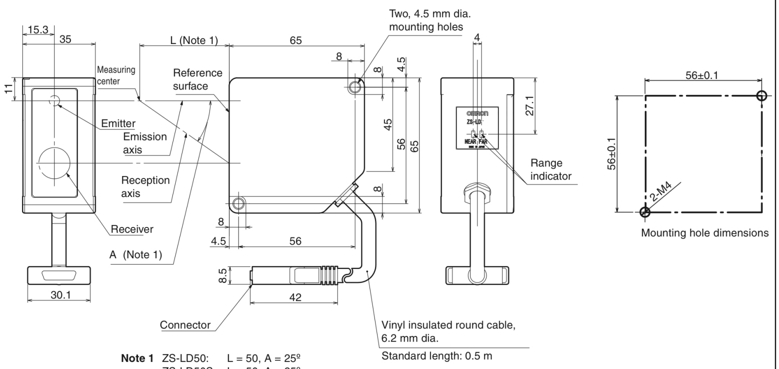
Controller link unit

Connection using the ZS-XCN



Sensor heads

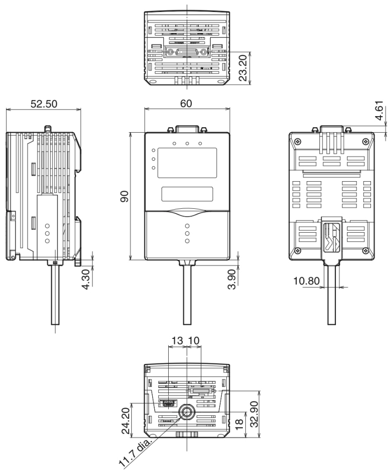
ZS-LD50/LD50S/LD80/ZS-LD130/LD200/ZS-LD350S



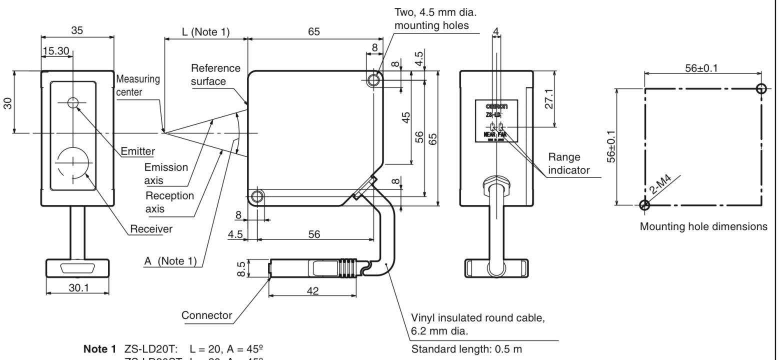
Note 1 ZS-LD50: L = 50, A = 25º
ZS-LD50S: L = 50, A = 25º
ZS-LD80: L = 80, A = 15º
ZS-LD130: L = 130, A = 12º
ZS-LD200: L = 200, A = 8º
ZS-LD350S: L = 350, A = 5º

Sensor controllers

ZS-HLDC11/HLDC41

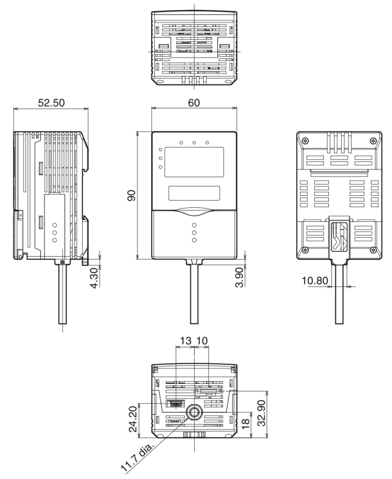


ZS-HLDS2VT/LD20T/LD20ST/LD40T

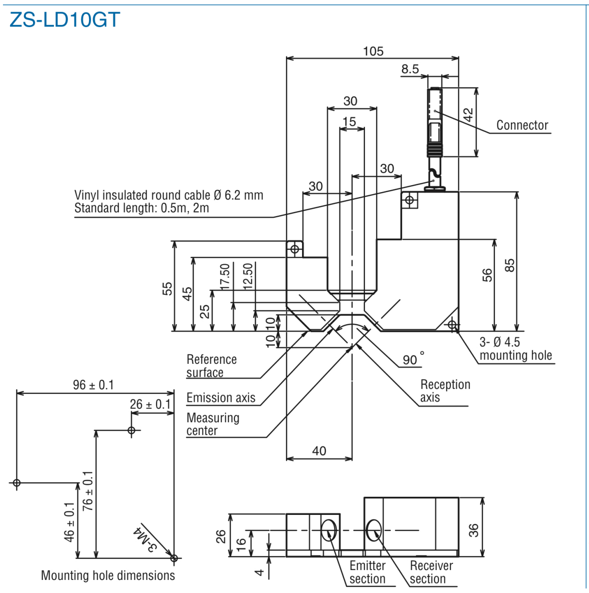


Note 1 ZS-LD20T: L = 20, A = 45º
ZS-LD20ST: L = 20, A = 45º
ZS-LD40T: L = 40, A = 32º

