FKMP

Semester II Session 2022/2023

Robotics

BDC40603

Lecturer: Dr. Cheng Yee Low

Report Title: Robotics Group Project

Date: 28 June 2023

Mattis Ritter (JD220003), Moritz Hoehnel (JD220004), Muhammad 'Iffat Syahami Bin Saroni (DD190037), Muhammad Radzman Hakim Bin Md Taufek (DD190055)

TABLE OF CONTENTS

1. Introduction

"The adoption of robots in Industries worldwide is on the high rise. […] The main aim of Robotics and Industry 4.0 is to improve productivity, produce high quality product at low price and meet customer expectation" [1]. While Industry 4.0 is aiming for a fully automated factoring process, nowadays robots are mainly used to support humans. Applications are for lifting objects or going to places that are too dangerous for humans. Characteristics of Industrial robots are, that they are kinetic automatons that can move into different directions along axes. A robot is specialized for one certain use case. They are programmable and have tools like grippers [2].

This report is covering an application for a robot. It will discuss how to transport goods through an area with constraints. The following pages will explain the theory and mathematics of the robot trajectory as well as the implementation of a robot in laboratory conditions.

1.1 Problem Statement

The scenario, where our robot is needed is a factory that produces printed circuit boards. In the production hall the factory has two machines. They are part of one production line. The first machine is thrilling the holes into the board, the second machine is printing the circuits onto the board. The machines are not physically connected. Workpieces that leave the first machine need to enter the second machine. Furthermore, there is air cleaning device in between the machine. This device is needed to make air conditions suitable to prevent breathing trouble of workers. The issue is that the device is an obstacle and the transport of boards between the two machines is hard. A new robot shall help in this situation. This robot needs to be able lift a board from the first to the second machine, over the obstacle.

1.2 Objectives

This project has two objectives, they are listed below.

- Provide a detailed description of the mathematical background of the robot trajectory planning with an obstacle.
- Provide a hardware demonstration of avoiding an obstacle with the Niryo One robot.

1.3 Group Member Responsibilities

The following table summarizes the responsibilities of each group member.

Name	Responsibility	
	Implementation	**Report**
Mattis Ritter	- Calculation - MATLAB Calculation	-Theory - Evaluation and Analysis
Moritz Hoehnel	- Hardware Implementation	- Introduction - Hardware Demonstration - Conclusion
Muhammad 'Iffat Syahami Bin Saroni	- Calculation - MATLAB Calculation	-Theory - References
Muhammad Radzman Hakim Bin Md Taufek	- Hardware Implementation	- Methodology - Hardware Demonstration

Table 1.1: Group Member Responsibility

2 Methodology

This chapter will explain the project plan. To have a better overview a flow chart shows all working steps. As they are performed in the order like the flow chart shows, the flow chart can also be seen as work schedule.

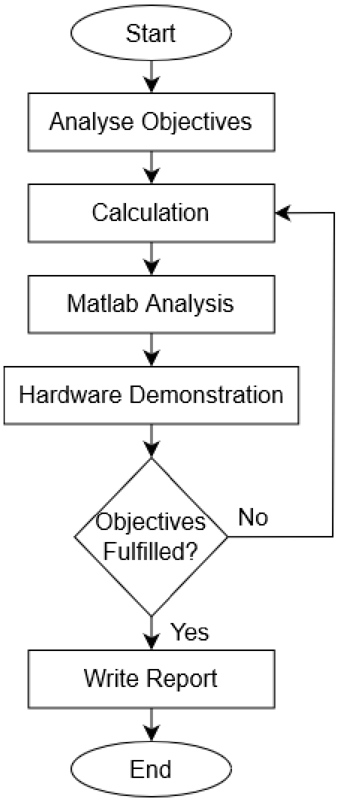


Figure 2.1: Flow Chart Project Structure

The project starts with an analysis of the objectives. Therefore, the task description of the group project got analysed. This was followed by working on the first objective, doing the calculation. In this step, theoretical trajectory planning was performed. Several calculations for different polynomial ranks were performed, more explanation can be found in chapter 4. Afterwards a MATLAB analysis was run to visualize position, velocity and acceleration. This contributes to find the best way to plan the trajectory.

After the planning and calculation was done, the laboratory sessions started to perform the hardware demonstration. A test environment was built for the Niryo One. The test environment included pick-up and drop-off locations, as well as the obstacle. The trajectory programming was done with the Niryo One Studio desktop application.

After the robot was running, the objectives were check if they were fulfilled. As soon as this was the case the team started to write the report.

2.1 Design Constraints

This subchapter introduces the limitations of the project. The hardware implementation will be held in the innovationslab of the Universiti Tun Hussein Onn Malaysia. The factory robot will be simulated by the Niryo One. This is a smaller robot than the one that will be used in the factory. But it is capable of performing similar tasks and to present the implementation in real world conditions.

3 Theory

3.1 Calculation of Trajectories

Since the scenario requires trajectory planning in order for the robot to avoid the obstacle. In this chapter trajectories that are polynomial functions are investigated. To simplify the problem a robot with a single rotary joint is used. This can easily be scaled up for robots with several rotary joints liked the Niryo One.

Firstly, trajectories without obstacles are considered.

