



University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program:	Spring 2023
Masters in Science and Engineering / Robot Technology and Signal Processing	Open
Author: Erlend Ettersti Thorkildsen	
Program coordinator: Morten Mossige	
Supervisor(s): Morten Mossige	
Title: Use of external sensors to modify robot path in real-time Credits: 30	
Keywords: Externally guided motion, EGM, real-time path correction, PixelPaint.	Pages: 52 + Attachments/other: Poster presentation The supervisor can be contacted for further source code Stavanger 15. June 2023

Abstract

This master thesis investigates the feasibility of using ABB's Externally Guided Motion (EGM) with path correction to adjust a pre-programmed robot path in real-time. The research explores whether the EGM controller, along with a displacement sensor, can improve the relative accuracy of robot movements with respect to an object. The primary objective is to maintain a constant distance from the surface and correct for any skew or unevenness during the robot's path execution. The research considers the potential of this system to be used in ABB's PixelPaint applications, where high accuracy with respect to objects is crucial.

The evaluation of the EGM system's performance considered two primary measures: relative accuracy and repeatability. Settling time, robot speed, and oscillations were also considered important to the performance. The performance was tested by subjecting the laser sensor to a step. Through an analysis of the results, the thesis demonstrates that the EGM system achieved high levels of repeatability for different robot speeds. However, the relative accuracy was insufficient. When using the sensor's analogue signal to correct the path, the correction suffered from oscillations. The thesis also highlights that when using the sensor as input, the correction stabilized at around half the measured displacement. The oscillations introduced a level of instability that affected the smoothness and accuracy of the robot's path. The fastest sampling rate for EGM Path Correction is 24ms and was used to maximize the robot's speed. However, when the speed exceeded 100mm/s the sensor struggled to measure a step and the robot's step response did not have time to settle before the trajectory ended.

In summary, this thesis contributes to the understanding of the use of external sensors for real-time path correction in robotics by evaluating ABB's EGM Path Correction. It highlights the constant displacement the controller did not correct, and it considers the presence of oscillations and the limitations imposed by the maximum sampling rate. These challenges require more research to improve the performance before EGM Path Correction could be applied to applications such as PixelPaint.

Preface

This thesis is the final work of the master's program in Robot Technology and Signal Processing at the University of Stavanger. The thesis is a collaboration with ABB Robotics Bryne, who has contributed with equipment and supervision throughout the project. All experimental work has been conducted at their facilities in Bryne.

First and foremost, I would like to thank my supervisor, Morten Mossige, for the help and feedback, and for putting me in contact with the correct experts during the process. This brings me to thank the experts at ABB Robotics Bryne who have taken the time to answer my questions on your field of expertise.

Contents

1	Introduction	1
1.1	ABB Robotics Bryne	1
1.2	Industrial Painting	1
1.3	Pixel painting	1
1.4	The idea of the project	1
1.5	Defining the tasks	2
1.6	Report structure	3
2	Prerequisites	4
2.1	Robotic Manipulators	4
2.1.1	Singularities in manipulators	5
2.1.2	Calibration and accuracy	6
2.1.3	External sensors on industrial robots	6
2.2	Paint robots	7
2.2.1	PixelPaint	7
2.3	Externally Guided Motion - EGM	9
2.4	Laser displacement sensors	11
2.4.1	The sensor for EGM Path Correction	12
2.5	Signal analysis tools	13
2.5.1	RobView for studying the robot's accuracy	13
2.5.2	Oscilloscope for sampling analogue displacement signals	14
3	Implementation	15
3.1	Work environment	15
3.1.1	The robotic manipulator	15
3.1.2	The end-effector and its coordinate systems	16
3.1.3	Work object	17
3.1.4	The path for testing	17
3.2	Sensor mount and considerations around the analogue input	18
3.3	EGM using Path Correction	21
3.4	The RAPID programs	23
3.4.1	Signal analysis program	24
3.4.2	EGM analogue output program	24
3.4.3	EGM analogue input program	25
3.5	Reading sensor data in Python	25
4	Experiments and results	27
4.1	Analogue Signal	27
4.1.1	Step response with 50mm/s speed	28
4.1.2	Step response with 100mm/s speed	29
4.1.3	Step response with 200mm/s speed	30
4.1.4	Analysis of the results	31
4.2	Externally guided motion with analogue output	32
4.2.1	EGM with analogue output with 50mm/s speed	32
4.2.2	EGM with analogue output with 100mm/s speed	34
4.2.3	EGM with analogue output with 200mm/s speed	35
4.2.4	Analysis of the results	36
4.3	Externally guided motion with analogue input	37

CONTENTS

4.3.1	EGM with analogue input with 50mm/s speed	37
4.3.2	EGM with analogue input with 100mm/s speed	40
4.3.3	EGM with analogue input with 200mm/s speed	42
4.3.4	Analysis of the results	44
5	Discussion	45
5.1	The workspace and setup	45
5.2	The analogue signal	46
5.3	EGM with analogue output	47
5.4	EGM with analogue input	47
6	Conclusion and future work	49
6.1	Conclusion	49
6.2	Future work	49
A	Appendix	53

1 Introduction

1.1 ABB Robotics Bryne

ABB Robotics Bryne is an advanced research and development center and holds responsibility over ABB's paint robot R&D. They have skilled developers with a wide area of knowledge. They specialize in embedded systems, robotics control, process control, and electronic design. The team is currently developing a new robot control system. ABB Robotics Bryne has over several years invited students to join their research and development team and this project is a part of this journey to develop the new robot controller.

1.2 Industrial Painting

Over the last few decades, the field of robotics has grown, and the area in which robots are being used has increased. Industrial robots have shown great potential in the field of painting. In particular, the use of robots for painting automobiles has been gaining popularity due to their efficiency, accuracy, and repeatability. Robots can move at a constant speed, which leads to a more even coating. Robots also work faster and companies can therefore become more cost-efficient on time and wasted paint. Another important factor of robotic painting is that the workers' safety increases with respect to harmful fumes and chemicals [1] [2] [3] [4].

1.3 Pixel painting

Pixel painting involves creating an image by painting individual pixels with a specific colour. ABB has developed what they call *PixelPaint* with an inkjet that features more than 1000 nozzles, each controlled individually. This improves the painting precision significantly which increases the paint transfer efficiency to save paint, and the use of masking becomes redundant [5].

This technology could open up more complex paint jobs in the automotive industry, among other industries. In recent years there has been a growing interest in more complex paintings like two-tone paint or even artistic endeavors, like in Figure 1. Pixel painting opens up possibilities for customer customization of automotive paint [5].

The accuracy required in PixelPaint is high, and small errors can result in significant deviations from the desired result. It is desired to keep a constant distance of 3mm to apply PixelPaint to a surface. ABB's robots have high accuracy and move with constant speed, but moving along an object introduces other challenges. The robot is then required to move accurately with respect to the object subjected to the paint. The automotive industry is one target group for pixel painting, and cars are seldom identical on a millimeter level. To increase the accuracy with respect to another object, Externally Guided Motion is suggested.

1.4 The idea of the project

ABB Robots have an optional feature called *Externally Guided Motion* (EGM) [6] which overrides the planned path based on external input. With input from

1.5 Defining the tasks

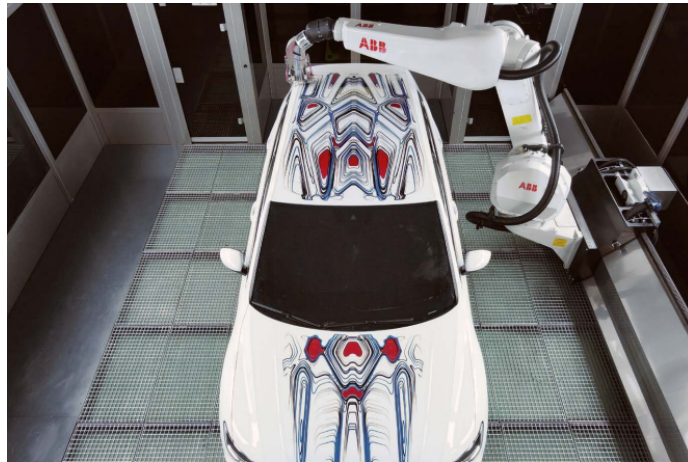


Figure 1: The figure shows an example of how ABB's PixelPaint work. The image is found in [5, Page: 2]

an external source into the control system, it provides partial control of the robot's movement in real-time. A problem in the automotive industry is that no automobile is exactly alike, and there will always be some deviance from car to car. The thought is that EGM could improve real-time accuracy, and facilitate for PixelPaint in the automotive industry. EGM could be used in PixelPaint applications to control the distance to the car subjected to the painting.

Through this project the accuracy of EGM will be studied. EGM's capabilities to keep a fixed distance from an object will be tested. Other factors resulting from EGM such as settling time, oscillations, error in path, smoothness, and repeatability will also be studied. The main focus will be to increase precision relative to an object and study the compensation performance of externally guided motion.

EGM offers solutions to the correction of offline generated trajectories, creating trajectories, and streaming the robot's position in real-time. For this project, a displacement laser sensor will be used as the external sensor. The idea is to study if EGM could improve the accuracy of a given ABB robot, and if it could keep a constant distance from the surface. Non-flat objects, in particular, step-responses, will be studied to see how EGM corrects the path in the most extreme situations.

1.5 Defining the tasks

The objective of the project is to explore the feasibility and advantages of using externally guided motion to improve the accuracy of ABB's robots in PixelPaint applications. The specific research questions that will be addressed in this thesis include:

- How is the analogue signal from the external sensor perceived and if it is affected by noise or other factors that need attention?
- Is externally guided motion suited for keeping a constant distance to an object whilst moving along the object?

1.6 Report structure

- Based on a given path along an object, how well can the robot adapt to changes in the path based on sensor input?
- How do settling time, path error, and smoothness affect the result, and how repeatable is the operation?
- How fast can the robot move along the object without compromising the quality of the result?

The outcome of this study could potentially have a significant impact on the field of pixel painting. By improving the precision of ABB robots in PixelPaint applications, this study could enable more accurate and efficient painting processes, leading to better results and reduced costs. In addition, the knowledge gained from this study could also have implications for other applications that require high-precision robotic systems.

1.6 Report structure

This report consists of six chapters. Chapter 1 is the introduction where the background and motivation are presented and the tasks are defined. In Chapter 2, the prerequisites, that are needed to read the report, are presented. Chapter 3 presents the implementation, including sub-chapters about the workspace, choice of externally guided motion functions, the analogue signal, and the RAPID programs. The experiments are described and the results are then presented in Chapter 4. Then the workstation and the results will be discussed in Chapter 5. and the report rounds off with a conclusion in Chapter 6.

2 Prerequisites

This chapter will present some prerequisites the reader will need before reading this report. It is assumed the reader has some knowledge of basic manipulator features and programming. This chapter is divided into sub-sections, covering some theory about robotics, and the hardware and software used in the project. The sub-sections cover the most important aspects relevant to this project. The theory in the chapter is based on application manuals and documentation for the different hardware and software. Articles and other reports have also been used as sources for the chapter.

2.1 Robotic Manipulators

Robots are being used in a wide field to solve different tasks and can be constructed vastly differently to fit a specific purpose. A robot can be defined as anything that works with some degree of autonomy, but through this report, the word robot will refer to an industrial robotic manipulator [3, Page: 1]. This chapter will offer a brief introduction to robotic manipulators, how they are built, and their area of application. Figure 2 offers an example of a 6 degrees of freedom (DOF) industrial robot with arrows explaining the joint movements.

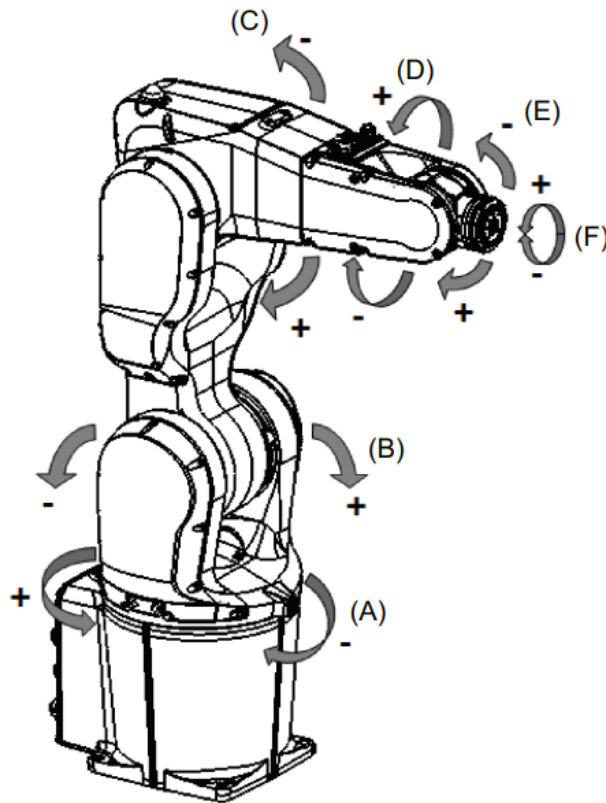


Figure 2: Example of a 6 DOF industrial robot [7, Page: 12]

2.1 Robotic Manipulators

An industrial robot manipulator is built by multiple rotational and/or prismatic joints, where the displacement between two consecutive joints is called the joint variable. The combination of rotational and prismatic joints characterizes the type of robot and its application area. Typically the manipulator has an arm to provide mobility, a wrist that provides dexterity, and an end-effector that performs the given tasks [4, Page: 4].

By exploiting accurate knowledge of the joint variables, the kinematic and dynamic models of the robot can be computed. Accuracy for robotic manipulators is the measure of how close the tool center point (TCP) can come to a given point. Through inverse kinematics, the joint variables of the robot can be computed, given a known end-effector point. To measure positional errors position encoders in the joints are used. There are typically no direct measurements of the position of the end-effector. The accuracy is therefore affected by computational errors, machining accuracy in the manipulator's construction, and flexibility in joints and structure [3, Page: 7-8], [4, Page: 87]. To further increase the accuracy of the robot, it is stated in [3, Page: 8] that *accuracy can only be increased by some sort of direct sensing of the end-effector position*. This project will study increased accuracy by using a laser displacement sensor at the TCP.

The workspace is every position and orientation the end-effector can reach when considering all available configurations –the combinations of joint variables. The workspace's shape and volume are characterized by the structure of the robot and its joint limits [3, Page: 5], [4, Page: 85-87]. In this report, the focus lies on a 6-axis robot from ABB, the IRB 1200, which is described in Chapter 3.1.1. ABB robots can be programmed in RobotStudio with the programming language RAPID.

In general, a robot is used to interact with an object within the workspace. For an object in three-dimensional space, the object has a position and an orientation, which sums to 6 DOF. For the robot to be able to reach any arbitrary point within the workspace and interact with the object, it needs at least 6 DOF. If the robot possesses more DOF than is the minimum required to complete the given task, it is considered to be redundant. A redundant robot is often preferred, with the intention to avoid singularities [3, Page: 4]. It also serves other purposes such as to increase dexterity, avoid reaching joint limits, and obstacle avoidance.

