



Hochschule Karlsruhe  
Technik und Wirtschaft  
UNIVERSITY OF APPLIED SCIENCES

**FESTO**

---

# **Iterative Learning Control (ILC) Implementation on Festo Pneumatic Joint ROPA 5 for Robotic Application**

---

prepared by

**LongHe Chew**

Schwabstraße 12  
73760 Ostfildern

Matrikelnummer: 89552

Hochschule Karlsruhe  
Fakultät für Maschinenbau und Mechatronik

Betreuer : Dr. Rainer Nitsche

Bearbeitungszeitraum: September 2023 bis Februar 2024

# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Festo Pneumatic Joint (pJoint) - Swivel Drive . . . . .	2
1.2. Iterative Learning Control (ILC) . . . . .	3
1.2.1. Stability and Convergency of ILC . . . . .	4
1.3. Slow Trajectory Motion Control . . . . .	5
<b>2. Methodology</b>	<b>6</b>
2.1. Test Bench . . . . .	6
2.2. Block Diagram Topology . . . . .	8
2.3. Simulink Model of The Controllers . . . . .	10
2.4. dSpace Control Desk . . . . .	11
<b>3. Results</b>	<b>12</b>
3.1. Low Amplitude with Slow Trajectory Testing . . . . .	13
3.2. High Amplitude with Slow Trajectory Testing . . . . .	16
<b>4. Discussion</b>	<b>17</b>
<b>5. Conclusion</b>	<b>18</b>
<b>Appendix</b>	<b>19</b>
<b>References</b>	<b>23</b>

# List of Figures

1.1.	Schematic Diagram of the pJoint. [1]	2
1.2.	Basic Idea of ILC [2]	3
1.3.	Control Loop	5
2.1.	Actual Test Bench	7
2.2.	Block Diagram Overview of the control loop. The green and orange arrows indicate parameters.	8
2.3.	Simulink Overview of the controller implementation.	10
2.4.	Interface for parameters tuning in dSpace Control Desk	11
2.5.	Time Plotter in dSpace Control Desk	11
3.1.	$\varphi_d$ and $\varphi_m$ at 3rd Iteration.	14
3.2.	measured error at 3rd Iteration $e_{\max} = 0.1081^\circ$ , $e_{\text{rms}} = 0.07158^\circ$ .	14
3.3.	$\dot{\varphi}_d$ and $\dot{\varphi}_m$ at 3rd Iteration.	14
3.4.	$\varphi_d$ and $\varphi_m$ at 33rd Iteration.	15
3.5.	measured error at 33rd Iteration $e_{\max} = 2.866 \cdot 10^{-3}^\circ$ , $e_{\text{rms}} = 7.187 \cdot 10^{-4}^\circ$ .	15
3.6.	$\dot{\varphi}_d$ and $\dot{\varphi}_m$ at 33rd Iteration.	15
A.1.	$\varphi_d$ and $\varphi_m$ at 3rd Iteration.	20
A.2.	Measured error at 3rd Iteration $e_{\max} = 0.3680^\circ$ , $e_{\text{rms}} = 0.07636^\circ$ .	20
A.3.	$\dot{\varphi}_d$ and $\dot{\varphi}_m$ at 3rd Iteration.	20
A.4.	$\varphi_d$ and $\varphi_m$ at 33rd Iteration.	21
A.5.	measured error at 33rd Iteration $e_{\max} = 0.01708^\circ$ , $e_{\text{rms}} = 2.025 \cdot 10^{-3}^\circ$ .	21
A.6.	$\dot{\varphi}_d$ and $\dot{\varphi}_m$ at 33rd Iteration.	21

# List of Tables

3.1. Parameter Table . . . . .	12
3.2. Low Amplitude Test Result . . . . .	16
A.1. High Amplitude Test Result . . . . .	22

# List of Symbols

Symbol	Description	Unit
$V_{dead,2}$	Dead volume at chamber 2 of swivel drive	$m^3$
$V_{dead,4}$	Dead volume at chamber 4 of swivel drive	$m^3$
$p_2$	Pressure in chamber 2	bar
$p_4$	Pressure in chamber 4	bar
$\varphi_d$	Desired angle	°
$\varphi_m$	Measured angle	°
$\dot{\varphi}_d$	Desired angular velocity	$\frac{°}{s}$
$\dot{\varphi}_m$	Measured angular velocity	$\frac{°}{s}$
$e_{max}$	Maximum Error in actual Iteration	°
$e_{rms}$	Root mean squared Error in actual Iteration	°
$k_p$	Propotional gain	varies depending on types of controller
$k_i$	Integration gain	varies depending on types of controller
$P_{mean}$	Pressure mean	bar
$k_{pressure}$	Pressure controller gain	$\frac{1}{s}$