As the robot has to be at zero velocity at the start and beginning a cubic polynomial is needed to fulfil all requirements of the trajectory. The joint angle θ and velocity $\dot{\theta}$ can be described as a function of time with the coefficients c_0, c_1, c_2, c_3.

$$\theta(t) = c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

$$\dot{\theta}(t) = 3c_3 t^2 + 2c_2 t + c_1$$

Furthermore, there are four boundary conditions, where the joint angles θ_0, θ_f and the end time t_f are known values. These conditions can be derived from the requirements to the movement that were described earlier.

$$\theta(0) = \theta_0$$

$$\dot{\theta}(0) = 0$$

$$\theta(t_f) = \theta_f$$

$$\dot{\theta}(t_f) = 0$$

With that a system of equations can be put into a matrix to deduce the coefficients. The inversion of M and the results for the coefficients are not shown here as these steps can be performed by a calculation program such as MATLAB.

$$\underbrace{\begin{bmatrix} \theta_0 \\ 0 \\ \theta_f \\ 0 \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ t_f^3 & t_f^2 & t_f & 1 \\ 3t_f^2 & 2t_f & 1 & 0 \end{bmatrix}}_{M} \cdot \underbrace{\begin{bmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{bmatrix}}_{c}$$

$$c = M^{-1}\theta$$

When using a cubic polynomial the start and end acceleration $\ddot{\theta}$ does not equal zero, which is a problem in reality, because a sudden acceleration or deceleration is impossible or lead to mechanical damage. To avoid this problem a quintic polynomial function is required. The functions for joint angle, velocity and acceleration are the following.

$$\theta(t) = c_5 t^5 + c_4 t^4 + c_3 t^3 + c_2 t^2 + c_1 t + c_0$$

$$\dot{\theta}(t) = 5c_5 t^4 + 4c_4 t^3 + 3c_3 t^2 + 2c_2 t + c_1$$

$$\ddot{\theta}(t) = 20c_5 t^3 + 12c_4 t^2 + 6c_3 t + 2c_2$$

Two more boundary conditions can be formulated for the start and end acceleration. The other four conditions for position and velocity as for the cubic polynomial are still valid.

$$\ddot{\theta}(0) = 0$$

$$\ddot{\theta}(t_f) = 0$$

This leads to another system of equations which can be solved for the coefficients.

$$\underbrace{\begin{bmatrix}\theta_0\\0\\0\\\theta_f\\0\\0\end{bmatrix}}_{\theta}=\underbrace{\begin{bmatrix}0&0&0&0&0&1\\0&0&0&0&1&0\\0&0&0&2&0&0\\t_f^5&t_f^4&t_f^3&t_f^2&t_f&1\\5t_f^4&4t_f^3&3t_f^2&2t_f&1&0\\20t_f^3&12t_f^2&6t_f&2&0&0\end{bmatrix}}_{M}\cdot\underbrace{\begin{bmatrix}c_5\\c_4\\c_3\\c_2\\c_1\\c_0\end{bmatrix}}_{c}$$

$$c=M^{-1}\theta$$

Next, trajectories with a via point are investigated. When there are obstacles in the workspace of a robot, it is often required to define via points for the trajectory in order to avoid collisions. Assume that the via point θ_1 is crossed at $t=t_1$ then there is one more condition.

$$\theta(t_1)=\theta_1$$

With the other four conditions from the cubic polynomial, a quartic polynomial is now required to fulfill all requirements.

$$\theta(t)=c_4t^4+c_3t^3+c_2t^2+c_1t+c_0$$

$$\dot{\theta}(t)=4c_4t^3+3c_3t^2+2c_2t+c_1$$

The coefficients can be derived from the following equation system.

$$\underbrace{\begin{bmatrix}\theta_0\\0\\\theta_f\\0\\\theta_1\end{bmatrix}}_{\theta}=\underbrace{\begin{bmatrix}0&0&0&0&1\\0&0&0&1&0\\t_f^4&t_f^3&t_f^2&t_f&1\\4t_f^3&3t_f^2&2t_f&1&0\\t_1^4&t_1^3&t_1^2&t_1&1\end{bmatrix}}_{M}\cdot\underbrace{\begin{bmatrix}c_4\\c_3\\c_2\\c_1\\c_0\end{bmatrix}}_{c}$$

$$c=M^{-1}\theta$$

The same logic applies when the boundary conditions for the angle acceleration being 0 at start and end are put into consideration as well. For that a hexic polynomial is needed.

$$\theta(t)=c_6t^6+c_5t^5+c_4t^4+c_3t^3+c_2t^2+c_1t+c_0$$

$$\dot{\theta}(t)=6c_6t^5+5c_5t^4+4c_4t^3+3c_3t^2+2c_2t+c_1$$

$$\ddot{\theta}(t)=30c_6t^4+20c_5t^3+12c_4t^2+6c_3t+2c_2$$

Solving for the coefficients as shown before.

$$\underbrace{\begin{bmatrix}\theta_0\\0\\0\\\theta_f\\0\\0\\\theta_1\end{bmatrix}}_{\theta}=\underbrace{\begin{bmatrix}0&0&0&0&0&0&1\\0&0&0&0&0&1&0\\0&0&0&0&2&0&0\\t_f^6&t_f^5&t_f^4&t_f^3&t_f^2&t_f&1\\6t_f^5&5t_f^4&4t_f^3&3t_f^2&2t_f&1&0\\30t_f^4&20t_f^3&12t_f^2&6t_f&2&0&0\\t_1^6&t_1^5&t_1^4&t_1^3&t_1^2&t_1&1\end{bmatrix}}_{M}\cdot\underbrace{\begin{bmatrix}c_6\\c_5\\c_4\\c_3\\c_2\\c_1\\c_0\end{bmatrix}}_{c}$$

$$c=M^{-1}\theta$$

Another way of connecting the via point to the trajectory is by using two separate cubic polynomials for the first and second part of the movement. It is required that the angle speed and acceleration at the via point are constant in order to ensure a smooth transition at the via point. θ_{01} is the joint angle for the first part of the movement until the via point and θ_{1f} is the joint angle for the second part from the via point to the end. The coefficients have another index that determines to which part they belong.

$$\theta_{01}(t) = c_{13}t^3 + c_{12}t^2 + c_{11}t + c_{10}$$

$$\dot{\theta}_{01}(t) = 3c_{13}t^2 + 2c_{12}t + c_{11}$$

$$\ddot{\theta}_{01}(t) = 6c_{13}t + 2c_{12}$$

$$\theta_{1f}(t) = c_{23}t^3 + c_{22}t^2 + c_{21}t + c_{20}$$

$$\dot{\theta}_{1f}(t) = 3c_{23}t^2 + 2c_{22}t + c_{21}$$

$$\ddot{\theta}_{1f}(t) = 6c_{23}t + 2c_{22}$$

The start time for the first polynomial is 0 and the end time is t_1. Assume that the start time of the second polynomial is 0 and the end time $t_{1f} = t_f - t_1$. This has to be handled carefully as the resulting coefficient are also valid for this time. When putting both together the coefficients have to be evaluated at these time intervals to get the trajectories, which are then connected to one.