2.1.1 Singularities in manipulators

If the robot moves close to its singularities, it can affect the motion. Hence the singularities should be mapped before defining the path. Therefore, a brief definition of singularities in industrial robotics will be offered. The focus lies on the practical meaning of singularities. For a more detailed explanation of either singularities or the Jacobian matrix, the reader is referred to [3, Page: 122-139]

An example of a singularity is a cannon aiming, and following an airplane cruising through the air. Imagine a cannon that can rotate and yaw 360° in the horizontal plane. It can simultaneously rotate and pitch 90° in the vertical plane, pointing the barrel straight up at the maximum. If the airplane is moving in the air beside the cannon, the movement can be followed smoothly by the cannon. But the moment the airplane flies straight above the cannon, it can

2.1 Robotic Manipulators

only follow the plane until it is straight above. Then the cannon must turn 180° before the motion can continue. This is how singularities are experienced by robots as well.

Through computations of the robot dynamics the Jacobian matrix, $J(q)$, can be computed. This matrix maps the joint velocities, \dot{q} , and the end-effector velocities, $\zeta = (v, \omega)^\top$:

$$\zeta = J(q)\dot{q} \quad (1)$$

The manipulator's Jacobian describes the relations of motion between the different joints and the end-effector. When the robot has a configuration where the rank of $J(q)$ is less than its maximum, it is called a singularity. Singularities tend to correspond to points on the boundary of the workspace. Singularities can also be defined as points where the configurations of the robot are not unique -the robot does not know which configuration to choose in these positions, and one might experience that the robot "jumps" between configurations [3, Page: 132-133]. The example of the cannon describes this occurrence when the plane flies straight above the cannon. Then the cannon have pitched until it aims straight upward. To keep aiming at the airplane, the cannon must yaw 180° in an instance. This is a typical "jump" from one configuration to another. Singularities could limit and/or separate the workspace, and the robot should not access a singularity from one region to another. Therefore, singularities should be mapped and avoided in the workspace to ensure feasible and non-interrupted robot movement [8].

2.1.2 Calibration and accuracy

At a pre-defined position, robots can typically achieve an end-effector accuracy of 0.005-0.10mm. This accuracy is at a given position and orientation but this does not necessarily apply to the movement along a trajectory. Improvements in accuracy are done by identifying a mathematical model that describes the joint variables in relation to the position of the end-effector. The new mathematical model has higher accuracy than the nominal mathematical model embedded in the robot controller. This type of calibration only improves the static accuracy of the robot and the path following precision is not improved. For applications like *pick and place*, the absolute precision is critical, but in applications like industrial painting, the relative accuracy is more important [9]. This will be revisited in chapter 2.2.

2.1.3 External sensors on industrial robots

When considering the accuracy of the robot and its environment, there are three factors that must be taken into account. The first is the robot's accuracy in itself, which was discussed in the last subsection. The second is the work object and its level of consistency with the CAD model. If the robot is supposed to work on an object, dissimilar to the model, the result could end up dissatisfactory. For instance, if a robot is welding an object that is supposed to be linear. If the object deviates from the linear model, the welded structure could end up weaker at dissimilar points in the object. The third factor is the accuracy in the position and orientation of the object. In automotive painting, the automobiles slide into the paint room on the assembly line, and position and/or orientation

2.2 Paint robots

could vary between each paint job. If the car is a few degrees, or maybe a few millimeters different from time to time, the paint job could deviate from the desired result.

The purpose of an external sensor on industrial robots is to provide the robot with information such that the controller can make "intelligent actions autonomously" [4, Page: 219]. The communication between the robot and the sensor contains information about the robot's position relative to the workspace. It is pivotal to detect objects in the workspace and know the distance to said objects. The area within which the sensor is able to detect is called *measuring or sensitivity range*. The data received from the sensor can be used to avoid obstacles, increase precision in tasks where the robot is interacting with the environment, recognize objects, or map the environment. Most frequently sensors based on sound propagation, *sonar*, or light propagation features, *laser*, are used in robot applications [4, Page: 2019]. Through this project, a laser displacement sensor has been used and the focus will therefore be on that type of sensor. This will be described in detail in Chapter 2.4.

2.2 Paint robots

As industrial robots are expensive, the industries that use paint robots often have a big production and/or need the precision and high-quality finish that robots can provide. The result of using painting robots is a uniform coating by moving with constant speed along the object. With human painting the movement cannot reach the same level of consistency, thus the coating becomes uneven. Robotic manipulators possess the ability of great repeatability, meaning the results of each paint job are equal. The speed of a robot also exceeds what is humanly possible. Robotic paint can be as much as 30 times faster than what humans are capable of. With all these benefits comes cost saving in terms of less wasted paint. Another important factor is that the health risk of the workers can be increased by operating less time in rooms with fumes and chemicals [1].

2.2.1 PixelPaint

ABB has developed PixelPaint, a technology that makes it industrially possible with customized paint jobs on automobiles. This simplifies two-tone paint jobs and makes more advanced paint jobs, such as individually designed images, possible. An example of such a customized paint job is shown in Figure 3. By being able to paint pixels independently with specific colours, pixel painting can recreate complicated designs onto an automobile. When painting in two colours with regular painting technologies, the base colour must dry before the masking can begin. Then the next colour can be painted, and the paint must dry again before the demasking can begin. PixelPaint removes the need for complicated and time-consuming masking, and demasking. This improves productivity, and as with other paint robots, the coating is more even [5].

2.2 Paint robots



Figure 3: The figure illustrates the possibilities in PixelPaint. The image is found in [5, Page: 1]

The PixelPaint inkjet has more than 1000 nozzles that can be controlled individually with high precision. This is a non-over-spray technology, transferring 100 per cent of the paint to the object, compared to the 80 per cent of electrostatic painting. The inkjet's droplet rate, pixel size, and nozzle pattern are shown in Figure 4. The droplets' sizes are $20\mu\text{m} - 50\mu\text{m}$ and are sprayed at a rate of up to 1000 droplets per second. This enables precise control of both overlapping, thickness of the coating, and sharp edges [5].

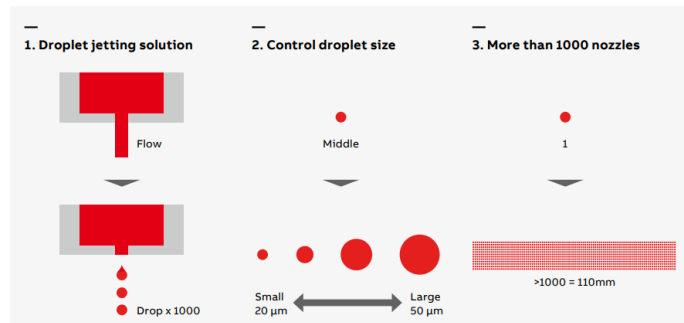


Figure 4: Descriptive diagram of the inkjet technology in PixelPaint, and an overview of pixel sizes. The image is found in [5, Page: 2]

To apply this new technology, it is required to move the inkjet precisely along the object subjected to the paint. The process control for the paint and the motion control of the robot must be controlled together. Poor accuracy in the robot movement, for spray painting applications, could affect the evenness of the coating, and even leave scars and bubbles in the paint. In the case of automotive painting, the automobile is different from the computer-aided design model, and each individual automobile deviates from the other [9]. The deviation could be large enough so that absolute accuracy offered from a pre-defined path is of no help. Even a small error can result in significant deviations from the design. However, the relative accuracy of the robot, with respect to the surface, can

2.3 Externally Guided Motion - EGM

be improved, as described in chapter 2.1. That is why the need for increased relative accuracy and motion control in real-time has been raised.

PixelPaint is also based on advanced process control technology, but this will not be discussed here, as it is outside the scope of the thesis. This report will rather study the relative accuracy of the robot when using externally guided motion.

2.3 Externally Guided Motion - EGM

Externally guided motion is a feature in robot control developed by ABB. Through an external device, such as a sensor, EGM offers low-level control of the robot's movement. EGM is connected directly to motion control, which bypasses the path planner, allowing the robot to react quickly to sensor input. Data from the external device can be read by the EGM-controller at multiples of 4ms when interacting with the environment [6], [10].

One way to consider EGM is as a control loop where the position of the robot, relative to the environment, is controlled. The data stream can flow in each direction based on the application. This stream may contain information about position and/or orientation, relative to the environment. ABB's technology offers three features for EGM [6]:

- **EGM - Position Stream:** The current and planned position of the robot is sent to external equipment.
- **EGM - Position Guidance:** The robot follows a path generated by an external source instead of the pre-programmed path from RAPID.
- **EGM - Path Correction:** The pre-programmed path from RAPID is modified/corrected using an external sensor.

Position Stream sends data packets, containing planned and current robot TCP position, to an external source, such as a data program. *Position Guidance* creates its own path in real-time based on an external device. This can be a camera feeding images to a data program that generates a path from the images. *Path Correction* follows a pre-programmed path while it corrects displacements along the y- and z-axis. For this project, the latter is used, and the remainder of this chapter will focus on Path correction.

Path correction will correct the pre-programmed path in real-time with respect to the environment. EGM increases the relative accuracy with respect to the environment and keeps the robot at a constant distance from objects. The external device is mounted on the robot's end-effector and generates correction data. The robot moves along the pre-programmed path with added measured corrections. The feature is used in applications such as seam tracking, or tracking objects moving near a known path. The maximum correction is $\pm 50\text{mm}$ from the original path [6].

Correction of the path can only be applied in the path coordinate system's y- and z-directions. The path coordinate system is defined in Figure 5 with X_P as the x-direction. This means that the correction is made vertically and/or horizontally, along Y_P and Z_P .

2.3 Externally Guided Motion - EGM

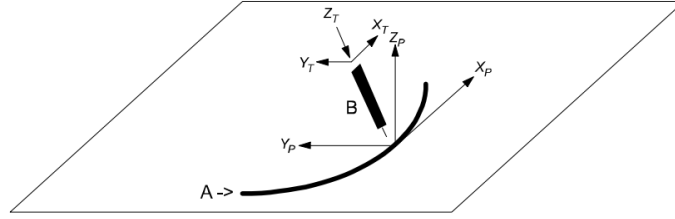


Figure 5: Definitions of path coordinate system and tool coordinate system [11, Page: 170].

Externally guided motion adds limitations to the implementation of the robot tasks. When using EGM, the robot path is constrained to start and finish at a fine point. The other points in EGM programs can be zones. This implies that EGM cannot be activated only during parts of the path. EGM could not suddenly be activated if something unexpected were to happen [10]. The external device has to be mounted on the robot, and cannot be mounted elsewhere in the environment. For EGM Path Correction, the sample time has to be a multiple of 24ms. The aforementioned sample time at a multiple of 4ms applies to Position Stream and Position Guidance. Another limitation of EGM is if the robot reaches a singularity. Then the EGM stops, and the robot must be moved out of the singularity manually to proceed with the task [6].

The approach for EGM Path Correction is written in Table 1 and found in [6, page: 22]

	Action
1	Move the robot to a fine point.
2	Register an EGM client and get an EGM identity. This identity is then used to link setup, activation, movement, deactivation, etc. to a certain EGM usage. The EGM state is still EGM_STATE_DISCONNECTED.
3	Call an EGM setup instruction to set up the position data source using signals or UdpUc protocol connection. The EGM state changes to EGM_STATE_CONNECTED.
4	Define the sensor correction frame, which always is a tool frame.
5	Perform the movement itself. Now the EGM state is EGM_STATE_RUNNING.
6	At the next fine point, EGM will return to the state EGM_STATE_CONNECTED.
7	To free an EGM identity for use with another sensor you have to reset EGM, which returns EGM to the state EGM_STATE_DISCONNECTED

Table 1: The table describes the basic approach to setting up EGM path correction and is found in [6, page: 22]

Figure 6 illustrates the data flow for the EGM Path Correction. The input data for EGM Path Correction can be selected through setup instructions. A

2.4 Laser displacement sensors

signal interface can be selected from analogue input, output, or group input. The sensor used for this project provides an analogue signal. EGM requests data from motion control and displacement data from the sensor through the I/O module. EGM then calculates the path correction and writes it to motion control [6, page: 26].

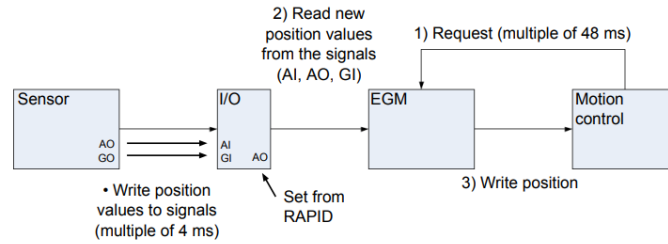


Figure 6: Externally Guided Motion simplified control loop. Source: [6, page: 26].

2.4 Laser displacement sensors

Laser sensors can either use the *time-of-flight* principal or the *triangulation* method to measure the distance. The *triangulation* method's accuracy can be influenced by the surface of the object, and/or the material in which the laser is reflected on. Differences, or change in colours or material of the surface can also affect the accuracy. This can be compensated by the sensor by automatically regulating the light intensity. With knowledge of the wavelength of the laser beam, it is possible to use selective filters to only allow the specific wavelength through. This reduces effects from other light sources and increases accuracy. *Time-of-flight* sensors are limited in accuracy because of the sensor's minimum observation time. The minimum distance observable is decided by the temporal accuracy of the receiver and the laser pulse width. This is not only a technological limitation but a budget limitation [4, Page: 219-225]. The sensor used in this project is the *Panasonic HL-G112-A-C5*, which uses the triangulation method to measure distance.

Measuring distance with a laser sensor based on trigonometric parameters is a simple principle but results in highly accurate measurements. This method uses a laser diode and a converging lens for the emitted beam. An optical filter, a receiving lens, and a position sensitivity detector (PSD) are used for the reflected beam. The trigonometry of the emitted and reflected laser beam results in the variables shown in Figure 7.

2.4 Laser displacement sensors

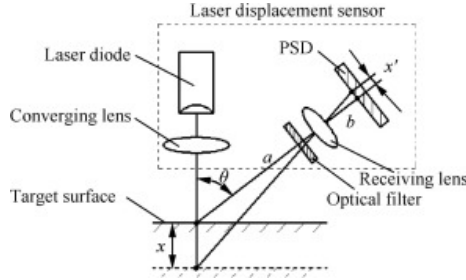


Figure 7: Measuring technique for optical trigonometry in displacement sensors [12]

The variables are further used to calculate the sensor's displacement, relative to the center of the measuring range. The following formula gives the displacement, x :

$$x = \frac{ax'}{bsin\theta - x'cos\theta}$$

The variables, θ and x' , are measured in the vertical plane. θ is formed by the angle between the emitted and the reflected laser beams. The displacement on the PSD is defined as the distance from the center of the PSD, created by the principal axis of the center of the measuring range, to the actual laser beam axis, projected onto the PSD. The parameters a and b are the constant length of the reflected beam before and after passing through the optical filter and receiving lens [12].