# 1. Introduction

This chapter introduces the topic of "Iterative Learning Control (ILC) Implementation on Festo Pneumatic Joint ROPA 5 for Robotic Application". The working principle of the Festo pJoint is provided, and the introduction of Iterative Learning Control (ILC) is presented, including its principles and how it can be applied to improve system performance. The key features and functionalities of ILC are highlighted. The aim of this chapter is to lay the foundation for understanding the implementation of ILC on the Festo pJoint ROPA 5 drive.

In the field of robotics at Festo, lightweight pneumatic robots for industrial use are developed. The joints of these robots are actuated pneumatically. An advantage of pneumatic joints, especially in collaborative robotics, is their ability to control both position and joint stiffness (multi-variable control). However, pneumatic robots have limitations, including limited precision and speed, lower force and payload capacities compared to other types of robotic systems, and a constant need for compressed air, resulting in higher energy consumption.

Therefore, various control strategies can be implemented to improve the performance of pneumatic robots due to their limitations. One such strategy is Iterative Learning Control (ILC), which can be applied to address the challenges associated with limited precision and speed. By utilizing the principles of ILC, the system can learn from previous iterations and make incremental adjustments to improve trajectory tracking and overall performance.

ILC is particularly useful in robotics because robots often perform repetitive tasks. By leveraging the repetitive nature of robotic operations, ILC enables robots to learn from each iteration, gradually reducing errors and improving performance with each repetition [4]. This iterative learning process helps robots achieve higher levels of accuracy and consistency in their repetitive tasks.

## 1.1. Festo Pneumatic Joint (pJoint) - Swivel Drive

A swivel drive is a pneumatically controlled double-acting rotary drive. Unlike an electric motor, a swivel drive does not require gears to perform rotary motion. It consists of a piston-cylinder with two chambers: Chamber 2 and Chamber 4. Each chamber has a port that allows mass flow to enter (aerate) and exit (de-aerate) across the chamber, creating a pressure difference between the two chambers. This pressure difference generates a torque on the piston, resulting in rotary motion. The rotation angle typically ranges from  $-135^\circ$  to  $135^\circ$ . Figure 1.1 illustrates the schematic structure of a swivel drive.

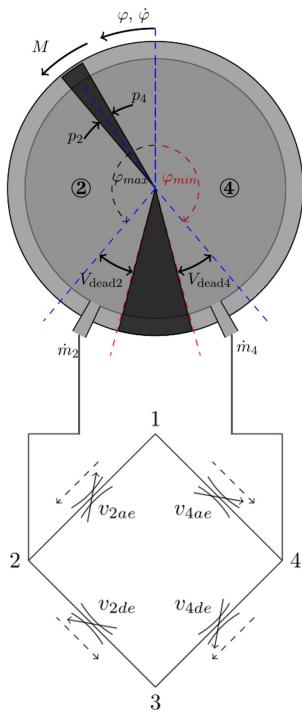


Figure 1.1: Schematic Diagram of the pJoint. [1]

From the schematic drawing, the connection of the valves can also be observed. The indications ② and ④ represent Chamber 2 and Chamber 4, respectively. Indication 1 is the inlet port, which supplies pressure to the valve, and Indication 3 is the outlet port, which is exposed to ambient pressure. The four valve symbols represent the piezoelectric valves, which are controlled by voltages to open and close in a proportional manner. The  $V_{dead,2}$  and  $V_{dead,4}$  represent the dead volume or the limit positions for the traveling piston. When the pressure in ②,  $p_2$ , is higher than the pressure in ④,  $p_4$ , the pistons will move in a (-) clockwise direction<sup>1</sup> (turning to the right based on the figure above), and vice versa.