The following conditions apply for this case. The last two conditions come from the continuity of angle velocity and acceleration at the via point.

$$\theta_{01}(0) = \theta_0$$

$$\theta_{01}(t_1) = \theta_1$$

$$\dot{\theta}_{01}(0) = 0$$

$$\theta_{1f}(0) = \theta_1$$

$$\theta_{1f}\left(t_{1f}\right) = \theta_f$$

$$\dot{\theta}_{1f}\left(t_{1f}\right) = 0$$

$$\dot{\theta}_{01}(t_1) = \dot{\theta}_{1f}(0) \rightarrow \dot{\theta}_{01}(t_1) - \dot{\theta}_{1f}(0) = 0$$

$$\ddot{\theta}_{01}(t_1) = \ddot{\theta}_{1f}(0) \rightarrow \ddot{\theta}_{01}(t_1) - \ddot{\theta}_{1f}(0) = 0$$

This leads to the following system of equations which can be solved for the coefficients of both cubic polynomials.

$$\underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ 0 \\ \theta_1 \\ \theta_f \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & t_{1f}^3 & t_{1f}^2 & t_{1f} & 1 \\ 0 & 0 & 0 & 0 & 3t_{1f}^2 & 2t_{1f} & 1 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & -1 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 & -2 & 0 & 0 \end{bmatrix}}_{M} \cdot \underbrace{\begin{bmatrix} c_{13} \\ c_{12} \\ c_{11} \\ c_{10} \\ c_{23} \\ c_{22} \\ c_{21} \\ c_{20} \end{bmatrix}}_{c}$$

$$c = M^{-1}\theta$$

The last method to find the trajectory is by connecting two quintic polynomials. By doing so the angle acceleration at the start and end of the movement is once again 0. The third and fifth derivation of the functions are required to find enough conditions to deduce all coefficients.

$$\theta_{01}(t) = c_{15}t^5 + c_{14}t^4 + c_{13}t^3 + c_{12}t^2 + c_{11}t + c_{10}$$

$$\dot{\theta}_{01}(t) = 5c_{15}t^4 + 4c_{14}t^3 + 3c_{13}t^2 + 2c_{12}t + c_{11}$$

$$\ddot{\theta}_{01}(t) = 20c_{15}t^3 + 12c_{14}t^2 + 6c_{13}t + 2c_{12}$$

$$\theta_{01}^{(3)}(t) = 60c_{15}t^2 + 24c_{14}t + 6c_{13}$$

$$\theta_{01}^{(4)}(t) = 120c_{15}t + 24c_{14}$$

$$\theta_{1f}(t) = c_{25}t^5 + c_{24}t^4 + c_{23}t^3 + c_{22}t^2 + c_{21}t + c_{20}$$

$$\dot{\theta}_{1f}(t) = 5c_{25}t^4 + 4c_{24}t^3 + 3c_{23}t^2 + 2c_{22}t + c_{21}$$

$$\ddot{\theta}_{1f}(t) = 20c_{25}t^3 + 12c_{24}t^2 + 6c_{23}t + 2c_{22}$$

$$\theta_{1f}^{(3)}(t) = 60c_{25}t^2 + 24c_{24}t + 6c_{23}$$

$$\theta_{1f}^{(4)}(t) = 120c_{25}t + 24c_{24}$$

There are four more conditions than in the previous method. Notice that even jerk and snap are now constant at the via point.

$$\ddot{\theta}_{01}(0) = 0$$

$$\ddot{\theta}_{1f}\left(t_{1f}\right) = 0$$

$$\theta_{01}^{(3)}(t_1) = \theta_{1f}^{(3)}(0) \rightarrow \theta_{01}^{(3)}(t_1) - \theta_{1f}^{(3)}(0) = 0$$

$$\theta_{01}^{(4)}(t_1) = \theta_{1f}^{(4)}(0) \rightarrow \theta_{01}^{(4)}(t_1) - \theta_{1f}^{(4)}(0) = 0$$

The following system of equations can then be used to find the coefficients.

$$\underbrace{\begin{bmatrix} \theta_0 \\ \theta_1 \\ 0 \\ 0 \\ \theta_1 \\ \theta_f \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\theta} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & t_{1f}^5 & t_{1f}^4 & t_{1f}^3 & t_{1f}^2 & t_{1f} & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5t_{1f}^4 & 4t_{1f}^3 & 3t_{1f}^2 & 2t_{1f} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20t_{1f}^3 & 12t_{1f}^2 & 6t_{1f} & 2 & 0 & 0 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 20t_1^3 & 12t_1^3 & 6t_1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 60t_1^2 & 24t_1 & 6 & 0 & 0 & 0 & 0 & 0 & -6 & 0 & 0 & 0 \\ 120t_1 & 24 & 0 & 0 & 0 & 0 & 0 & -24 & 0 & 0 & 0 & 0 \end{bmatrix}}_{M} \cdot \underbrace{\begin{bmatrix} c_{15} \\ c_{14} \\ c_{13} \\ c_{12} \\ c_{11} \\ c_{10} \\ c_{25} \\ c_{24} \\ c_{23} \\ c_{22} \\ c_{21} \\ c_{20} \end{bmatrix}}_{c}$$

$$c = M^{-1}\theta$$

Now there are two different solutions for a simple trajectory without via points and even four solutions for trajectories with via points.