2.4.1 The sensor for EGM Path Correction

The sensor used for this study is Panasonic HL-G112-A-C5. This is a CMOS laser displacement sensor with a resolution of $8\mu\text{m}$ and a sampling rate of $200\mu\text{s}$ - 2ms , depending on the configuration and which type of object the laser is reflected from. For dark, absorbing surfaces, the sample rate is 2ms , and for lighter surfaces, the sample rate is $200\mu\text{s}$. The laser beam's diameter is $1.0 * 1.5\text{mm}$ and has a constant wavelength of 655nm . Knowing the wavelength of the laser allows the sensor to use a selective filter to stop other wavelengths. The sensor measures $60 \times 57 \times 20.4\text{mm}$ and weighs 70 grams (380 grams with the 5-meter cable weight). The light weight makes it suitable for use on robot arms, without adding much load to the arm. The operating range is $120\text{mm} \pm 60\text{mm}$. The sensor offers three channels of digital output and analogue voltage and current [13], [14].

Noise on the signal can affect the measurement's accuracy and should be considered when installing the sensor and placing the cable. The sensor should be mounted as separately as possible from noise-generating sources, and the cable should be at least 100mm from other wires and not be in parallel with them. This is especially important for the analogue output which is easily influenced by noise in the power supply. If affected by such noise, a noise filter or noise-cut transformer should be used. The cable should also be kept as short as possible. In general, it is recommended by the product manual to use shielded cables [14].

2.5 Signal analysis tools



Figure 8: Panasonic laser displacement sensor HL-G1. Source: Panasonic HL-G1 User's Manual [14, Page: frontpage]

2.5 Signal analysis tools

Throughout the research, two different tools have been used to analyse and sample data from the experiments. The tools are *RobView* and an oscilloscope. These offer something different from each other and are therefore used to display the relevant data for the experiments. This chapter will briefly present the tools and how they perform. Both are used to sample the sensor input, which is used to analyse the performance of externally guided motion. The main takeaway is the sample rate difference and the fact that RobView offers to sample the robot's position and the laser signal simultaneously.

2.5.1 RobView for studying the robot's accuracy

RobView is a program developed by ABB Robotics and can be used to view and log signals and data from industrial robots. It offers access to robot status, and I/O signals, display the robot's position, views RAPID variables, and lists robot messages. In other words, the program provides a GUI to monitor and analyse the robots and their performance. It can connect to the IP address of the robot and store signals with a time stamp in a database. Through the database, the program can retrieve the sampled signals and display multiple signals simultaneously. Operators can use it to compare, add, or subtract signals from each other [15].

A feature in RobView is called Signal Analyzer. In Signal Analyzer the robot's position and internal and external signals, connected to the I/O module, can be displayed. The signals stored in the database do not have a fixed sample rate but are sampled as soon as they are available. This can vary from 38ms to 62ms and it can also vary between the different signals.

The logged data in RobView can be analysed within the program, or exported to a CSV file. This allows the user to manipulate the data and then import files to the program. The exported data is retrieved from the database where it could suffer from double sampling where multiple samples have the

2.5 Signal analysis tools

same value. The CSV files can be read through a Python program for further calculations, analysis, and filtering [15].

2.5.2 Oscilloscope for sampling analogue displacement signals

The oscilloscope, Rohde & Schwarz RTM3004, is a powerful tool used to analyse and visualize electrical signals. One of the oscilloscope's advantages is the high bandwidth and sample rate. The bandwidth is from 100MHz to 1GHz, and the sample rate is up to 5GSamples/s. The oscilloscope also has a 10-bit ADC resolution, which is four times superior to conventional 8-bit analog-digital converters. The oscilloscope also offers a sensitivity of $500\mu V/div$ on its display. This enables users to capture and analyse signals with high precision and accuracy. When exporting the sampled data to a CSV file through the IP address, the data do not get a time stamp, but only a sample number[16].

3 Implementation

This section covers the method used to study if Externally Guided Motion can be used to follow an object at a constant distance. First, the workspace and its implementation are presented. The work environment is presented by describing the robot, the work object, the tool, and the path used for testing. Then the mounting of the sensor onto the tool is described, and how the analogue signal is converted to logical values is presented. The analogue signal is also checked for excessive noise which can disturb the later experiments or the implementation of EGM. The configuration and implementation of EGM Path Correction are then described in detail by showing the setup in RobotStudio and RAPID. Some EGM functions are also explained briefly. Lastly, the RAPID programs for three different experiments are described. These will be run in chapter 4 where the results are presented.

3.1 Work environment

The workspace consists of an IRB 1200 robot with a tool mounted and a table placed in front of the robot. Figure 9 shows the environment used for the project. The different parts of the environment will be described in this section to provide an understanding of how the different parts are connected. It also explains why some details in the results came to be.



Figure 9: Work environment with the robot, the tool with sensor, and the box step in the middle of the table.

3.1.1 The robotic manipulator

The robot used during this study is the IRB 1200, produced by ABB. This is a six-degree-of-freedom (DOF) industrial robot with a payload capacity of

3.1 Work environment

7 kg and a range of 0.7 meters. Its pose accuracy is 0.02mm and the robot's repeatability is 0.02mm. For linear path accuracy and repeatability, the robot's numbers are 0.77mm and 0.02mm, respectively [7, Page: 47]. Figure 10 displays the robot without its tool mounted at the end-point. During this project, the robot will have a laser displacement sensor, described in Chapter 2.4, mounted on the tool. The tool which holds the laser sensor is described in Chapter 3.1.2, and Figure 11 displays the tool. The robot controller is the OmniCore controller. OmniCore's I/O module is configured to sample the input every 4ms to ensure that it is sufficiently fast compared to the other sample rates. The controller can be equipped with optional software for increased functionality, such as Externally Guided Motion [17].



Figure 10: ABB Robotics IRB 1200. Source: [7, Page: 1, frontpage].

3.1.2 The end-effector and its coordinate systems

For this project, a 3D printed tool was used, as shown in Figure 11, designed by *ABB Robotics Bryne*. It allows for two different positions for the sensor to be mounted, and the sensor is mounted on the side of the tool, unlike Figure 11. The laser sensor's position, [66.641, 17.5, 152.536], and the orientation, 60° around the y-axis, are both with respect to the robot end-point coordinate system. The orientation is given as quaternions, which the reader can get more information about in [18, Page: 13-14]. Two tool-coordinate systems were defined as shown in listing in 3.1.3. The first coordinate system represents the laser sensor's position and is called "tLaser" in the RAPID program. The position and orientation of the sensor itself define this coordinate system. The other coordinate system is called "tEGM" in the RAPID program and is defined to be 120mm from the laser, along the emitted laser's direction, acting as the center of the measuring range of the laser. The latter coordinate system is the one used relative to the work object in the motion commands, which will be described in

3.1 Work environment

Chapter 3.4.

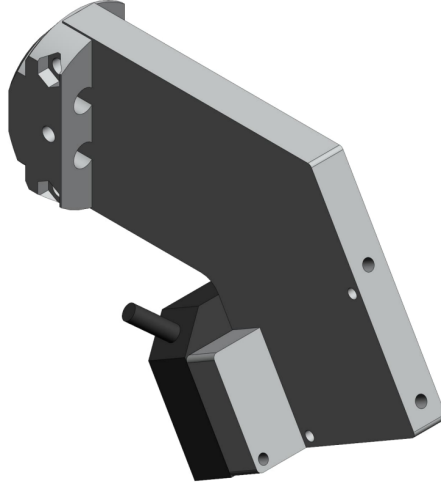


Figure 11: The tool used for this project. The laser sensor is moved to the side of the tool to satisfy the suggested mounting. Source: ABB Robotics Bryne.

3.1.3 Work object

The table is defined as a work object, *Workobject_1*, and is 218 millimeters tall relative to the base coordinates of the robot. This is the height used in RobotStudio to define the work object's height. All the targets in RobotStudio were defined relative to this. The work object's definition is shown in code listing 3.1.3. However, the tabletop is uneven. The table was measured to be 0.9 and 1.4mm higher in the middle than on the edges. This is visible in the plots in the later chapter, but the interesting features in the plots are still clearly represented.

Listing 1: Tool and work object definitions.

```
1 TASK PERS wobjdata Workobject_1:=[FALSE, TRUE, "", [
    [-120,0,0], [1,0,0,0]], [[376.515, -311.198, 218],
    [1,0,0,0]]];
2 PERS tooldata tEGM:=[TRUE, [[170.564, 17.5, 212.536],
    [0.480764431, 0.026773932, -0.876317842, 0.014688682]]
    , [0.3, [0,0,1], [1,0,0,0], 0,0,0]];
3 PERS tooldata tLaser:=[TRUE, [[66.641, 17.5, 152.536],
    [0.480764431, 0.026773932, -0.876317842, 0.014688682]]
    , [0.3, [0,0,1], [1,0,0,0], 0,0,0]];

```

3.1.4 The path for testing

The path used for the experiments is defined as seen in Figure 12. It is circular to exploit as much of the robot's range as possible without moving close to singularities. A long path was needed to study the settling times of the step responses in the later experiments. The path correction allows corrections at

3.2 Sensor mount and considerations around the analogue input

$\pm 50mm$ with respect to the original path. The program will stop with an error if the input signal suggests a correction that exceeds this. The EGM controller does not consider if its output to motion control will move the robot into any singularities. Therefore, it is important to ensure that the path is feasible for the maximum corrections. A simple test was performed to see if the robot manages to execute the same path with maximum offset on the z-axis in the positive and negative directions.

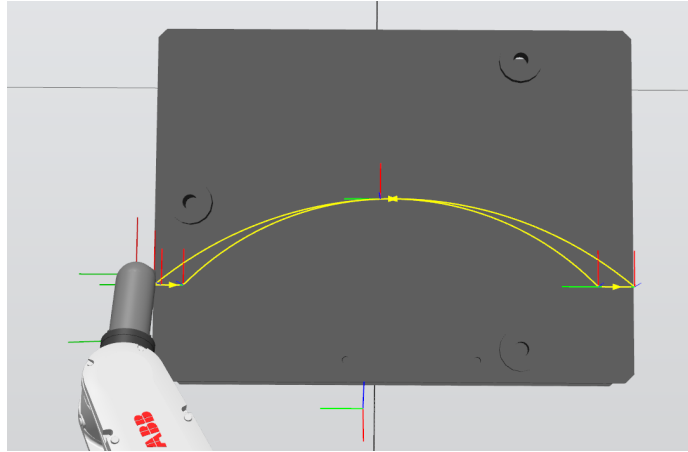


Figure 12: The figure shows the circular path which is used throughout the project. The path is made circular to avoid moving close to singularities and maximize path length and speed.

As described, the EGM controller does not consider potential singularities. Therefore, it is important to define a feasible trajectory with room for corrections for the robot to follow when testing EGM. When the path is corrected close to singularities, it affects the robot's movement and the speed of the end-effector. The robot struggles to follow the assigned speed through certain points when multiple axes are moving simultaneously. This could be sudden large rotations for the joints while the path is being followed at the assigned speed. This happens especially in situations where the robot is forced through a point close to a singularity. When the original trajectory is defined these factors should be taken into consideration to achieve a feasible trajectory.

The path moves from the left in a short linear path to enable EGM and get into position. Then it moves in a circular path toward the right. The path is again linear for a few millimeters at the right of the table before it moves in a circular path back to the starting point. During testing, a box was placed in the middle of the table to perform experiments on extreme cases, such as a step response.

3.2 Sensor mount and considerations around the analogue input

The mounting of the sensor plays a role in the result. Because of the trigonometry of the emitted laser beam, the orientation of the sensor, with respect to the

3.2 Sensor mount and considerations around the analogue input

direction of movement, becomes important [14, Page: 19-21]. Ideally, the sensor is mounted like in Figure 13, and then the robot is programmed to always orient the tool with respect to the robot's movement in real-time. However, orienting the tool is not possible for EGM Path Correction. Only position correction in the y- and/or z-direction [6, Page: 15]. As mentioned in chapter 3.1.2, there are two mounting options for the sensor on the tool, and the sensor has been mounted as suggested by Figure 13. This ensures that the triangulation of the laser measurement does not get intercepted.

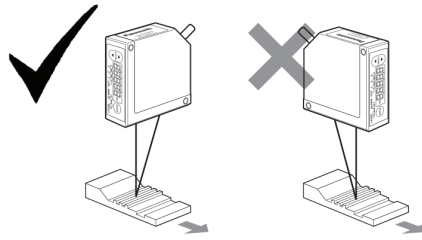


Figure 13: Sensor mounting directions are important to achieve the best result [14, Page: 20].

Through this project the analogue voltage is used as the reference signal. The analogue signal is connected to the OmniCore controller through an analogue add-on device, *DSQC1032* [17]. Through this device, the signal is directly accessible in RobotStudio, as shown in Figure 14. The signal was named *LaserSignal* and the physical signal was related to logical values. To achieve a precise displacement sensor, the calibration needs to be done correctly. The controller interpreted the input signal as a physical values ranging from 0 to 10V and then converted them to logical values ranging from -77.699 and up to 52.603 millimeters. Small errors in the calibration can result in incorrect measurements and thus inaccurate path corrections later on.

To convert and calibrate the analogue values to logical values, which can be accessed and used in RAPID, the following equation was used:

$$y = ax + b \quad (2)$$

The values were calibrated to be as accurate as possible close to zero, the middle of the measuring range. The voltage was measured to 5.1958V at $-10.01mm$,

Name	Value
Name	LaserSignal
Type of Signal	Analog Input
Assigned to Device	DSQC1030
Signal Identification Label	DSQC1032
Device Mapping	32-47
Category	
Access Level	All
Default Value	0
Analog Encoding Type	Unsigned
Maximum Logical Value	52,6026421
Maximum Physical Value	10
Maximum Physical Value Limit	10,5
Maximum Bit Value	4095
Minimum Logical Value	-77,6992035
Minimum Physical Value	0
Minimum Physical Value Limit	0
Minimum Bit Value	0

Figure 14: The analogue input is configured in RobotStudio. The maximum and minimum values are related to a maximum and minimum logical value.

3.2 Sensor mount and considerations around the analogue input

and $6.732V$ at $10.02V$. The laser will mainly be working within this measuring range in the experiments. These values were further inserted into equation 2 and then solved as a set of equations with two unknown values.

$$\begin{aligned}5.1948a - b &= -10.01 \\6.7320a + b &= 10.02\end{aligned}$$

The result of this is:

$$\begin{aligned}a &= 13.03 \\b &= -77.70\end{aligned}$$

These values are inserted into equation 2 and solved for the maximum and minimum physical values, $0V$ and $10V$ respectively. The result, for logical values, that is, are:

$$\begin{aligned}MaximumLogicalValue &= 52.60 \\MinimumLogicalValue &= -77.70\end{aligned}$$

After the calibration the system had to be rebooted and these values were changed at the third decimal. The values above are the calculated calibration, and the numbers in Figure 14 were changed by the system. The values are in millimeters, and the third decimal is on a micrometer level. Thus the values were accepted after the reboot.

