

BSc Advanced Technology
Bachelor Assignment

Implementation of Inverse Dynamic Controller into Parallel SCARA Robot

Brunon Kazimierz Bojków

Supervisors:
Chair: Dr.Ir. R.G.K.M. Aarts
Daily Supervisor: Ir. Alex Budding
External member: Dr.Ir. A.Q.L. Keemink

July, 2024

Applied Mechanics & Data Analysis
University of Twente

maxon Benelux

Abstract

This report describes the assignment of implementing an inverse dynamic controller into a parallel, dual-arm SCARA robot made to demonstrate the pick-and-place operation. Creation of the dynamical model, inverse dynamic solver and state estimator of the passive angle position and subsequent derivatives was described.

To evaluate this controller, its performance was measured and compared to the previously used Feed-Forward Proportional-Integral-Derivative controller that was part of the inbuilt software - ApposC - of the controller hardware - MiniMac6. The results of the improvements were 50% path execution time reduction, displacement error improvement, settling time reduction below 35 ms and setpoint precision of 0.035° , which all contribute to fulfilling the majority of the requirements set for the controller.

At last, further improvements in the form of frequency domain analysis, software changes, the addition of rotary encoders for passive angles and the addition of external current amplifiers have been suggested.

Acknowledgements

I would like to thank maxon Benelux for giving me the opportunity to do this internship. I am grateful to Roland Aarts for accepting me as a thesis candidate and for all the guidance provided.

Special thanks to Roel Munster and Alex Budding for the daily guidance and the help you provided during the work on this thesis.

And thank you, Mum and Dad, for allowing me to study and inspiring me. Without you, I would not have made it here.

List of Abbreviations

BLDC Brushless Direct Current. 10

DOF Degree of Freedom. 1–3

FF Feed-Forward. 4, 6, 20

FF-PID Feed-Forward Proportional Integral Derivative. 4, 12, 14–17, 19, 20, 25

IDC Inverse Dynamic Controller. 5, 6, 14–22

PID Proportional Integral Derivative. 4–6, 12

List of Symbols

l Length of individual arm link

l_{motor} Distance between SCARA axis motors

θ Angle position of the SCARA joint

$\dot{\theta}$ Velocity of an angles of the SCARA joint

$\ddot{\theta}$ Acceleration of an angle of the SCARA joint

m Mass

I Inertia

τ Torque

\mathcal{M} Mass matrix in the motion equation context

\mathcal{C} Coriolis and centripetal matrix in the motion equation

\mathbf{M} Mass matrix in the dynamic solver context

\mathbf{C} Coriolis and centripetal matrix in the dynamic solver context

K_t Motor torque constans

\mathcal{I} Resulting motor current

X_e X coordinate of the end effector in cartesian coordinates

Y_e Y coordinate of the end effector in cartesian coordinates

Contents

Abstract	i
Acknowledgements	ii
List of Abbreviations	iii
List of Symbols	iii
Contents	iv
1 Introduction	1
1.1 Industrial robots and SCARA	1
1.2 Issue of control	2
1.3 Goals of the assignment	4
2 Theory	5
2.1 Literature research	5
2.2 Inverse dynamics controller	6
2.3 State observer	6
2.3.1 Inverse kinematics for passive angles	7
2.3.2 Position derivative estimator	7
2.4 Inverse dynamic model	8
2.4.1 Inverse dynamic solver	8
3 Methodology	10
3.1 Robot setup	10
3.2 Measurement setup	11
3.2.1 High speed measurements	12
3.3 Controller parameter tuning	12
4 Results	14
4.0.1 Position tracking performance	14
4.1 Comparison between the FF-PID and IDC	15
4.1.1 IDC at increased speed	15
4.2 Performance between runs	16
4.2.1 Deviation between runs	17
4.3 Measurements of motor currents	19
4.4 Performance in the frequency domain	20
5 Discussion	21
5.1 Performance comparison	21
5.2 Faster movement	21
5.3 Importance of feed-forward	21
5.4 Performance bottleneck	21
6 Conclusion	22
6.1 Argument on goal confirmation	22
6.2 Recommendation	22

A Appendix: Block diagram of the FF-PID controller	25
B Appendix: Bode plots	25
B.1 Frequency domain	25

1 Introduction

This report is about a bachelor's graduation assignment done in collaboration with maxon Benelux. The assignment was centred on a parallel SCARA robot used as a pick-and-place demonstration. The robot was designed and built by Roel Munster to demonstrate the capabilities of the company's motors and controller hardware. Since the robot is meant to appear at trade fairs, it was designed to be movable, fit on half of a standard delivery pallet and operate on a standard power connection available in Europe and North America.



Figure 1: Double SCARA robot built by maxon Benelux

1.1 Industrial robots and SCARA

Industrial robots have been a crucial part of manufacturing for the last 50 years. There are different types of robots, each built differently to serve its purpose. Some robots are built to move heavy objects, others are made to move small parts, but they all need to work repeatedly and fast. The main purpose of industrial robots is to substitute or assist humans in performing tedious, repetitive, and often dangerous work. Assembly robots are a sub-category within industrial robots, and their primary purpose is to automate the process of putting objects together in a desired manner. Their most common use case is a pick-and-place machine that picks up elements and places them in the designated spot.

Selective Compliance Assembly Robot Arm - SCARA is an example of an industrial assembly robot. The SCARA robot was developed in Japan in the '80s by Hiroshi Makino [1] [2]. It was designed to remove the possibility of assembly components being inserted into the assembly at an angle, thus locking it in an incorrect position. This assembly error has been illustrated in Figure 3. SCARA has a set of links that fold on themselves, forming an arm. That allows for easy movement in the horizontal plane, blocking the possibility of any rotation, resulting in two Degree of Freedom (DOF). This principle is called selective compliance, which is part of the robot's name.

Movement in the vertical axis is performed by a separate mechanism mounted at the end of the arm, called an end-effector. The end-effector is designed to move precisely and interact with the working environment, for example, grabbing a screw.

The first SCARA robot was of a serial type, meaning the linkages were positioned one after another. Another type of SCARA robot that was developed later is a parallel type, meaning it has a pair of linkages positioned in parallel and connected at the end. This type of SCARA robot is the subject of this assignment. A comparison of the two different types can be seen in Figure 2.

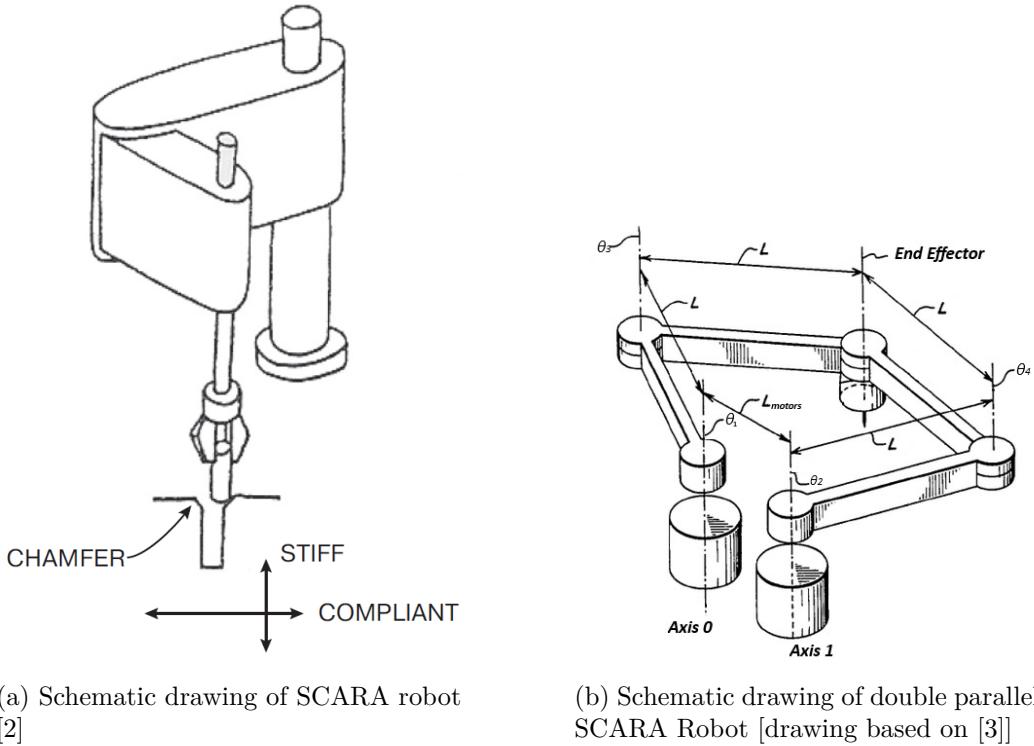


Figure 2: Types of SCARA robot

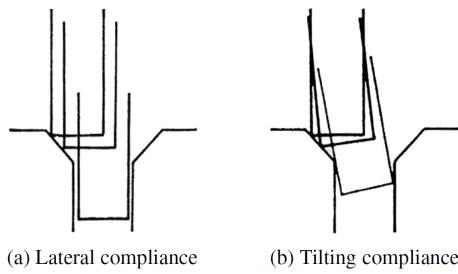


Figure 3: Effects of selective compliance [2]

1.2 Issue of control

The layout of the linkages in parallel configuration causes a control problem named singularities to occur [4]. In general terms, singularity is a joint-arm configuration where the robot loses one or more of its DOF. Parallel SCARA robot exhibits two types of singularities:

- Type 1 - Serial singularity - position configuration where the end-effector loses one of its DOFs. This happens at the boundary of the working space - one or both arms are fully folded, or one or both arms are fully extended.
- Type 2 - Parallel singularity - position configuration where the motors can not resist the forces applied to the end-effector. This happens inside the working space - the distal links are in line with each other.