3.2 Realization and Analysis in MATLAB

A MATLAB script *trajectory_planning* is developed to find and visualize the trajectories and their first and second derivatives. In order to deduce the coefficients for the polynomial functions, MATLAB functions are written where the previously shown equations are implemented.

The functions for the cubic and quintic trajectories *coefficients_cubic* and *coefficients_quintic* take θ_0, θ_f and t_f as inputs and return an array of the coefficients. with the function *polyval* the trajectory can be deduced and by using *polyder* the derivatives of the trajectory can be found.

This is performed with the sample parameters $\theta_0 = 15°, \theta_f = 65°$ and $t_f = 10s$. In the following figure 3.1 are the MATLAB plots for angle position, velocity and acceleration for the cubic and quintic trajectory.

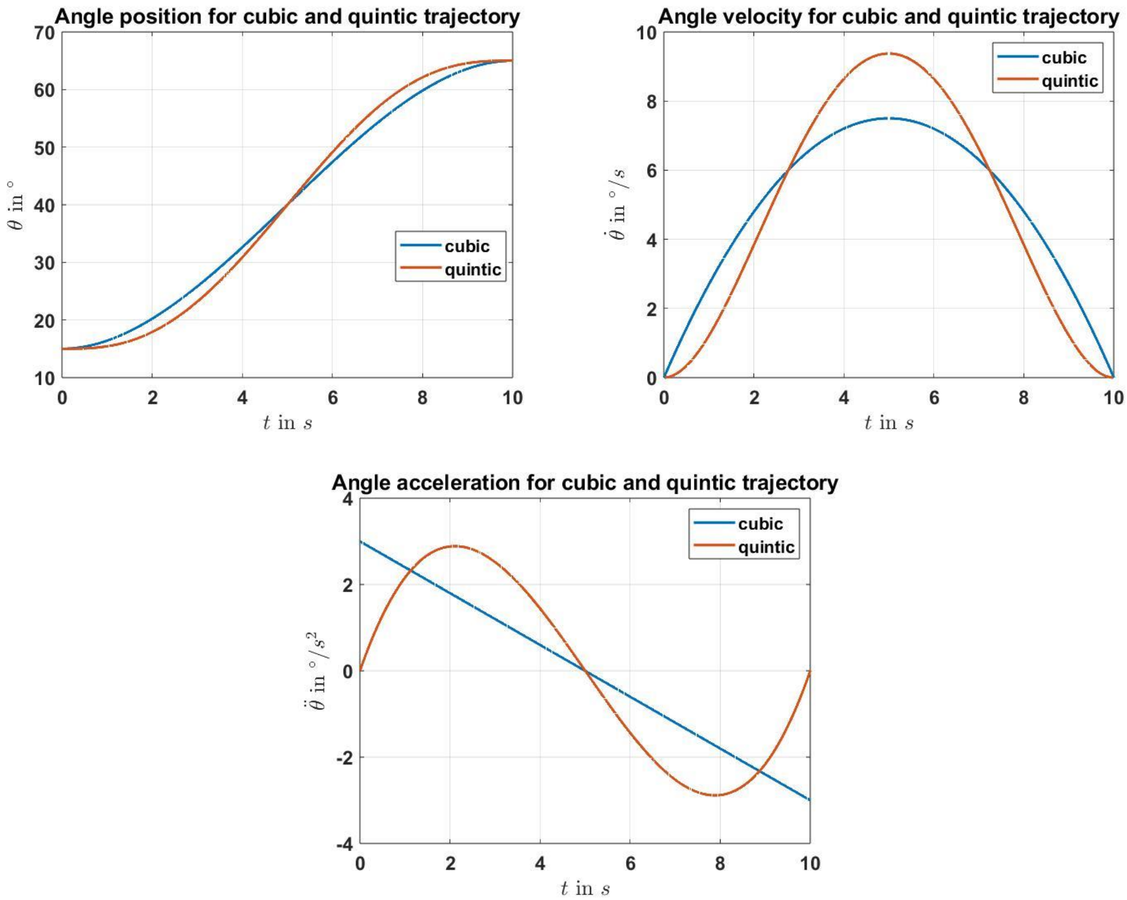


Figure 3.1: Angle Position, Speed and Acceleration

There is not much difference for the angle position. But for the angle velocity it is obvious that quintic starts and ends tangential to the x-axis and the maximum is higher than for cubic. Most notable are the differences in the angle acceleration as cubic starts with a positive value and ends with a negative value while quintic starts and ends with 0. The comparison between the two methods shows that both have advantages and disadvantages, but overall quintic is better for real applications as the angle acceleration is continuous at the start and end of the movement.

For the trajectories with via points the functions *coefficients_quartic, coefficients_hexic, coefficients_cubic_via* and *coefficients_quintic_via* take $\theta_0, \theta_1, \theta_f, t_1$ and t_f as inputs and returns the coefficients. For the first two functions there is one set of coefficients while the second has one set for the first and one for the second part of the trajectory.

The via point is at $\theta_1 = 5°$ and $t_1 = 2s$ and the other parameters stay the same. Figure 3.2 shows the results of all methods.

