Universitetet
i Stavanger

FACULTY OF TECHNOLOGY AND SCIENCE

MASTER THESIS

Study program: Master in Science and Engineering / Robot Technology and Signal Processing	Spring 2023 Open
Author: Alide Irene Gundersen	
Course responsible: Morten Mossige Supervisor(s): Morten Mossige	
Title: Improved automatic testing of events on robot-path measurements used in robots Credits: 30	
Keywords: events on robot path, I/O synchronization PixelPaint, ABB, RobotStudio	Pages: 42 + Attachments/other: 5 Stavanger 15. June 2023

Abstract

This master's thesis investigates the activation time of electrical signals along a robot's path and explores the potential impact of calculation errors in robot position and speed. The study emphasizes the need for reliable and consistent results in systems that require high precision and accuracy. Two primary experiments were conducted to examine the existence of calculation errors.

The first experiment involved altering the robot's path while maintaining a constant speed. Activation times of the electrical signal were recorded and compared to a stationary reference point, aiming to identify potential calculation errors in position. The findings suggest the presence of calculation errors, but limited data points resulted in no definitive conclusions.

The second experiment focused on traversing the same path with varying robot speeds. Activation times were recorded and compared to the stationary reference point to identify potential calculation errors in robot speed. Although indications of calculation errors were observed, conclusive results were not obtained.

While no definitive conclusions regarding the existence or extent of calculation errors were reached, this thesis establishes a framework for future experimentation. The framework provides a structured approach for further investigations, enabling researchers to understand the relationship between calculation errors of robot position and speed, and activation time.

Preface

This thesis was in collaboration with ABB Robotics Bryne. It was through their expertise and equipment this thesis was possible. I would like to thank my supervisor Morten Mossige and all the people at ABB Robotics for being of great help during this thesis.

Lastly, I want to thank all my fellow students in the master's program for Robot Technology and Signal Processing at the University of Stavanger. I would not have come this far without their help and companionship through challenging assignments. I'm happy it's over, but looking back at the memories I've made with fondness.

Contents

Chapter 1

Introduction

1.1 ABB robotics Bryne

This thesis is in collaboration with ABB Robotics Bryne. Over the years, they have invited hundreds of students to participate in their research, which has benefited both the firm and the students. The research facility specializes in embedded systems, robotic control, and electronic design. ABB Robotics Bryne has the primary responsibility for the industrial paint robots developed by the company. They are developing a new robot controller, and this thesis aims to aid their research.

1.2 PixelPaint

Industrial paint robots have been used in the car painting industry for many years. Paint robots for the car industry have increased in popularity as opposed to manual painting due to their precision, speed and even coating [8]. As the demand for custom-painted cars increases, the automation of this process becomes more relevant. Previously, this process involved applying masking tape to achieve the design, spraying over it, and then demasking. ABB's PixelPaint creates the desired image by painting individual pixels with a specific color. The robot's inkjet head has more than 1000 nozzles that can all be controlled individually. This makes the masking process redundant and increases the paint transfer efficiency to save paint [2] [3]. Figure 1.1 shows PixelPaint in operation.

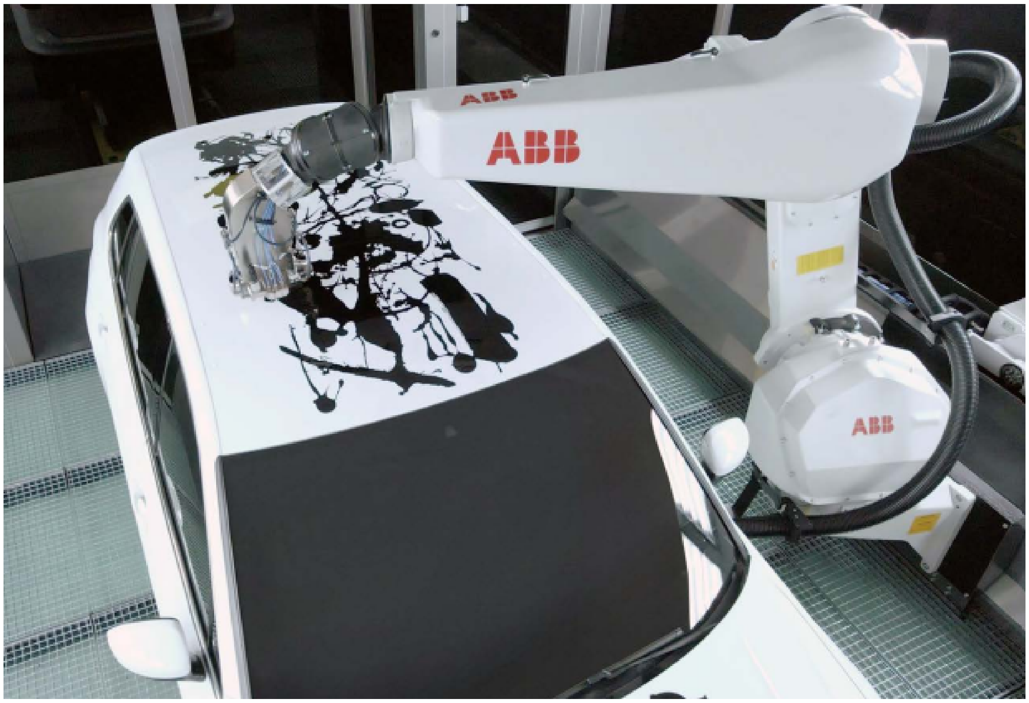


Figure 1.1: Picture of custom car paint using PixelPaint. Figure provided by ABB robotics Pixel Paint manual, page 1 [3].

PixelPaint requires high precision, meaning the margin of error is small, as minor mistakes can lead to significant deviations in the design. The paint nozzles must be activated precisely over the correct area to create the desired design. ABB's robots have high precision and can move at a constant speed. However, activating an electrical signal, like a paint nozzle, at the correct time and place can be challenging. When activating an electrical signal along a path, the robot calculates when to activate the signal based on its velocity and position. However, each run's activation time may vary, even on the same path. The difference in activation time is studied in this thesis.

1.3 Thesis focus

Moving a robot along a path while activating an electrical signal requires synchronizing the motion system and the input/output (I/O) system. Testing this synchronization can be challenging, but it is important to further improve robotic systems such as PixelPaint.

Throughout this thesis, a framework for testing I/O synchronization was built. By studying the difference in activation time, one can determine a time interval Δt and express how precise the synchronization is. This will be done through a series of tests on a robot in motion while activating an electrical signal.

ABB robots have a built-in function *TriggL* which activates an electrical signal while moving on a linear path. This function will be utilized in this thesis. To properly track when the signal was activated while the robot is in motion, a stationary reference point is needed. A laser fork sensor serves this purpose, tracking when the robot passes through a point. The *TriggL* command will activate an electrical signal at approximately the same time as passing through the laser fork. An oscilloscope will track the electrical signal from the robot and the laser fork sensor.

Other important factors in I/O synchronization, such as clock synchronization will also be explained briefly.

1.4 Report structure

The report consists of six chapters. Chapter 1 briefly introduces the motivation and background of the thesis. Chapter 2 tackles the prerequisites required to properly understand the report. Chapter 3 describes the equipment used, the work area, and the data acquisition software. Chapter 4 handles the experiments conducted as well as the results. The results will be discussed in Chapter 5, and finally, a conclusion will be presented in Chapter 6.

Chapter 2

Prerequisites

This chapter contains the theoretical information required to comprehend the thesis. The reader is expected to have some basic knowledge of robotic manipulators before reading this thesis. This chapter is split into several sections explaining some basics of robotic manipulators and paint robots including PixelPaint. The precision timing protocols for the operating system are also described, as the results rely on accurate timing.

2.1 Six degrees of freedom robot

For many years, robots have been utilized as substitutes for human workers in repetitive or potentially hazardous tasks. Typically, industrial robots are programmed to operate within highly structured environments, where most of the decision-making and variability are carefully engineered out of the workplace. Such tasks include welding, pick-and-place operations, and spray painting. Industrial robots are cited to have decreased labor costs, and increased precision and productivity, compared to the same task performed by a human worker [15] [16]. This chapter focuses on explaining some basic information about robotic manipulators, as well as some terms used in this thesis.

A robotic manipulator consists of several links connected by either rotational or prismatic joints. They usually have an end-effector at the robotic manipulator's final joint, enabling it to interact with the environment and perform a given task. The center point of the end-effector is referred to as the tool center point (TCP). A robot's accuracy is calculated by how close the TCP gets to a given point [11] [16].

The displacement between two joints is called the joint variables. If all the joint variables are known, the position and orientation of all the robot links can be inferred. An object has six degrees of freedom (DOF) in three-dimensional space. This means the robot needs at least six DOF to interact with the object, three for position and three for orientation. As the number of axis (DOF) increases, the calculations for all the joint variables become more complex. More DOF than what is necessary is called a kinematically redundant manipulator. The total volume swept out by the end-effector as the robotic manipulator executes all possible motions is called the workspace [16].

Robotic manipulators have a limited range of motion. Singularities refer to configurations where certain ranges of motion become unattainable. These singularities can manifest as points on the boundary of the robotic manipulator's workspace, representing targets that the robot cannot reach [16]. In practice, the robot will slow down to maneuver around the difficult configurations, or it could completely stop close to singularities. Therefore, there is a need to configure the robot path in a manner that avoids singularities.

2.2 Industrial paint robots

The car painting industry has been using robots to coat cars in an even layer of paint since 1980. Paint robots have increased the painting speed by up to 50% compared to human workers. The surface finish of the car also is more even. Paint loss is reduced by about 0.5 liters per car when using a paint robot, compared to a human worker. Working in a paintshop is physically demanding and one is constantly exposed to toxic fumes. Therefore, using a paint robot greatly reduces the health hazard for the workers [5] [9].

2.2.1 PixelPaint

Previously, applying a contrast color or pattern onto a car required a time-consuming procedure involving masking, spraying, and demasking. As customer demand for customized cars increases, ABB's PixelPaint significantly reduces this time with its non-overspray, two-tone painting application. PixelPaint can paint complicated designs onto vehicles by painting pixels independently with specific colors, which removes the need for manual masking/demasking. Waste from the overspray is then reduced, with no paint lost to the masking, resulting in reduced cost [3]. Figure 2.1 displays a car painted with the PixelPaint technology.

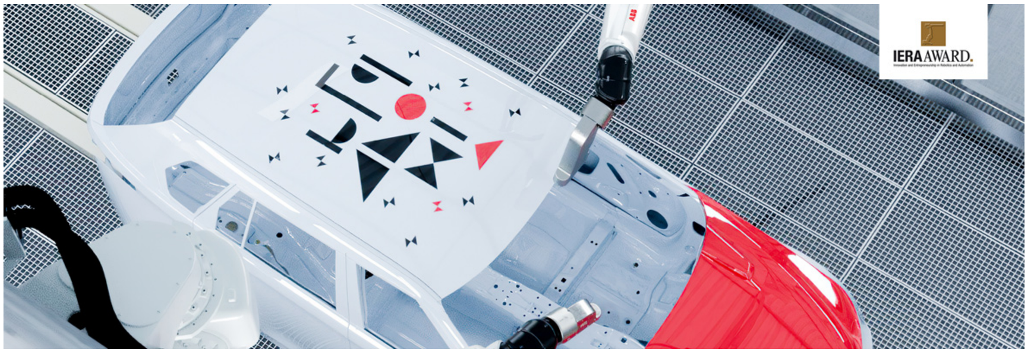


Figure 2.1: Picture of custom car paint by using PixelPaint. The figure is provided by ABB robotics from their article about PixelPaint [2].

For the image to have the desired design, the paint nozzle must be activated at the right time to paint the correct area. Small errors in activation time may lead to significant deviations in the paint job. Figure 2.2 illustrates an example of painting a straight line in PixelPaint. If the paint nozzle is not activated at the correct time each time it paints the line, the result might look crooked.