While working, parallel SCARA robots can be set into one of the four working modes, which means that the joints and linkages of the robot can be arranged in four different ways, and most of the positions can be achieved by more than one configuration. Subsequently, each working mode has its individual singularities [5]. All four of the working modes and their singularities have been illustrated in Figure 4.

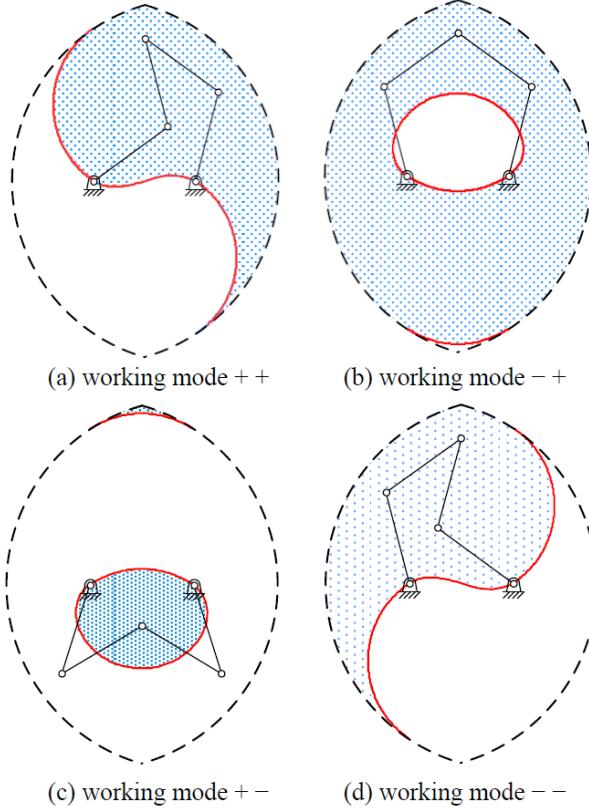


Figure 4: Four working modes with their respective singularity-free zones (marked with dots), with type 2 singularities occurring at the continuous red lines and type 1 singularities occurring at the dashed line. Mechanical interference had been disregarded [4].

Another difficulty in controlling any arm manipulator is the effective change in inertia experienced by the driving motors. Due to the change in effective arm length, the position of the center of mass of each arm element moves, varying the torque needed to make a movement.

Robot arm manipulators are moving structures, meaning they lie in the subject of robot dynamics, and associated motion dynamic motion equations are needed to describe them. As they are multi-link mechanisms, those dynamics are more complex than just a change of position due to applied force, as Coriolis, gravity, and friction terms play a

key role in the movement. All of those terms change depending on each joint's individual positions and velocities and must be accounted for when demanding a controlled movement [6].

1.3 Goals of the assignment

Since the main purpose of this robot is to demonstrate pick-and-place operations, it needs to move swiftly and accurately and withstand the run time of a trade fair with minimal maintenance. Previously, to control this SCARA robot, a simple Proportional Integral Derivative (PID) with a simple Feed-Forward (FF) of acceleration term had been utilised. That controller did not perform as intended, showed underdamped oscillatory behaviour, had jerky movement and did not move the arms at desired velocities. To top that, it also had trouble controlling the arms in their whole range of motion - at the further borders of the working area, it became inaccurate, and oscillations around the set points became even more present.

Therefore, a new controller had to be created to get better out of the SCARA robot. This resulted in the following specifications: With fast and accurate movement, no oscillation during movement or at a standstill, and the controller has to work repetitively. The requirements mentioned above have been rephrased in the following quantified terms:

- steady state position error below 1°
- tracking error while moving below 5°
- decreases settling time below 100 ms
- reduces the path execution time by 50 %
- Increase the repeatability by 100 %

The previously used - Feed-Forward Proportional Integral Derivative (FF-PID) controller, with its designated movement path, shown in 8, was used as a baseline for comparison and performance quantification.

To summarise, the main purpose of this assignment was to create a robust and dynamic control scheme that would allow for fast and impressive movement.

2 Theory

This chapter will present the literature study and introduce the required theoretical background needed to understand the inner workings of the new controller. It will start with a description of the controller and be followed by an explanation of every element necessary to implement the controller into the robot.

2.1 Literature research

The problem of singularities in parallel SCARA robots has been known and studied. The biggest problem faced is the presence of type 2 singularities, as they create an inaccessible zone within the working area [7].

The most obvious solution to avoid this type of singularity is to forbid the controller from accessing the area around the singularities, thus reducing the work area. This is the easiest method, yet it reduces the robot's functionality.

Another seemingly simple solution to the singularity problem is the addition of an actuator [8] to counteract the parallel singularities, but this increases the costs and complexity of the robot and can be hard to execute with a robot that is already assembled or is a premade product.

Because the use case of this robot is relatively predictable, limiting to one working mode had been chosen, as for demonstration purposes, the robot will pick and place objects within one working zone. Also, addressing the singularities is not the scope of this assignment because the intended path of the demonstration does cross the singularity in the middle, but that is done by a pre-programmed sequence, where the motors are controlled and commanded individually. It has been programmed before the start of this assignment. Thus, this will be omitted further, and attention will be given to the other control matter - dynamic changes in the system.

Controlling a parallel SCARA robot can be achieved using a control scheme named model predictive control [9]. This control scheme has the big benefit of choosing complicated constraints of the system and considering complex dynamics, but it requires vast computational power, which is not offered by the used hardware.

Different approaches have been proposed to control the parallel arms with complex dynamics, one of them being an adaptive PID controller that uses neural networks [10]. It has shown better performance when compared to standard PID, but the complexity of neural networks is too big for the hardware chosen for this SCARA.

The book Modern Robotics[6] provides insight into the kinematics, dynamics and control strategies of robot arms and has been used as a guideline. More precisely, Inverse Dynamic Controller (IDC) scheme, presented in chapter 11, has been chosen and implemented. This approach was chosen as it seemed to yield sufficient results and would be possible to incorporate into the controller architecture available within the chosen hardware. This architecture has been described in appendix B.1.

2.2 Inverse dynamics controller

The key component in this assignment is the IDC. It has been created based on the methodology shown in [6].

The key working principle behind the controller is as follows: a torque required to change the position is being calculated by the solver using commanded acceleration, current position, current velocities, mass and centripetal-Coriolis matrixes. The resulting torque is converted into the required current using motor torque constant (see eq.8). This current is then summed with the result of the position PID controller and forwarded into the amplifier. A diagram of the controller has been illustrated in the figure 5. The FF element of the controller acts before the position error occurs - increasing the response time of the system. It incorporates elements responsible for centripetal torques and the Coriolis effect. Those are notable elements of the dynamics of the system, yet the main contribution is from the commanded acceleration. This signal is multiplied by a mass matrix that represents the inertia of the system and includes the geometry of the robot (see Eq.5). A more detailed description of the dynamic model can be found in the corresponding section 2.4.

The PID element of the controller responds to the error in position - the difference between commanded and actual position. This is a key element of the controller as it has the ability to adjust the position, respond to disturbances and correct for steady-state error. This term accounts for the imperfect modelling of the system and environment.

Those two elements work in combination to achieve a fast-responding system that is able to reject disturbances and counteract inaccuracies in the model.

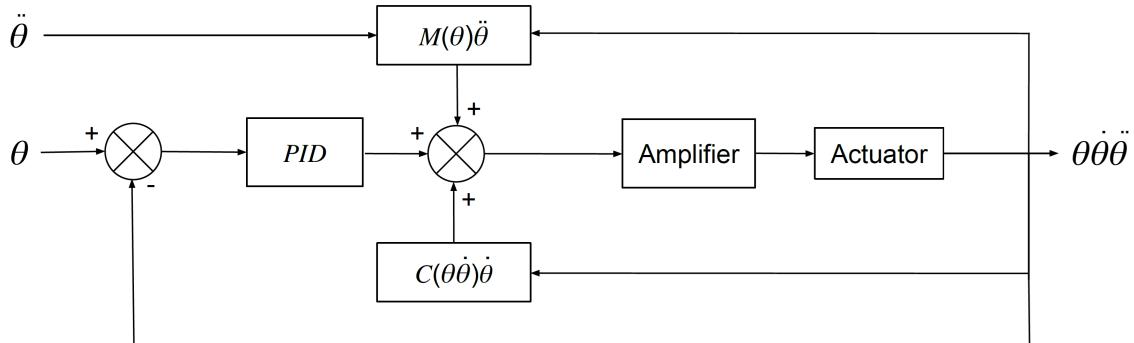


Figure 5: Block diagram of the controller IDC

2.3 State observer

Information about position, velocities and accelerations is crucial to the modelling of dynamic behaviour. The utilised controller provides only information on the angular position, velocity, and desired acceleration of the active joints, as well as the cartesian coordinates of the end effector. The following subsections are dedicated to the functions that had to be added to generate information about the states of the passive angles - θ_3 and θ_4 shown in 6.