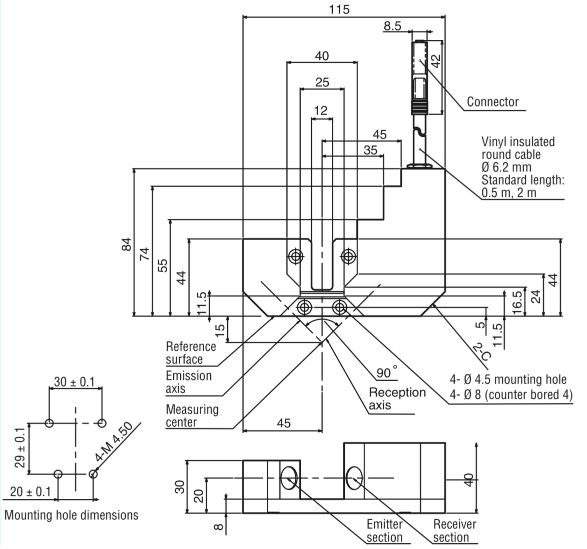
ZS-MDC11/MDC41 Multi-Controllers



ZS-LD10GT



ZS-LD15GT



I - Preliminary study report

NTNU – Trondheim
Norwegian University of
Science and Technology

Preliminary study report
Improvement of the positioning accuracy of industrial robots

Stud.techn. Vegard Johnsrud

Spring 2014

Department of Production and Quality Engineering
Norwegian University of Science and Technology

Main supervisor: Professor Knut Sørby
External supervisor: Prof. Ph.D. Trygve Thomessen

Preface

In the 10th and final semester of the 5-year master degree in Mechanical Engineering at NTNU, the students will write their master thesis (course code: TPK4900). The master thesis is rewarded 30 units of credit, which corresponds to an approximate workload of 48 hours per week. The thesis will result in a written report, with additional appendixes and other material produced through the project. The thesis is evaluated based on the written report, but also the project management throughout the thesis will be given a lot of consideration. The report will be submitted to the Department of Production and Quality Engineering prior to its due on the 10th June 2014. Prof. Ph.D. Trygve Thomessen, managing director of PPM AS and Prof. Knut Sørby at IPK (NTNU), has created the problem description. Thus, the project is cooperation between the Department of Production and Quality Engineering and high-tech robot system integrator PPM AS. The thesis will be carried out by stud.techn Vegard Johnsrud on behalf of NTNU and PPM AS, and will contain both theoretical studies as well as practical implementations.

The problem concerns the positioning accuracy of industrial robots. Because of Industrial robots low-accuracy, compared to high accuracy machines such as CNC machines and CMM's, tasks usually consist of material handling and spot welding operations where it utilizes the robots high repeatability. In the mentioned operations the locations the robot is moving between are manually taught to the robot through *teach-in-programming.* The accuracy problem arises when the robot motion is specified from Cartesian coordinates rather than taught positions, example of these kinds of operations is slot milling and arc-welding operations, where the robot is programmed to follow a specific line, not just move between positions. The goal of this thesis is to be able to perform milling operations with a dimensional accuracy better than ± 0.1 mm on the NACHI MC70 Industrial robot in PPM AS research facility in Fossegrenda, Trondheim.

The preliminary report is a mandatory part of the master thesis. The purpose of the preliminary study is to analyze the project description, acquire early overview of the different tasks at hand and to develop a schedule for the future work. The schedule shall outline the approximate work effort of different tasks and state the project milestones. The report will serve as an important tool in the project management aspect, and help the candidate to progress throughout the project.