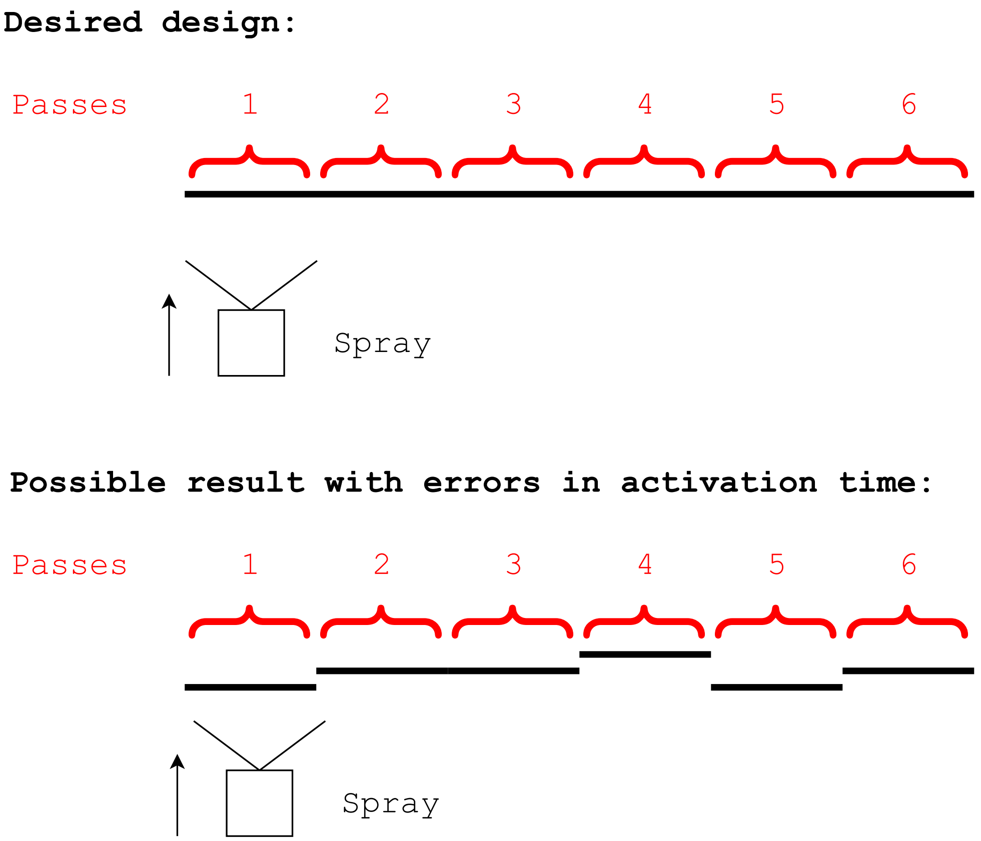


Figure 2.2: Example of possible errors while painting a straight line with PixelPaint.

Some internal research at ABB has revealed that faults within the paint job equal to or larger than 0.1 mm are visible to the human eye. This implies that when the paint nozzle is traveling at 100 mm/s the margin of error is less than 1 ms. This margin decreases as the traveling speed of the robot increases, as shown in Figure 2.3.

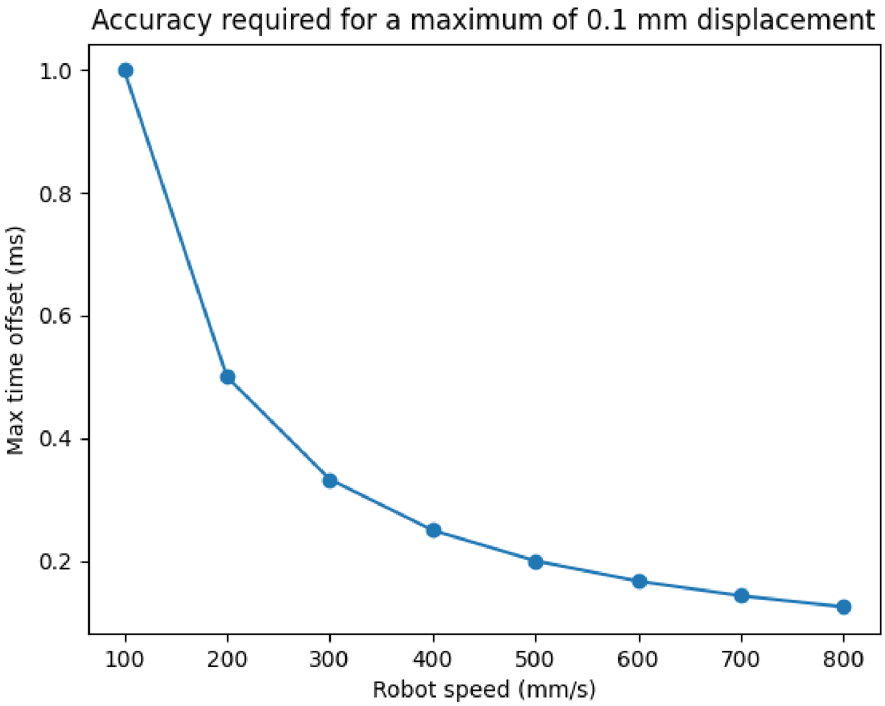


Figure 2.3: Time differential vs. robot speed with a maximum displacement of 0.1mm.

2.3 Precision Timing Protocol - IEEE1588 standard

When activation timing needs a high level of accuracy and precision, there needs to be a framework in place that ensures this. Several industry standards are in place for precision clock synchronization for control systems. The one in this control system between the microcontroller in the robot and the I/O system is the precision timing protocol (PTP) IEEE1588 standard.

The IEEE 1588 standard is a precision timing protocol for precise clock synchronization in network measurement and control systems, offering high accuracy at the sub-microsecond level without relying on GPS receivers. The connected computer systems are referenced to a master clock reference time, and the slave clocks estimate their offset from the master clock time. The master clock periodically sends synchronization messages to the slave. The master clock records the time it sends the synchronization message (m_t), and the slave records the time it receives the synchronization message (s_t). The master to slave delay d_{m2s} is defined by equation 2.1 [6][10].

$$d_{m2s} = m_t - s_t \tag{2.1}$$

The synchronization protocol periodically updates the recorded delay. The IEEE1588 standard also accounts for the time delay by sending the messages from slave to master (d_{s2m}) and from master to slave (d_{m2s}). Then, the message propagation delay can be calculated as shown in equation 2.2 [10].

$$d_{prop} = \frac{d_{m2s} - d_{s2m}}{2} \tag{2.2}$$

Then, the PTP system estimates the time difference between master and slave clocks, which is the master-to-slave delay with corrections for message propagation delay. The offset from the master clock is calculated as shown in equation 2.3.

$$Offset = d_{m2s} - d_{prop} \tag{2.3}$$

2.3.1 IEEE1588 on robot motion

In the context of robot motion, robot movement instructions and electrical activation times are timestamped in reference to the master clock. The slaves calculate the time difference according to equation 2.3 and determine when to perform their respective tasks according to the master. Figure 2.4 illustrates this concept.

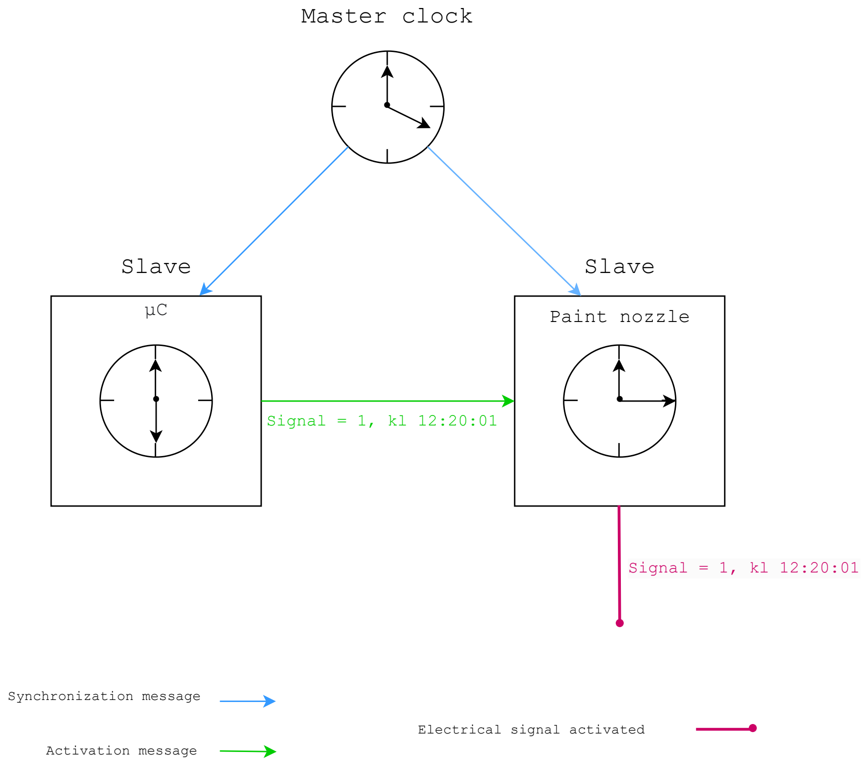


Figure 2.4: Example of the Precision Timing Protocol IEEE1588 on a robot system.

Figure 2.4 illustrates the operational mechanism of the control system in a robot. The robot controller performs calculations to determine the timing for activation of an electrical signal, which is then compared with the master clock. A message is transmitted to synchronize the activation of the electrical signal based on the timing provided by the master clock. Similarly, the paint nozzle calculates its deviation from the master clock and triggers the signal accordingly, ensuring synchronization with the master clock.

Errors in clock synchronization might lead to uniform distribution of the data, as the slave clock constantly tries to update to be closer to the master clock. This is explored in appendix A.

2.4 Example of electrical signal activation on robot path

Although the clock synchronization is highly accurate, there might be some internal calculation errors of when to activate the electrical signal. To further explain the main aim of the thesis, a simple example is provided. The numbers and points have no real-world connection and are chosen at random as a way to explain the concept better.

As shown in Figure 2.5, a robot moves from point p_1 to point p_2 with constant speed v. In point p_x, a paint nozzle needs to be on. The paint nozzle takes 200 ms to be fully active from the time the nozzle was activated. This implies that the nozzle needs to be activated 200 ms before p_x, in the point p_{on}. The robot's control system then calculates when to activate the electrical signal. The calculations are based on several factors, two of which were explored; the position of p_{on} and time until point p_{on}. The position is calculated based on the distance between point p_1 and p_{on}. Activation time is calculated based on the robot's speed v and the distance between point p_1 and p_{on}.

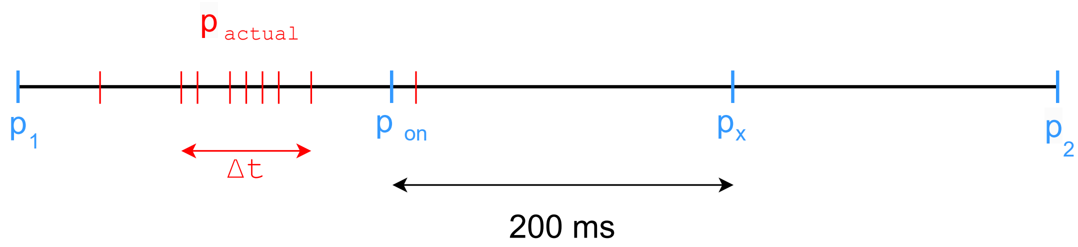


Figure 2.5: Electrical signal activation on robot path.

In an ideal world, the signal would be activated precisely at p_{on} each time the robot completed the path. However, due to minor errors in the calculation position and time, the actual activation time of the signal may vary. In the figure, these variations are denoted as p_{actual} and as shown as the red points. The average activation time can be found by analyzing p_{actual} over multiple runs. How much variation there is around the average activation time denoted as Δt.

Chapter 3

Equipment

This chapter provides an overview of the equipment utilized and the physical work area in which the research was conducted. It aims to demonstrate how these elements contribute to investigating the thesis problem. The equipment description includes technical specifications and capabilities, highlighting its suitability for capturing accurate data and facilitating reliable analysis.

3.1 Work area

Figure 3.1 shows the work area with the equipment utilized. The robot is stationary as it is mounted to the ground. The tool with the metal rod is attached to the end joint of the robot. In front of it is a small table on which the laser fork is placed. In the figure, there is also a black plastic plate in the foreground, which belongs to another thesis working on the same robot.