2.3.1 Inverse kinematics for passive angles

Forward and inverse kinematics between the arm angles and the end effector have been described [11] and are a premade feature in controllers software[12]. Since the values of passive angles θ_3 and θ_4 are important information required for further calculations, and there was no information about them, a solver had to be created and implemented. Equations for passive angles had been derived using trigonometry and were based on figure 6. Equation 1 has been programmed into the controller. X_e and Y_e correspond to the position of the end effector in cartesian coordinates, where the center of the left motor is the center of the coordinate system. l and l_{motor} correspond to the length of the individual linkage and distance between motors, respectively, and their values have been described in table 1.

$$\begin{aligned}\theta_3 &= \arctan\left(\frac{Y_e - \sin\theta_1 \cdot l}{X_e - \cos\theta_1 \cdot l}\right) \\ \theta_4 &= 180^\circ - \arctan\left(\frac{Y_e - \sin\theta_2 \cdot l}{\cos\theta_2 \cdot l - X_e + l_{motors}}\right)\end{aligned}\quad (1)$$

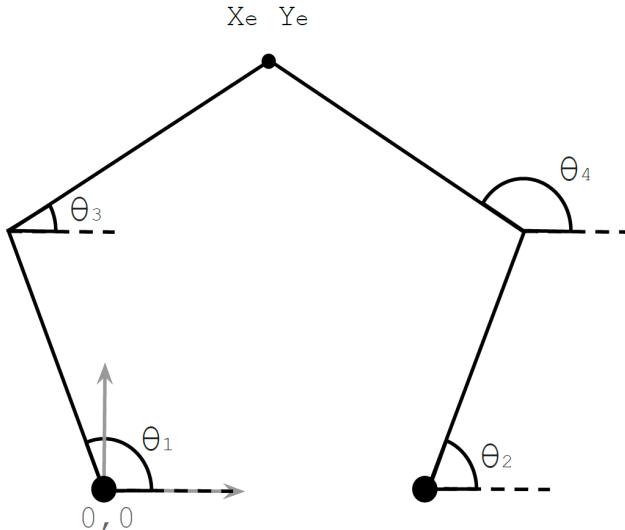


Figure 6: Kinematics of the active angles θ_{1-2} and passive angles θ_{3-4}

2.3.2 Position derivative estimator

To perform the dynamic calculation, information on velocities and accelerations is required. Therefore, a derivative estimator was created. It follows the Newton quotient formula 2.

$$\dot{\mathcal{G}}(t) = \frac{\mathcal{G}(t + \Delta t) - \mathcal{G}(t)}{\Delta t} \quad (2)$$

Where \mathcal{G} is a chosen variable, and Δt is a time step. In this case, \mathcal{G} was either position or velocity, and the time step was 5ms.

A moving average filter has been implemented to grant smoother values. It follows the formula 3.

$$avg_n = \frac{value_n + avg_{n-1} + value_{n-1}}{3} \quad (3)$$

To prevent numerical calculation errors when using the filter, initial values were set to nearly zero values ($0.0001 [{}^{\circ}s^{-1}$ or ${}^{\circ}s^{-2}$ depending if it was velocity or acceleration]). The filter was used to lessen sudden changes in acceleration that would cause spikes in commanded current. The drawback of this technique is that the velocities and accelerations have a lagging component.

All of the state estimator calculations are performed in real time. Since they require information on previous steps and take time to compute, all of the results have a delay. This poses a problem and causes inaccuracies. A possible improvement will be suggested at the end of the report.

2.4 Inverse dynamic model

The equation of motion in relation to robot dynamics is:

$$\tau = \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta})\dot{\theta} \quad (4)$$

where τ is the vector of joint torques, θ is the vector of the joint variables and its derivatives, \mathcal{M} is the mass matrix, and \mathcal{C} are combined torques resulting from Coriolis and centripetal effects. Gravity and viscous friction were omitted because their influence on the system is neglectable [6].

2.4.1 Inverse dynamic solver

Since there are only two active joints in this SCARA robot, only two of them can be controlled. Therefore, only two torques can be applied to those joints, and those need to be computed by the solver. That influences the dimensions of \mathcal{M} and \mathcal{C} matrixes, making them the following:

$$\mathcal{M} = \begin{bmatrix} m_{11} & 0 & m_{13} & 0 \\ 0 & m_{22} & 0 & m_{24} \end{bmatrix} \quad \mathcal{C} = \begin{bmatrix} c_{11} & 0 & c_{13} & 0 \\ 0 & c_{22} & 0 & c_{24} \end{bmatrix} \quad (5)$$

where:

$$\begin{aligned} m_{11} &= \frac{1}{4}m_1l^2 + m_3l^2 + I_1 & m_{13} &= \frac{1}{2}m_3l^2\cos(\theta_1 - \theta_3) \\ m_{22} &= \frac{1}{4}m_2l^2 + m_4l^2 + I_2 & m_{24} &= \frac{1}{2}m_4l^2\cos(\theta_2 - \theta_4) \\ c_{11} &= \frac{1}{4}m_3l^2S_{13}\dot{\theta}_3 & c_{13} &= -\frac{1}{4}m_3l^2S_{13}\dot{\theta}_1 + \frac{1}{2}m_3l^2S_{13}\dot{\theta}_3 \\ c_{22} &= \frac{1}{4}m_4l^2S_{24}\dot{\theta}_4 & c_{24} &= -\frac{1}{4}m_4l^2S_{24}\dot{\theta}_2 + \frac{1}{2}m_4l^2S_{24}\dot{\theta}_4 \\ S_{13} &= \sin(\theta_1 - \theta_3) & S_{24} &= \sin(\theta_2 - \theta_4) \end{aligned} \quad (6)$$

where m_{1-4} correspond to the mass of the arm, I_{1-2} correspond to the inertia of the arms, and l corresponds to the length of the arm. All of the arms are the same length, which simplifies the calculations found in [10]. Those values will be specified in the section 3.1.

Those matrixes are combined to solve for the torques:

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \mathcal{M} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix} + \mathcal{C} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad (7)$$

Said torques are further converted into currents \mathcal{I} values via the motor torque constant T_k , as shown in equation 8, and forwarded to the amplifier, as shown in diagram 5.

$$\mathcal{I} = \frac{\tau}{T_k} \quad (8)$$

3 Methodology

In this chapter, a description of the SCARA robot and key specifications will be given. Later, the measurement procedure was described, followed by the description of the controller parameter tuning.

3.1 Robot setup



Figure 7: Exploded view of the Dual SCARA

Figure 7 shows the exploded view of the SCARA robot. This robot comprises two Brushless Direct Current (BLDC) electric motors (EC frameless DT 85 M)[13] controlled by a MiniMACS6-AMP-4/50/10 controller[14] with built-in BLDC motor drivers. The controller operates on 24 V and has a maximum deliverable current of 22 A. Its two arms are each composed of two links connected with bearings and pins. Each arm is connected to a motor, forming an axis. There are two axes, numbers 0 and 1. The end effector is positioned at the common connection point of the two distal links.

There are a few key metrics of the robot that were necessary to determine in order to create the dynamic model. Those are the length of the links, the distance between the links, the mass of each link, and the inertia of the two proximal links. Since the arms combine multiple parts, it was necessary to define certain groups and combine their masses and their inertias. The mass of link 1 combines the mass of the aluminium body of the link, the screw that connects it to the shaft, one of the bearings at the joint, with link 3 and half of the mass of the metal shaft at that joint. Link 3 combines the other half of the mass of the shaft, the mass of the link itself, the masses of two bearings at each of the ends, half of the mass of the shaft connecting the distal links, half of the mass at that joint, and the mass of the screw at the joint with link 1. Masses of links 2 and 4 were calculated analogously. Inertia calculations also required establishing groups. This was done for the same objects as for the masses, but only for links 1 and 3, with an addition of the inertia of the motor shaft, which was identical in both cases. The vertical axis going through the centre of mass was used as the inertia axes. CAD software was used to

perform the calculations. Table 1 presents the results of the said calculations. The last very important parameter used in the controller is the motor torque constant K_t , which is equal to 139 mNmA^{-1} for each of the motors according to their specification [13].

Mass of link group 1	m_1	0.267	kg
Mass of link group 2	m_2	0.322	kg
Mass of link group 3	m_3	0.155	kg
Mass of link group 4	m_4	0.165	kg
Inertia of group 1	I_1	2.6×10^{-3}	kgm^2
Inertia of group 2	I_2	2.2×10^{-3}	kgm^2
Arm length	l	0.2	m
Distance between motors	l_{motors}	0.245	m

Table 1: Link parameters

3.2 Measurement setup

A series of measurements was performed to determine the performance of the new control scheme. To enable the comparison, the new control scheme was first run on the same paths with the same velocities as the previous controller. Secondly, to assist the maximum possibilities of the new system, the controller was run on the same paths but with increased velocities, acceleration, and jerks. Measurements were recorded using the built-in oscilloscope featured in ApposC software. To ensure the repeatability of the performance, each of the measurements was repeated 5 times and individual measurements were averaged. The robot's test sequence was programmed to be a rectangle with diagonal edges connected. The pattern, its dimensions, and the direction of the movement are illustrated in the figure8.