Before implementing Externally Guided Motion on the robot, an analysis of the external signal was performed. If the analogue signal is inaccurate or is subjected to excessive noise, the EGM functions will also be inaccurate. Analogue signals can be subject to noise from other electronics or wires and cables. This was considered when connecting the laser to the I/O module and throughout the wiring in general. The signal was sampled and tested for excessive noise. The laser signal has been tested by positioning the robot still above the table whilst measuring the analogue signal over a period of time. The height was set to be $120mm$, measured from the table to the laser, as that is the center point of the laser's measuring range. Figure 15 displays the analogue signal converted to logical values in millimeters. The signal has a calculated mean value of $4.582e^{-5}mm$ and a standard deviation of $0.189mm$. The calibration of the logical values was, based on these measures, considered acceptable for the project.

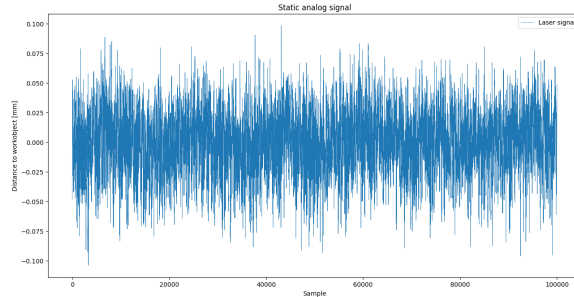


Figure 15: The analogue signal measured with the oscilloscope when standing still.

3.3 EGM using Path Correction

3.3 EGM using Path Correction

EGM Path Correction was chosen for the project as its specifications fit the purpose of this research. This was the clear choice for this kind of task compared to Position Guidance and Position Stream. The latter is dependent on the external device receiving position and path plans, and Position Guidance generates its own path based on an external device. Path Correction is designed for tasks where the robot follows a trajectory and corrects the distance along the y- and/or z-axis. While the robot follows a pre-defined trajectory, EGM Path Correction operates as Figure 16 presents. The following steps describe the figure:

1	Measure the displacement at the sensor's sampling rate $200\mu s$ -2ms.
2	The I/O module is configured to sample the analogue signal at 4ms. The signal is converted to logical values.
3	The EGM controller samples the logical values at a rate of 24ms. Then it calculates the new z-position which is fed directly to motion control. It also reads the real-time position of the robot at multiples of 48ms.
4	Motion control moves the robot with the updated coordinates

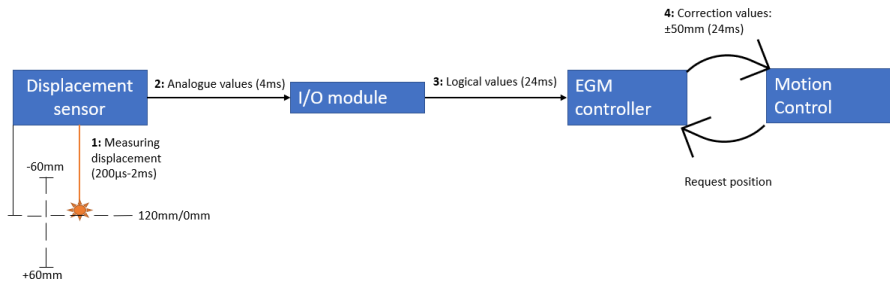


Figure 16: Configurations for EGM Path Correction

To enable EGM Path Correction requires a configuration in the topic *Motion* in RobotStudio. Figure 17 shows *External Motion Interface Data*, which is located within the aforementioned topic. Through this configuration, the *Level* must be set to "Path" to enable Path Corrections. The other values are not used for Path Correction [6, Page: 28]. Other control parameter values can therefore not be decided for Path Correction. The unused choices such as gain, ramp time, and low pass filter can be used for the other EGM features.

3.3 EGM using Path Correction

Name	Value	Information
Name	Test	
Level	Path	
Do Not Restart after Motors Off	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Return to Program Position when Stopped	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Default Ramp Time	0,0500000007	
Default Proportional Position Gain	5	
Default Low Pass Filter Bandwidth	10	

Figure 17: Configurations for EGM Path Correction in RobotStudio

The implementation of the robot with EGM consists of a regular RAPID program and the EGM set up for Path Correction. The RAPID program contains some targets and a path for the robot to follow. The targets are defined in the same way as for any other robot movement. The only difference lies in the movement commands, where the specific EGM-id must be specified. There can be multiple EGM setups within the same program, which is why they are assigned an EGM-id. However, only one was used for this project.

The EGM documentation [6], contains descriptions of EGM functions, and program examples to get started using EGM. The listing in 3.3 shows how the EGM setup can be done for path correction in the z-direction with a sample rate at 24ms. The first command releases the EGM-id in case the program was stopped last time without releasing the EGM-id. Then the EGM-id is retrieved again.

Listing 2: EGM setup for path correction with analogue input. The listing show line 69-72 of the program.

```

1 EGMReset egmID1;
2 EGMGetId egmID1;
3 EGMSetupAI ROB_1, egmID1, "Test" \PathCorr \APTR \
  aiR3z := LaserSignal;
4 EGMActMove egmID1, tLaser.tframe \SampleRate:=24;

```

The following table explains how *EGMAnalogueAI* is used to set up externally guided motion for an analogue input signal.

	EGMSetupAI:
1	The function first connects the mechanical unit <i>ROB_1</i> with an EGM-id, <i>egmID1</i> .
2	Path Correction is chosen with " <i>Test</i> " and <i>\PathCorr</i>
3	The corrections are set up to be <i>At-Point TRacker</i> by using <i>\APTR</i> . This means that the values read from the sensor correct the current position, and not some future position.
4	<i>aiR3z</i> is the analogue input for the z-position. The laser signal is assigned to correct along this axis

EGMActMove activates the specific EGM process defined as *egmID1*. The sensor frame is defined as *tLaser*. When defining this, it is important to consider the orientation of the coordinate system of *tLaser*. EGM corrects the path

3.4 The RAPID programs

in the path coordinate system. The path coordinate system was described in Chapter 2.3. By defining the laser coordinate system with the positive z-direction upwards, the corrections produced by EGM will match the logical values provided by the analogue signal.

The sampling rate for Path Correction has to be in multiples of 24ms, where 24ms is the fastest [6, Page: 52]. This was also the chosen sample rate for the controller.

Lastly, a couple of EGM functions used to move the robot are *EGMMoveL* and *EGMMoveC*. The use of these is presented in the code listing in 3.3. These work in the same way as the regular move commands in RAPID. The difference is that they specify what EGM setup the move-commands apply to. Move commands can be applied to a specific tool with respect to a given work object. *tEGM* and *Workobject_1* are used in the move-commands allowing all the targets' coordinates to be defined with respect to the table. *tEGM* is a path coordinate system defined to be 120mm below the laser. It is defined to be the center of the measuring range, and it has a positive z-direction upwards. This also prevents EGM corrections from crashing the laser into the work object since it can only correct ± 50 mm before the robot stops with an error. The targets the move-commands use are defined by *p3*, *pc1*, *pc2* and *p5*. *Speed* is defined as a variable so that all can be changed simultaneously and *z5* is an acceptance zone with a 5mm radius from the targets.

Listing 3: EGM move functions for linear and circular movement.

```
1 EGMMoveL egmID1, p3, Speed, z5, tEGM\WObj:=  
   Workobject_1;  
2 EGMMoveC egmID1, pc1, pc2, Speed, z5, tEGM\WObj:=  
   Workobject_1;  
3 EGMMoveL egmID1, p5, Speed, z5, tEGM\WObj:=  
   Workobject_1;
```

3.4 The RAPID programs

Three different RAPID programs was used for the experiments throughout the project. All of them are based on the same targets, path, tool and work object. However, they are a bit different in the type of functions and setup. Throughout the report, the programs will be referred to as *signal analysis program*, *EGM analogue output*, and *EGM analogue input*. In the following three subsections, the different programs are described.

The programs use a trigger function to separate the test runs in the CSV-files. The functions activate an analogue signal which is detected by the oscilloscope. Through this signal, each test was separated. This will not appear directly in the results, but the triggered signal is used to calculate the values in the results.

The main difference between the signal analysis program and the EGM programs is that the height of the TCP is constant for the signal analysis program, and the height varies for the EGM programs. The move functions for EGM programs include path corrections, as set up in Chapter 3.3. Otherwise, the programs do the same.

3.4 The RAPID programs

3.4.1 Signal analysis program

The first program is the simplest and only moves with standard linear and circular motion whilst keeping constant height, without using EGM. Through this program, the path and targets were verified not to contain any singularities that could interrupt the experiments. The robot can move freely at a constant speed and the path length is maximized in the given workspace. By moving the robot at constant speed across the table as shown in Figure 12, the analogue signal can be analysed. The idea was to first verify that the signal is reliable and does not contain excessive noise. Another interesting possibility to consider is if the signal, when subjected to a step, measures a step, ramp, or oscillations.

In the code listing 3.4.1, *MoveC* is used to create the circular path. The tool, *tEGM*, is defined 120mm from the laser and is moved with respect to *Workobject_1*. The work object is the table in front of the robot, and when the height of the targets *pc11* and *pc12* is zero, the laser's output will be zero. It is the displacement between *tEGM* and *Workobject_1* that is displayed in the results.

Listing 4: An example of the MoveC-function.

```
1 MoveC pc11, pc12, speed, z5, tEGM\WObj:=Workobject_1;
```

3.4.2 EGM analogue output program

The second program uses EGM Path Correction with analogue output as the source for the corrections. This program uses the same path and targets defined for the *Signal analysis* program. In a way, this is a simulation of the analogue signal with a box step. The analogue value is changed at the same coordinate as the box was placed in the other program. The EGM setup is a little different for analogue output than the analogue input, which was presented in 3.3. Code listing 3.4.2 shows the setup. There is only a few details difference, where the setup is specified for analogue output, and within this command, the output port is connected with *aoR3z*. This defines that the z-axis is to be corrected by the values received from *DSQC1030_02_AO1*, the analogue gate. The sampling rate is set to 24ms to have as many measurements of the displacement as possible.

Listing 5: EGMSetupAO for analogue output.

```
1 EGMSetupAO ROB_1, egmID1, "Test" \PathCorr \APTR \
  aoR3z := DSQC1030_2_AO1;
2 EGMActMove egmID1, tLaser.tframe\SampleRate:=24;
```

The value can be changed within the RAPID program as the code listing 3.4.2 shows. The EGM controller reacts thereafter. As this is a RAPID-defined value, it can be set as a logical value in millimeters and the program converts it to analogue values.

Listing 6: How to set an analogue output value.

```
1 SetAo DSQC1030_2_AO1, -10.5;
2 EGMMoveC egmID1, pc1, pc21, Speed, z5, tEGM\WObj:=
  Workobject_1;
```

3.5 Reading sensor data in Python

```
3 SetAo DSQC1030_2_A01, 0;  
4 EGMMoveC egmID1, pc22, pc2, Speed, z5, tEGM\WObj:=  
    Workobject_1;
```

3.4.3 EGM analogue input program

The third program uses EGM with analogue input from the laser sensor. The program is similar to the second program, but now the EGM controller receives the reference value from the laser sensor. The setup and configuration for this program were shown in Chapter 3.3. The path and targets are the same as the two previous programs, and the box is again placed in the middle of the path. The analogue value is continuously read at the input, and the EGM controller corrects for the displacement. The sample rate is again set to be the fastest measure as many details as possible and be able to move as fast as possible without decreasing the accuracy.

3.5 Reading sensor data in Python

Python was used to read the CSV files from RobView and the oscilloscope. Two separate programs were used because the structure of the files was different from the two sampling methods. The idea of the Python programs was to split the comma-separated values (CSV) and instead store the values in separate arrays. The CSV files contain 4-16 test runs repeated after each other. The number of test runs are varying because the oscilloscope was set up to have the same sample rate for each experiment and stop sampling after 100 000 samples.

When the experiments were running, the test runs repeated themselves without the robot stopping and each test run had to be separated in Python. Test runs sampled by the oscilloscope were separated by a triggered signal from RAPID which showed up in the scope. Figure 18 shows how the laser signal is separated. Each time the triggered signal appeared, the test run started/stopped. These points are marked by black dots. The files from RobView used the y-position of the robot to determine when the test starts. Figure 19 shows how the test runs were separated in the RobView data. When the y-position peaked, the robot was 300mm to the left of the center of the table. This is the leftmost robot position and was used as the start/stop for the test runs. These points are marked by blue dots.

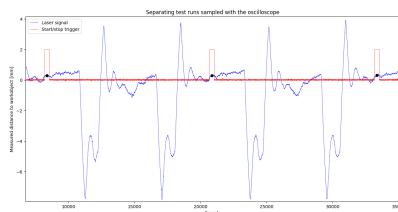


Figure 18: Separated test runs sampled with the oscilloscope.

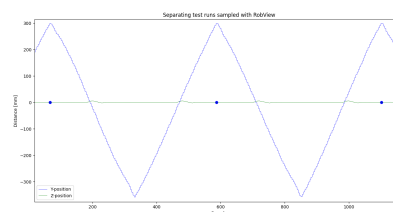


Figure 19: Separated test runs sampled with the RobView.

3.5 Reading sensor data in Python

Based on the start and stop of the test run, the mean and standard deviation were calculated. The mean and standard deviation of each separate test run were stored in arrays. An overall value for these was also computed. The standard deviation within the array of mean values was computed to serve as a measure of how repeatable the tests were. If each test run has the same mean value, it would indicate that the signal is consistent and the robot can repeat the task with the same level of accuracy each time.

The signals' graphs are plotted to visualize the performance. The accuracy can also be studied visually. From the oscilloscope data, the laser signal's data was plotted with respect to time, and for the RobView data, the z-position of the robot was plotted with respect to the sampling number. The laser signal was supposed to be in the plot with the robot position, but it was sampled at another rate and the signals did not align.

4 Experiments and results

This section describes the experiments and the results from them. The *signal analysis*, *EGM output*, and *EGM input* programs are presented separately. Each of these experiments is repeated three times, where only the speed is changed. The idea is to study how the signal and the EGM controller's performance are affected by increasing speed. Each subsection will first describe the experiment and then they are divided into three smaller sections again. These sections present the results for the different speeds for which the experiments have been tested. Lastly, there will be a short analysis to compare the results of the experiment.

In the figures two signals are visible. The raw analogue signal is gray in the figures which is unprocessed like the I/O module sample the signal. The orange signal is filtered in Python for visual purposes by using the Savitzky-Golay filter [19] with a window length of 102 and a fitting 2-degree polynomial. The filtered signal is only added for improved visualization, and not for increased performance. Therefore, the filtering is not explained in more depth here. The data and calculations made in Python are all based on the raw signal to preserve the values the EGM controller uses.

The oscilloscope was set up to do 100 000 samples, at a rate of 833 samples per second for each experiment. The sampling will therefore be approximately 120 seconds. This implies that the different experiments, for different robot speeds, will have different numbers of completed test runs in the 120-second time interval.