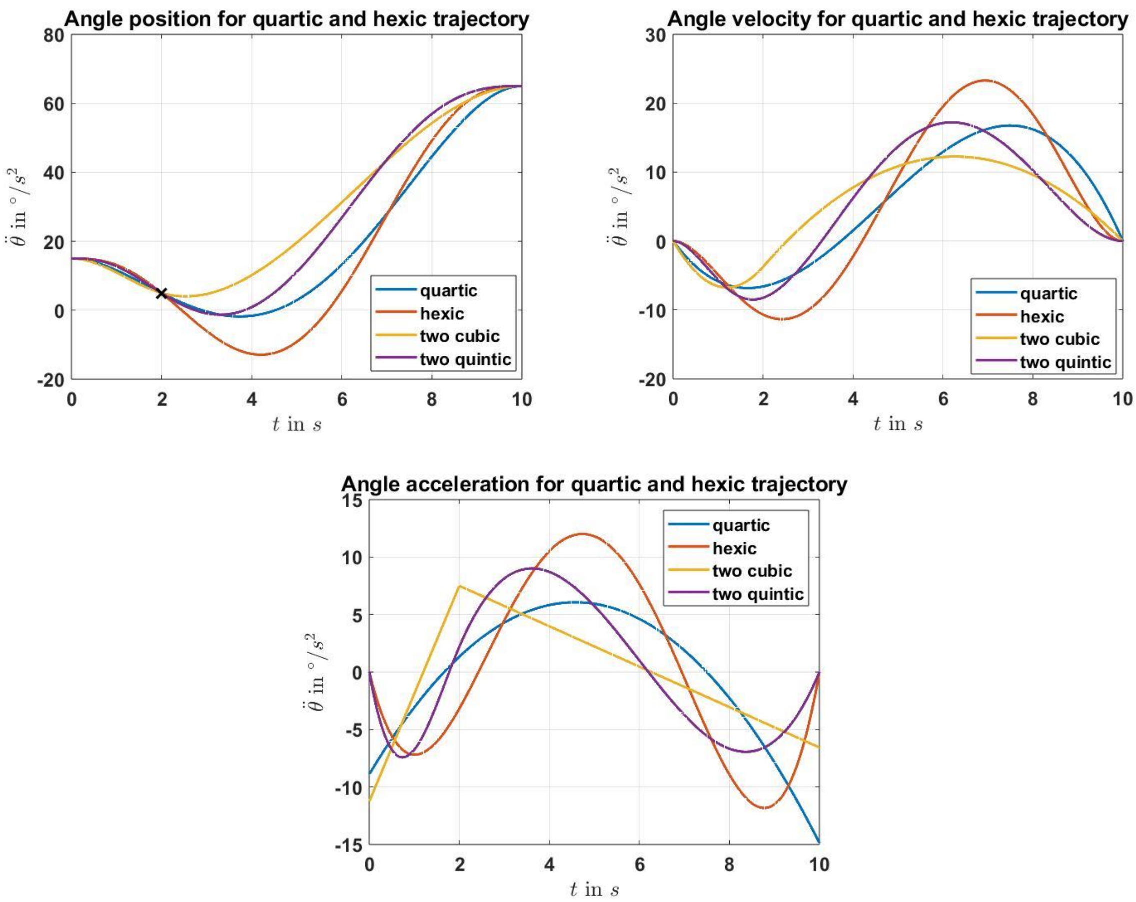


Figure 3.2: Angle Position, Speed and Acceleration with Via Point

Hexic overshoots the angle position the most after crossing the via point and also has the highest angle velocity and acceleration. Quartic and two quintic have similar overshoot of the angle position and also similar maximum angle velocity, but for the angle acceleration two quintic is better as the start and end acceleration are 0. Two cubic has the lowest overshoot and the lowest angle velocity, but there are the expected problems with the angle acceleration.

In conclusion two quintic performs best in this setup, because of its advantages in the angle acceleration compared to two cubic and quartic and the lower overshoot and angle velocity compared hexic. A smooth and continuous acceleration is most important for the trajectories of a real robot.

3.3 Application on a simplified Niryo One Model

For the application of the trajectory planning a very simplified model of the Niryo One robot is created in the MATLAB script *robot_data*. It only contains the two main joints for reaching out with the arm and is constrained to two dimensional movements only. The offset for the shoulder joint is $0.183m$,the length of the arms $r_1 = 0.21m$ and $r_2 = 0.2452m$. θ_1 describes the angle between the upper arm and the ground. θ_2 is the relative angle between the upper arm and the lower arm. The coordinate origin is at the bottom of the foot.