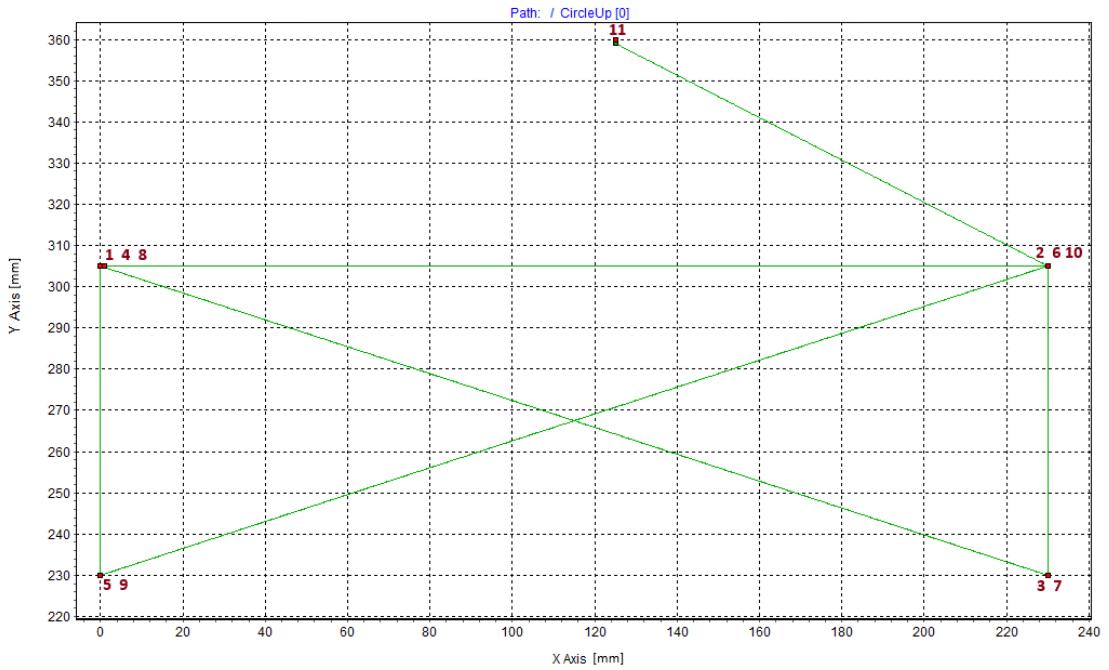


Figure 8: Test path used during the testing. Numbers next to points correspond to the order of end effector movements. X and Y axes correspond to the cartesian position in the working mode, where point (0;0) is the centre of motor/axis 0.

3.2.1 High speed measurements

To allow the inverse dynamic controller to command higher accelerations and increase velocities, the software limitation had to be increased. There are three key parameters that influence the path execution time: maximum velocity, maximum acceleration, and maximum jerk. Said parameters were increased, as shown in table 2.

	Previous value	Increased value	Unit
Velocity	3000	20000	°s^{-1}
Acceleration	6000	50000	°s^{-2}
Jerk	70000	500000	°s^{-3}

Table 2: Increases in the maximum allowable values

The increased values changed the execution of the test path and made it not executable by the FF-PID controller. Therefore, the comparison was only made at the lower value limits.

3.3 Controller parameter tuning

Proper tuning is a crucial part of a controller design. The previous controller had been tuned prior to the start of this assignment, and there is no information on how the values were chosen. At first, the PID element of the new controller had been tuned using the Ziegler–Nichols method as advised in [15]. It was done for each of the arms separately, which resulted in different values for each of the arms. The arms remained connected, but only one was controlled at a time. This tuning proved inappropriate due to the high overshoot it allowed. Therefore, it was used only as a starting point. Further tuning was

done using the method described in the controller documentation [12], which had a more heuristic approach. This resulted in the final gains described in table 3.

	Arm 0	Arm 1
Proportional Gain K _p	25300	22300
Integration Gain K _i	400	400
Derivative Gain K _d	300000	500000

Table 3: PID Gain factors

4 Results

In this chapter, the outcome of the measurements of arm position, tracking error, current supplied to the motor and response of the system to disturbance will be presented and briefly described. The main emphasis will be put on presenting the difference between the previously used FF-PID and the newly implemented IDC.

4.0.1 Position tracking performance

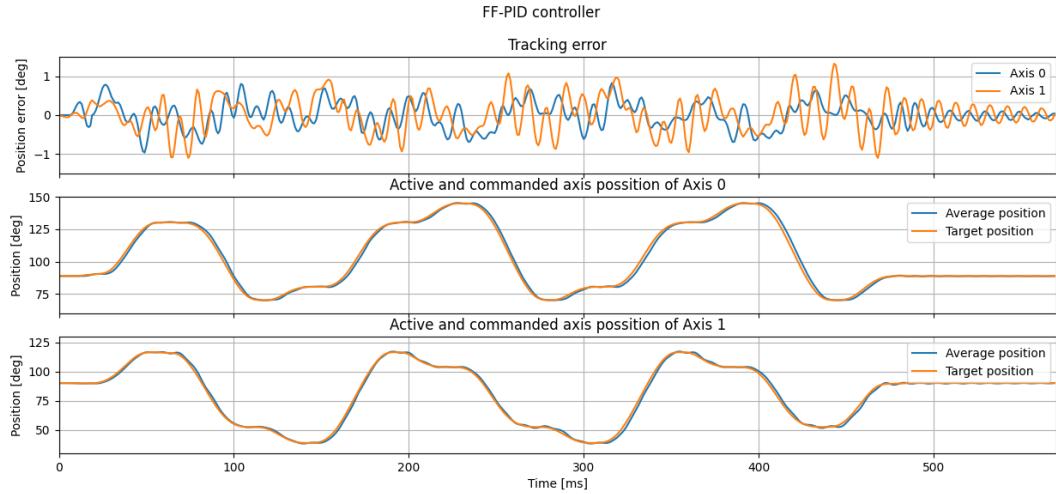


Figure 9: Average tracking error and position of Axis 0 and 1 of the FF-PID

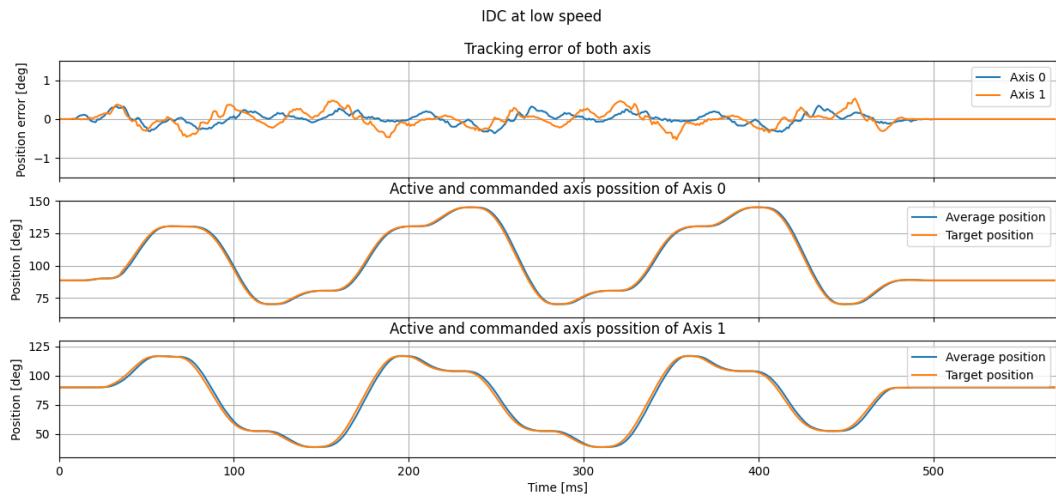


Figure 10: Average tracking error and position of Axis 0 and 1 of the inverse dynamic controller at low speed

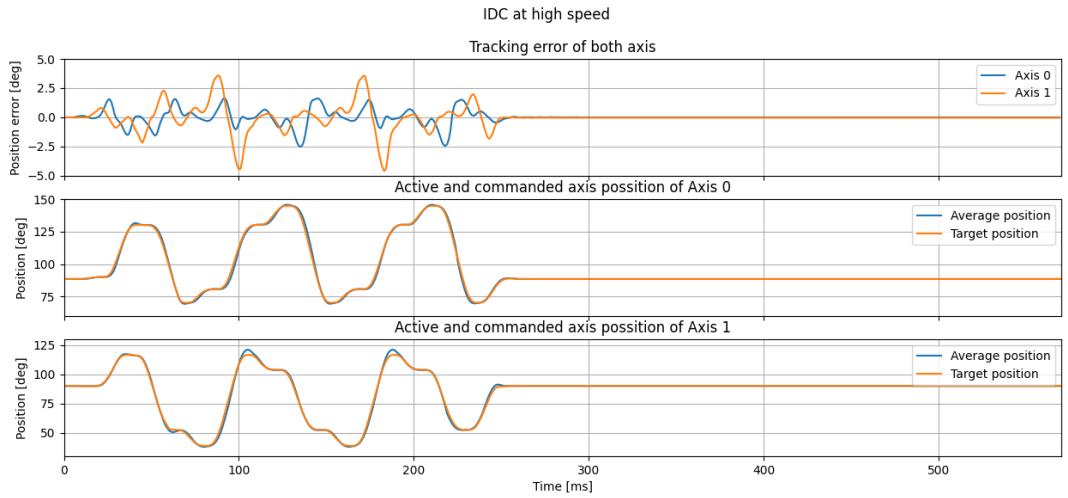


Figure 11: Average tracking error and position of Axis 0 and 1 of the IDC at high speed

4.1 Comparison between the FF-PID and IDC

Figure 9 displays the performance of the FF-PID controller. That controller had notable oscillatory behaviour that can be observed throughout the whole movement, but especially at the final set point. The test sequence took 480 ms to execute, but the underdamped oscillations did not settle within 100 ms after the commanded movement ended. At its peak, the average tracking error reached -0.95° for axis 0 and 1.33° for axis 1.

Figure 10 displays the performance of the IDC. The test had been performed with the same velocity and acceleration limiting factors as the one made with the FF-PID controller. Therefore, the sequence took 480 ms to execute, but the settling at the final point took less than 5 ms for axis 0 and 10 ms for axis 1. At its peak, the average tracking error reached -0.35° for axis 0 and -0.53° for axis 1.