4.1 Analogue Signal

The signal was tested when subjected to a step of 10.5mm to investigate how the signal reacts in extreme cases. The EGM will later in the project be subjected to the same extreme case. Therefore, it was interesting to see how the laser, subjected to a step, will behave. The *signal analysis* RAPID program was used and tested for 50, 100, and 200mm/s. The robot is moved across the table at a constant height. It starts from the left of the table and moves to the right, and then returns to the starting point, as shown previously in Figure 12. There was a box in the middle of the path with a height of 10.5mm and a length of 170mm. What was interesting to survey was if the signal behaved as a step, a ramp, or if it possibly overshoots. The figures in the following subsections display the step response when the robot moves at a constant height above the table and with constant speed.

To analyse the accuracy and repeatability of the signals, the experiments are repeated multiple times, separated as *test runs*. The figures that will follow only display one run where the robot moves back and forth across the table. The exact number of runs varies because of the robot's speed. The sampling time of the oscilloscope was kept constant and therefore the speed of the robot decided how many repetitions were done within the total amount of samples the oscilloscope was configured to make.

The mean and standard deviation of each run have been calculated. The mean values were put in an array, and then the standard deviation between each of the mean values was calculated as well. This is to gain knowledge of the

4.1 Analogue Signal

analogue signal's accuracy and repeatability.

4.1.1 Step response with 50mm/s speed

Two steps are displayed in Figure 20 as ramps. As presented in Chapter 3.1 the table has been measured to be between 0.9 and 1.4mm taller in the middle than on the two edges, respectively. The table skew is visible and the table height varies. The test run starts in the "pit" of the signal, at the left of the figure. When the laser measures the step, the signal is not at the zero line, which is dashed horizontally. From the figure, the height of the step is -9.4mm. However, the signal is 1.1mm from the zero line before reading the step. The relative height is in fact 10.5mm, as the box is measured to be. The analogue signal behaves like a ramp and there are few indications of oscillations in the signal apart from some noise.

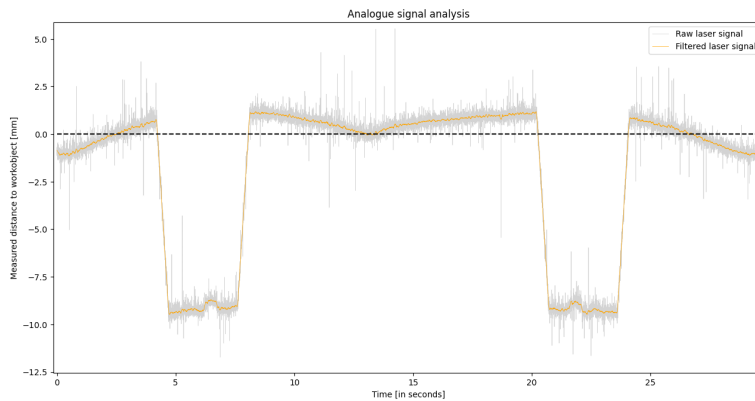


Figure 20: Laser signal subjected to a 10.5mm step at a robot speed of 50mm/s.

Figure 21 shows parts of Figure 20 in more detail. It takes 532ms for the ramp, which the signal creates, to reach the step height. The settling time is decided by the amount of time it takes the signal to reach the 5% acceptance region, represented by the dashed lines. The lines are $\pm 5\%$ of the height of the step.

4.1 Analogue Signal

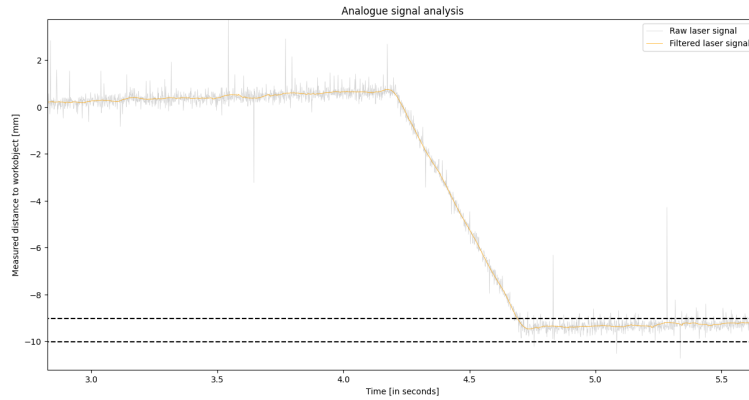


Figure 21: The settling time with robot speed at 50mm/s is 532ms.

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	-1.865	3.956	532
2	-1.864	3.959	544
3	-1.868	3.955	525
Overall	-1.866	3.956	534

Table 2: Mean, standard deviation, and settling time from the experiment with robot speed at 50mm/s.

This experiment for robot speed at 50mm/s was repeated 3 times. Each repetition is measured separately and then compared. This was the slowest speed the experiment was tested for. Table 2 shows the calculated means, standard deviations, and settling time for each run. The mean value does not directly represent the accuracy and is rather used to compare the repeatability by calculating the change in the mean values between each repetition. The standard deviation between the mean of each run gives a repeatability of 0.002mm. The standard deviation values in the table are quite high because the steps deviate from the table's height. The graph itself shows the accuracy when it measures the height of the step and it also shows the table skew clearly. The settling times variate for each run, but they are in the same region.

4.1.2 Step response with 100mm/s speed

Figure 22 shows the analogue signal for robot speed at 100mm/s. The signal measures the box to be -9,4mm, and the table is approximately 1.1mm by the box. This is the same as when the speed is 50mm/s. The box is however covered by fewer samples and appears shorter because the robot moved faster. There are few indications of oscillations in the signal apart from some noise. The signal behaves like a ramp and it takes 532ms to settle on the height of the step.

4.1 Analogue Signal

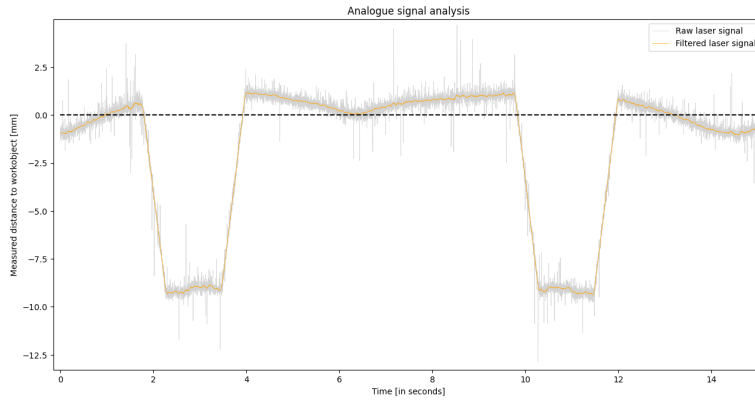


Figure 22: Laser signal subjected to a 10.5mm step at a robot speed at 100mm/s.

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	-1.823	3.778	532
2	-1.815	3.773	524
3	-1.858	3.795	516
4	-1.797	3.766	517
5	-1.823	3.773	533
6	-1.827	3.777	527
7	-1.824	3.776	530
Overall	-1.824	3.776	526

Table 3: Mean, standard deviation, and settling time from the experiment with robot speed at 100mm/s.

This experiment was run 7 times because the robot moved twice as fast as before. Table 3 contains the calculated mean and standard deviation from each run. These values are not directly used in the evaluation of accuracy and repeatability. They are both quite high and in Chapter 4.1.4 the evolution of these numbers will be compared for the different speeds. The mean values vary more in the different repetitions of the test runs. With robot speed at 100m/s, the standard deviation between each test run's mean is 0.017mm.

4.1.3 Step response with 200mm/s speed

When increasing the speed of the robot, less detailed information shows up in the analogue signal's plot. Figure 23 still displays a clear box step, but it is narrower. The measured height of the step is -9.11, measured from approximately 1.16mm, adding up to a relative height of 10.26mm. This is a bit lower than the two previous experiments. The signal uses 520ms to reach its peak, which is in the same region as the other speeds.

4.1 Analogue Signal

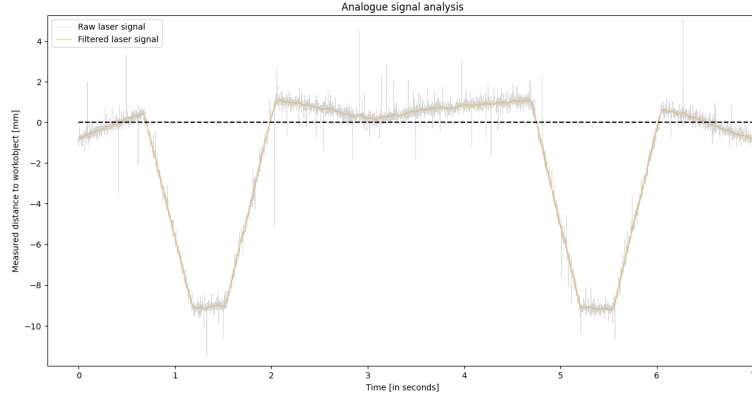


Figure 23: Laser signal subjected to a 10.5mm step at a robot speed of 200mm/s.

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	-1.800	3.435	520
2	-1.793	3.437	533
3	-1.799	3.434	518
4	-1.803	3.436	524
5	-1.797	3.436	531
6	-1.800	3.444	527
7	-1.799	3.433	526
Overall	-1.798	3.438	525

Table 4: Mean, standard deviation, and settling time from the experiment with robot speed at 200mm/s.

The experiment included 16 runs, but the table is cut short. Therefore, only the first 7 runs are included in Table 4. When increasing the speed of the robot to 200mm/s, the standard deviation of the mean values in the table is 0.003mm. This is quite low and implies good repeatability for the analogue signal.

4.1.4 Analysis of the results

First of all, the settling time is in the same region for each of the experiments. It implies that the signal uses the same amount of time to measure the height, regardless of robot speed. The interesting aspect of the results to consider is how the signal performs in terms of accuracy and repeatability. As it is supposed to be the source for the externally guided motion, which will be tested in Chapter 4.3.

The differences in the repeatability between the experiments were expected to decrease as the speed was increased. From 50mm/s to 100mm/s, the repeatability decreases, while the repeatability for 200mm/s is better than for 100mm/s. The repeatability for 200mm/s is almost as good as for 50mm/s.

4.2 Externally guided motion with analogue output

This is believed to be a coincidence, as it is natural that the accuracy and repeatability decrease when the speed increases.

The reason why the overall value for the mean and standard deviation decreased between the test runs, is because of the robot's speed. When the robot moves faster, it will spend less time above the box, relative to the time it spends above the table. Even though the table is skew, it is closer to the zero line and the mean value will be lower for faster speeds. These values themselves are not the most interesting, but they are compared with each other for every test run to study the change. If the mean value changes for each test run, for the same speed, it would indicate low repeatability. The same goes for the standard deviation. If the standard deviation is high, or if it changes for each test run for the same speed, it serves as information on the accuracy and repeatability of the signal.

The accuracy for the robot speeds at 50mm/s and 100mm/s is high. The sensor measures the box accurately for these speeds, but when the speed was increased to 200mm/s, the measurement deviated from the actual height.

4.2 Externally guided motion with analogue output

The experiment studies the basics of externally guided motion and its setup in RobotStudio. EGM Path Correction is configured as described in Chapter 3.3 and uses the *EGM analogue output* program described in Chapter 3.4. The program simulates a step with the same height as the 10.5mm high box. EGM is active in both directions and is tested for 50, 100, and 200mm/s. The results could indicate how the EGM controller performs and how increasing speed affects the results.

The signal in the figures in this section will take a step in the opposite direction compared to the previous section of results. The output value was set to -10.5mm in RAPID. If objects are closer than the middle of the sensor's measuring range, the distance is negative. This is what happens in the result in Chapter 4.1. If a box of 10.5mm in height and length of 170mm was placed in the path, the result would, ideally, be that the signal's graph appears flat. However, the experiment is performed without the box in the path, and the signal's graph is a step in the positive direction.

The results will be used to set up externally guided motion with analogue input in the next step. The three following subsections present the results of the experiments for the different speeds.

4.2.1 EGM with analogue output with 50mm/s speed

Figure 24 presents the sampled signal from the laser when performing externally guided motion with analogue output as the source for the corrections. The robot did three runs with 50mm/s speed, and the figure displays one run. The path starts to the left of the figure and ends in the "pit" on the right of the figure. In the middle of the figure, the robot turns around and repeats the correction on its way back to the start/stop point.

The RAPID program sets the analogue output to -10.5mm and then back to zero. This simulates a box step with the height and length of the actual box. Because of the skewed table, the signal is never completely flat in between the steps. The graph is approximately 1mm before and after the step. It then jumps

4.2 Externally guided motion with analogue output

to around 11.5mm, resulting in the relative height at -10.5mm. The signal does not oscillate other than regular noise in the signal.

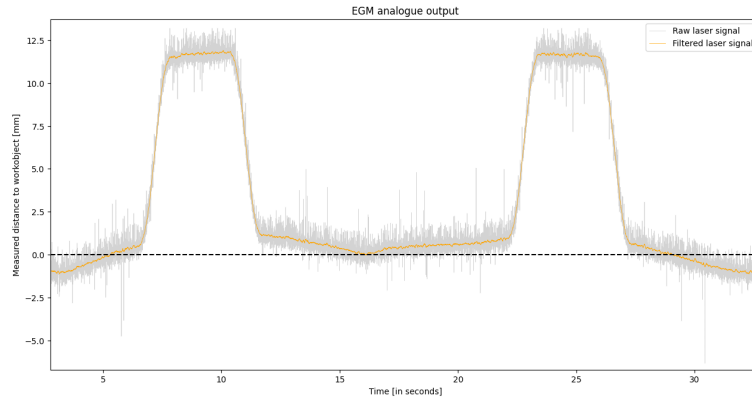


Figure 24: Path Correction with analogue output with robot speed at 50mm/s.

The signal still behaves as a ramp, and it takes 1.272 seconds for the robot to reach the acceptance region. Figure 25 shows the step response of the path correction with analogue output as the source. The dashed lines represent the region which is $\pm 5\%$ of the step height. The settling time was measured with the oscilloscope using cursors. The step takes around twice the time that the signal itself used to reach the step height, as shown in Chapter 4.1. This is even though the analogue output is an "ideal" step, set in the RAPID program.

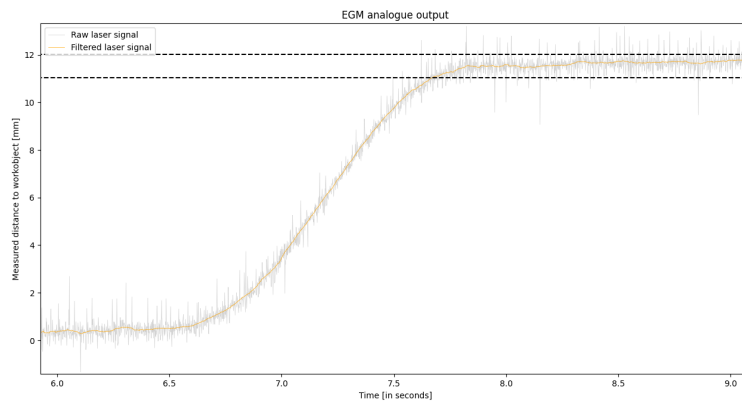


Figure 25: The settling time is 1.272 seconds with robot speed at 50mm/s

4.2 Externally guided motion with analogue output

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	3.305	4.784	1 272
2	3.298	4.786	1 247
3	3.299	4.787	1 200
Overall	3.301	4.784	1 240

Table 5: Mean, standard deviation, and settling time from the experiment with robot speed at 50mm/s.