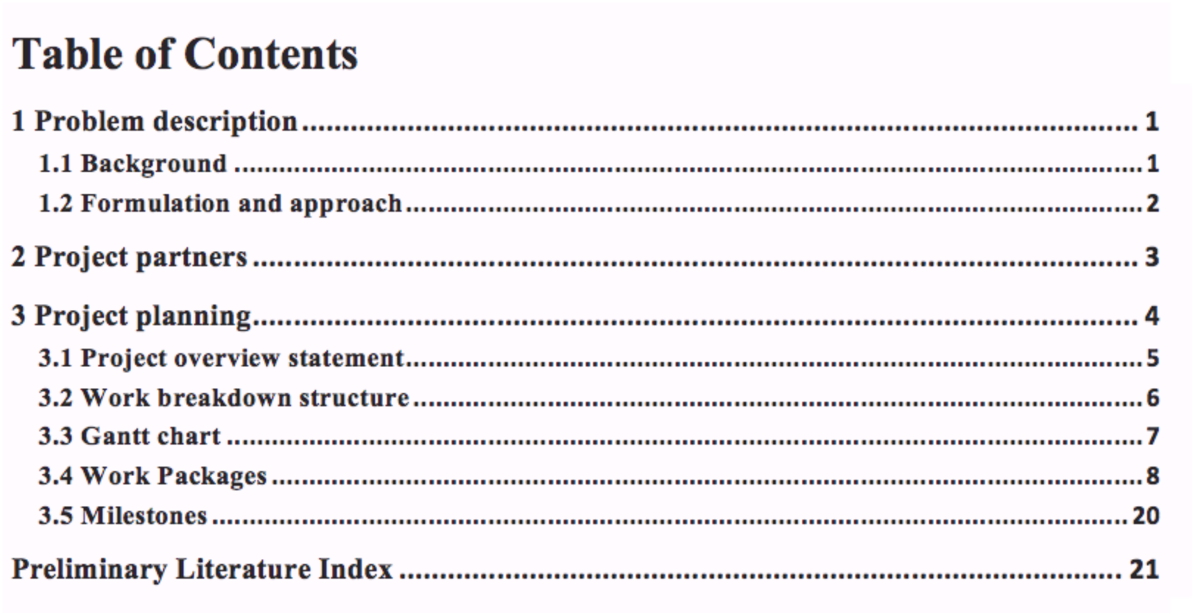


Table of Contents

Acronyms

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CMM	Computer Measurement Machine
CNC	Computer Numerical Control
IPK	Institutt for Produksjon og Kvalitetsteknikk (English: Department of Production and Quality Engineering)
PPM	Productive Programming Methods
WBS	Work Breakdown Structure
WP	Work Package

1 Problem description

This section of the preliminary study report, contain the description of the problem this master thesis is to solve. The description is divided into two sections the first being the background for the problem and finally the problem is formulated and the way the candidate will approach the problem is described.

1.1 Background

Industrial robots are an important component of modern industrial automation. They are capable of several operations, but mainly used on highly repetitive tasks such as pick and place operations. Also, robots are used in more complex operations such as welding and assembly operations. One thing these operations have in common is that they can be taught manually to the robot through “Teach-in-programming”. What makes the robot ideal for this kind of operations is its high repeatability and that the robots position between the points do not require high accuracy, as long as collisions is avoided.

When it comes to more complex operations such as part milling, “teach-in-programming” becomes too complex and time- consuming, therefore CAM data is used. This implies programming the robot using Cartesian coordinates instead of positions manually taught by the operator. Also, the robot is programmed to follow a path rather than move between points. The robots ability to follow a programmed path is governed by the accuracy of the robot, where the accuracy is the maximum deviation between the programmed path and the actual path measured in the Cartesian plane.

Compared to dedicated CNC machining centers the robots accuracy is poor, but what the robot lacks in accuracy it makes up in flexibility. As of this many industries will benefit from having a robot to perform highly repetitive task together with high accuracy task in the same cell.

1.2 Formulation and approach

The master thesis will be a continuation of the candidate's specialization project (course code: TPK4510), carried out in the fall of 2013, also on behalf of PPM AS and NTNU. In this project the candidate, together with Hungarian B.Sc. Nyirő Péter, set up a robot-milling cell in PPM's research facility, where the NACI MC70 industrial robot was used to mill parts drafted in CAD software, using cutting trajectories generated with CAM software. The parts milled in the cell disclosed dimensional errors from the CAD model in the range of 0.5 to 1.1 mm. As of this, the master thesis will focus on finding and eliminating the sources of this error. Thus, the goal of the thesis is to be able to perform milling operations with a dimensional accuracy better than $\pm$ 0.1 mm on the NACHI MC70.

To achieve this goal a literature study is to be conducted on robot accuracy and robot calibration methods. In the accuracy part of the study, reason of inaccuracies and the influence of these will be analyzed. ISO standards defining how to measure and assess robot performances will be thoroughly studied and used as a framework for the practical work throughout the project. In the second part of literature study, robot calibration will be studied. Robot calibration is the tool to help improving the robots accuracy, and the study will involve the various calibration methods. In the practical study the accuracy testing- and calibration methods will be studied in a more practical manner with respect to the NACHI MC70. The practical study will result in a list of accuracy testing- and calibration methods to perform on the NACHI MC70 at PPM AS' research facility.

Before testing the positioning accuracy and calibrating the robot, the candidate must verify that the robots repeatability and path accuracy is not a influencing the positioning accuracy. This will be verified experimentally. Further on the positional accuracy will be tested before and after the calibration in order to compare the results. As a final verification the robot will perform a milling operations, which will be measured in the CMM machine at IPK to verify if the thesis goal has been reached.

2 Project partners

This chapter gives a brief description of the project partners involved in this project.