4.1.1 IDC at increased speed

Figure 11 displays the performance of the IDC at increased speed, which increased due to the change in the limiting values described in table 2. The same test sequence took 245 ms to execute, and the settling at the final point took 35 ms for both axes with an oscillation amplitude of $\pm 0.035^\circ$ (1 quad count). At its peak, the average tracking error reached -2.5° for axis 0 and -4.6° for axis 1.

4.2 Performance between runs

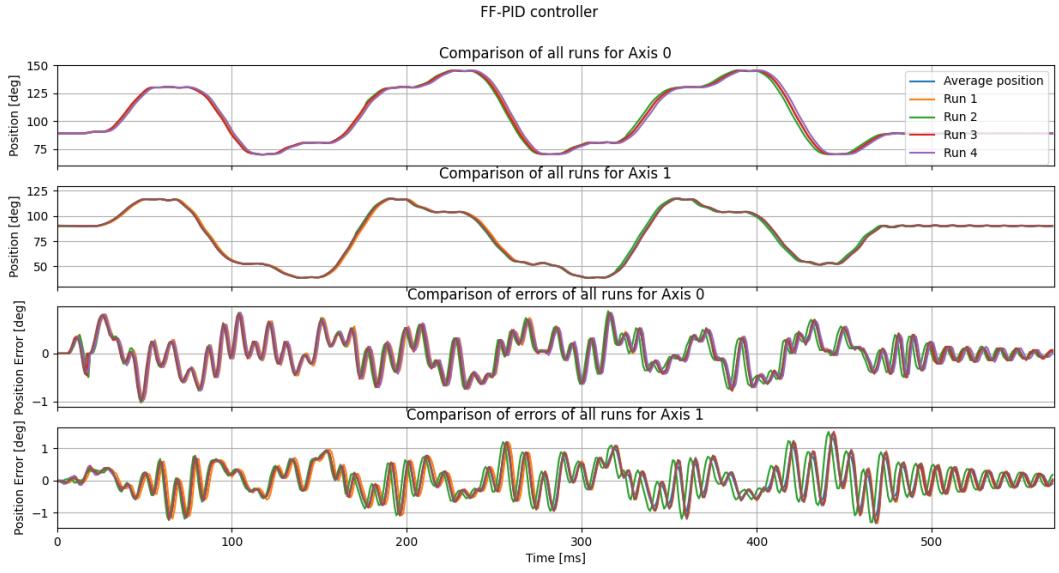


Figure 12: Position and error comparison between individual runs of Axis 0 and 1 of the FF-PID

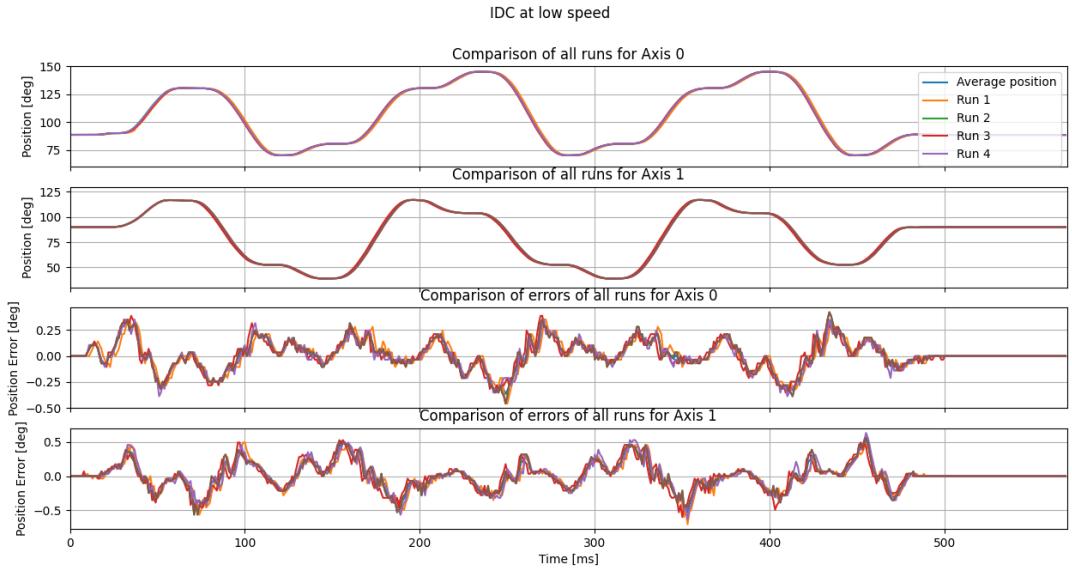


Figure 13: Position and error comparison between individual runs of Axis 0 and 1 of IDC at low speed

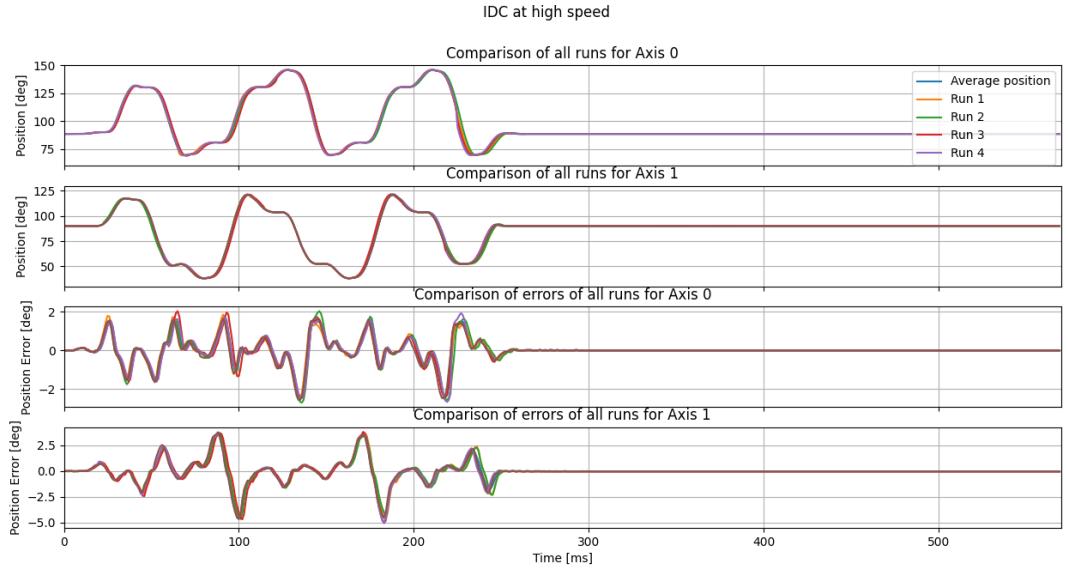


Figure 14: Position and error comparison between individual runs of Axis 0 and 1 of IDC at increased speed

Figures 12 and 13 illustrate the differences between individual runs of respective control schemes with the same velocity and acceleration constraints. There is an apparent overlap between the runs in both cases, yet the IDC has less deviation between the runs, especially when comparing axis 1.

Figure 14 illustrates the differences between individual runs of IDC at higher velocities. Once again, overlap between the runs is clearly seen, yet there is a noticeable deviation between the runs of axis 0.

Those deviations will be further studied in the following subsection.

4.2.1 Deviation between runs

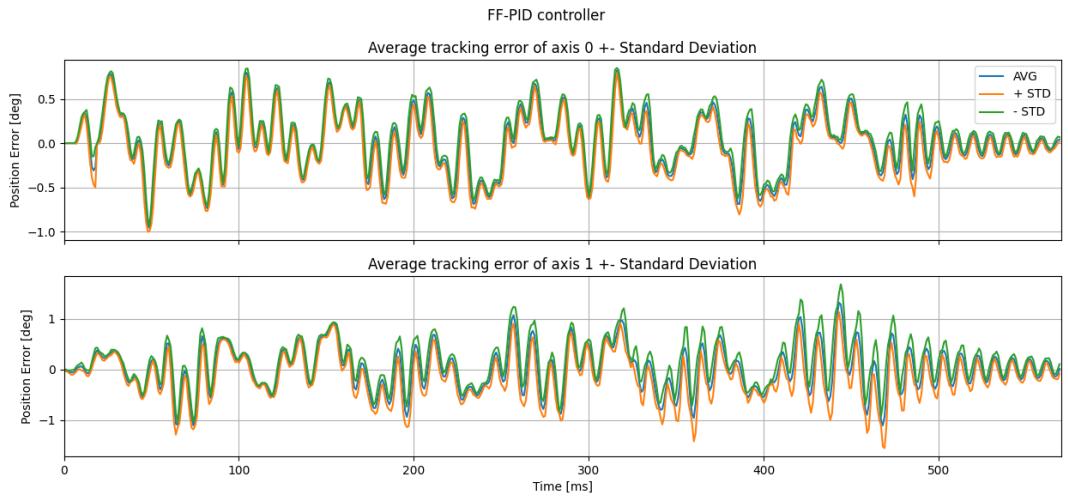


Figure 15: Average tracking error and standard its deviation visualisation - FF-PID