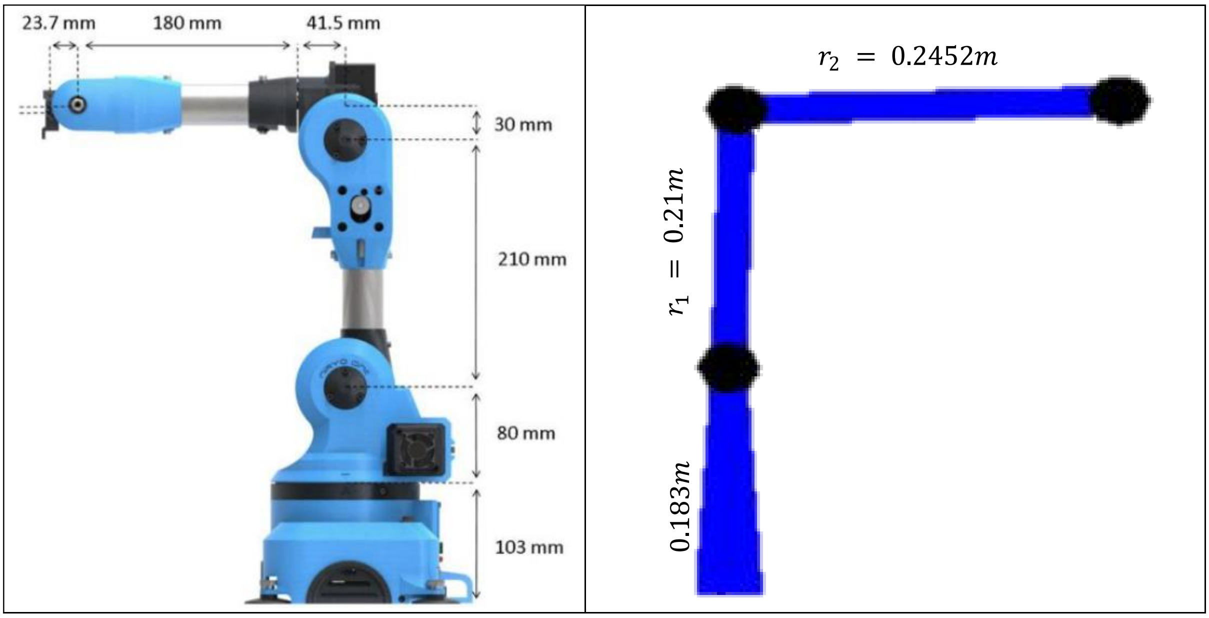


Figure 3.3: Simplified Model of Niryo One

A use case is created by adding an obstacle in front of the robot at x-position $0.2m$ and with a height of $0.1m$. The goal is to pick up an object in front of the obstacle and place it behind it. The start and end point can be defined in the coordinate system as well as a via point to avoid the obstacle and the timing of the movement. For this use case the tool center point at the beginning is $P_0(0.10m, 0.00m)$, at the end it is $P_f(0.30m, 0.00m)$ and the via point is $P_1(0.15m, 0.15m)$. Furthermore, the type of trajectory can be defined.

To find the corresponding joint coordinates an inverse kinematic function has to be applied. The function *joint_coordinates* takes the x and y coordinates, the arm lengths and the sign of θ_2, which is -1 for the robot, as inputs and returns the joint angles. Now the trajectory for the joints can be calculated as described in the previous chapter.

To visualize the respective trajectory of the tool center point a plot of it with the start, via and end point as well as the obstacle is created. With quintic polynomials it looks like shown in figure 3.4 with the previous settings.

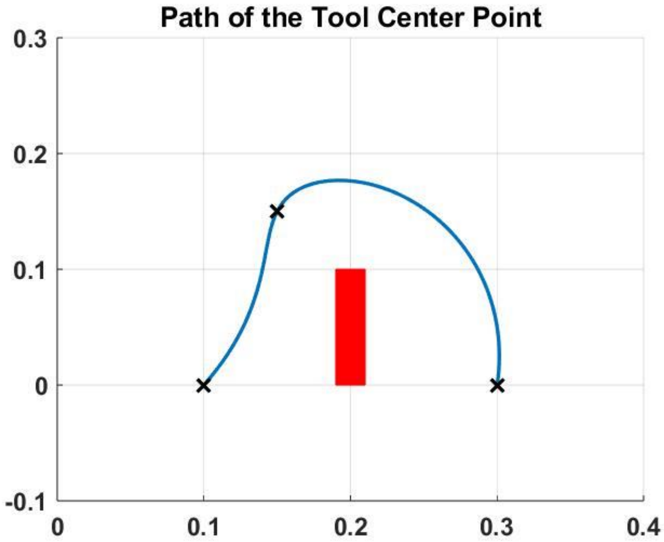


Figure 3.4: Path of Tool Center Point

As the path of the tool looks like it does not hit the obstacle the simulation for the robot can be made by executing the MATLAB Simulink file *robot*. The animation shows the movement of the whole robot for every time step. Figure 3.5 displays the end position.

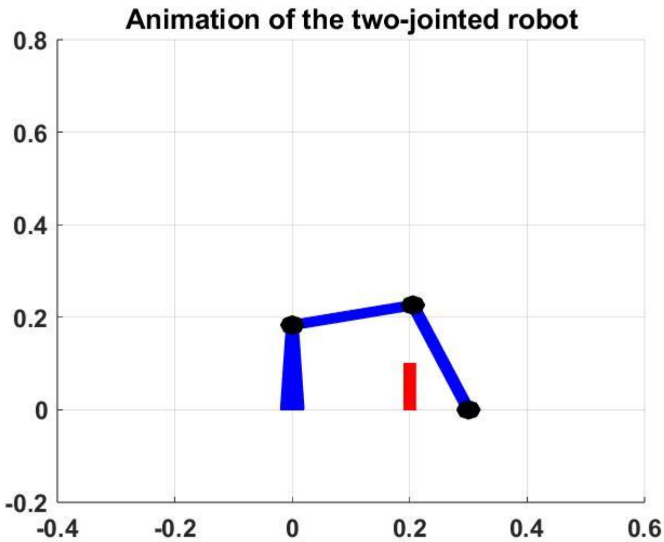


Figure 3.5: Animation of the Two-Jointed Robot

4 Hardware Demonstration

This chapter shows the physical execution of the project. An explanation of the setup is followed by an execution description and an insight into the code.

4.1 Hardware Setup

This chapter will discuss how to implement a robot into the environment described in the introduction. To show a solution how to move the boards around an obstacle with a robot, the following situation model got created in the laboratory.

A robot picks up the circuit boards, which are simulated by UTHM student cards, as they have a similar thickness. The robot moves the cards over the cardboard box, which reflects the air cleaning device. The cards get laid down on the right side of the box. A detail explanation of the set up follows later in this chapter.

The project is conducted with the Niryo One. It is a six-axis robot. This gives the robot enough degrees of freedom to move around the obstacle.

Picking up the cards is an issue. The standard claw is not capable of clamping a card. By not being accurate enough, the gripper could easily miss the card and do the lifting without attached object. It is also wished to just take the top card. The gripper has the risk of taking several cards with one take. Therefore, it was decided to use a vacuum gripper.

The vacuum gripper and its working principal is shown in detail in the following figure.