This experiment was repeated 3 times for 50mm/s and Table 5 presents the calculated mean and standard deviations. As was seen in Chapter 4.1, the values for mean and standard deviation are quite high. When the robot corrects the height, based on the RAPID command, it moves away from the zero line, and thus the mean value and the standard deviation are high. These values do not directly inform about the accuracy and repeatability. However, the graphs show how accurately the EGM analogue output performs, and the standard deviation of the mean values in the table can be interpreted as a number for repeatability. This was calculated to be 0.003mm.

4.2.2 EGM with analogue output with 100mm/s speed

Figure 26 shows the analogue signal measured when the robot is moving 100mm/s with path correction based on analogue output. The path begins at the left of the table, which can be recognized by the "pit", at the left in the figure. When the simulated step is activated, the signal behaves like a ramp. It settles at 11.5mm before it steps down again to the table's surface. The box shape is still clearly visible with ramps up and down, only with some table skew in between. Even though the signal settled at 11.5mm, the relative height is indeed 10.5mm, just as defined in the RAPID program.

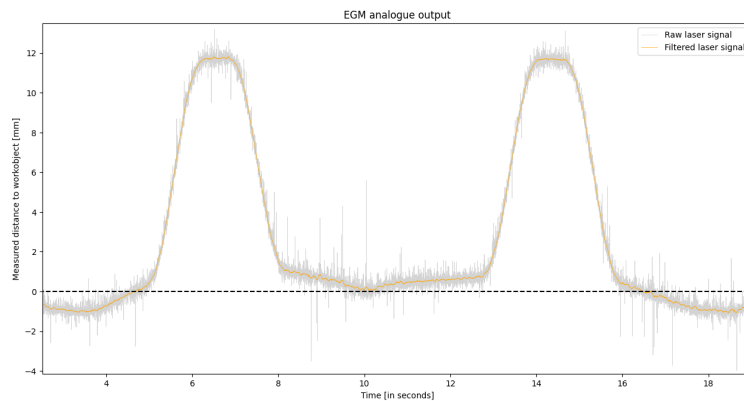


Figure 26: Path Correction with analogue output with robot speed at 100mm/s.

The signal used 1.199 seconds to settle at this height. The average settling

4.2 Externally guided motion with analogue output

time when the robot moves at 100mm/s is 110ms faster than the average settling time for the robot moving at 50mm/s.

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	3.172	4.487	1 199
2	3.166	4.479	1 118
3	3.180	4.488	1 122
4	3.177	4.485	1 082
5	3.176	4.489	1 117
6	3.175	4.487	1 153
7	3.177	4.488	1 117
Overall	3.175	4.486	1 130

Table 6: Mean, standard deviation, and settling time from the experiment with robot speed at 100mm/s.

In Table 6 the calculated mean and standard deviations from the experiment are presented. The mean and standard deviation from the zero line are still high, and not directly parameters describing the accuracy. There is an increase in the mean values compared to the 50mm/s experiment. This is the opposite trend of what was seen in Chapter 4.1, where the mean and standard deviation decreased when the speed increased. The accuracy is shown in the graph in the previous figure. The mean is varying more than for the lower speed. This is also proved by the standard deviation of the mean value of each test run, 0.004mm.

4.2.3 EGM with analogue output with 200mm/s speed

Figure 27 shows the signal from the experiment with 200mm/s speed and path correction based on analogue output. Because of the speed, the signal reaches 10.7mm, 0.8mm less than the two slower speeds. Because of the table skew, the steps should reach 11.5mm. However, it never reaches the full height before it steps down again because of the speed and length of the simulated step. This results in a graph that does not look like a box ramp, but rather an impulse. There are no clear oscillations except for the regular noise in the signal.

4.2 Externally guided motion with analogue output

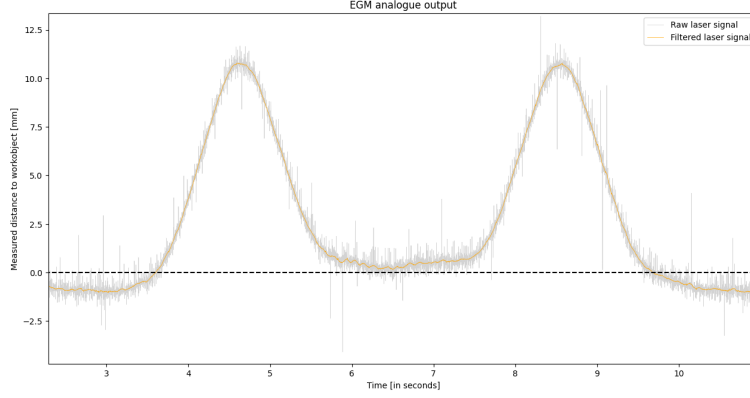


Figure 27: Path Correction with analogue output with robot speed at 200mm/s.

The signal reaches its peak height after 1.135 seconds, but there are uncertainties if the signal actually is settled at this point. It steps down again before it is really settled at the step's height. Anyhow, if it is said to be settled, it is 101 milliseconds slower than it was for 50mm/s.

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	2.991	3.880	1 135
2	2.990	3.881	1 091
3	2.988	3.885	1 135
4	2.994	3.877	1 169
5	2.985	3.877	1 167
6	2.987	3.881	1 175
7	2.988	3.888	1 134
Overall	2.988	3.883	1 144

Table 7: Mean, standard deviation, and settling time from the experiment with robot speed at 200mm/s.

Table 7 presents the calculated mean and standard deviation for the experiment at 200mm/s. The high values are due to the aforementioned reasons. For this speed, the accuracy has dropped, and the decrease in the mean and standard deviation values, compared to the two lower speeds, does not indicate higher precision. The repeatability, measured by taking the standard deviation between each mean value in the table, is 0.005mm. The values in the table follow the trend where the increase in speed results in decreased mean and standard deviation when compared with the lower speeds.

4.2.4 Analysis of the results

For EGM Path Correction with analogue output, the settling time was 110ms faster for 100mm/s than it was for 50mm/s. For the fastest experiment, the

4.3 Externally guided motion with analogue input

signal does not reach the full height. Therefore, the settling time for that experiment can not be compared with the others.

The differences in the repeatability between the experiments were expected to decrease as the speed was increased. From 50mm/s to 100mm/s, the repeatability decreases, which it continues to do from 100mm/s to 200mm/s. The repeatability is decreasing by 0.001mm for each tested speed, making it less repeatable. The mean values of each test run is varying more when the speed increases, which could be expected.

It was seen in Chapter 4.1 that the mean and standard deviation, compared to the zero line, decrease when the speed increase. This was not the situation comparing the result from the 50mm/s and 100mm/s experiments. But when taking the 200mm/s experiment into consideration, even though it never reached its full correction, the trend continues. That is when looking at the change in values from 50mm/s to 200mm/s.

The mean values are higher for the experiment with *EGM analogue output* than it was for the *analogue signal* program. The step is the same for both experiments. The analogue signal measures the actual step in the first experiment, and the path corrections are correcting the same height. Because of the higher settling time for EGM corrections, the mean values are higher for the EGM experiment with analogue putout values.

Lastly, the fact that the simulated correction of 10.5mm never reaches its full correction, for the highest speed, that is, could be a challenge before the test for the EGM Path Correction with analogue input at this speed.

4.3 Externally guided motion with analogue input

This experiment uses the knowledge from the two previous experiments of both the analogue signal and externally guided motion. The experiment runs the *EGM analogue input* program, which is the same setup as before, but now the EGM controller uses analogue input signal from the laser sensor. The laser measures the distance to the object, and the controller works to keep the distance constant. The nominal distance is set to be 120mm, as it is the middle of the measuring range. The EGM controller will try to keep a constant distance of 120mm for any table skew or boxes in the path.

This section will present the oscilloscope measurements of the analogue signal and it will present data sampled in RobView from the same experiment. The oscilloscope will still present the laser signal while the RobView plot shows the z-position with respect to the table. In the figures with data from RobView, the x-axis is labeled "Sample", but the exported data files have doubled sampled. Therefore, the same value was sampled four times in a row and then plotted. The x-axis does not represent the actual samples in the case of the RobView plots.

4.3.1 EGM with analogue input with 50mm/s speed

Figure 28 shows the signal from the test where the robot uses EGM path correction with analogue input as the source for the corrections. The signal drops downwards in the plot because the laser is closer to the surface than the nominal distance. This is the same as what was seen in Chapter 4.1. The signal oscillates around -5mm before the signal jumps upwards again. It then oscillates around

4.3 Externally guided motion with analogue input

0.5mm before settling. After the step down to zero, the only visible feature in the signal is the table skew before the robot returns, which it does in the middle of the figure. The settling values -5mm and 0.5mm are half of the height of the results in chapter 4.1, where the full height is 10.5mm and 1mm for the box and table skew, respectively.

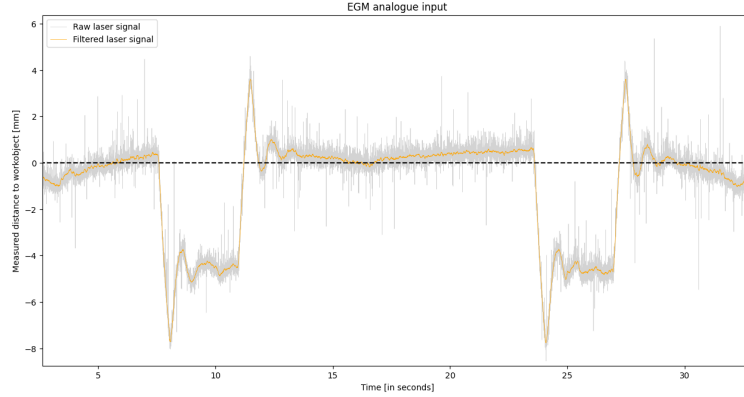


Figure 28: EGM with path correction with analogue input at 50mm/s.

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	-0.953	2.201	1 691
2	-0.952	2.213	1 733
3	-0.950	2.205	1 660
Overall	-0.952	2.206	1 694

Table 8: Mean, standard deviation, and settling time from the experiment with robot speed at 50mm/s.

Table 8 presents the means and standard deviations of each test run for the experiment. The values in the table could not be used directly in the previous tests for the *analogue signal* and *EGM analogue output*. When considering EGM with analogue input, these values can tell more about the performance. For this experiment, the signal was supposed to be as close to the zero line at all times. The mean values should be close to zero, and the standard deviation should also be smaller. However, the standard deviation can be affected when the sensor measures the step. The steps were never completely corrected, and the values in the table could have been lower. Three repetitions were completed for 50mm/s and the standard deviation between the mean values of each test run is 0.001mm, indicating high repeatability.

Now the RobView plot will be presented. It is visible in Figure 29 that the sampling rate is slower than the oscilloscope. The test run starts at the blue dot to the left and finishes at the blue dot to the right. After the robot starts the

4.3 Externally guided motion with analogue input

run, it corrects the table's skew before reaching the step. The robot oscillates above the box, at around 4mm. It steps down from the box to oscillate around -1mm. Then the figure displays some table skew before the robot turns around in the middle of the figure to repeat the step. This was previously seen for the laser signal as well. The z-position's graph should be shaped after the table skew before and after the step, and then it should take the shape and height of the box. That would indicate high accuracy.

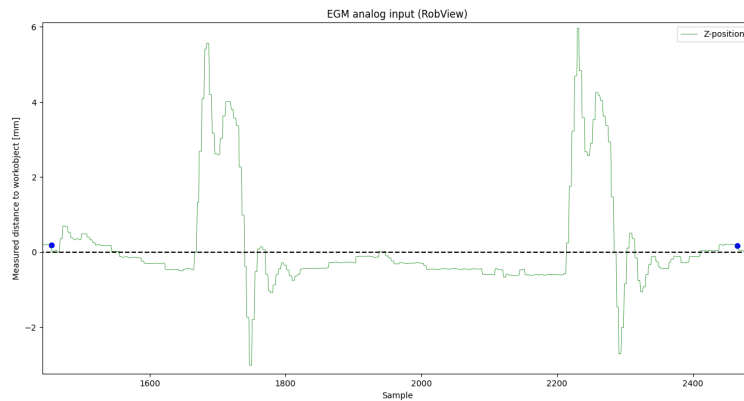


Figure 29: RobView data displaying the robot's z-position when using path corrections with robot speed at 50mm/s.

Figure 30 shows the same step as in Figure 28, but it is zoomed to focus on the oscillations and settling time. The oscillations become more clear, and the settling time can be measured in the oscilloscope. In the figure, the dashed lines represent the 5% acceptance region, and the signal is considered to have settled when entering and staying in the region. The settling time is measured on the step down to ensure that it has enough time to settle completely. It is worth noticing that when the signal steps down, it does so from around -5mm. The robot takes 1.691 seconds to settle at around zero. This correction is half of what the table skew was measured to be in previous results.

4.3 Externally guided motion with analogue input

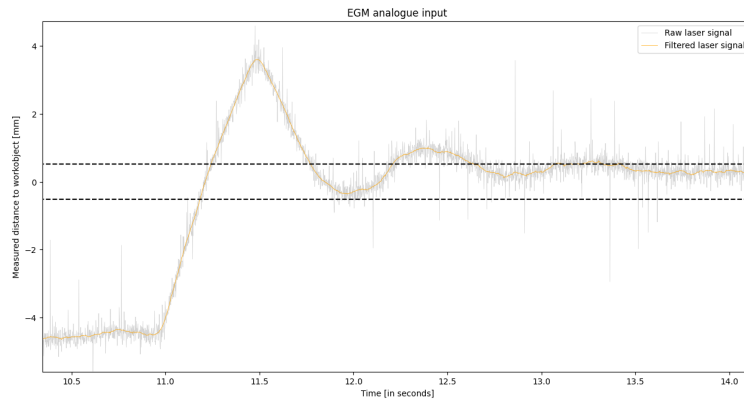


Figure 30: Settling time of the corrected step with robot speed at 50mm/s

4.3.2 EGM with analogue input with 100mm/s speed

The signal is clear in Figure 31, where the speed is 100mm/s. The experiment was repeated 7 times, but the figure is zoomed in on one run. As seen before as table skew is now rectified. When the sensor measures the step, the signal starts oscillating around -5mm, but never settles. Instead, the robot steps down and begins to oscillate around 0.5. The signal settles before the robot turns around in the middle of the figure. The robot then repeats the step on its way back to the starting point.