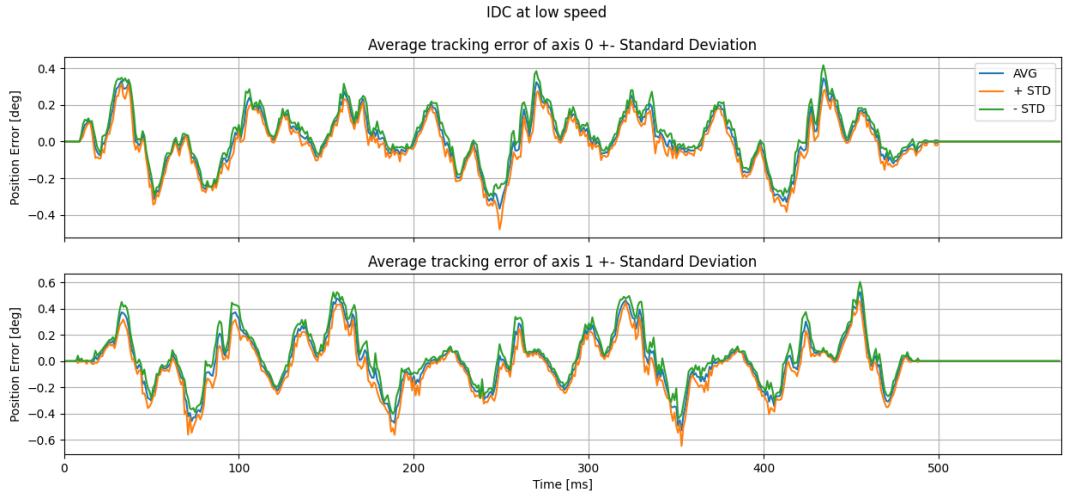


Figure 16: Average tracking error and its standard deviation visualisation - IDC at low speed

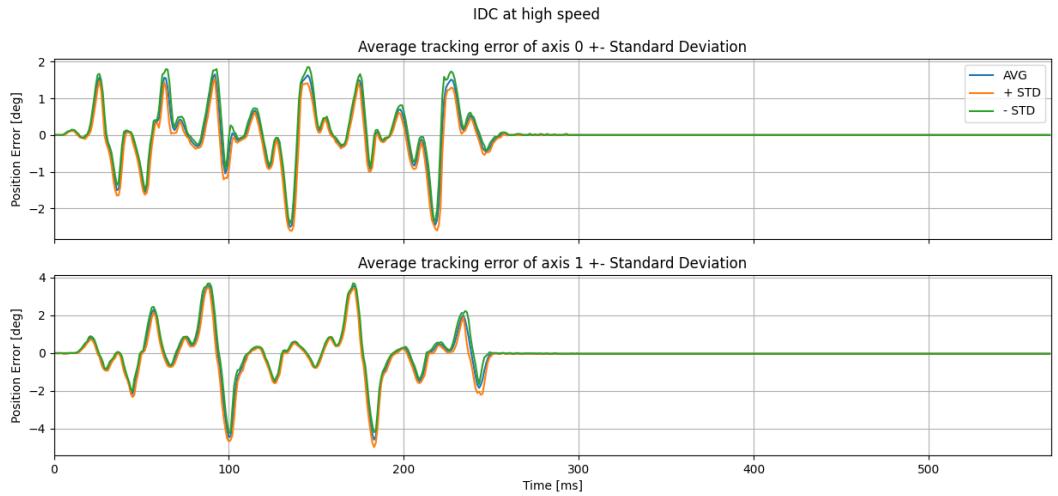


Figure 17: Average tracking error and its standard deviation visualisation - IDC at high speed

To further assess the performance of the new controller, a standard deviation of tracking error and positions between the individual runs was calculated. Visualisation of that comparison can be seen in figures 17, 16 and 15. Further, the maximum values of standard deviations were found and displayed in table 4 to provide another source of performance measurement.

	FF-PID	IDC low speed	IDC high speed
Max. dev. position Axis 0	3.5 °	1.53 °	6.33 °
Max. dev. position Axis 1	2.3 °	1.61 °	3.85 °
Max. dev. error Axis 0	0.27 °	0.11 °	0.90 °
Max dev. error Axis 1	0.64 °	0.15 °	0.91 °

Table 4: Comparison of maximum values in standard deviations of position and tracking error

4.3 Measurements of motor currents

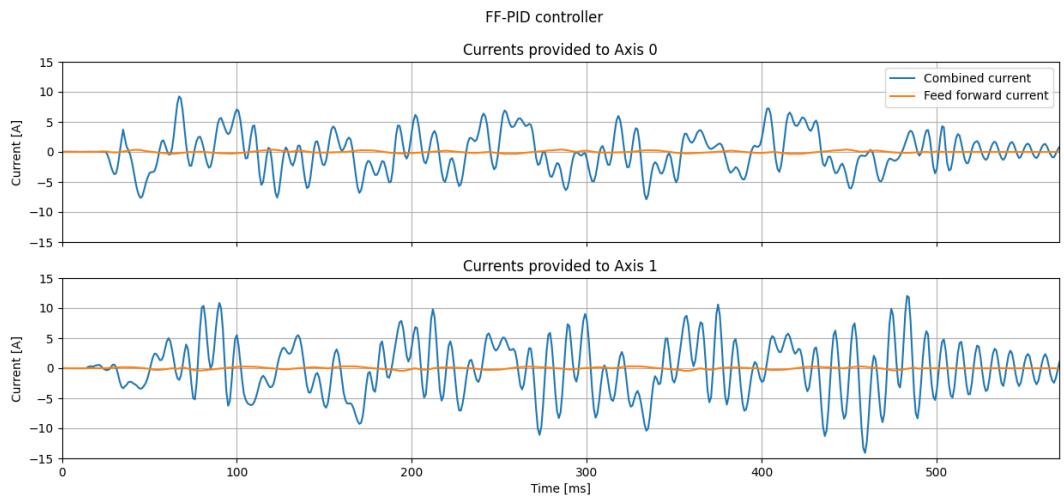


Figure 18: Currents supplied to both motor axis - FF-PID

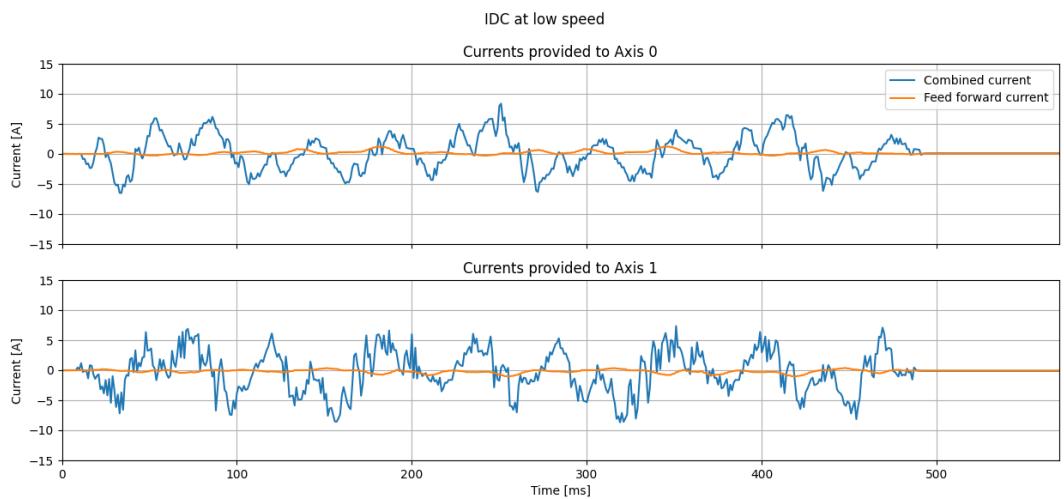


Figure 19: Currents supplied to both motor axis - IDC low speed

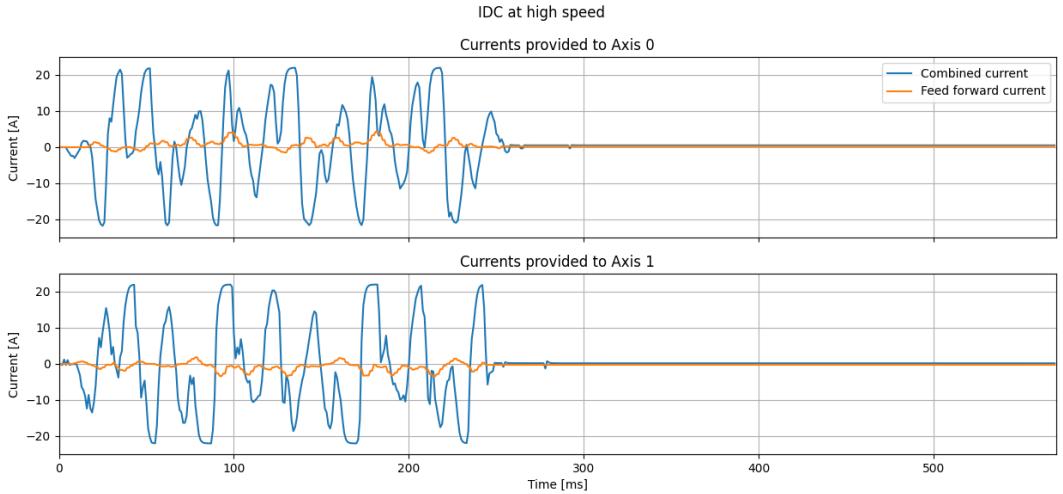


Figure 20: Currents supplied to both motor axis - IDC high speed

Figures 18, 19 and 20 present the currents delivered to the motors during the test operation. The currents are separated into feed-forward currents and actual currents supplied to the motor. There are negative currents present on the graphs mentioned above; this is an indication that the motors were commanded to spin in the clockwise direction - a positive current indicates a counterclockwise direction. To simplify the analysis of the results, an absolute value was used. Key features of the comparison are as follows:

- The presence of the FF current is clearly visible with the new IDC controller at increased velocities. The FF component had provided 4.2 A of current to motor 0 at the peak when the combined current was 12.3 A and 3.4 A for motor 1 when the combined current was 18 A. The combined current reached its maximal allowed value of ± 22 A on both of the motors during the operation at higher velocities.
- At the lower velocities, the presence of the FF is also visible in the IDC. FF current component peaked at 1.2 A when the combined current was 2.4 A for motor 0 and 1.0 A for motor 1 when the combined current was 6.0 A. At this velocity, the combined current peaked at 8.3 A for motor 0 and at 7.3 A for motor 1.
- The FF-PID controller commanded a maximum current of 14A for motor 1 and 9.2 A for motor 0, where the FF current component peaked at 0.4 A for both motor 0 and 1 at a moment when the combined current reached 2A on motor 0 and 6.4 A on motor 1.