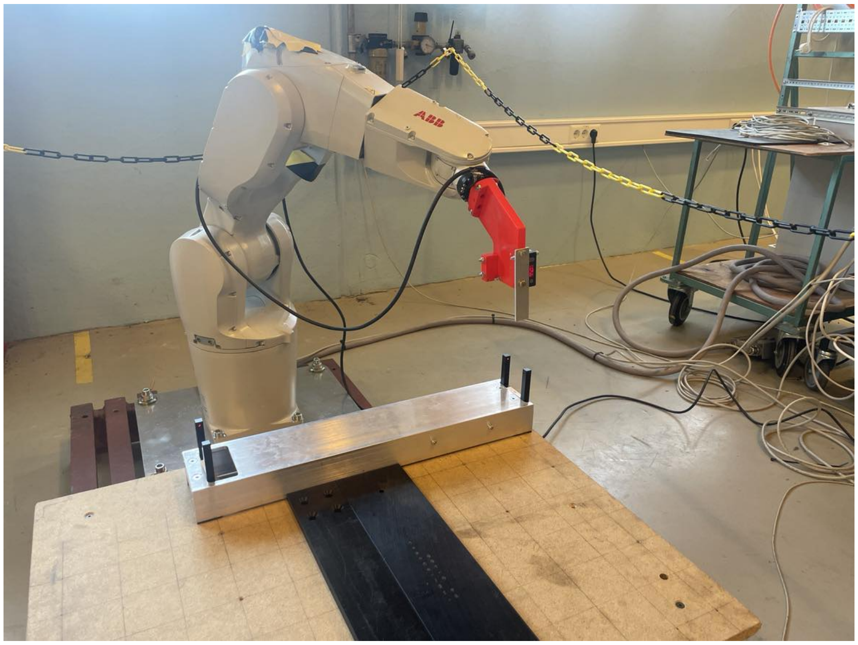


Figure 3.1: Image of the physical area with the robot, tool, and laser fork placed on the table.

3.1.1 ABB robot

The IRB 1200 robot produced by ABB Robotics is a 6-axis industrial robot and is the one pictured in Figure 3.1. It can carry up to 7 kg with a 0.7 m reach. The robot was specifically designed for manufacturing industries that require flexible robot-based automation. The robot has a pose repeatability and accuracy of 0.02 mm. It can communicate with external systems and is equipped with the OmniCore robot control software [4].

3.1.2 Robot tool

The tool attached to the end joint of a robot is a 3D-printed design, as shown in Figure 3.2. Extending from the 3D-printed plastic part of the tool, there is a thin metal plate with a length of 53mm and a width of 25mm. The tool center point (TCP) is marked in the Figure. The extending metal bar allows the TCP to pass smoothly through the laser fork sensor. The tool can be seen in Figure 3.1 as the red object mounted at the end joint of the robot.

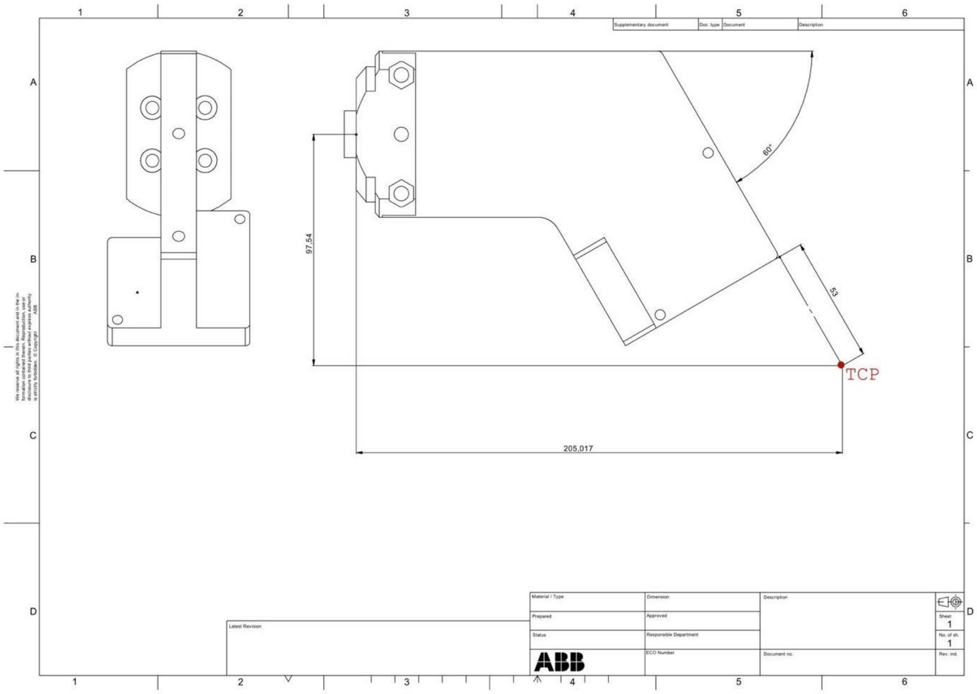


Figure 3.2: Schematic drawing of the tool, provided by ABB.

3.1.3 RobotStudio

RobotStudio is software made to handle the operations of the ABB robot. Within this software is the programming language RAPID. In this program, the tool was defined according to Figure 3.2, and the work table according to its measurements. This information was then used to create a path the robot could follow with a set speed. This program has several important built-in functions, where MoveL, TriggL, and TriggEquip are the most relevant for this thesis.

- **MoveL:** Linearly moves the robot to a set point with a set speed.
- **TriggL:** Linearly moves the robot to a set point with a set speed, with events on the path.
- **TriggEquip:** Defines what signal is set, how the signal is set (high/low), and when the signal is set.

TriggEquip was used to define where the event happens in mm, when in seconds, and which signal does what. This was combined with the TriggL command, which moves the robot to a point with a set speed [1]. For example, the code below sets the analog signal A1Needle high 0.1s before the TCP reaches a fictitious point 233mm before point p2.

Kode 3.1: Example of the TriggL command

```
TriggEquip TriggOn, 233, 0.1\DOp:=A1Needle,1;
TriggL p2, speed, TriggOn, z50, TCP\WObj:=Workobject_1;
```

3.2 Laser fork sensor as a reference point

The laser fork sensor used in this thesis is by di-soric which has a high accuracy switching point between transmitter and receiver achieved over the complete fork opening. It is also suitable in high ambient light conditions, which is preferable as it is placed right near a window. When the laser is intercepted it produces a 12V signal [7]. Figure 3.3 shows a picture of the laser fork utilized.

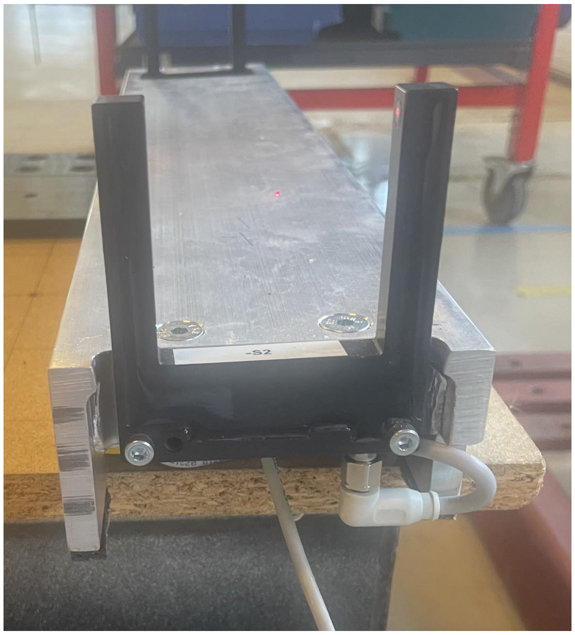


Figure 3.3: Picture of the laser fork used in the thesis.

The stationary laser fork serves as a reference to the analog output signal. The robot achieves its most consistent speed in the middle of its path, as the robot will accelerate or decelerate near a start or end point. Therefore, the laser fork is placed near the midpoint of the robot path to ensure constant speed when passing through the sensor.

3.3 Oscilloscope for analog signal sampling

The Rohde & Schwarz RTM3004 oscilloscope is a powerful instrument used to capture and analyze electrical signals. It has several advantages, including high bandwidth and sample rate. The bandwidth spans from 100 MHz to 1 GHz, and the sample rate can be as high as 5 GSamples/s. Additionally, the oscilloscope has an input sensitivity of 500 μV/div, which gives high accuracy and precision when capturing and analyzing electrical signals. Furthermore, the oscilloscope has a 10-bit ADC resolution, four times larger than the conventional 8-bit analog-digital converters [12].

Another advantage of the oscilloscope is its ability to trigger on an electrical signal, meaning it only captures the relevant part of the signals. A trigger only occurs if certain user-defined conditions are met. The oscilloscope continuously records the signal before and after the trigger, based on a user-defined interval, after which the signal is displayed. Once triggered, the oscilloscope does not record another trigger until the acquisition is complete [13].

The oscilloscope has multiple trigger modes, which define the trigger conditions. The "norm" mode is used in this thesis, which records data only if a trigger is detected. Additionally, various trigger types are available; the "rising edge" type was utilized in this thesis [13].

Another benefit of this oscilloscope is its network interface, which can be connected to an Ethernet local area network (LAN). This LAN connection provides the capability to control the instrument remotely, enabling control and data logging through a Python program [13].

3.3.1 Data acquirement from the oscilloscope to a Python program

The oscilloscope measured the electrical signals from the laser fork and the electrical signal *A1Needle*. The oscilloscope was controlled and monitored with a Python program. The program read the time difference between when the laser fork signal activated and when *A1Needle* activated. The laser fork was stationary throughout the experiment, acting as the reference point. The information was then stored in a JSON file, as shown in the code below.

Kode 3.2: The data the python program saves

```
 {"time": "2023-04-28 15:08:35.954", "trigtime": -0.0705985, "valid": true}
{"time": "2023-04-28 14:31:55.673", "trigtime": 9.91e+37, "valid": false}
```

The first variable is "time", which is the date and time the measurement was done. The following variable is "trigtime", which is the time difference between the laser fork sensor and *A1Neelde* activation. If the "valid" variable is false, the laser fork signal is recorded, but not the *A1Needle* signal. All false data points, if any, will be regarded as outliers.

Chapter 4

Experiments and results

This chapter highlights the experiments conducted to examine the time differential Δt. The activation timing of the electrical signal by the robot relies on calculations of its position and speed; therefore, both of these factors will be researched. By outlining the methodology and experimental design, this chapter sheds light on the approach to investigating the relationship between the robot's position, speed, and the resulting activation time.

4.1 Experiment with different path start points

An experiment was conducted to investigate how calculation errors in robot position might influence Δt. By moving the start point of the robot path and analyzing the results, one could gain valuable information on how Δt is affected by position. In total, 28 different start points along a rectangular path were tested. The robot moves in a repeating rectangular pattern at a constant speed of 100 mm/s, as demonstrated in Section 4.1.1. A RAPID program that automatically switches the start point after a set amount of repetition was developed for this experiment.

4.1.1 Program for experiment with different path start points

As mentioned in Section 2.1 the robot has a limited range of motion. When the robot approaches a singularity it will slow down or completely stop. To ensure the robot maintains a constant speed at the trigger point, singularity configurations should be avoided during the testing process. The robot has the most consistent speed at the midpoint of its path, as it is not accelerating or decelerating near a turn. Therefore, it is advantageous to determine the largest path while avoiding singularities. The final path used in the experiment was found through testing.

The robot follows a rectangular pattern, as shown in figure 4.1. The robot moves from point $p0$ to point $p3$ following the yellow line in the figure. The distance from $p1$ to $p2$ is 440 mm. The analog signal *A1Needle* is activated in p_on and deactivated in p_off. The laser fork is placed approximately in the same position as p_on. After 1000 repetitions of the same path the start point $p1$ is moved closer to p_on, as shown in the flowchart in Figure 4.3. Figure 4.2 is a zoomed-in view of $p1$ in figure 4.1, to highlight the displacement of the point.