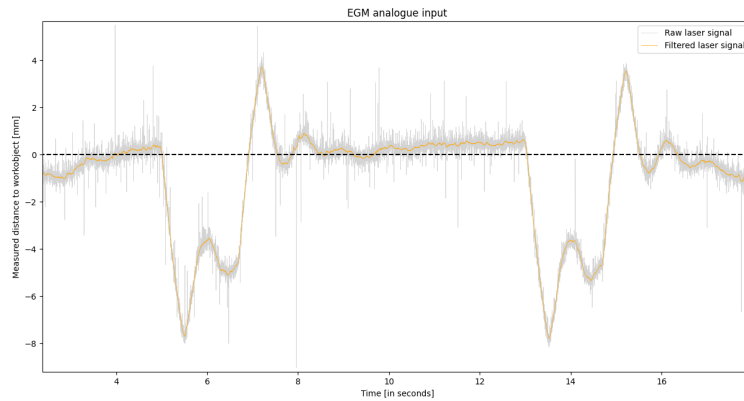


Figure 31: EGM with path correction with analogue input with robot speed at 100mm/s.

4.3 Externally guided motion with analogue input

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	-0.939	2.331	1 591
2	-0.937	2.340	1 622
3	-0.939	2.350	1 639
4	-0.933	2.331	1 616
5	-0.937	2.323	1 639
6	-0.932	2.333	1 664
7	-0.934	2.337	1 628
Overall	-0.936	2.335	1 628

Table 9: Mean, standard deviation, and settling time from the experiment with robot speed at 100mm/s.

Table 9 presents the calculated means and standard deviations of the experiment. The standard deviation between the mean of each test run is 0.003mm, an increase compared to the result from the experiment with a speed of 50mm/s. It is expected that the mean values vary more for higher speeds. The mean is still decreasing when the speed increases. This can still be explained by the fact that the robot spends relatively less time above the box than above the table's surface. Had the controller corrected all the displacements, the mean value should have increased when the speed was increased. Lastly, the standard deviation has increased compared to before, which is natural.

Figure 32 shows one run for 100mm/s, sampled with RobView. The test run starts at the blue dot to the left and ends at the blue dot to the right. The table skew is visible at the beginning of the run. The robot does not oscillate when it steps up. When it steps down again, the robot oscillates around -1mm a couple of times. In the middle of the figure is the turning point, and the robot repeats the step on its way back to end the test run.

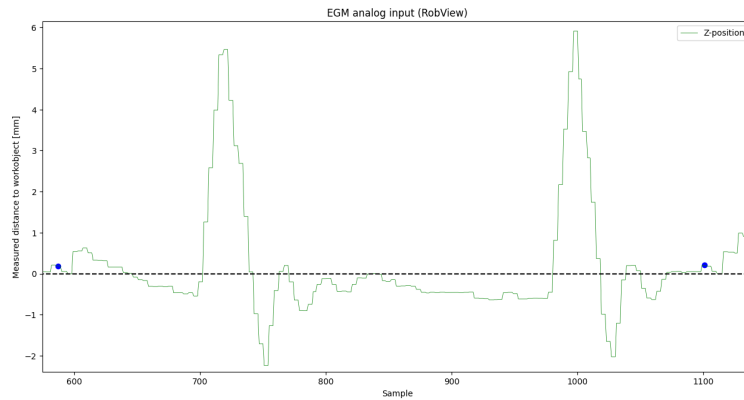


Figure 32: RobView data displaying the robot's z-position when using path corrections with robot speed at 100mm/s.

4.3 Externally guided motion with analogue input

Figure 33 is the zoomed-in result of the step. It takes 1.59 seconds for the movement to settle, less than for the 50mm/s experiment.

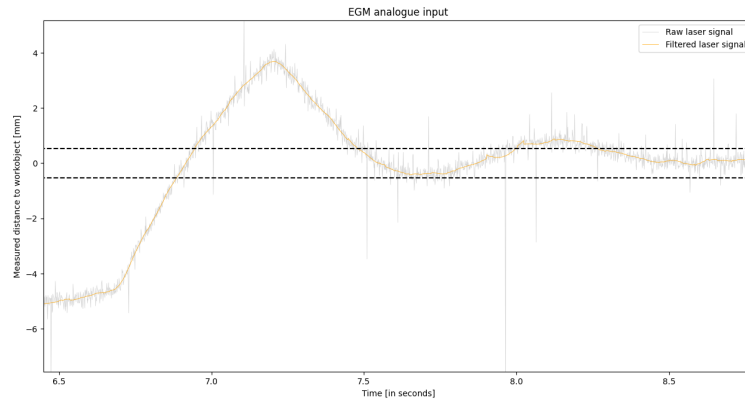


Figure 33: Settling time of the corrected step at 100mm/s

4.3.3 EGM with analogue input with 200mm/s speed

Figure 34 shows the experiment for EGM with analogue input where the speed is 200mm/s. The experiment was repeated 15 times, and the figure shows one test run. The test run starts at the left in the figure, rectifying the table skew. This experiment is twice as fast as the last one and the robot moves so fast that the correction never stabilizes after the step. It has passed the box and steps down again before the correction is settled above the box. The oscillations after it steps down continue into the returning point in the middle of the figure. On the return, the controller repeats the step up and down and it does not stop oscillating before ending the test run.

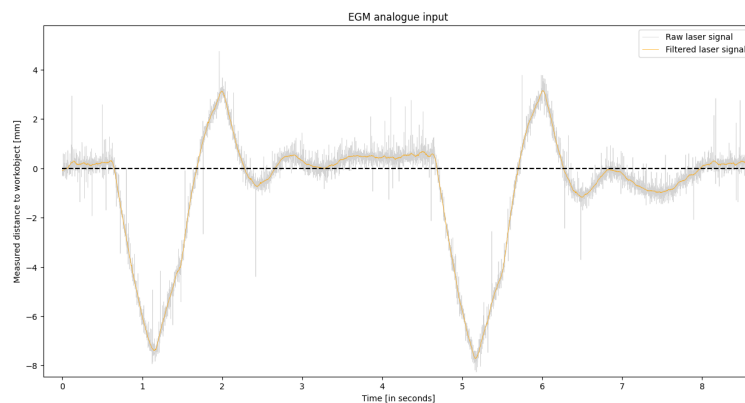


Figure 34: EGM with path correction with analogue input with robot speed at 200mm/s.

4.3 Externally guided motion with analogue input

Run	Mean [mm]	Standard deviation [mm]	Settling time [ms]
1	-0.898	2.466	x
2	-0.901	2.465	x
3	-0.903	2.459	x
4	-0.905	2.467	x
5	-0.903	2.489	x
6	-0.897	2.467	x
7	-0.896	2.467	x
Overall	-0.899	2.471	x

Table 10: Mean and standard deviation from the experiment with robot speed at 200mm/s. The correction never settled and the settling times are not measured.

Table 10 presents the calculated means and standard deviations of the experiment. The table only shows the seven first runs. The standard deviation between the mean of each test run is 0.004mm. The mean value has decreased from the two previous results, but the standard deviation has increased. The overall trend has changed from the two previous experiments. The mean value decrease with increased speed, but the standard deviation is now increased with the speed. The settling times are "not a number" in the table because the correction never settles sufficiently.

RobView sampled the same experiment as the oscilloscope and Figure 35 shows one of the runs. The test run starts at the blue dot to the left and stops at the dot to the right in the figure. The table skew is not visible anymore as the oscillations from the previous run still dominate. The robot steps up to 5mm, and then down again where it oscillates until the robot turns around in the middle of the figure. Then the step is repeated on the way back.

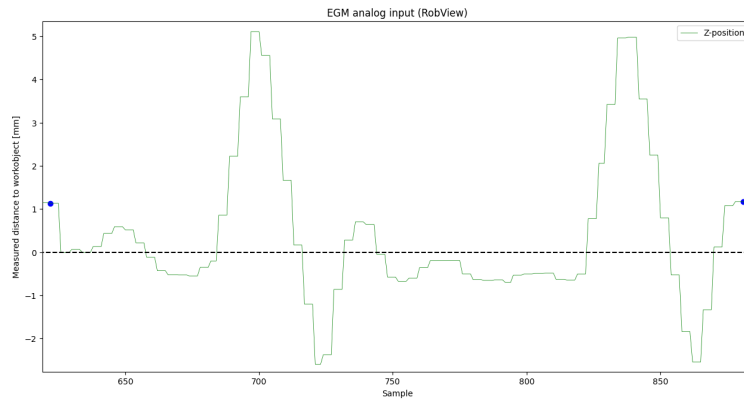


Figure 35: RobView data displaying the robot's z-position when using path corrections with robot speed at 200mm/s.

When looking closer at the signal for the highest test speed, the range of the

4.3 Externally guided motion with analogue input

robot was too short for the correction's oscillations to settle. The signal neither settles at the top of the step nor does it settle after stepping down. Figure 36 shows the result of the step, and how it never settles before it is stopped and has to return. The robot is already moving in a circular motion to exploit as much of the table as possible, but for the given workspace, the correction does not have space enough to settle.

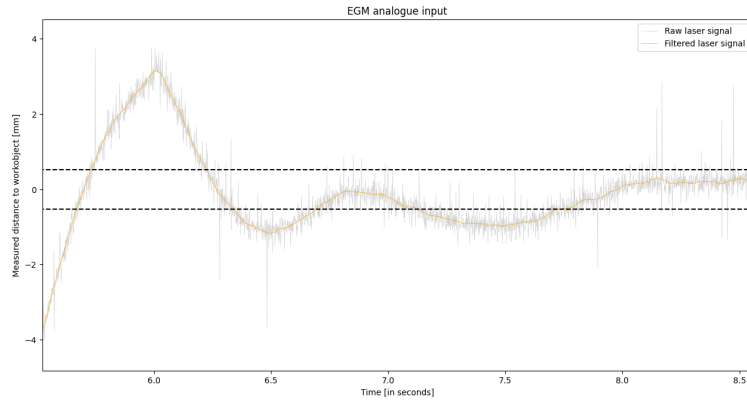


Figure 36: Settling time of the corrected step with robot speed at 200mm/s.

4.3.4 Analysis of the results

The settling times are now varying more than for the two other experiments. The correction takes an average of 66ms longer for 50mm/s than it did for 100mm/s. For the fastest experiment, the signal does not reach the full height. Therefore, the settling time for that experiment can not be compared with the others.

The differences in the repeatability between the experiments were expected to decrease as the speed was increased. The repeatability decreases for all speed increases. The mean values of each test run is varying more when the speed increases, which could be expected.

The overall mean value for EGM analogue input, decreases when the speed increases, and at the same time, the overall standard deviation increases with the speed. As mentioned, the mean value decreasing as the speed increases only indicates that the robot spends more time close to zero. It is more interesting to look at the variance between the different mean values within one experiment. The increase in standard deviation is caused by the speed, and it is natural that the samples variate more for higher speeds.

5 Discussion

Throughout this chapter, the different results will be discussed and compared with the objectives defined in Chapter 1.5. Some thoughts on the general setup will be discussed first, and then the results will be discussed.

5.1 The workspace and setup

The IRB 1200 robot, and the workspace used, proved to be suitable for certain experiments in this project. The range of the robot was maybe too short to carry out the experiments at the fastest speeds. It struggled to settle its correction for faster speeds within the given range and workspace. The table and its wooden top surface were skew, which affected the results. It was concluded early in the project that this would not be critical during the experiments because the important features were still clearly visible. However, the results could be more precise if the table was completely flat, or if the results were compared with a CAD model of the tabletop and box instead of comparing the signal to zero.

There were multiple sample rates to consider when configuring the system. The displacement sensor's sample rate is $200\mu\text{s}$ - 2ms , and the I/O module was configured to read the analogue input signal at 4ms . By having a faster sampling rate in the sensor than in the I/O module, it will always read a new sample value from the sensor. Externally Guided Motion with Path Correction has been configured to use the fastest sample rate at 24ms . The EGM controller will therefore always get a new sample. However, the space between each sample in the controller is much larger than the I/O module. In the case where the robot moves at 200mm/s , the controller only samples the displacement each 4.8mm , but in the case of 50mm/s , the sensor samples the displacement each 1.2mm . This results in a loss of accuracy relative to the surface, and the EGM controller does not "see" all the contours on the surface.

Because of the low sample rate in RobView, the oscilloscope was used as the main data sampling source. That led to a workaround where a trigger in the RAPID code was used to separate test runs in the data files. This workaround led to uncertainties in the results, but it was necessary to increase the sample rate in the figures to be able to analyse the performance. The implemented Signal Analyzer program in RobView, sample at a rate of 38 - 62ms , and the oscilloscope sampled, in these experiments, at 1.2ms . The sample rate of the oscilloscope was set to 1.2ms to ensure that all the sampled signals were included, as the sensor's sample rate is at least 2ms . The oscilloscope can start its sampling on a step in a signal. The trigger function in RAPID might be delayed in time, and for different robot speeds, the trigger's activation point can vary in the path. The problem with this is that the sampling does not start at the same point for different robot speeds. However, the speed for each experiment is constant. For one given robot speed, the trigger was activated at the same point for each repetition, and the time between each trigger was constant. The downside of not using RobView is that the laser signal can not be directly compared with the robot's position along the z-axis. The z-position and the laser signal did not align with each other in RobView, and the z-position was therefore plotted alone.

5.2 The analogue signal

The analogue signal is stable, appears smooth and has a sampling rate of at least 2ms. There is not much noise in the signal that affects the result. The Noise's order of magnitude is 0.1mm, as shown in Chapter 3.2, which is considered acceptable for this purpose. The internal signal processing in the sensor affects the signal, creating a ramp of the signal when moving over the step. The results in Chapter 4.1 give an idea of how the sensor's processing works, as the signal contains a settling time just north of 500ms to settle at the step height. This applies to all the tested speeds.

To measure settling time, the acceptance value for when the signal is settled must be defined. Usually, this is within 2-5% of the step's height, but because of the table skew, 5% of 10.5mm has been applied like the following: -9.5 ± 0.525 mm, as this is the height the signal in the practical application settles around.

The mean value can not directly imply accuracy for the analogue signal. First of all, the table's skew is a source of standard deviation, which is not corrected by the robot. Secondly, the signal's value could have been compared to a CAD drawing of the table skew with a box step in the middle, replicating the actual environment. Then the mean value would indicate how accurate the signal is. However, it would be challenging to replicate the table skew in a data program, and then use it to compare the measured signal to analyse the accuracy. Without doing this, the graphs and mean values can still be used by comparing each separate repetition to indicate how it performs and its repeatability.

The standard deviation between the mean in the different test runs for 50mm/s was calculated to be 0.002mm while the robot's repeatability is 0.02mm. This implies that the sensor can provide a signal which is repeatable to a satisfactory degree for the given speed. The standard deviation between the mean was measured to 0.017mm for the runs with 100mm/s speed, and 0.003mm for 200mm/s. The natural result would be if the repeatability decreased as the speed increased. The repeatability of the 200mm/s experiment being better than the 100mm/s experiment is considered to be a coincidence. From 50mm/s to 100mm/s, the results clearly indicate that the laser's repeatability and accuracy decrease along the z-axis when the speed increases. The repeatability of the robot on linear paths is 0.02mm, and the results are all more precise than that.