---

<sup>1</sup>mathematically positive sense of rotation is counterclockwise

## 1.2. Iterative Learning Control (ILC)

Iterative Learning Control (ILC) is a feedback control strategy that improves the performance of dynamic systems by learning from previous iterations. It is highly recommended to apply ILC in applications where repetitive tasks need to be performed with high precision and accuracy, such as in robotics arm applications.

The first occurrence of ILC appears to be in a US patent application for Learning control of actuators in control systems [2]. The basic idea behind ILC is to iteratively adjust the control inputs based on the error between the desired output trajectory and the actual output trajectory. By incorporating the knowledge gained from previous iterations, the control inputs are refined to reduce tracking errors and improve the system's ability to follow desired trajectories. The basic idea is visualized in Figure 1.2 below.

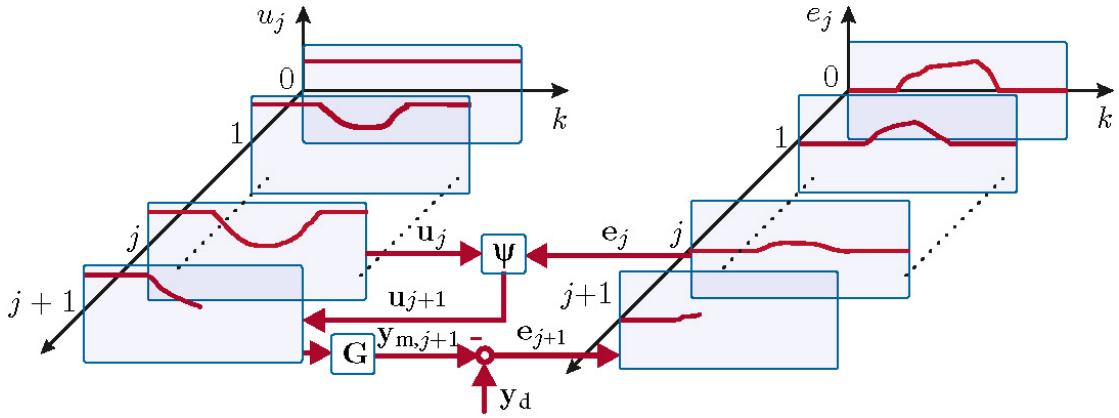


Figure 1.2: Basic Idea of ILC [2].

The ILC concept from Figure 1.2 can be expressed in the following equation:

$$u[k + 1] = u[k] + \psi e[k + 1] \quad (1.1)$$

In (1.1), the left hand side  $u[k + 1]$  represents the control signal at the  $(k + 1)$ th iteration, while  $u[k]$  represents the control signal at the  $k$ th iteration. The learning gain, denoted as  $\psi$ , determines the extent to which the control signal is adjusted based on the error  $e[k + 1]$ . The error, calculated as the difference between the desired output or reference signal and the actual output of the system, serves as the driving force for learning and improvement [5].

### 1.2.1. Stability and Convergency of ILC

When it comes to control, stability and convergence play important roles in ensuring that the control system remains stable and does not diverge. Stability is crucial to prevent any instability that could lead to unpredictable or undesirable behavior, ensuring that the control system stays on track with the desired trajectory. Convergence refers to the ability of the ILC algorithm to reduce the error between iterations over time. By ensuring stability and promoting convergence, ILC can effectively improve the performance of repetitive tasks over time [7].

According to Timo Heubach's bachelor thesis [3], a specific MATLAB script can be used to investigate the stability and convergence of ILC. The script is shown in Script 1.1 below. For further understanding, the detail can be found in his bachelor thesis.

```
1 while(finish ~= 1)
2     switch state
3         case 1 % kp
4             disp('kp ermitteln... ')
5             for kp = kp_test
6                 L = kp * E;
7                 A = Q*(E-L*G);
8                 rho(n) = max(abs(eig(A)));
9                 sigma(n) = max(svd(A));
10                if rho(n) < 1 && sigma(n) < 1
11                    kp_sol(end+1) = kp;
12                    e_inf_seq(:,n) = (E - G/(E-A)*Q*L) *
13                        reshape(y_d,numel(y_d),1);
14                    e_inf_max(end+1) = max(abs(e_inf_seq
15                        (:,n)));
16                    e_inf_rms(end+1) = rms(e_inf_seq(:,n)
17                        );
18                end
19                n = n + 1;
20            end
21            if isempty(kp_sol)
22                n = 1;
23                state = 2;
24                disp("Keine passenden Werte fuer kp
25                    gefunden")
26            else
27                finish = 1;
28                disp("Passende Werte fuer kp gefunden")
29            end
```

Script 1.1: Stability and Convergency Analysis [3].