NTNU

The Norwegian University of Science and Technology is one of the largest university in Norway and the university in Norway with the main responsibility for higher education in the field of science and technology (NTNU 2013). Of the 22 000 students, about half is studying technological subjects. Situated underneath the faculty of Engineering Science and Technology, is the Department of Production and Quality Engineering. The institute focuses their research and education within four areas; Production systems, Production Management, Project and Quality Management and Reliability, Availability, Maintainability and Safety (RAMS) (NTNU no date). The department has strong relationships with the industry and broad experience in cooperation with the industry for the student's project- and master thesis.

PPM AS

In December 2000, Dr.ing. Trygve Thomessen and Siv.ing Per Kristian Sannæs founded PPM (Productive Programming Methods) (PPM 2007). The company focuses on R&D projects with respect to productivity improvement in low batch production. The company has close relations to several universities and research institutes, and has been involved in several national and international R&D projects.

3 Project planning

This chapter focuses on the project management aspect of the master thesis. In order to get sufficient overview of the various tasks and generate an initial project plan, several project-planning tools have been used:

Project overview statement
An overview of the thesis problem, goals, success criteria's, conditions, risks and obstacles.

Work Breakdown Structure (WBS)
In this chart the project is broken down into smaller work packages that needs to be completed, in order to complete the project. The chart shows an overview of the project in a systematical and logic manner, and is a helpful tool in order to identify the important tasks of the project.

Work Packages
This is used to give a more comprehensive view of selected work packages (WPs) in the WBS.

Gantt
A chart, showing the various activities of the project versus time. The Gantt chart is created using Microsoft Project 2010.

Milestones
Listing the milestones of the thesis.

These project management tools are described in Rolstadås (2006) and Pinto (2010).

3.1 Project overview statement

Project Overview Statement

Project:	Improvement of the positioning accuracy of industrial robots	**Responsible:**	Stud.techn. Vegard Johnsrud

Problem:

In the candidates (see Responsible) specialization project carried out in the fall of 2013, on behalf of PPM AS and NTNU, a robot-milling cell was set up in PPM AS' research facility, where the NACI MC70 industrial robot was used to mill parts drafted in CAD software, using cutting trajectories generated with CAM software. The parts milled in the cell disclosed dimensional errors from the CAD model in the range of 0.5 to 1.1 mm. As of this, the master thesis project will focus on finding and eliminating the sources of this error.

The project will consist of both theoretical studies and practical implementation in the fields of accuracy testing and calibration methods for industrial robots. The ISO standards for robot testing will serve as the framework for the practical implementations and testing throughout the project.

Main Goal:

Perform milling operations with a dimensional accuracy better than ± 0.1 mm on the NACHI MC70 Industrial robot in PPM AS research facility in Fossegrenda, Trondheim.

Secondary Goals:

- Write a preliminary report
- Develop a thorough procedure for testing the positioning accuracy on the NACHI MC70
- Develop a thorough procedure for calibration of the NACHI MC70
- Be able to follow the progress in the same pace outlined in the project plan
- Write a status report
- Write a final report

Success Criteria:

- The produced material meet the expectations of the partners in the project
- The produced material shall be adaptable to a paper that can be published.
- Get top grade

Conditions, Risks and Obstacles:

Conditions:

- The candidate can get sufficiently knowledge about the subject
- The candidate has access to required hardware