The displacement sensor is capable of reading precise values, and it measures the step's height precisely. When the speed is 200mm/s, the signal never reaches the full 10.5mm, and the accuracy has decreased. The robot should not move much faster for the given setup and configuration if the accuracy is to be conserved.

The results from the analogue signal alone seem promising before EGM is tested. These experiments indicate that 200mm/s might be too fast for the given sampling rate of the sensor. The settling of about 500ms should be considered when choosing the speed, as the robot could move far in half a second. At the same time, the robot will seldom have to correct for perfect steps in a Pixel-Paint application. The repeatability and accuracy of the signal are sufficient to theoretically improve the precision along the z-axis.

5.3 EGM with analogue output

By testing the path correction capabilities with analogue output, the controller's precision and settling time can be investigated for an EGM program. The gained knowledge can be used in a path correction program which uses the analogue input.

The EGM controller corrects the height smoothly and precisely, and the signal does not contain any oscillations when subjected to step responses. The robot's corrections rather undershoot compared to the analogue signal itself, as it uses around twice the time to settle. For the two slowest speeds, the corrections are indeed 10.5mm, as the actual box. The settling time has increased from what was seen in chapter 4.1, even though receiving a momentary change in height.

As for the analogue signal's mean values, neither can the mean of these experiments directly imply accuracy. The table skew is not corrected and the corrections move the robot farther from the surface. The measured values accumulate, and the mean values do not indicate accuracy. However, the standard deviations of the mean values for each test run can suggest how repeatable the EGM controller is. By visual inspection of the graphs, the accuracy can be determined instead. For 50mm/s and 100mm/s, the accuracy is good and the path is corrected precisely.

When the robot is moving 50mm/s the standard deviation between the test runs' mean is 0.003mm. This repeatability is better than that of the robot at 0.02mm. The settling time averages 1.24 seconds. When the speed increases to 100mm/s, the standard deviation between the means of each test run is 0.004mm, and the settling time averages 1.13 seconds. EGM with analogue output is quite repeatable for these two speeds. However, the settling time is double what is the case for the displacement sensor. Again, for the fastest experiment, the signal never reaches the step's full height. The space between the step up and down is 170mm and at a speed of 200mm/s the time the robot spends 0.85 seconds above the simulated box. So it is clear that it never will reach its full height when the settling time is above 1 second. Hence, moving the robot 200mm/s was for this experiment too fast.

Repeatability of EGM analogue output has, through these experiments, proved to be good for lower speeds. By taking this performance and changing the signal to analogue input, the correction should also be precise and repeatable. The analogue input behaves like a ramp, and the EGM output is a somewhat slower ramp. The corrections based on the analogue input should then be able to achieve something similar.

5.4 EGM with analogue input

Before the actual path correction was tested, the analogue input signal and the EGM controller were surveyed. These results, for the slower speeds, were promising for both accuracy and repeatability. No oscillations were seen in the corrections based on the analogue output. When introducing the EGM Path Correction with analogue input, the results were quite different. The corrections suffered from overshooting and oscillations, and the controller only corrected about half the displacement.

5.4 EGM with analogue input

The calculated repeatability for 50mm/s is 0.001mm and 0.003mm for 100mm/s, both highly repeatable. For the 200mm/s the repeatability is 0.004mm/s, but as seen previously, it might move too fast to capture all the details, and the oscillations never stops. These results are valuable because it says something about the sensor, the robot, and the EGM controller's capability of achieving the same result every time.

To measure settling time, the acceptance value for when the signal is settled, must be defined. Usually, When the signal is within 5% of the step's height, it is considered accepted. Since the signal is never corrected completely, the settling time is measured by looking at the step down from the box. This ensures that the oscillations have enough time to cease. The 50mm/s experiments used 1.694 seconds to settle, 66ms more than for 100mm/s, which settled after 1.628 seconds. The 200mm/s experiment never settled and is therefore considered to be moving too fast for real-time path correction for the given configuration.

High accuracy was expected of the path corrections if the speed was kept around 50-100mm/s. This would imply low mean values, favorably as close to zero as possible. The controller is supposed to correct the robot's position, seeing the laser signal move toward zero at all times. The mean values are naturally closer to zero than they were for the previous experiments. The values would be even closer to zero if the controller managed to correct the displacement all the way. Considering the standard deviation, which increases along with the speed, can be contemplated as natural. However, the mean values decrease when the speed increases. This can be explained by the fact that the robot moves faster, and therefore spends relatively more time above the table than the step. The system did not correct the signal all the way back to zero and instead stabilized with a constant deviation from the zero line.

The configuration was the same for *EGM analogue output* and *EGM analogue input*. The only difference was in the RAPID function to set up for either output or input. Therefore, it was expected accurate corrections for the *EGM analogue input* program. However, there was a constant displacement even after the correction was settled. This constant displacement could be seen in RobotStudio, in the signal's graph, and in the laser sensor's digital display. The source for the constant displacement was not located, and could not be rectified. The correction was tried for different step heights to see if this could affect the results. Nevertheless, the displacement continued to be around half of the step's height.

6 Conclusion and future work

This thesis has studied the feasibility of ABB's externally guided motion for improved accuracy relative to an object. The robot was equipped with a laser displacement sensor to increase the TCP's relative accuracy and test the EGM controller's performance.

The thesis is built around the theory behind defined research objectives, and the method and the results from the experiments. Then a discussion about the method and results was offered. To conclude the thesis, the research objectives will be considered before a few thoughts about future work will be discussed.

6.1 Conclusion

The project started by studying the analogue signal before testing Externally Guided Motion for analogue output. The results from these experiments were promising for the lower speeds. The analogue signal from the external sensor is perceived as a ramp with a settling time north of 500ms. From the result achieved through experiments, the signal's accuracy in itself is high with a high resolution. When using EGM with the analogue output configuration, the EGM controller corrects the simulated step with high accuracy. The EGM controller itself possesses the ability to control and correct the path in real-time with a settling time of just above 1 second.

When the analogue input configuration was implemented for the Path Correction, the performance did not satisfy the requirements. The robot can follow a path and correct for changes in the path, such as a step. However, the controller does not correct the displacement all the way to zero. Based on the experiment's results the robot struggles to keep a constant distance from the object because of the challenges with correcting the laser signal to zero. Oscillations are not wanted for applications requiring high accuracy. However, the repeatability is high for all three experiments. The error in the path after an attempted correction is significant. The accuracy achieved with this configuration, and this RAPID setup is not sufficient for a system that is being surveyed to be used in PixelPaint applications.

6.2 Future work

The first thought for future work is to study why the correction is not completed and why the correction is settled around half of the physical step. The configurations for *EGM analogue output* and *EGM analogue input* are the same, and the signal the robot controller receives is not zero. It measures what is shown in the graphs, and the controller should correct this signal.

An interesting behavior to study is how corrections along the z-axis possibly affect the accuracy in the x-y plane. PixelPaint requires high accuracy in all directions, and if real-time corrections affect the precision in the horizontal plane, this will need corrections as well.

Another way to follow a surface with constant distance could be to use Position Guidance to create a trajectory in real-time. The external sensor could be set up as a *look ahead tracker*, and an external program could generate targets and a trajectory in real-time. This could possibly open up to re-orient the end-effector with respect to the direction of movement and shape of the object. The

6.2 Future work

sampling rate of the EGM Position Guidance could be 4ms, six times faster than for Path Corrections.

REFERENCES

References

- [1] Mark Fairchild. *Paint robots: Benefits, Applications and How to Source Them. How To Robot*, 2021.
- [2] Udo Grohmann. *Paint robots in the automotive industry – process and cost optimization. ABB Review*, 1996.
- [3] Spong & Vidyasgar Hutchinson. *Robot Modeling and Control*. John Wiley & Sons, INC, 2010.
- [4] Siciliano Sciavicco Villani and Oriolo. *Robotics: Modelling, Planning and Control*. Springer, 2006.
- [5] ABB Robotics. *PixelPaint - Unmasks the future of customized automotive painting*, 2022.
- [6] ABB Robotics. *Application manual - Externally Guided Motion*, g edition, 2022. Document ID: 3HAC073318-001.
- [7] ABB Robotics. *Product Specification - IRB 1200*, k edition, 2019. Document ID: 3HAC046982-001.
- [8] Zhang & Wang Wu, Gao. *Workspace and dynamic performance evaluation of the parallel manipulators in a spray-painting equipment. Robotics and Computer-Integrated Manufacturing*, 44(1):199–207, 2017.
- [9] & Bonev Khaled, Akhrif. *Dynamic Path Correction of an Industrial Robot Using a Distance Sensor and an ADRC Controller. IEEE/ASME Transactions on Mechatronics*, 26(3):1646 – 1656, 2020.
- [10] Ørjan Mæhre. *Following Moving Objects Using Externally Guided Motion (EGM)*. Master’s thesis, Univeristy of Stavanger, 2016.
- [11] ABB. *Technical reference manual RAPID Instructions, Functions and Data types*, f edition, October 2017.
- [12] Li & Guo Bi, Fang. *Extrinsic calibration of a laser displacement sensor in a non-contact coordinate measuring machine. Chinese Journal of Aeronautics*, 30(4):1528–1537, 2017.
- [13] Panasonic Electric Works. *Compact Laser Displacement Sensor: HL-G1*, October 2010. Number: CE-HLG1-7.
- [14] Panasonic Electric Works. *Panasonic Laser Displacement Sensor: HL-G1 User’s Manual*, May 2010. Number: MEUEN-HLG1V1.
- [15] ABB Robotics Bryne. *RobView: Robot status monitoring*, Unknown.
- [16] Rohde & Schwarz. *R&S RTM3000 Oscilloscope*, 08 edition, 2022.
- [17] ABB Robotics. *Application manual - Scalable I/O*, j edition, 2022. Document ID: 3HAC059109-001.
- [18] Eivind Sandve Haus. *The use of motion tracking for teaching paint robots*. Master’s thesis, Univeristy of Stavanger, 2018.

REFERENCES

- [19] Ronald W. Schafer. What is a savitzky-golay filter? [lecture notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

A Appendix

The poster from the presentation is on the next page.

Use of external sensors to correct robot path in real time

Erlend Ettersti Thorkildsen
Robot technology and signal processing, University of Stavanger

Abstract

The thesis studies the feasibility of using ABB's Externally Guided Motion feature to correct a robot path in real time. In particular, the thesis focuses on the relative accuracy of the robot by measuring the TCP distance to an object and trying to keep this constant. The research explores the technology's relative accuracy and repeatability and considers the potential of applying this to PixelPaint.

The experiments and results suggest that the repeatability of EGM is dependent on velocity, but in general is high. It also put forward that the relative accuracy is not very good, partly because of oscillations, but mainly because of incorrect setting point. The sampling rate of 24ms limits the velocity.

In summary, the study contributes to understand the possibilities in EGM and highlights some limitations the system experiences. The limitations could be further researched to enhance the performance of the real time corrections.



Introduction

The use of robots for painting automobiles has been gaining popularity due to their efficiency, accuracy, and repeatability.

Pixel painting involves creating an image by painting individual pixels with a specific colour. ABB's inquit head features more than 2000 nozzles, that can be controlled individually. This improves the painting precision significantly which increases the paint transfer efficiency to save paint, and the use of masking becomes redundant.

The accuracy required in PixelPaint is high, and small errors can result in significant deviations from the desired result. It is desired to keep a constant distance of 5mm to apply PixelPaint to a surface.

ABB Robots have an option feature called Externally Guided Motion (EGM) which overrides the planned path based on external input. With input from an external source into the control system, it provides partial control of the robot's movement in real time.

Through this project the accuracy of EGM will be studied. EGM's capabilities to keep a fixed distance from an object will be tested. Other factors resulting from EGM such as time lag, oscillations, error in path, smoothness, and repeatability will also be studied. The main focus will be to increase precision relative to an object and study the compensation performance of externally guided motion.

Methods and Materials

During the project, the IRB 1200 robot has been used along with Panasonic HI-G112-A-C5 sensor. The setup and work-environment looks like Figure 1. The laser sensor is mounted on the robot's end-effector and measures the distance to the table. The correction works along the z-axis.

The correction was tested for the extreme case to study the step response/ settling time and possible oscillations. The experiments were repeated several times and for multiple velocities.

Results

The figures below show the result from the experiment where the robot moves with 100mm/s. The laser measures the step and starts correcting. The controller tries to always correct the signal back to the zero line. The interesting part is that it oscillates around -5mm instead of 0. Then it steps down and starts oscillating around the zero line. It then turns around and repeats the step onto the box.

The second image is zoomed in on the first figure to show more data on the oscillation and the settling time. The settling was measured to 1.572 seconds.

The repeatability was studied by looking at the differences in the mean values for every run. It was measured to be 0.003.

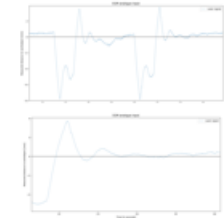


Table 1. Measured results from EGM analogue input at 100mm/s.

Run	Mean	Deviation
1	-0.939	2.314
2	-0.937	2.325
3	-0.939	2.335
4	-0.933	2.314
5	-0.937	2.309
Overall	-0.936	2.320

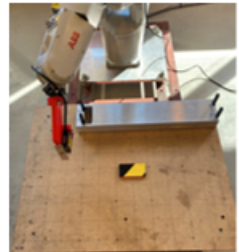


Figure 1: The work environment

Discussion

The results presented is only for 100mm/s, but the same experiment was repeated for 50mm/s and 200mm/s to study how the velocity affected the results.

The oscillations are undesired for applications requiring high accuracy. However, the most concerning part is that the controller can not correct the signal all the way to zero. The mean value presented in the table should have been closer to zero, as the graph should always be corrected towards zero. However, it only corrects half the displacement. This has been studied without finding an answer. When performing the same experiment, using an analogue output value as the source for corrections, the EGM controller corrects perfectly.

When increasing the velocity, the sampling at each 24ms becomes to slow to read the surface detailed. For lower velocities, the sample rate is sufficient.

The repeatability in both the analogue signal and the EGM controller is high. The tasks can be performed with the same quality each time.

Conclusions

Throughout this project EGM's Path Correction, and its capabilities to correct a path in real time, has been studied. The results are presented with graphs of the laser signal and calculations of the means and deviation of each test run. The repeatability is measured by comparing the different mean values.

The repeatability is sufficient. The robot and the EGM controller manages to repeat the same task multiple times with the same result.

However, there are unanswered questions, like why the controller only corrects half the displacement. The ability to move with a constant distance to an object is not good enough to ensure sufficient accuracy for PixelPaint. With further research, the challenges could be solved, and the accuracy could be evaluated again.

Contact

Erlend Ettersti Thorkildsen
University of Stavanger, ABB Robotics Bryce
Email: erlend.etterstithorkildsen@gmail.com
Phone: 98635410