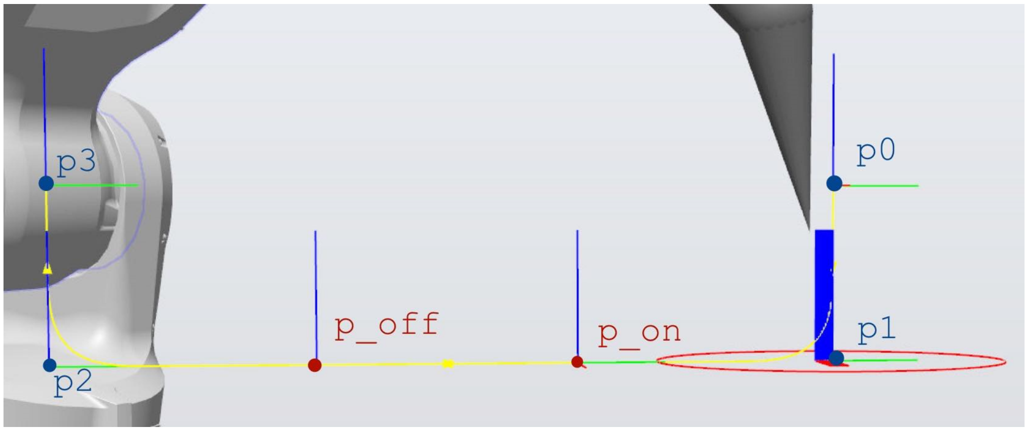


Figure 4.1: Robot path for position test, shown in RobotStudio simulation tool.

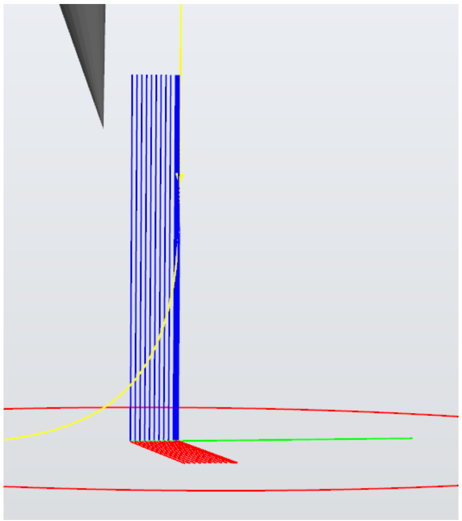


Figure 4.2: Zoomed in to show the different points from $p1$, shown in the RobotStudio simulation tool.

Figure 4.3 illustrates how the RAPID program works. The robot follows the same path shown in Figure 4.1. After 1000 repetitions the robot stops for five minutes, and the start point is moved closer to the trigger point. The wait time is purely for data identification purposes, as all the data will be collected in one continuous JSON file. This way, the different data sets were identified by creating a Python program that searches for a five-minute gap between two data points.

Three different displacement intervals were tested in this experiment. One where the start point was moved 0.01 mm closer to p_on until it reached 0.1 mm displaced from the original start point p1. Then the start point moved in intervals of 0.1 mm until the displacement was 1 mm. Finally, it moved in 1 mm increments until it reached a total displacement of 10 mm.

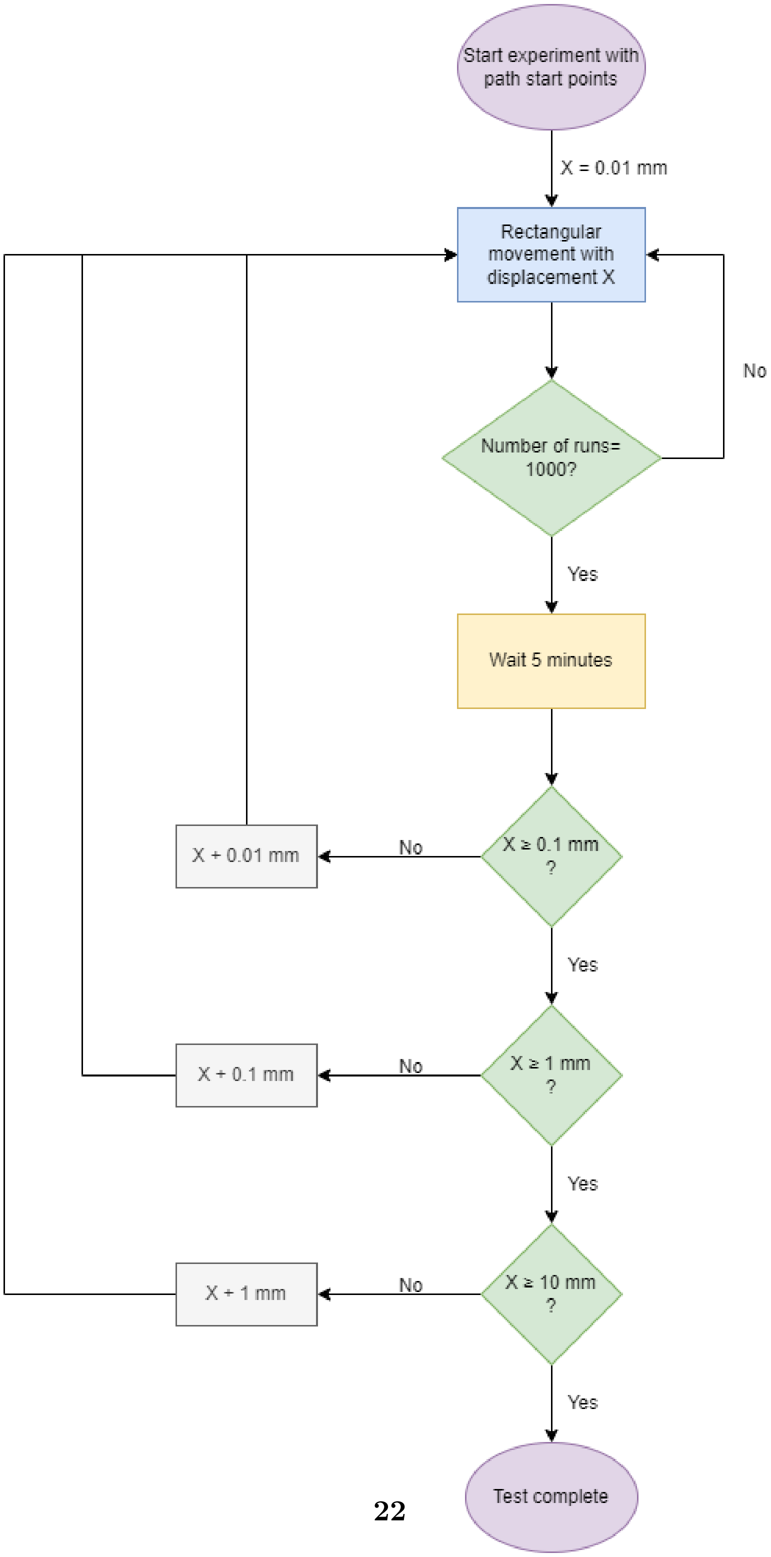


Figure 4.3: Flow chart for robot start point program.

4.1.2 Average triggtime from start point experiment

Figure 4.4 depicts the average triggtime observed in the experiment with 0.01 mm increments in offset. A total of 1000 tests were conducted for each offset value. The average triggtime varied between 5.867 ms before the laser fork and 5.668 ms. Thus, the maximum difference observed was 0.199 ms. There appears to be a pattern of decreasing then increasing average triggtime in the graph.

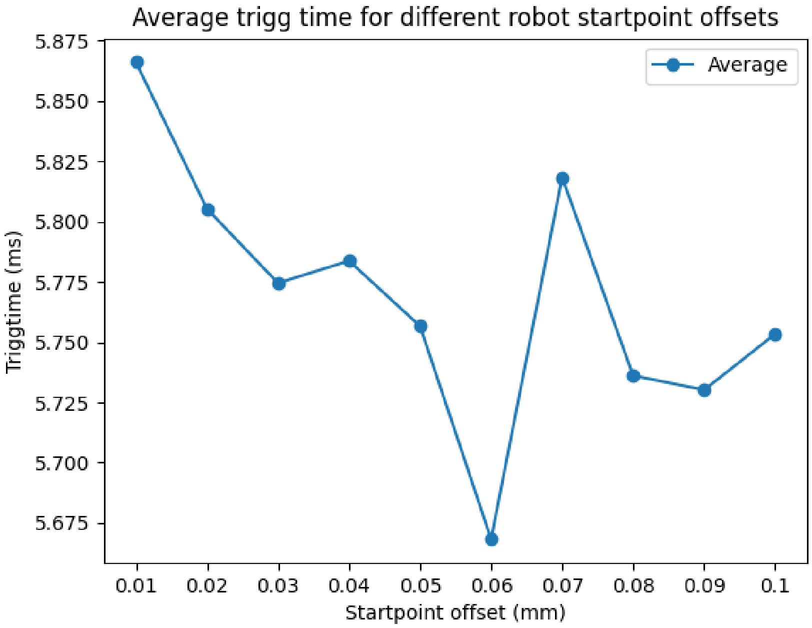


Figure 4.4: Average triggtime for 0.01 mm offset increments.

Figure 4.5 depicts the average triggtime observed in the experiment with 0.1 mm increments in offset. This test was repeated 1000 times per increment. The average triggtime varied from 5.753 ms to 5.956 ms, which gives a difference of 0.203 ms. A repeating pattern of increasing and decreasing average triggtime can also be seen here.

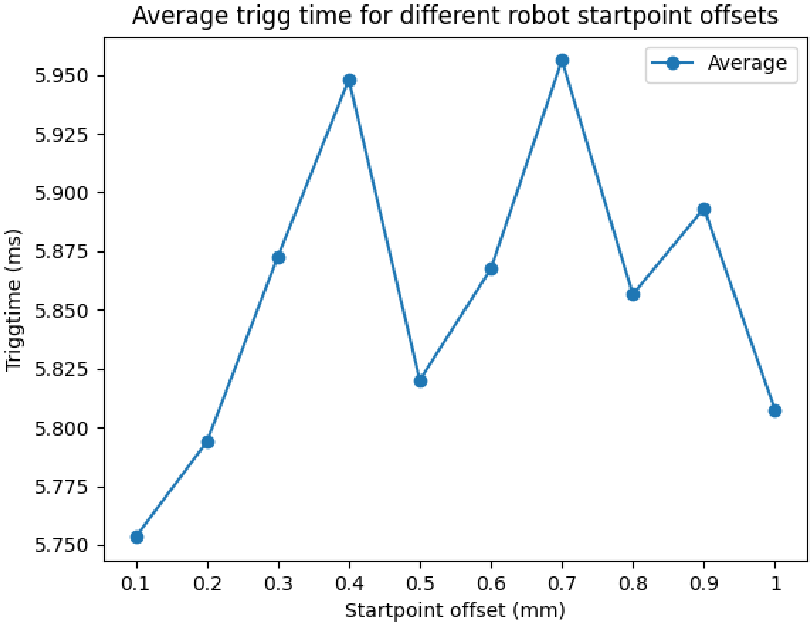


Figure 4.5: Average triggtime for 0.1 mm offset increments.

Figure 4.6 depicts the average triggtime observed in the experiment with 1 mm offset increments. As with the other experiments, each point was tested 1000 times. The average triggtime varied from 6.060 ms to 5.690 ms, a difference of 0.37 ms. This also has a pattern of sinking and rising average triggtime.

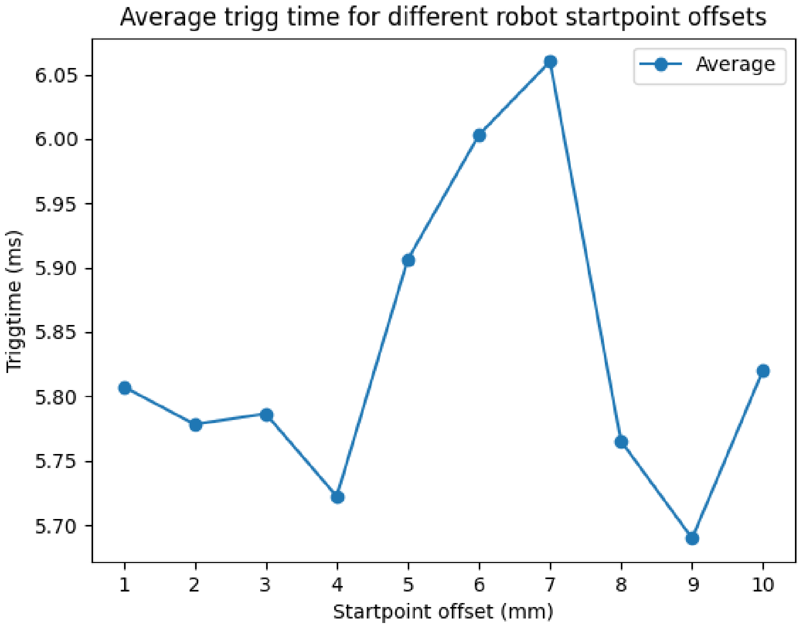


Figure 4.6: Average triggtime for 1 mm offset increments.