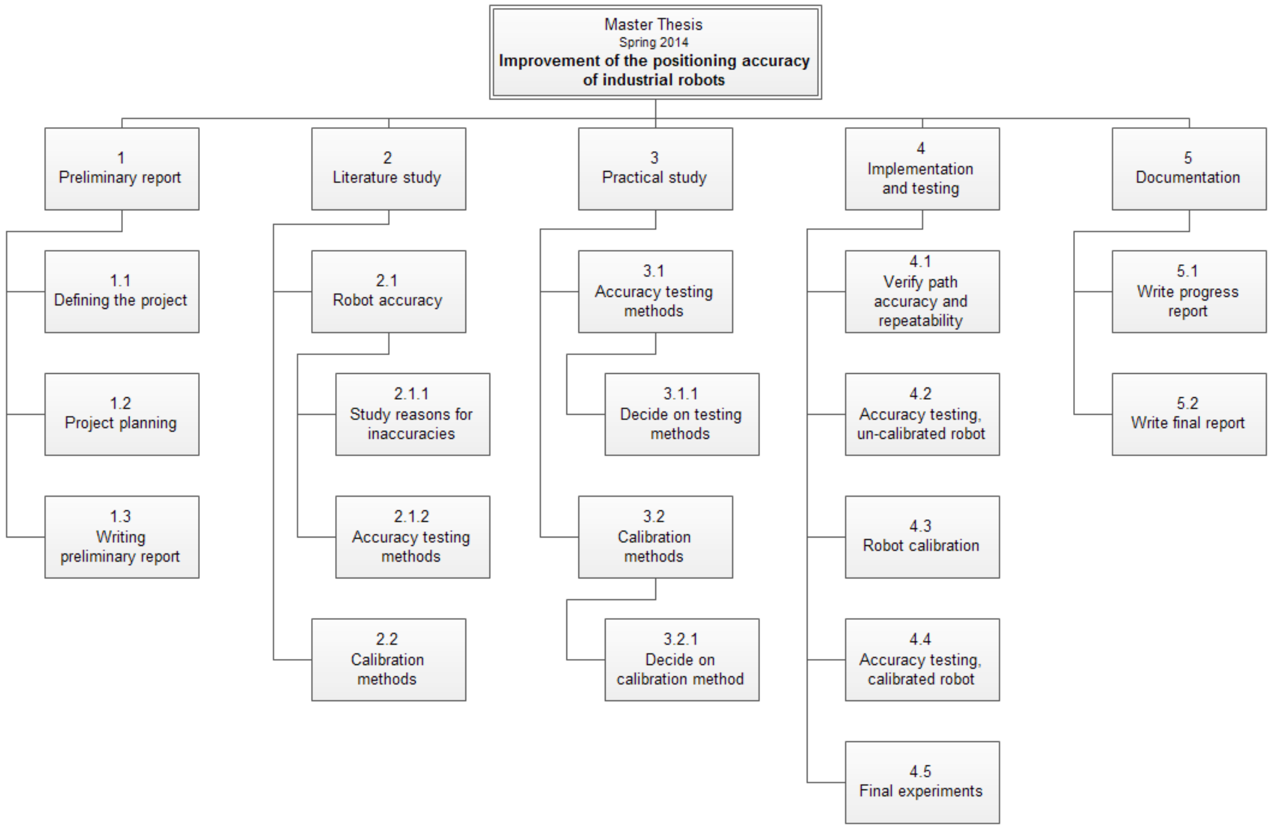
Risks:

- Sickness
- The magnitude of the project is proved to be too comprehensive

Obstacles:

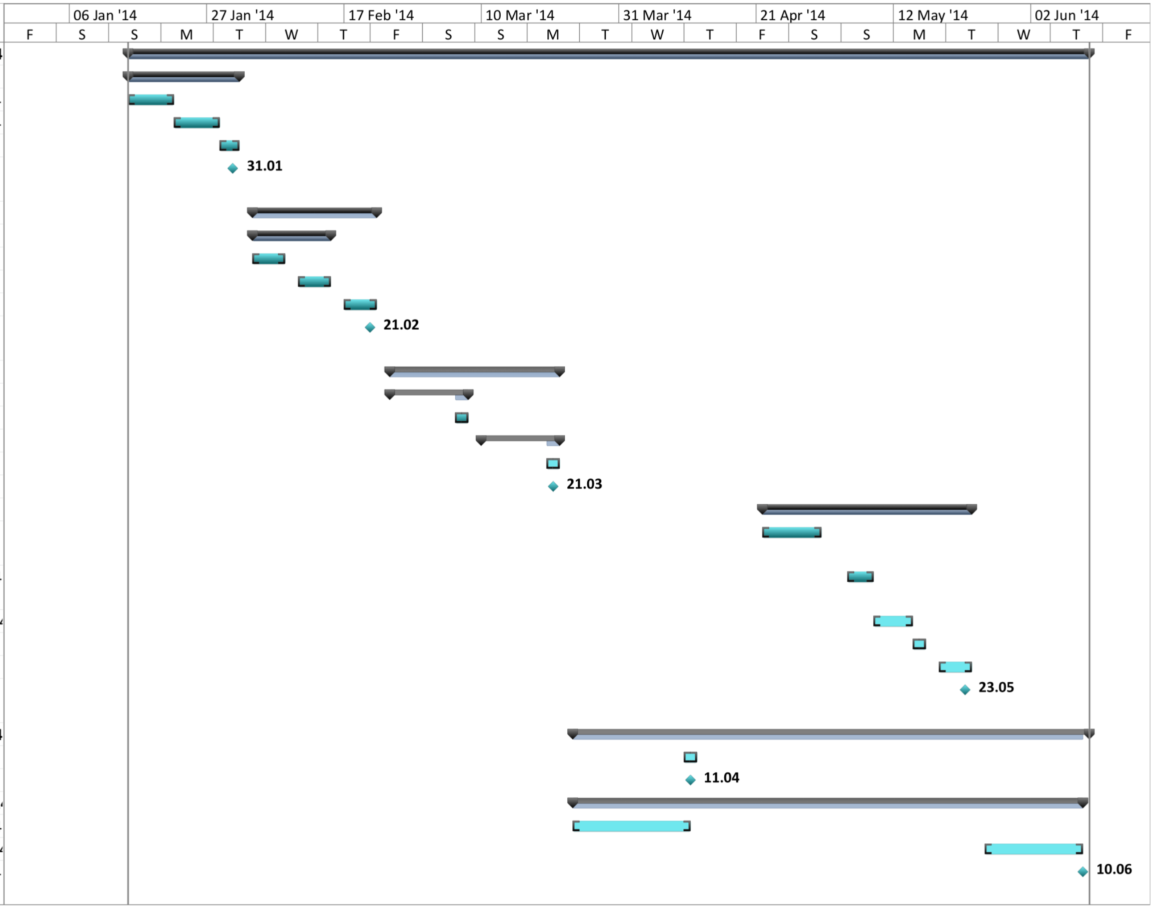
- Logistics

3.2 Work breakdown structure


Master Thesis
Spring 2014
Improvement of the positioning accuracy of industrial robots
1 Preliminary report
1.1 Defining the project
1.2 Project planning
1.3 Writing preliminary report
2 Literature study
2.1 Robot accuracy
2.1.1 Study reasons for inaccuracies
2.1.2 Accuracy testing methods
2.2 Calibration methods
3 Practical study
3.1 Accuracy testing methods
3.1.1 Decide on testing methods
3.2 Calibration methods
3.2.1 Decide on calibration method
4 Implementation and testing
4.1 Verify path accuracy and repeatability
4.2 Accuracy testing, un-calibrated robot
4.3 Robot calibration
4.4 Accuracy testing, calibrated robot
4.5 Final experiments
5 Documentation
5.1 Write progress report
5.2 Write final report

3.3 Gantt chart

ID	Task Name	Duration	Start	Finish
1	**Master Thesis, Spring 2014**	**105 days**	**Wed 15.01.14**	**Tue 10.06.14**
2	**1. Preliminary report**	**13 days**	**Wed 15.01.14**	**Fri 31.01.14**
3	1.1 Defining the project	5 days	Wed 15.01.14	Tue 21.01.14
4	1.2 Project Planning	5 days	Wed 22.01.14	Tue 28.01.14
5	1.3 Writing the report	3 days	Wed 29.01.14	Fri 31.01.14
6	*Milestone 1: Preliminary report finished*	0 days	Fri 31.01.14	Fri 31.01.14
7	**2. Literature study**	**15 days**	**Mon 03.02.14**	**Fri 21.02.14**
8	**2.1 Robot accuracy**	**10 days**	**Mon 03.02.14**	**Fri 14.02.14**
9	2.1.1 Study reasons for inaccuracies	5 days	Mon 03.02.14	Fri 07.02.14
10	2.1.2 Accuracy testing methods	5 days	Mon 10.02.14	Fri 14.02.14
11	2.2 Calibration methods	5 days	Mon 17.02.14	Fri 21.02.14
12	*Milestone 2: Literature study finished*	0 days	Fri 21.02.14	Fri 21.02.14
13	**3. Practical Study**	**20 days**	**Mon 24.02.14**	**Fri 21.03.14**
14	**3.1 Accuracy testing methods**	**10 days**	**Mon 24.02.14**	**Fri 07.03.14**
15	3.1.1 Decide on testing methods	2 days	Thu 06.03.14	Fri 07.03.14
16	**3.2 Calibration Methods**	**10 days**	**Mon 10.03.14**	**Fri 21.03.14**
17	3.2.1 Decide on calibration method	2 days	Thu 20.03.14	Fri 21.03.14
18	*Milestone 3: Practical study finished*	0 days	Fri 21.03.14	Fri 21.03.14
19	**4. Implementation and testing**	**24 days**	**Tue 22.04.14**	**Fri 23.05.14**
20	4.1 Verify path accuracy and repeatability	7 days	Tue 22.04.14	Wed 30.04.14
21	4.2 Accuracy testing, un-calibrated robot	4 days	Mon 05.05.14	Thu 08.05.14
22	4.3 Robot calibration	4 days	Fri 09.05.14	Wed 14.05.14
23	4.4 Accuracy testing, calibrated robot	2 days	Thu 15.05.14	Fri 16.05.14
24	4.5 Final experiments	5 days	Mon 19.05.14	Fri 23.05.14
25	*Milestone 5: Implementation and testing finished*	0 days	Fri 23.05.14	Fri 23.05.14
26	**5. Documentation**	**57 days**	**Mon 24.03.14**	**Tue 10.06.14**
27	5.1 Status report	2 days	Thu 10.04.14	Fri 11.04.14
28	*Milestone 4: Status report finished*	0 days	Fri 11.04.14	Fri 11.04.14
29	**5.2 Final report writing**	**56 days**	**Mon 24.03.14**	**Mon 09.06.14**
30	Part 1	14 days	Mon 24.03.14	Thu 10.04.14
31	Part 2	11 days	Mon 26.05.14	Mon 09.06.14
32	*Milestone 6: Master thesis delivered*	0 days	Tue 10.06.14	Tue 10.06.14