### 1.3. Slow Trajectory Motion Control

A slow trajectory motion control testing is being carried out in the testing. As the name suggests, the swivel drive will be tested in slow motion to better study the accuracy and stability of the control system. By understanding and implementing slow trajectory motion control techniques, researchers and practitioners can achieve improved precision, smoothness, and stability. The study of slow trajectory motion control focuses on achieving smooth and stable motion profiles. In scenarios where jerky or abrupt movements can result in damage, instability, or reduced performance, this aspect becomes critical. By implementing slow trajectory motion control strategies, researchers can ensure that systems move in a smooth and stable manner, minimizing disruptions and optimizing overall system performance. The trajectory will be fed into the controller as shown as Figure 1.3 below.

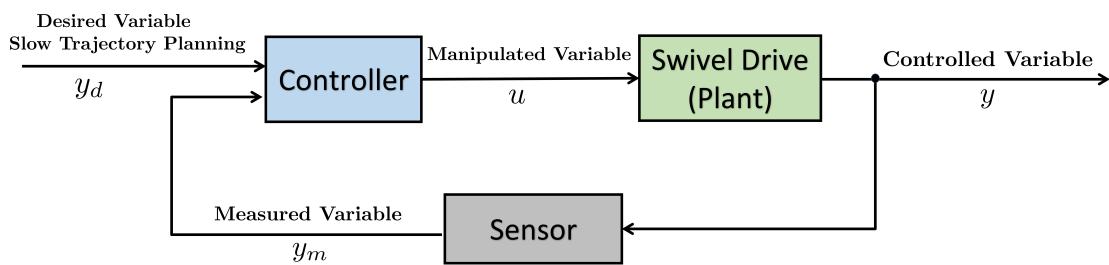


Figure 1.3: Control Loop

## 2. Methodology

This chapter explains the methodology for conducting the test, focusing on the test bench setup and the utilization of the dSPACE<sup>1</sup> software and hardware. The test bench includes the Festo pJoint ROPA 5 robotic system, sensors, and actuators. The dSPACE system serves as the software and hardware platform for real-time control, data acquisition, and interfacing with the test bench.

### 2.1. Test Bench

This section explains the purpose and configuration of the test bench. A test bench refers to a simulation environment used for testing the functionality of digital circuitry or electronic components. It is essential to verify whether the developed design meets the required specifications and performs as expected. A test bench typically includes the design under test along with test inputs, output waveforms, and measurement outputs.

The purpose of the test bench built for this project is to control a single rotating pneumatic joint, known as the swivel drive. The test bench is used to test and control the position and speed of the pneumatic joint. Figure 2.1 shows the configuration of the test bench. Only the labeled components were used in this specific test bench setup.

---

<sup>1</sup>dSpace is a Rapid Control Prototyping System

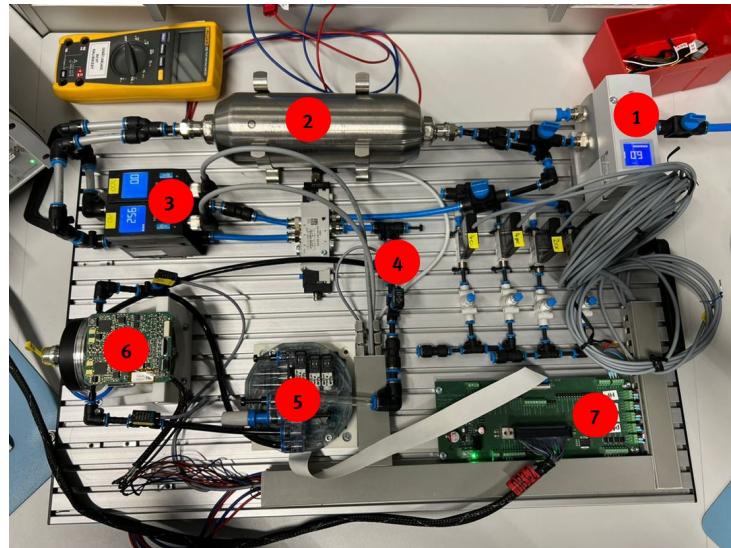


Figure 2.1: Actual Test Bench

1. Pressure Regulator Valve
2. Air Reservoir
3. Volume Flow Sensor
4. Pressure Sensor (Pressure Supply)
5. PiezoValve Unit
6. Swivel Drive with Pressure Sensor
7. Festo to dSpace PCB