4.1.3 Standard deviation and distribution from start point experiment

The distribution of the data for the experiment with 0.08 mm offset was plotted, and shown in Figure 4.7. The histogram in the figure demonstrates how triggtime approximates a normal curve around the average triggtime of 5.736 ms. This was done for all data points, and the normal curve was observed for all of them.

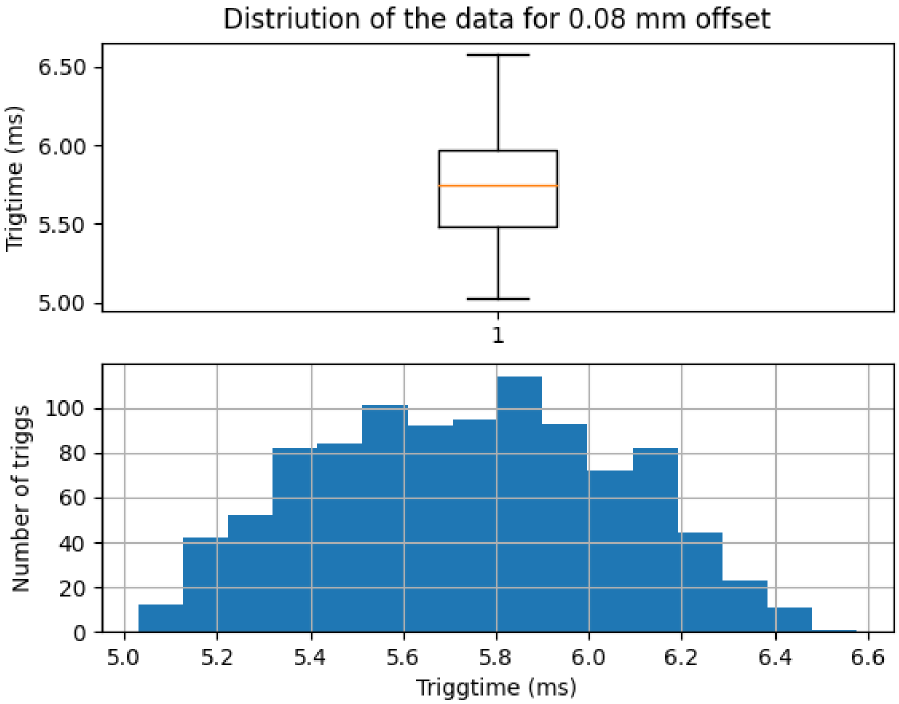


Figure 4.7: Distribution of the data for 0.08 mm offset.

The upper subplot of Figure 4.7 is a box plot. The orange line represents the median, which for this figure is 5.746 ms. The main boxes surrounding the line are the upper and lower quartile, where 50 % of the data lies. In Figure 4.7, this is 5.973 ms and 5.480 ms. The lines coming from the box are the maximum and minimum values the data has, which represent 25% of the data each. The maximum in this data set is 6.578 ms, and the minimum is 5.029 ms.

Figures 4.8, 4.9 and 4.10 displays the standard deviation from the experiments with 0.01 mm offset, 0.1 mm offset, and 1 mm offset respectively. The standard deviation from the experiments all vary between 0.348 ms and 0.304.

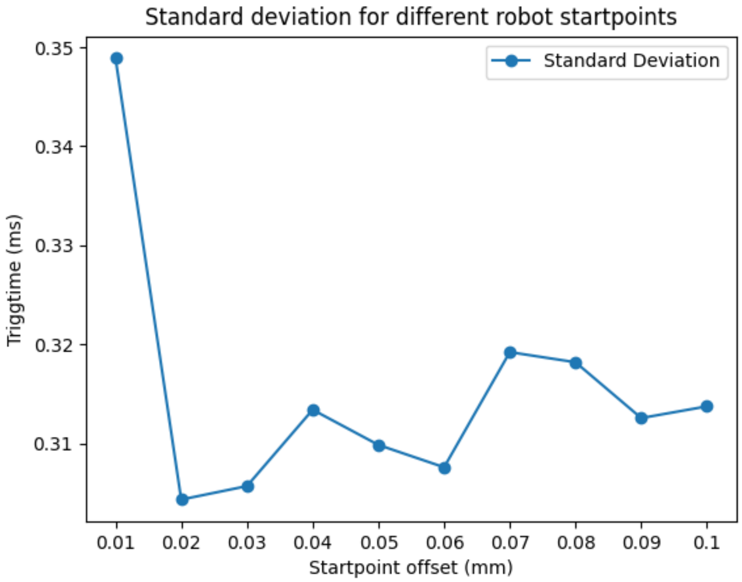


Figure 4.8: Standard deviation of triggtime for 0.01 mm offset increments.

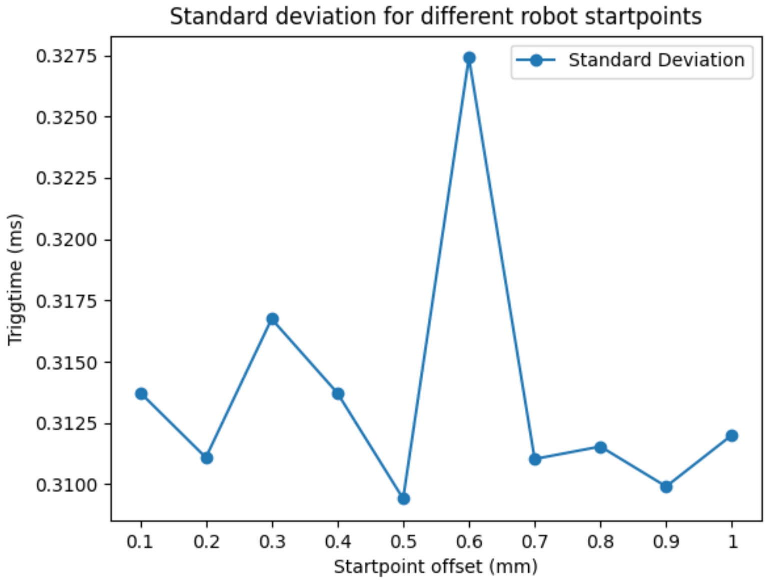


Figure 4.9: Standard deviation of triggtime for 0.1 mm offset increments.

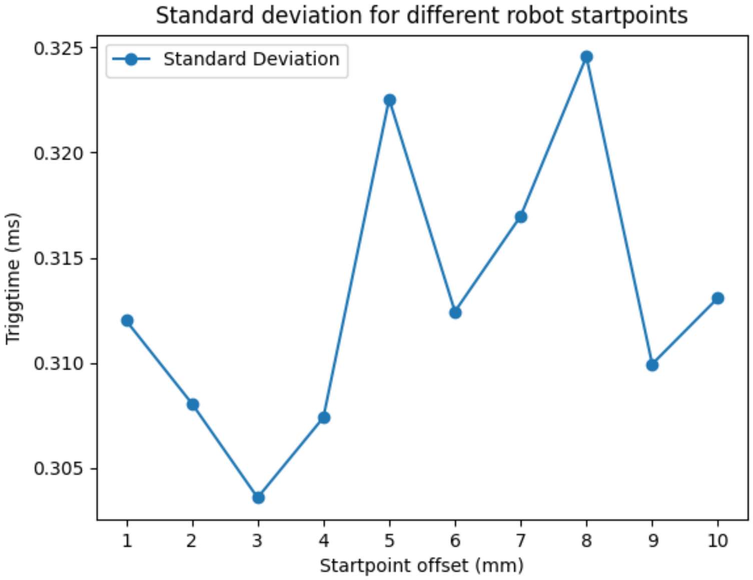


Figure 4.10: Standard deviation of triggtime for 1 mm offset increments.

4.1.4 Analysis of the results

The observed pattern resembling a mountain peak with alternating increases and decreases in average triggtime during the experiments suggests the possibility of a calculation error in the robot's control system. It appears that the calculation error in position gradually accumulates before resetting, and then repeats the process, resulting in a cyclical pattern. However, the sample size is too small, so nothing can be definitively concluded.

The normal distribution of the data indicates that the variation around average triggtime (Δt) is relatively consistent. According to the Empirical Rule, 95% of the data lies within $MeanValue \pm 2 \cdot StandardDeviation$ [14]. With a standard deviation of approximately 0.3 ms, 95% of the triggtimes would be within ± 0.6 ms of the average triggtime. This is less than the margin of error of 1 ms for visible deviations in the paint job, as described in section 2.2.1.

The experiment was repeated at a robot speed of 200 mm/s, figures shown in the appendix. Some of the same patterns in average triggtime can be seen in this experiment. A standard deviation of around 0.3 ms can be seen here too. However, this would result in a distribution larger than the margin of error, and faults with the paint job could be visible.

4.2 Experiments with different robot speeds

An experiment was conducted to investigate the potential calculation error in speed. By changing the robot's speed while maintaining the same path, information about how robot speed changes Δt could be gained. Seven different robot speeds were tested in this experiment. A RAPID program that automatically increases the speed after a set amount of repetitions was developed for this experiment.

4.2.1 Program for experiment with different robot speed

The experiment involved the robot moving in the same rectangular pattern, as depicted in Figure 4.11. The robot will move from $p0$ to $p3$ as shown in the figure. The yellow line represents the robot's path for a single run. The distance from $p1$ to $p2$ is 440mm. In p_on the analog signal *A1Needle* activates, and in p_off the signal deactivates. The laser fork is placed in the same position as p_on.

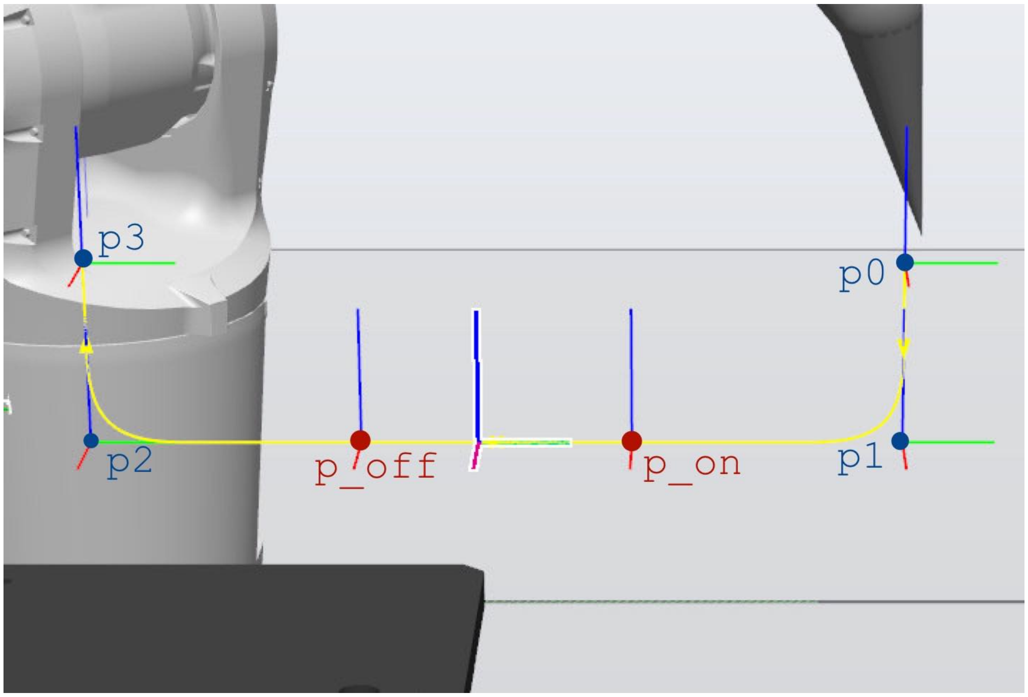


Figure 4.11: Robot path for speed test, shown in RobotStudio simulation tool.