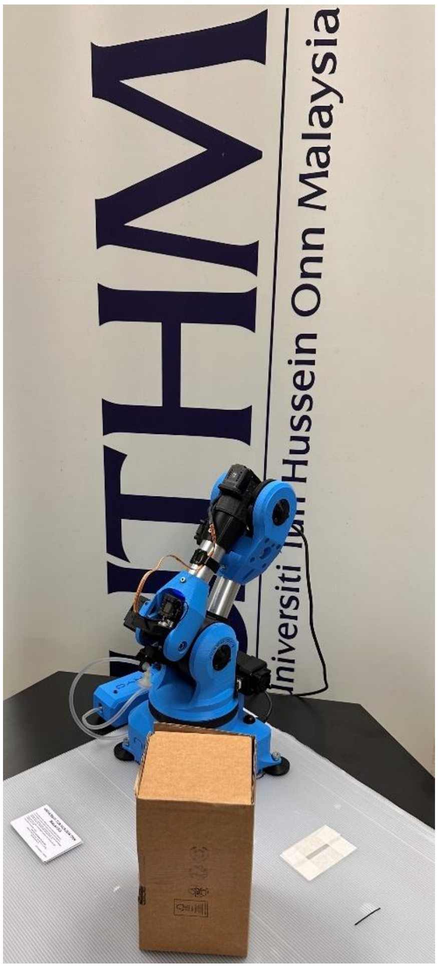


Figure 3.1: Overview Lab

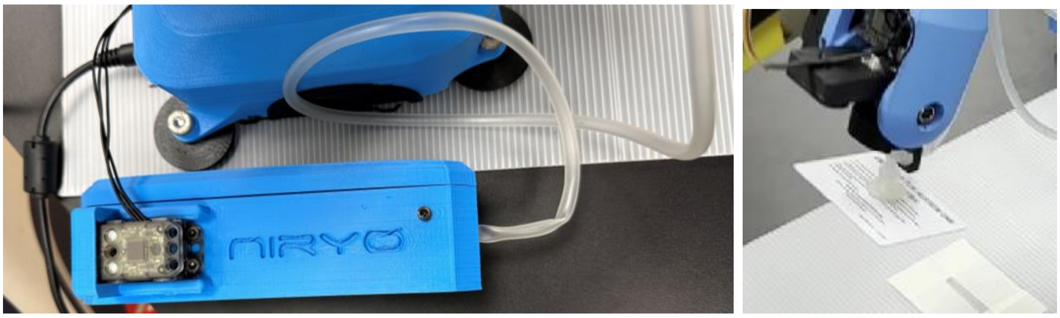


Figure 3.2: Vacuum gripper

The vacuum gripper consists of two parts. The pump, that is show on the left side of the figure. It generates the low pressure. The low pressure is then transferred to the gripper, shown on the right side. To attach an object, the robot arm needs to position the vacuum cup abut to the object before the vacuum gets generated.

The exact model assembly is shown in the figure below:

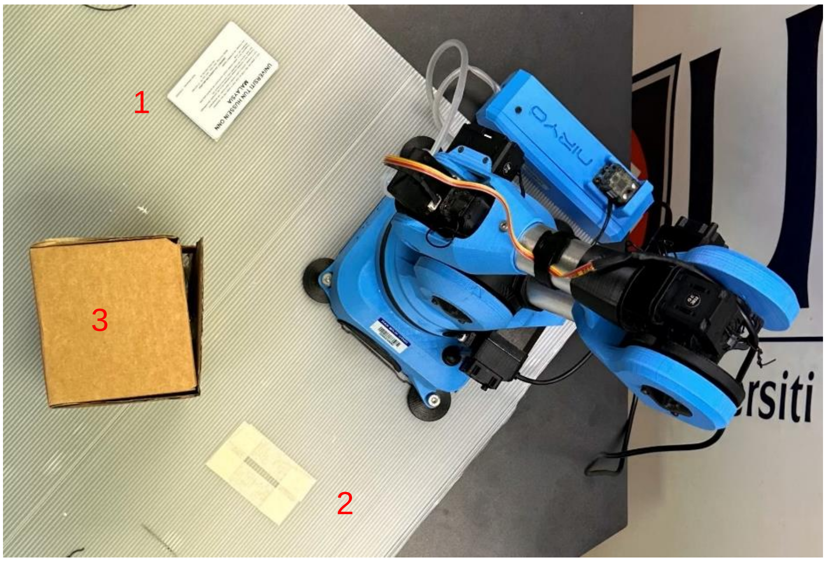


Figure 3.3: Topview assembly

Marked with number one is the area, where the first machine puts the finished cards. It is the pick-up location for the robot. A sample of cards already are available for pick-up. The drop-off area for the robot is marked by the white rectangle numbered with the two. It is the start position for the second machine. Marked by number three is the obstacle box, that represents the air cleaner.