4.4 Performance in the frequency domain

An effort was put into measuring the performance of the IDC in the frequency domain, but it did not provide meaningful results. It was most likely due to hardware limitations, but the ApposC firmware also does not provide any support for frequency analysis. This case was further described in appendix B.

5 Discussion

The following chapter will discuss the findings of the measurements presented in the previous chapter. A comparison between the controllers will be shown, followed by describing the improvements made and the importance of the newly added algorithm. At last limitations of the system will be discussed.

5.1 Performance comparison

Comparison between figure 9 and 10 shows that the new controller removes the oscillating behaviour present in the previous controller - settling time had been reduced to 10 ms. It also illustrates that the displacement error has been reduced by a factor of 2.7 for Axis 0 and 2.5 for Axis 1. With the IDC, there is significantly less deviation between the runs - the maximum deviation in position is 2.3 times smaller for Axis 0 and 1.4 times smaller for Axis 1. Deviation in tracking error had also been reduced.

5.2 Faster movement

Thanks to reducing the limitations, the IDC completed the given test path in 245 ms, almost half the original time of 480 ms. The tracking errors increased because of the increased acceleration. This was to be expected, as the acceleration increased by a factor of 7. Due to that, the settling time also increased, but the oscillations are in the range of single counts of the rotary encoders (that corresponds to 0.035°) and last for a maximum of 35 ms, which is significantly less than the oscillations in the position of the ff-pid controller that ranged between $\pm 0.5^\circ$ and lasted over 100ms.

5.3 Importance of feed-forward

In the previously used controller, the forwarded part of the current was marginal, corresponding to 6% of the total current at its peak. Figures 19 and 20 show the noticeable presence of the forwarded currents that are a result of the inverse dynamic solver. During the peak, forwarded current contributed to a third of the delivered current during the high-speed operation and to half of the delivered current during the low-speed operation.

5.4 Performance bottleneck

During high-speed operation, at peaks, the controller commands the delivery of the maximum allowed current of 22A. This is the maximum current that the controller can handle in this amplifier configuration. Because of this limitation, the gain of the controller is at the limit, which poses a limit on the maximum torques that can be generated by the motors. This corresponds to the maximum accelerations that can be handled by the system. Because of the system gain, the jerk had to be limited so as not to command changes in the accelerations that the controller would not be able to control. That being said, the maximum current delivered by the amplifier is the bottleneck of the system.

6 Conclusion

This chapter concludes the report's findings and argues whether the goal has been met. Further, a suggestion for further points of improvement for the robot is made.

6.1 Argument on goal confirmation

The Inverse Dynamic Controller outperforms the previously used controller in all notable ways when operated at the same velocities and allows for their increase. The controller has been proven to be accurate - it reaches the final position with a precision of $\pm 0.035^\circ$ and has a maximum tracking error of 4.6° during movement. It has executed the path in 245ms, which is almost a 50% improvement. The oscillations had been effectively removed - with minimal amplitude of $\pm 0.035^\circ$, settling time being almost a third of the assigned maximum. The standard deviation tracking error had been reduced more than twice, where the deviation in position was decreased by 1.4 and 2.3, depending on the axis, which proves the increase in the repeatability of the system. Unfortunately, the new controller has a noticeable drawback. Due to the higher commanded accelerations, the robot became noticeably louder, which might pose a problem, but considering its intended purpose, it will make the robot more noticeable. Another important observation is the difference between the axis of the robot. Axis 0 tends to follow the set position more accurately, but it has reduced repeatability when comparing between the runs.

6.2 Recommendation

As for further improvement, there are several subjects that could be looked into. First, the inverse dynamic solver should be moved from the application layer of the software to the firmware layer. This would result in an increase in algorithm frequency and, by that, make the calculation more precise. The same should be done to the passive angles state estimators. Ideally, a pair of encoders would be installed on the passive joints and replace the inverse kinematic equations for the passive joint angles. Derivative calculation could also be improved - the same algorithm could be used as the one that is used for active angle calculation. Unfortunately, the user does not have access to this algorithm or the layer of the software where it is written. That all combined would increase the refresh rate of the dynamic solver, possibly increasing the performance.

Another key improvement would be to increase the maximum deliverable current. This could be done by using the twin mode of the current controller or by adding an external amplifier to power the motors. It would allow the motors to generate more torque, which would increase the possible accelerations.

The last point of improvement worth suggesting is analyzing the system in the frequency domain. This would require changing the controller's hardware from MiniMACS6 to a different controller or using an external oscilloscope to record data at a higher frequency.

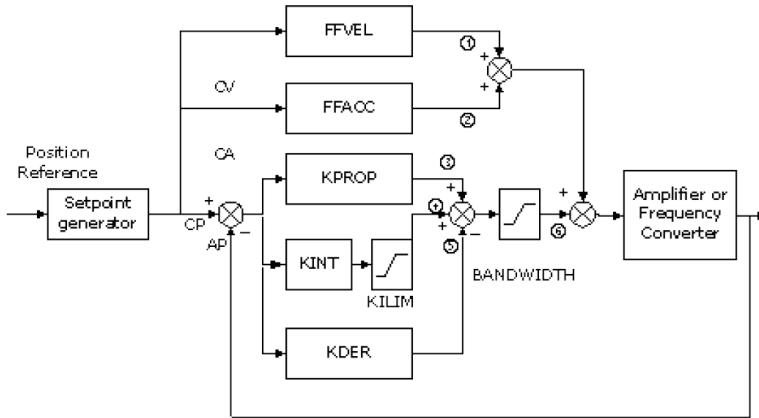
References

- [1] Nobuyuki Furuya and Hiroshi Makino. “選[F]的コンプライアンス構造をもつ組立用ロボットの開[F]研究(第1報)系の特性.” In: 精密機械 46.12 (1980), pp. 1525–1531. DOI: 10.2493/JJSPE1933.46.1525.
- [2] Hiroshi Makino. “Development of the SCARA.” In: *Journal of Robotics and Mechatronics* 26.1 (2014), pp. 5–8. DOI: 10.20965/jrm.2014.p0005.
- [3] 3: (a) *A schematic of the five-bar robot proposed in US patent No.... / Download Scientific Diagram.* URL: https://www.researchgate.net/figure/a-A-schematic-of-the-five-bar-robot-proposed-in-US-patent-No-4-712-971-5-and-b-a_fig3_277299749.
- [4] Lucas Campos, Francis Bourbonnais, Ilian A. Bonev, and Pascal Bigras. “Development of a Five-Bar Parallel Robot With Large Workspace.” In: *Proceedings of the ASME Design Engineering Technical Conference* 2 (2011), pp. 917–922. DOI: 10.1115/DETC2010-28962. URL: <https://dx.doi.org/10.1115/DETC2010-28962>.
- [5] Francis Bourbonnais, Pascal Bigras, and Ilian A. Bonev. “Minimum-time trajectory planning and control of a pick-and-place five-bar parallel robot.” In: *IEEE/ASME Transactions on Mechatronics* 20.2 (2015), pp. 740–749. DOI: 10.1109/TMECH.2014.2318999.
- [6] Kevin M Lynch and Frank C Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge: Cambridge University Press, 2017, Ch. 8–Ch.11. DOI: DOI : 10.1017/9781316661239. URL: <https://modernrobotics.northwestern.edu/>.
- [7] Shamik Sen, Bhaskar Dasgupta, and Asok Kumar Mallik. “Variational approach for singularity-free path-planning of parallel manipulators.” In: *Mechanism and Machine Theory* 23.11 (2003), pp. 1165–1183. DOI: 10.1016/S0094-114X(03)00065-X.
- [8] S. Kock and W. Schumacher. “A parallel x-y manipulator with actuation redundancy for high-speed and active-stiffness applications.” In: *Proceedings - IEEE International Conference on Robotics and Automation* 3 (1998), pp. 2295–2300. DOI: 10.1109/ROBOT.1998.680665.
- [9] Branimir Mrak, Taranjitsingh Singh, Quentin Docquier, and Joris Gilis. “Model Predictive Control of a Highly Dynamic Parallel SCARA Robot.” In: *9th 2023 International Conference on Control, Decision and Information Technologies, CoDIT 2023* (2023), pp. 2027–2031. DOI: 10.1109/CODIT58514.2023.10284063.
- [10] V-T. Nguyen, C-Y. Lin, S-F. Su, and Q-V. Tran. “Adaptive PID tracking control based radial basic function networks for a 2-DOF parallel manipulator.” In: *2017 International Conference on System Science and Engineering (ICSSE)*. 2017, pp. 309–312. DOI: 10.1109/ICSSE.2017.8030887.
- [11] Zhili Hou, Ruiqin Li, Wenge Wu, and Huibin Qin. “Kinematic analysis and experimental verification of a controllable five-bar parallel kinematic manipulator.” In: *ICCAIS 2015 - 4th International Conference on Control, Automation and Information Sciences* (Nov. 2015), pp. 486–490. DOI: 10.1109/ICCAIS.2015.7338718.
- [12] Maxon | ZUB. *ApposC Help manual v 7.00.95*. June 2023. URL: https://www.maxongroup.com/medias/sys_master/root/9139076956190/ApposIDE-7-00-95.zip.