Figure 4.12 shows a flow chart of the RAPID program developed for this experiment. The robot moved along the path shown in Figure 4.11 multiple times with a set speed. After 10000 repetitions through the same path, the robot stops for five minutes. The pause is implemented as a data separation approach, similar to the position experiment described in Section 4.1.1.

After the wait time, the robot's speed increased. The experiment was done with robot speeds of 100 mm/s, 200 mm/s, 300 mm/s, 400 mm/s, 500 mm/s, 600mm/ and 800 mm/s. 700 mm/s was not tested as RobotStudio has no setting for this robot speed.

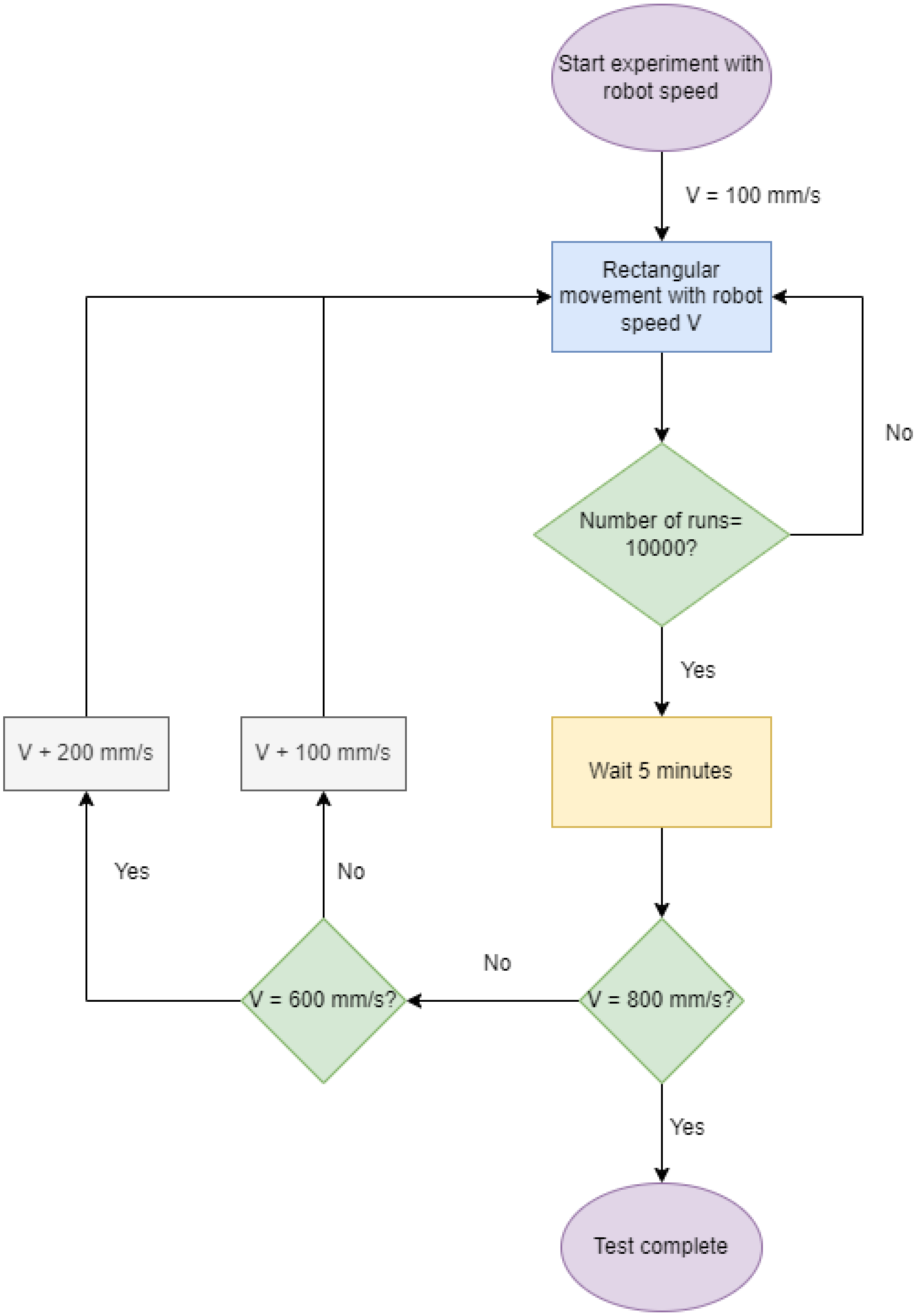


Figure 4.12: Flow chart for robot speed program.

4.2.2 Results from the speed test

Figure 4.13 shows how the average triggtime changes for each robot speed tested. Each test repeated the loop shown in Figure 4.11 10000 times per robot speed tested.

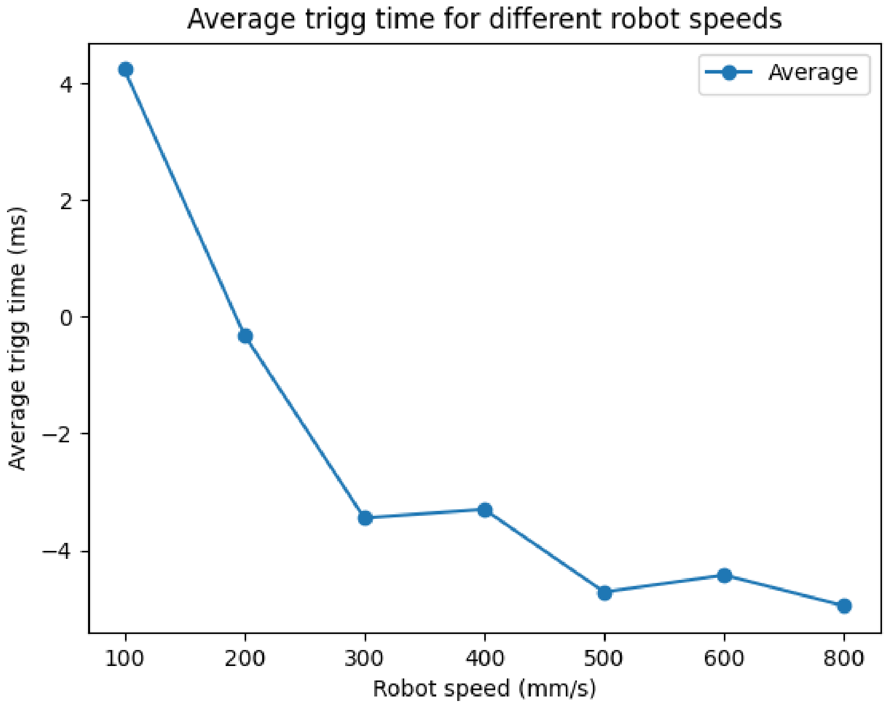


Figure 4.13: Average triggtime for each robot speed tested.

Figure 4.13 shows how the average triggtime changes based on robot speed. It goes from 4.25 ms before the laser fork at 100 mm/s to 4.96 ms after passing through the laser fork at 800 mm/s. The activation time of the electrical signal *A1Needle* is causing this increasing delay.

Figure 4.14 presents the standard deviation for each robot speed tested. The standard deviation varies from 0.296 ms to 0.325 ms.

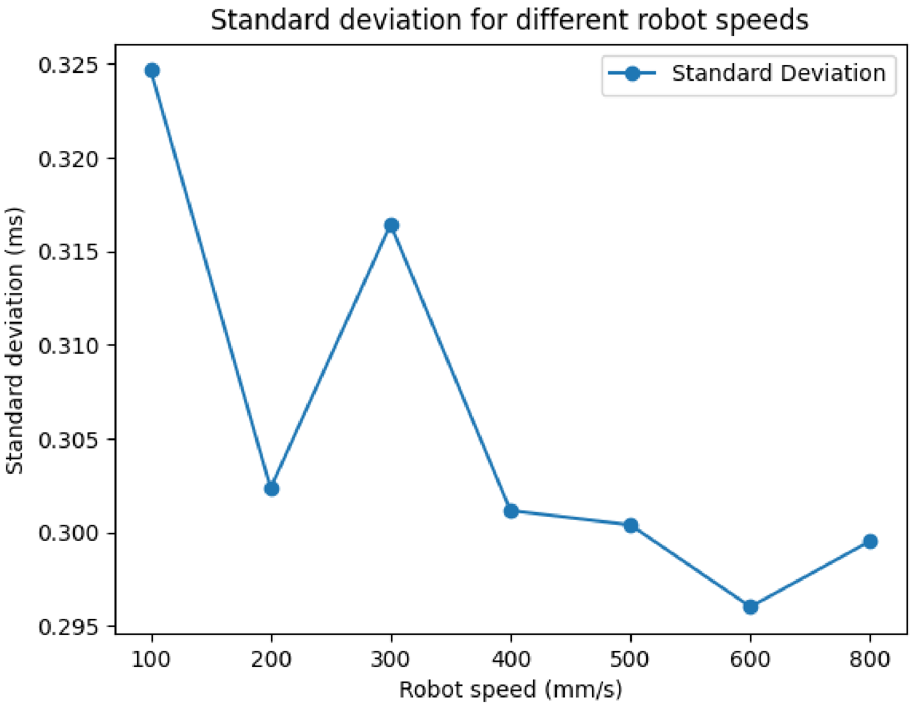


Figure 4.14: Standard deviation of triggtimes for each robot speed, after 10000 loops through the same path.

Figure 4.15 shows a box plot and a histogram for one robot speed tested, 300 mm/s. In this figure the point is to find Δt, meaning the average variation of triggtime. To find this, the average triggtime was subtracted from the data and then plotted in a box plot. In this data set, the average was 3.448 ms, meaning the orange center line of the box plot and histogram is the average triggtime. The histogram appears to be an approximation of a normal distribution.

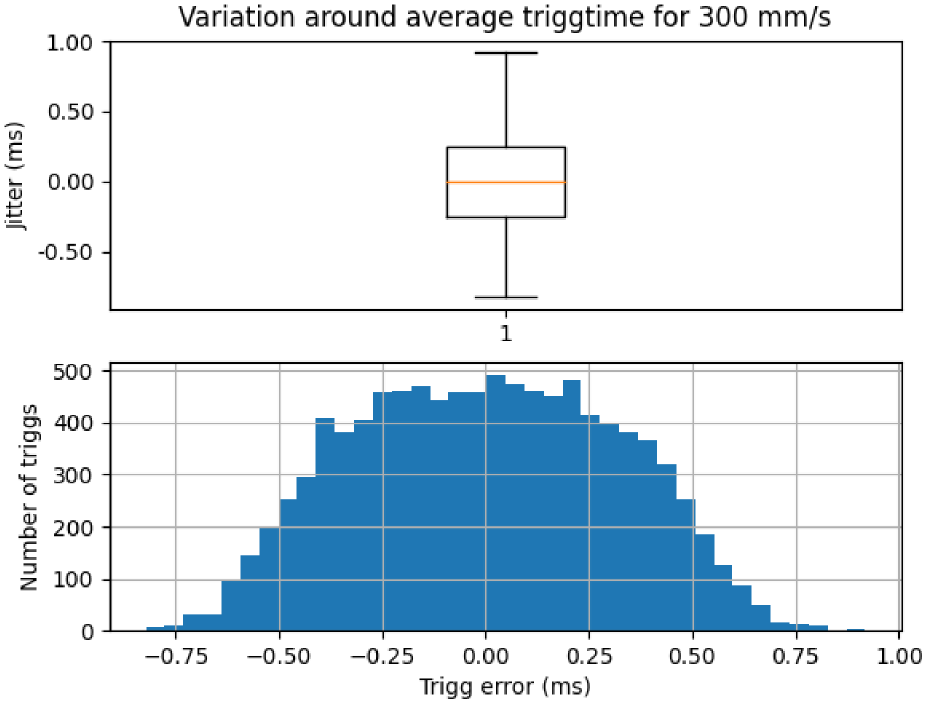


Figure 4.15: Box plot and distribution of data, 300 mm/s.