4.2 Execution

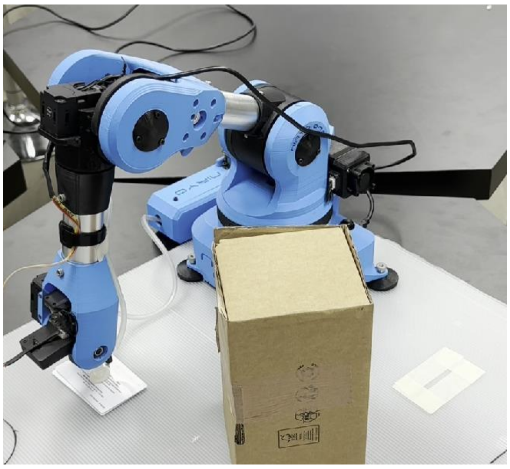
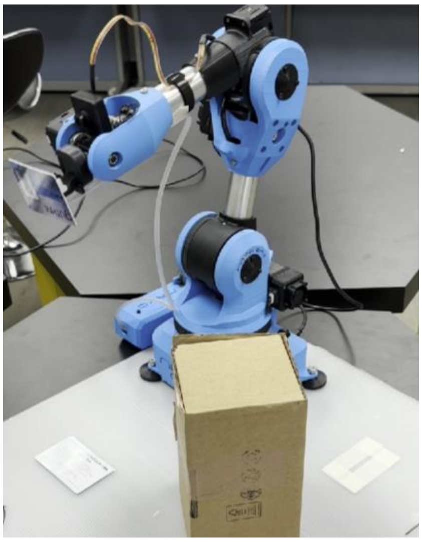
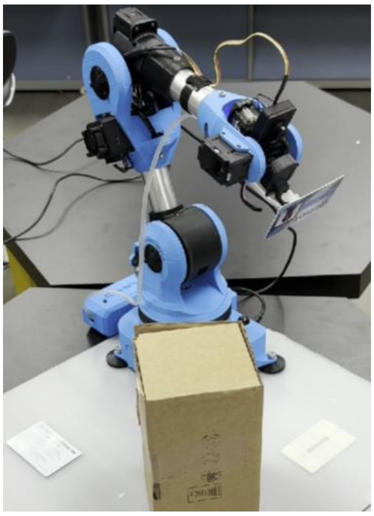
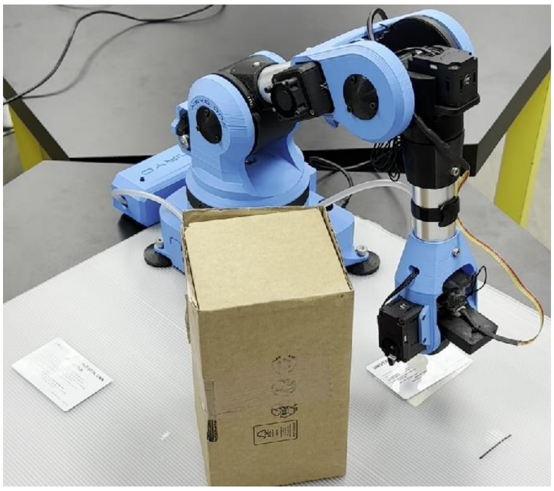
	
Pick-up	Before obstacle
	
After obstacle	Drop-off

Figure 3.4: Execution sequence

After starting from its origin position the robot position itself into the pick-up location like shown in the above figure. The pump of the vacuum gripper than gets activated. The card is now attached to the vacuum cup. The robot starts the lifting. It takes a path around the obstacle. Figure 3.4 shows a position before and after the obstacle. This shows that after the card got gripped the robot needs to lift it over the obstacle. In the end the object gets dropped off int the wished destination.

For picking up the next card, the robot has to take the same route to return to the first machine.

4.3 Programming

The programming is done with Niryo One Studio desktop application. This is a block programmable tool. The user has to create a set of coordinates. Each coordinate is a block. The coordinate blocks get attached in order of the path. The orange blocks activate and deactivate the vacuum pump. All steps are within a repeat loop. The robot redoes the steps as often as wished. As soon as the loop is ended the robot is send to the start position.

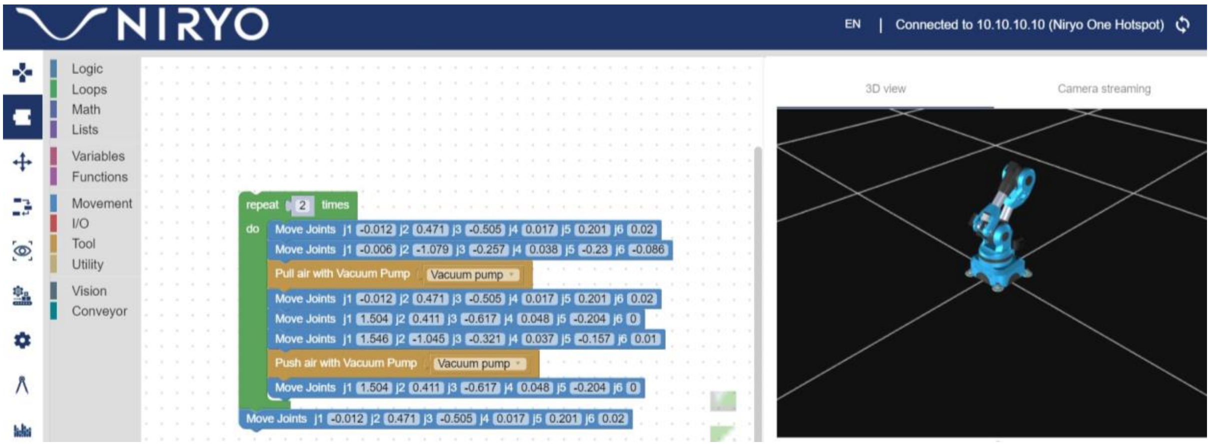


Figure 3.5: Niryo One Studio

The coordinates are determined by pushing the robot arm into the destined positions and safe the coordinates. To safe a position, one has to press the button on the side of the robot. The software tool than automatically creates a block. This block can be used like shown above. It can be duplicated to be used several times.

It is possible to limit the moving speed of the joints. The achieve the fastest possible production, the arms are running at a speed of 100%.

5 Evaluation and Conclusion

The project had two objectives. The first goal was to provide a detailed description of the mathematical background of the robot trajectory planning with an obstacle. Chapter 3 is covering this objective. The chapter is showing the mathematical derivation of the polynomials needed to plan the trajectory. It is even comparing different approaches of calculation and they get compared. In summary, it can be said that the first project objective has been fulfilled.

The second objective was to provide a hardware demonstration of avoiding an obstacle with the Niryo One robot. This goal was also reached by establishing a work environment in the laboratory, programming the robot and observing the execution.

For the future it would be great to not just have a hardware execution under laboratory conditions. This means to build a real size robot that can perform in the production environment that was analysed in the problem statement. Which means to have a transport robot that can connect two factoring steps while having to avoid an obstacle.

6 References

[1] A. Nayyar & A. Kumar, "A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development", Springer Cham, 2019

[2] K. H. Grote & others, "Dubbel", 25th edition, Springer Vieweg Berlin, 2018

[3] Niryo, "Niryo One documentation", *nyrio.com*, Feb. 13, 2018. Available: https://niryo.com/docs/niryo-one/ [Accessed: Jun. 07, 2023]