3.4 Work Packages

Work Package 1	
Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Preliminary report
Title: Preliminary report	**Work Place:** PPM and IPK
Description of Work: The purpose of writing this report is to create an early overview of the project, to identify problems to be solved, list goals and limitations and to create a detailed project plan. Project planning is an important part of the thesis evaluation, and shall include a Work Breakdown Structure (WBS), a Gantt chart and a list of the project milestones. The preliminary study report is to be handed in to the supervisor before it's due on Monday 04.02.14. The report will also be included in the appendix of the final report.	
Planned start: 15.01.14	**Planned finish:** 31.01.14
Revision number: V0.1	**Date:** 22.01.14

Work Package 2.1

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Literature study
Title: Robot accuracy	**Work Place:** PPM and IPK

Description of Work:

In this work package the candidate will conduct a theoretical study in the field of robot accuracy. A literature search will be performed in order to find sufficient information. The work package is divided into two sub-packages. In the first sub-package the candidate will study the different factors that contribute to inaccuracies and the magnitude of their contribution. The different kind of accuracy definitions and criteria's will also be studied. In the second sub-package a study of the accuracy testing study will be conducted, this will create the foundation for the practical study of same topic. Study of ISO standards is of special importance in this work package.

Planned start: 03.02.14	**Planned finish:** 14.02.14
Revision number: V0.1	**Date:** 22.01.14

Work Package 2.2

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Literature study
Title: Robot calibration	**Work Place:** PPM and IPK

Description of Work:

In this work package the candidate will conduct a theoretical study in the field of robot calibration. The candidate will explore the different calibration methods available on the market, evaluate their strengths and weaknesses, and consider the possibility for implementation to the robot cell in PPM AS' research facility. A literature search will be performed in order to find sufficient information. This will create the foundation for the practical study of the same topic.

Planned start: 17.02.14	**Planned finish:** 21.02.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 3.1

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Practical Study
Title: Accuracy testing methods	**Work Place:** PPM and IPK

Description of Work:

In this work package the candidate will conduct a practical study of the available testing methods. This will serve as a continuation of the literature study performed in Work package no.2 The work related to this WP involves usage and learning of the test equipment. The main focus will be on positioning accuracy testing, but path accuracy and repeatability testing will also be studied as these parameters needs to be tested prior to the positioning accuracy testing. An important task of this WP is how to implement and perform the test procedures in PPM AS' research facility. Lacking hardware must be revealed and acquired as efficiently as possible, in order to progress according to the schedule.

The WP will result in the accuracy testing methods performed on the NACHI MC70 industrial robot.

Planned start: 24.02.14	**Planned finish:** 07.03.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 3.2

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Practical Study
Title: Calibration methods	**Work Place:** PPM and IPK

Description of Work:

In this work package the candidate will conduct a practical study of the available calibration methods. This will serve as a continuation of the literature study performed in Work package no. 2 The work related to this WP involves usage and learning of the calibration equipment. An important task of this WP is how to implement and perform the calibration procedures in PPM AS' research facility. Lacking hardware must be revealed and acquired as efficiently as possible, in order to progress according to the schedule.

The WP will result in the calibration methods used on the NACHI MC70 industrial robot.

Planned start: 10.03.14	**Planned finish:** 21.03.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 4.1

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Implementation and Testing
Title: Verify path accuracy and repeatability	**Work Place:** PPM

Description of Work:

In this work package the candidate will verify the path accuracy and repeatability of the NACHI MC70. It is important to verify that the robots performance on these criteria's is within the value stated by NACHI and thereby not affecting the positioning accuracy.

Experiments will be conducted to verify the values. If these values prove unsatisfactory, they need to be taken into account when testing the positioning accuracy and calibrating the robot.

Planned start: 22.04.14	**Planned finish:** 30.04.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 4.2

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Implementation and Testing
Title: Accuracy testing, un-calibrated robot	**Work Place:** PPM
Description of Work: In this work package the candidate will perform the positioning accuracy tests studied in WP no. 2.1 and 3.1. During this WP the robot will not be calibrated.	
Planned start: 05.05.14	**Planned finish:** 08.05.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 4.3

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Implementation and Testing
Title: Robot calibration	**Work Place:** PPM

Description of Work:

In this work package the candidate will perform the calibration methods chosen in WP no. 3.2 on the NACHI MC70 in PPM AS' research facility.

Planned start: 09.05.14	**Planned finish:** 14.05.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 4.4

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Implementation and Testing
Title: Accuracy testing, calibrated robot	**Work Place:** PPM
Description of Work: In this work package the candidate will perform the positioning accuracy tests studied in WP no. 2.1, 3.1 and performed in WP 4.2. The robot will now be fully calibrated; as of this the calibrated test results can be compared to the un-calibrated test results.	
Planned start: 15.05.14	**Planned finish:** 16.05.14
Revision number: V0.1	**Date:** 22.01.14

Work Package no. 4.5

Project:

Improvement of the positioning accuracy of industrial robots

Sub-project:

Implementation and Testing

Title:

Final experiments

Work Place:

PPM & IPK

Description of Work:

In the final experiments, the calibrated robot will perform selected milling operations and the dimensional tolerance of the finished parts will be measured in the CMM at IPK. This experiment will reveal if the project reached its goal. The main tasks of this WP is:

- Creating the CAD models and CAM cutting trajectories for the experiments
- Performing the experiments in the cell
- Document the experiments with a video camera; the videos shall be appended to the final thesis in a DVD.