The main box in the figure represents the upper and lower quartile, where 50 % of the data lies around the median. In figure 4.15, the upper and lower quartiles are 0.245 ms and -0.250 ms. The lines coming from the box are the data's maximum and minimum values, excluding outliers. The maximum in this data set was 0.919 ms, and the minimum was -0.822 ms.

Figure 4.16 shows all box plots for all the different robot speeds. As the average will move based on the robot's speed, it is subtracted from the figure to only account for the variation around the average triggtime for each speed.

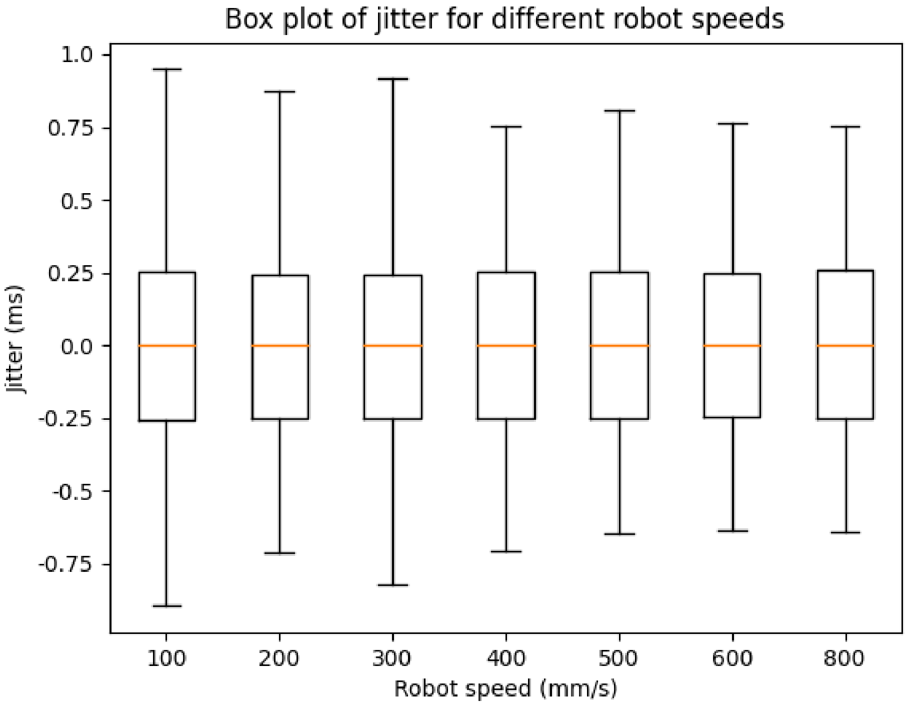


Figure 4.16: Box plots for all robot speeds tested.

From the figure it can be observed that the variation about the average value is consistent across the different speeds. 50% of the data varies around ± 250 μs of the average triggtime. The maximum from + 948 μs to + 754 μs, and the minimum varies from -895 μs to -635 μs.

4.2.3 Analysis of the results

In the experiment, it was shown that the average trigger time lagged when increasing the speed. This is expected as the electrical signal activation time remains consistent while the robot speed increases. As the robot moves faster, the robot will have passed through the laser gate before the electrical signal could be fully activated. This is why the average of each robot speed tested was subtracted in the results in figure 4.16. In this experiment, the aim was to see how much Δt varies.

The results shown by this experiment show approximately how much Δt varies. It seems to be relatively independent of robot speed, however more testing should be conducted to draw definitive conclusions. However, as explained in section 2.2.1, the margin of error becomes smaller as the robot speed increases. This means that Δt needs to decrease as the robot speed increases, this was not the case and can be seen in two different ways.

One way is by looking at the box plots in figure 4.16. The minimum and maximum at robot speed 100 mm/s were 948 μs and -895 μs respectively. This is below the maximum margin for error visible error in the paint job at 1000 μs as calculated in Chapter 2.2.1. At 200 mm/s the minimum and maximum are -713 μs and 872 μs respectively. This is more than the calculated margin for error of 500 μs, meaning that errors in the paint job might already become visible at robot speeds of 200 mm/s.

This can also be calculated based on the standard deviation. According to the Empirical Rule, 95% of the data lies within $MeanValue \pm 2 \cdot StandardDeviation$ [14]. The standard deviation for 100 mm/s was 0.325 ms, meaning 95% of the data is within ± 0.65 ms, which is less than the 1 ms margin of error. The standard deviation for 200 mm/s was 0.302 ms, meaning 95%

Chapter 5

Discussion

A framework for testing events on a robot path was set in place throughout this thesis. Two factors in the calculation of events on a robot path were tested; position and speed. Although no definitive expression or value for Δt was found, further research with the framework set in place might be useful.

The placement of the laser fork proved to be a difficult problem, as placing something by hand with millimeter accuracy is quite challenging. Ideally, it would be placed exactly at the same point that the electrical signal was activated to see how activation time varies around a set point. The laser fork was not moved between experiments to lower the impact of this placement error. This way, the placement error remains the same for all the experiments.

The time required for one experiment is also quite significant. The experiment in section 4.2 required more than 3.5 days to run. And the experiment in Section 4.1 took 3.3 days to complete. This does not consider the times the Python program got interrupted and the tests had to be repeated.

The cyclical pattern shown in Section 4.1 might indicate a calculation error. Perhaps a rounding error that changes based on the calculations done. However, due to few data points, there is insufficient evidence to state this confidently. With more time, each point should have been tested more. It would also be interesting to experiment with 0.01 mm offset from point $p1$ to p_on to truly see how much the results are affected by miscalculations in position.

The experiment and results in Section 4.1 show how robot speed affects precision. As the robot moves faster the activation time of the electrical signal will happen later. The electrical signal is not activating fast enough to activate at the exact time it passes through the laser fork. The mean value for each speed tested was subtracted to account for this. As stated earlier, the margin of error decreases as robot speed increases. From the experiment, an argument could be made that from the robot speeds tested, only 100 mm/s achieves a paint job without flaws visible to the human eye. The variation of activation time Δt seems relatively independent of robot speed, as it remains fairly consistent throughout the experiment, as shown in Figure 4.16. This could indicate that the possible calculation error in robot speed is constant for all robot speeds, but more experiments should be conducted to draw definitive conclusions.

5.1 Further work

To further investigate how calculation errors of the position affect an electrical signal's activation time, more experiments should be conducted. Each point should be repeated more times and an interval of 0.01 mm could be used until it reaches p_on. If the repeating pattern of decreasing and increasing average triggtime persists, a conclusion of a calculation error could be drawn with more confidence.

Another interesting experiment that could be done is investigating how the trajectory from point $p1$ to point p_on affects the results. Both experiments in this thesis had a robot trajectory in a straight line along the x-axis. It could be interesting to see if a circular movement affects the results. Or moving along the z-axis, up and down over the laser fork sensor. This might highlight the system's sensitivity to trajectory changes and further test for the possible calculation error in position.

Chapter 6

Conclusion

Experiments were conducted to analyze activation time on a robot path. As activation time is calculated based on robot position and speed, these factors were the main focus. The variation around average activation time (Δt) was studied to gain knowledge of possible calculation errors of robot position and speed.

The observed repeating pattern of increasing and decreasing average triggtime in the position experiment suggests the presence of a calculation error. A rounding error may be occurring, which might vary depending on the specific calculations done. Further experiments should be conducted to obtain more definitive information and confirm this hypothesis.

The experiment results from varying robot speeds indicate that the variation in activation time (Δt) does not significantly vary based on the robot's speed. This observation suggests that the calculation error remains consistent across different robot speeds. However, further experiments should be conducted to strengthen the confidence in this assumption and validate its accuracy. Additionally, the experiment indicates that visible errors in the paint job might occur at robot speeds higher than 100 mm/s.

Overall, more experiments should be conducted to draw conclusions confidently. A framework for automatic testing was developed and could be utilized further to improve the precision and accuracy of ABBs PixelPaint.

Bibliography

[1] ABB. *Technical reference manual RAPID Instructions, Functions and Data types*, f edition, October 2017.

[2] ABB. Pixelpaint. `https://new.abb.com/products/robotics/functional-modules/pixelpaint`, 2022.

[3] ABB. *PixelPaint Datasheet*, 3 edition, 2022.

[4] ABB. *Product specification IRB 1200*, s edition, September 2022.

[5] Scott Baldwin. Robotic paint automation: The pros and cons of using robots in your paint finishing system. *Metal finishing*, 105:172–175, 2008.

[6] Kendall Correll and Nick Barendt. Design considerations for software only implementations of the ieee 1588 precision time protocol. 01 2006.

[7] di-soric. *Laser fork light barriers LGUP*.

[8] Mark Fairchild. Painting robots: Benefits, applications and how to source them. `https://howtorobot.com/expert-insight/painting-robots`, 2021.

[9] Udo Grohmann. *Industrial Robot: An International Journal*, volume 23. MCB UP Ltd, 1996.

[10] Jiho Han and Deog-Kyoon Jeong. A practical implementation of ieee 1588-2008 transparent clock for distributed measurement and control systems. *IEEE Transactions on Instrumentation and Measurement*, 59(2):433–439, 2010.

[11] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation.* CRC press, 1994.

[12] Rohde & Schwarz. *R&SRTM3000 Oscilloscope Data Sheet*, 07 edition, 2017.

[13] Rohde & Schwarz. *R&SRTM3000 Oscilloscope User Manual*, 09 edition, 2021.

[14] Sheldon M Ross. *Introduction to probability and statistics for engineers and scientists.* Academic press, 2020.

[15] Balkeshwar Singh, N Sellappan, and P Kumaradhas. Evolution of industrial robots and their applications. *International Journal of emerging technology and advanced engineering*, 3(5):763–768, 2013.

[16] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*, volume 1. JOHN WILEY & SONS, INC, 2005.

Appendix A

Appendix

A.1 Clock synchronization flaw

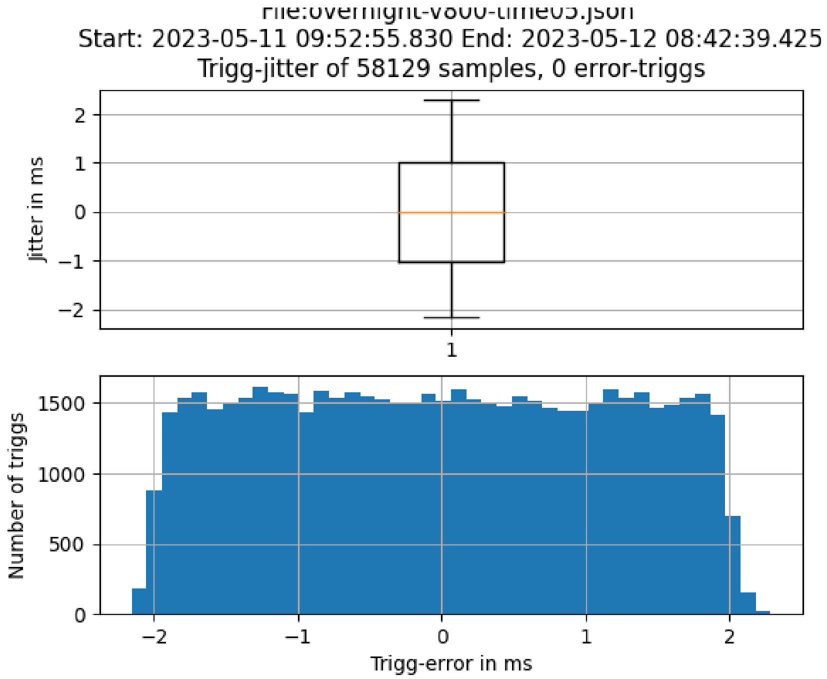


Figure A.1: Uniform distribution due to flaw in clock synchronization.