- [13] Maxon Benelux. *EC frameless DT - Installation Manual*. URL: https://www.maxongroup.com/medias/sys_master/root/9239182606366/ec-frameless-dt-en-03-2024.pdf.
- [14] Maxon | ZUB. *MiniMACS6-AMP-4/50/10 Hardware Reference*. May 2023. URL: https://www.maxongroup.com/medias/sys_master/root/9139075940382/MiniMACS6-AMP-4-50-10-Hardware-Reference-en.pdf.
- [15] Kaiser David. *Fundamentals of Servo Motion Control*. URL: <https://www.parker.com/parkerimages/emn/ServoFundamentals.pdf>.

During the preparation of this work, the author used Grammarly to find and correct spelling and grammar mistakes. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the work.

A Appendix: Block diagram of the FF-PID controller



AV actual velocity, calculated as difference of the actual velocity minus the last velocity

AP actual position (calculated from encoder feedback) in qc.

CP current position set point in qc.

CV velocity set point in qc/ms; it is calculated as difference of the actual set point position minus the last set point position.

CA acceleration set point

Figure 21: Block diagram of the FF-PID controller [12]

B Appendix: Bode plots

B.1 Frequency domain

A set of tests was performed to visualise the controller's performance in the frequency domain. Unfortunately, this did not yield meaningful results. One of the reasons is the low record frequency of the hardware, which is only 100 Hz. The second potential cause of the error is that the hardware and associated software does not have any features of the frequency analysis, so it had to be done using external software, which might have introduced some flaws. Selected elements of the results have been attached in the appendix, displaying the difference in the plan of the system with different controllers, which was expected to remain the same between the runs. This further proves that no information can be gained from the frequency domain with the current hardware.

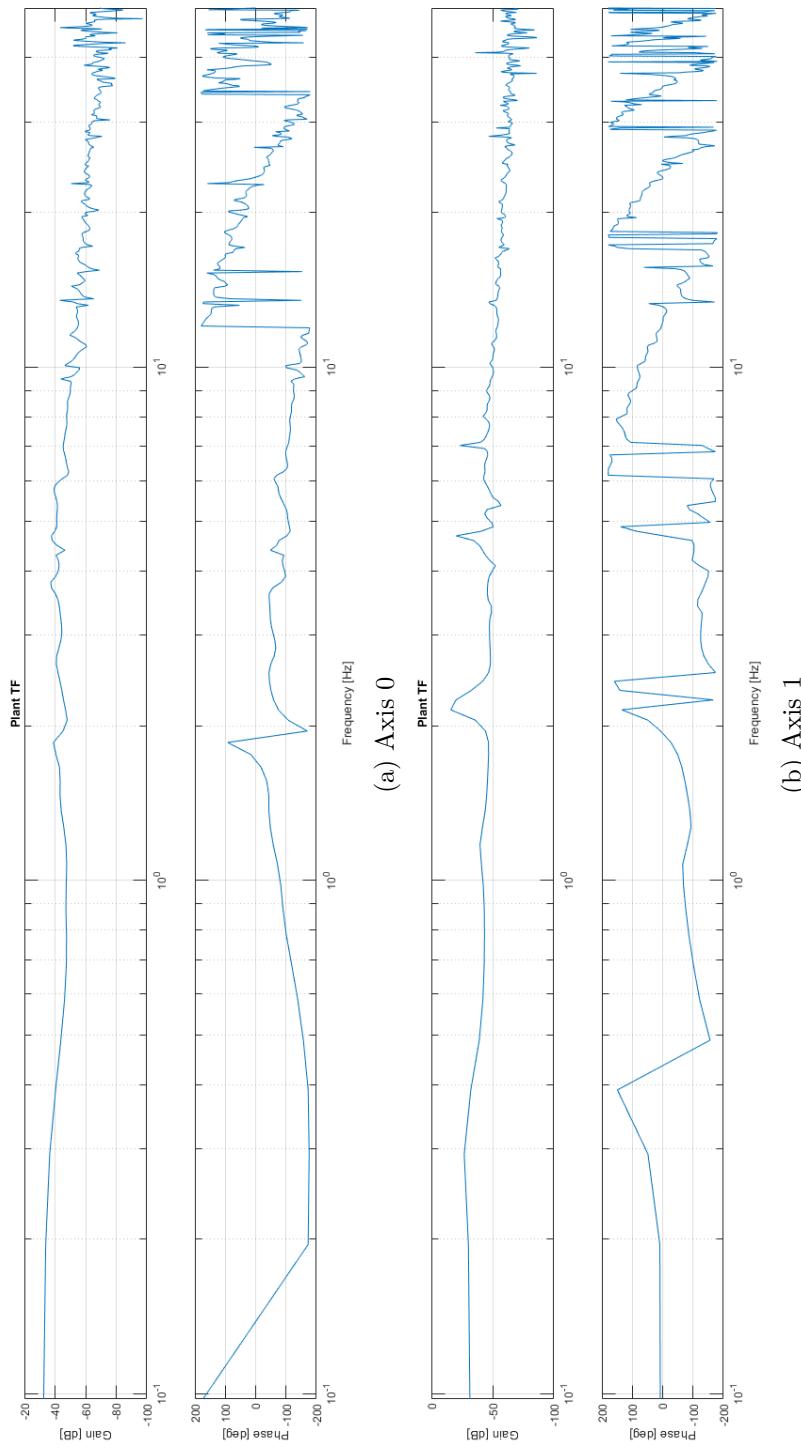


Figure 22: Bode plot of the plant - FF-PID controller test

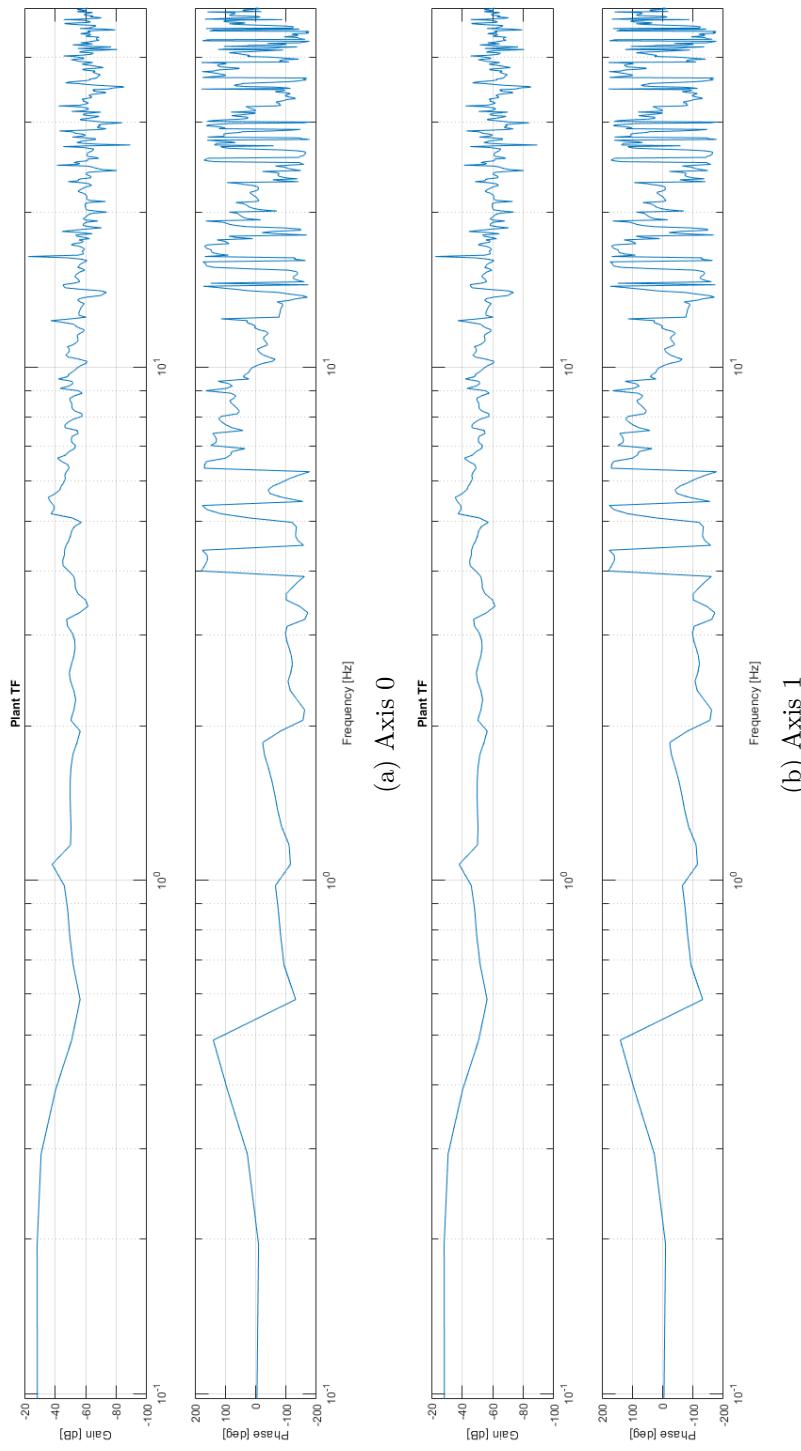


Figure 23: Bode plot of the plant - IDC controller test