Planned start:

19.05.14

Planned finish:

23.05.14

Revision number:

V0.1

Date:

22.01.14

Work Package no. 5.1

Project: Improvement of the positioning accuracy of industrial robots	**Sub-project:** Documentation
Title: Write status report	**Work Place:** PPM & IPK

Description of Work:

In this package the candidate will write a report disclosing his progress together with an updated project plan. The report shall uncover the work that is completed and any changes and delays in the project plan. All the changes and deviation from the project plan needs to be approved by the supervisor.

The progress report is to be handed in to the supervisor during the thesis work, as well as being included in the appendix of the final report.

Planned start: 10.04.14	**Planned finish:** 11.04.14
Revision number: V0.1	**Date:** 22.01.14

<table>
<tr><td colspan="2"><h1>Work Package no. 5.2</h1></td></tr>
<tr><td><b>Project:</b>
Improvement of the positioning accuracy of industrial robots</td><td><b>Sub-project:</b>
Documentation</td></tr>
<tr><td><b>Title:</b>
Write final report</td><td><b>Work Place:</b>
PPM & IPK</td></tr>
<tr><td colspan="2"><b>Description of Work:</b>

The final report will summarize and present the research performed in the first stage of project (WP no. 2 and 3), show the test and calibration procedures performed on the NACHI MC70 (WP no.4) and present and discuss the results.

The report will be written in two parts; Part 1 will be written after the literature and practical study and Part 2 will be written after the final experiments.

The report is to be handed in before it’s due on Tuesday 10.06.14. The hand in consists of two bound paper copies and a digital (.pdf) copy.

The final report shall strive to meet the standard of a scientific report.</td></tr>
<tr><td><b>Planned start:</b>
Part 1: 24.03.14
Part 2: 26.05.14</td><td><b>Planned finish:</b>
Part 1: 10.04.14
Part 2: 10.06.14</td></tr>
<tr><td><b>Revision number:</b>
V0.1</td><td><b>Date:</b>
22.01.14</td></tr>
</table>

3.5 Milestones

#	Event	Date
1	Preliminary study report finished	31.01.14
2	Literature study finished	21.02.14
3	Practical study finished	21.03.14
4	Status report	11.04.14
5	Implementation and testing finished	23.05.14
6	Master Thesis delivered	10.06.14

After milestone no. 2,3 and 5 the candidate will schedule a "Milestone-meeting" where the work and progress is presented and discussed with the supervisors of the project.

Preliminary Literature Index

NTNU (2013). *Fakta om NTNU*. Retrieved 21.01.2014 from http://www.ntnu.no/tall-og-fakta

NTNU (no date). *Departement of Production and Quality Engineering*. Retrieved 21.01.2014 from http://www.ntnu.edu/ipk

Pinto J. K. (2010). *Project Management achieving competitive advantage*. 2nd ed. Pearson Education Inc.

PPM (2007). *Om PPM*. Retrieved 21.01.2014 from http://www.ppm.no/section.php?section=ppm&subsection=information

Rolstadås A. (2006). *Praktisk prosjektstyring*.

J – Changes made to preliminary study report

After the candidate had finished the scheduled literature study, a meeting was scheduled between candidate and the supervisors. On this meeting the various available measurement methods for measuring robot accuracy was discussed. It was pointed out that in order to gain a valid result, the measurement equipment had to have accuracy at a certain level above the robot and if this was not the case no conclusion could be drawn from any experiments. The meeting concluded with that performing accuracy measurements would be given a lower priority and be mainly covered by a theoretical study. This meant that the testing of robot repeatability and path accuracy, as described in the preliminary report, would not be performed.

On the other hand, the focus was shifted to robot calibration, which was given a higher priority. After the above-mentioned meeting a new meeting was arranged with Ph.D. Morten Lind, Senior Researcher at SINTEF Raufoss Manufacturing, who had implemented a joint offset calibration routine for Universe robot UR 5, using a tool mounted laser sensor and a calibration plane. It was therefore decided to implement his calibration software, written in Python, to a setup in PPM's research facility. This implementation includes a hardware setup and a software development, as the Python calibration software is to be translated into National Instruments LabVIEW, which is a more common programming language at PPM AS. Due to the amount of work with the practical implementations, part of the theoretical studies concerning robot accuracy and testing methods has been given lower priority, and should therefore by evaluated as such.

Due to the big changes of the project focus, the work packages of the preliminary report is less meaningful, and the project work should therefore not be evaluated on the basis of the preliminary report.