A.2 Position test with a robot speed of 200 mm/s

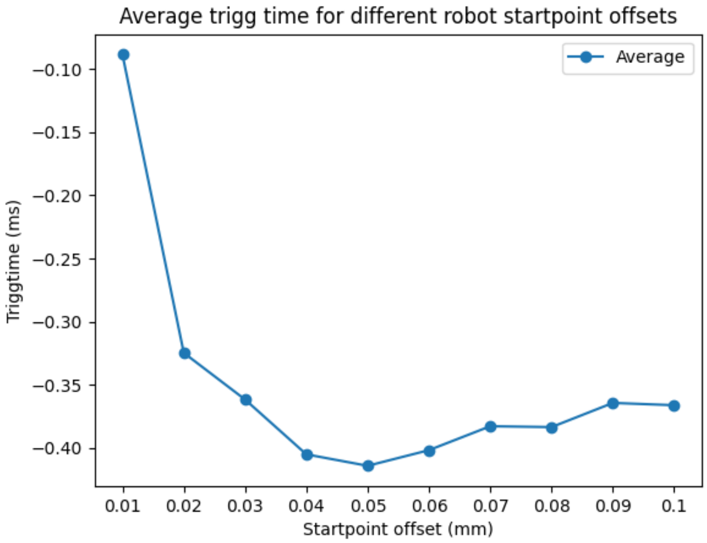


Figure A.2: Average triggtime for 0.01 mm offset increments.

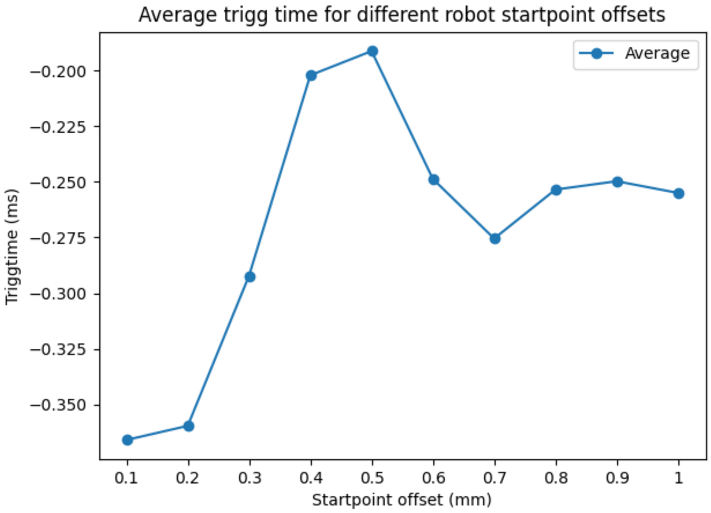


Figure A.3: Average triggtime for 0.1 mm offset increments.

A.2 Position test with a robot speed of 200 mm/s

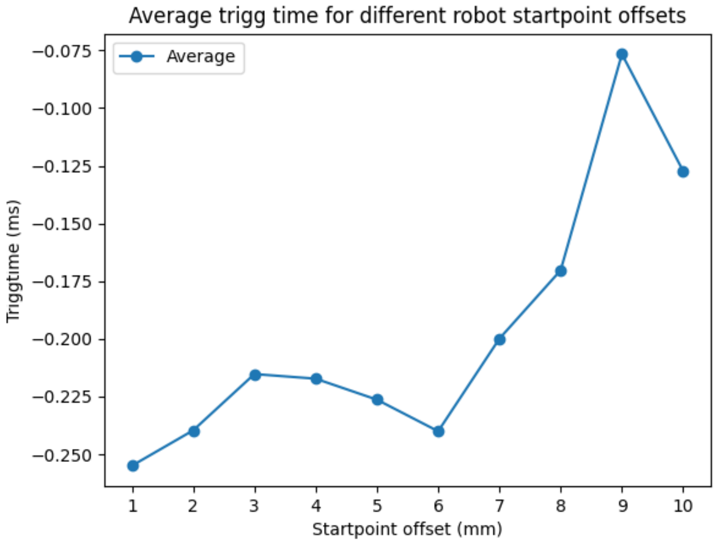


Figure A.4: Average triggtime for 1 mm offset increments.

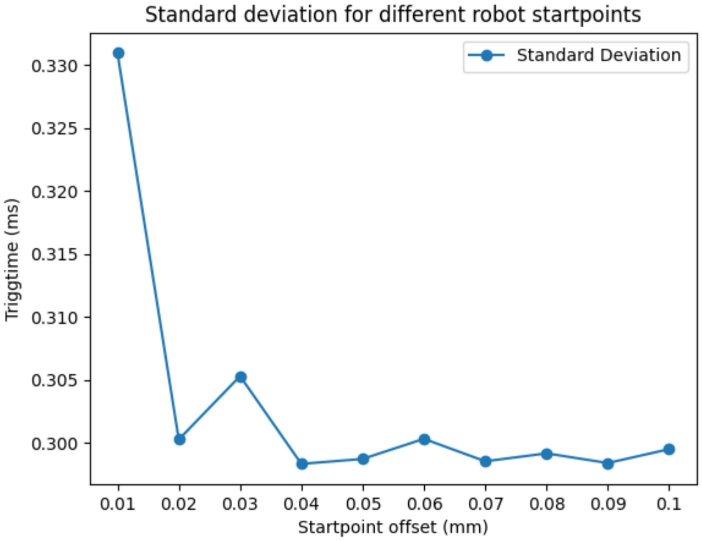


Figure A.5: Standard deviation of triggtime for 0.01 mm offset increments.

A.2 Position test with a robot speed of 200 mm/s

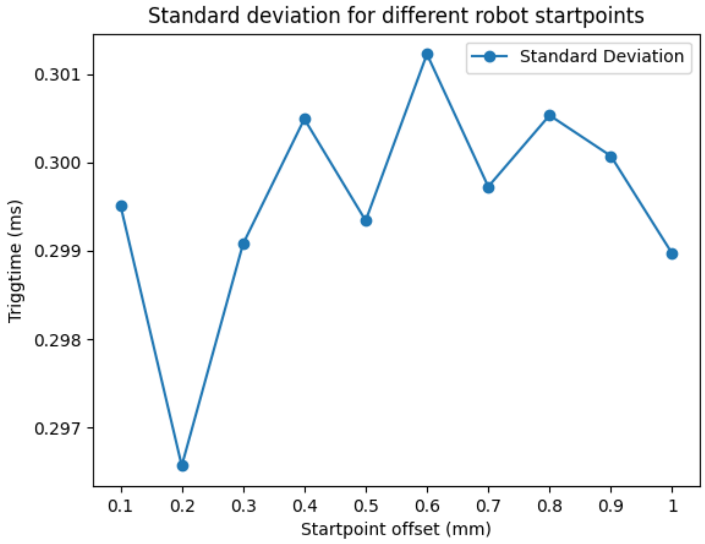


Figure A.6: Standard deviation of triggtime for 0.1 mm offset increments.

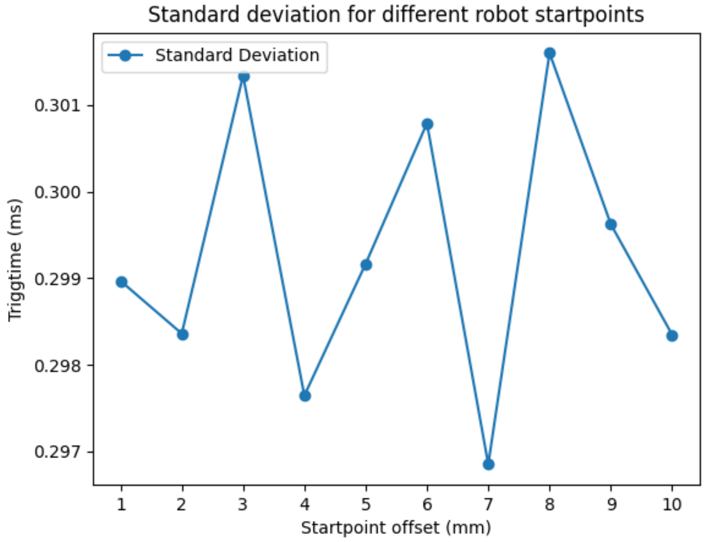


Figure A.7: Standard deviation of triggtime for 1 mm offset increments.

A.3 Poster presentation

Improved automatic testing of events on robot-path measurement used in robots

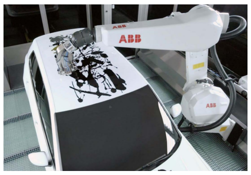
Alide Irene Gundersen
Robot technology and signal processing, University of Stavanger

Abstract

This thesis investigates the activation times of electrical signals while the robot is in motion.

When moving along a path, internal calculations are done based on position and velocity of when to activate the signal. Minor errors in the calculations lead to different activation time, this error in activation time is called Δt, and this study focuses on investigating Δt .

The studies done was aimed to find a way to accurately predict Δt in order to further the testing of systems used in ABB's PixelPaint.



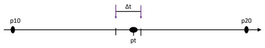
Introduction

The want for custom painted cars has recently increased, and ABB aims to find a robotic solution to replace the previous time-consuming task of paining custom images. PixelPaint is this solution, which involves creating the image by painting individual pixels with the desired color. ABB's inkjet head features more than 1000 nozzles, that can be controlled individually. This improves the painting precision significantly which increases the paint transfer efficiency and save paint as none is lost to masking tape [1].

In order to create the desired design with high precision, the margin for error is small, as tiny errors can lead to significant deviations. The paint nozzles needs to be activated at precisely over the correct time so the image to look correct.

ABB Robots have a built-in function called TriggL, which activates an electrical signal on a linear robot path [2]. This function calculates when to activate an electrical signal in order to be on at the correct position while the robot is in motion.

When moving a robot from point p10 to point p20, with the wish to activate an electrical signal at point pt the robot calculates when to activate the signal based on velocity and position. However, this calculation might not be the exact same one each time, due to some small errors. By investigating the difference (Δt) one can find an expression for the errors in calculation.



Methods and Materials

The IRB 1200 robot and a laser fork was used for this thesis. They were set up as shown in Figure 1.

The robot looped through the same rectangular pattern, with the laser fork placed in the middle of the path. An electrical signal was activated in the same point as the laser fork. An oscilloscope measured and logged the time differential between the laser fork signal and the electrical signal, where the stationary laser fork is used as the reference.

This test was done for several different speeds to test how velocity affects the accuracy.

Results

The figures below show the result from the experiment where the robot moves at 300mm/s. The robot completed the loop 1000 times.

The average trigg time was subtracted from each sample, to see the jitter around the average. This was plotted in the figure below, with a box plot showing the jitter around zero (Δt). 50% of the samples lies around -0.75ms to 0.25ms, the upper 25% is up to 0.75ms and the lower 25% is down to 0.6ms jitter.

Below there is a histogram showing the distribution of the samples. The distribution is approximately a normal distribution.

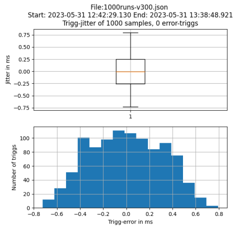


Table 1. Measured result for seven different robot speeds

Speed (mm/s)	Mean (ms)	Deviation (ms)
300	-3.94203	0.3146
400	-3.62558	0.2982
500	-4.9747	0.2976
600	-4.5983	0.2933
800	-5.0349	0.2988

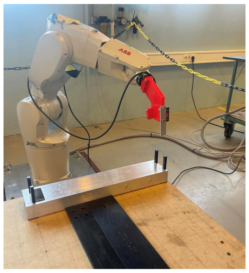


Figure 1: The work area

Discussion

The image only shows the results for 300 mm/s, but the same experiment was repeated for 400 – 800 mm/s as well, to determine if velocity has affected the result.

The normal distribution of the data and the box plot shows how Δt varies around -25ms to +25ms , this is also (about) the same for the rest of the velocities as well.

The table shows that the standard deviation was not much affected by the change in velocity.

The normal distribution over the different velocities and the consistent standard deviation might indicate that the internal calculations error is rather consistent, regardless of robot velocity.

The average activation time does change based on velocity, probably due to the errors in the internal calculations. It might be that the robot tries to compensate for the increasing speed by activating at an earlier time. However, more research into this subject is needed to draw a definite conclusion.

Conclusions

Throughout this thesis activation time of an electrical signal on a robot path has been studied. An expression to describe Δt has not been made yet . With more data, and another different test the expression can be found by further analyzing the results.

Contact

Alide Irene Gundersen
University of Stavanger, ABB Robotics Bryne
Email: ai.Gundersen@stud.uis.no
Phone: 94033744

References

[1] ABB robotics pixel paint manual
[2] ABB robotics rapid instruction manual

Figure A.8: Poster presentation