The pressure supply of the system is precisely adjusted using the pressure regulating valve (Position 1). A large-volume compressed air reservoir (Position 2) is used as a buffer tank to ensure that there is always enough air in the system. The flow sensors (Position 3) enable precise measurement of the volume flow and offer the possibility to manually switch between two measuring ranges: High Flow and Low Flow. The pressure sensor for power supply (Position 4) measures and indicates the pressure supply of the system. The flow of air is controlled by the Piezo Valve Unit (Position 5), which can aerate or deaerate the air of the swivel drive. The Swivel Drive (Position 6) is the joint of a Festo pneumatic Cobot. Lastly, the Festo to dSpace PCB (Position 7) allows the dSpace hardware, MicroLabBox, to send signals to the test bench.

## 2.2. Block Diagram Topology

A topology of the block diagram in the control system is first drafted. It provides a high-level representation of the system's structure and the flow of information or signals between the blocks. Block diagram topologies are important and helpful because they provide a visual representation of a system's architecture, aiding in understanding and communication between the interfaces. They allow for the analysis of system behavior, troubleshooting, and identification of potential issues. Additionally, block diagrams serve as a documentation tool for system design and modification. Then, a simulink model is built based on the topology.

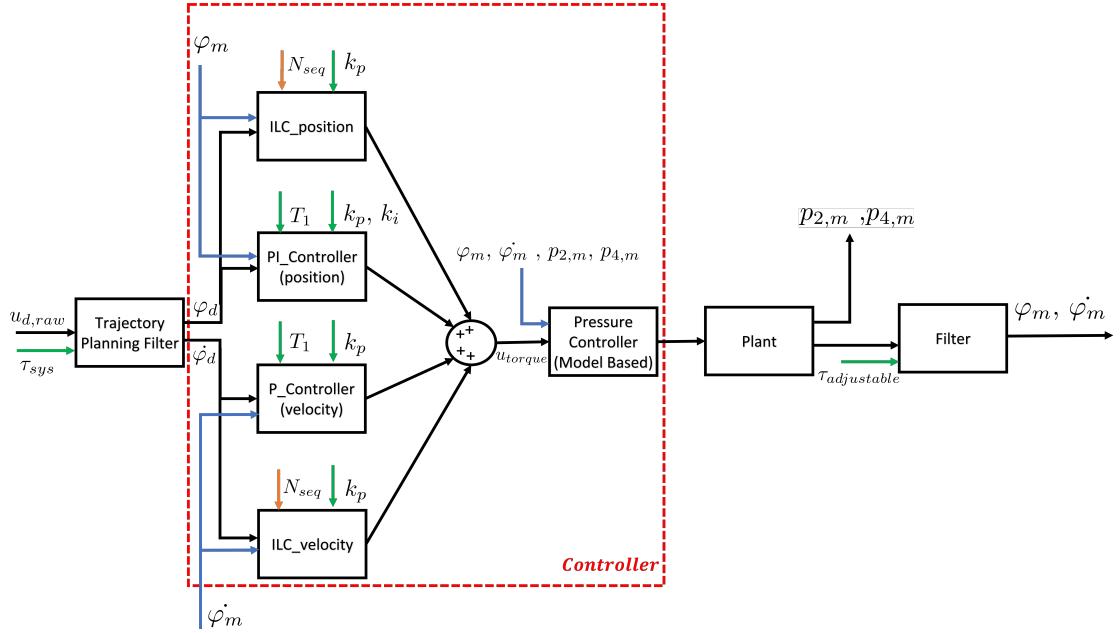


Figure 2.2: Block Diagram Overview of the control loop. The green and orange arrows indicate parameters.

Figure 2.2 above illustrates the block diagram used in this test. The desired input signal is first filtered, and the outputs of the filter are  $\varphi_d$  and  $\dot{\varphi}_d$ . The  $\varphi_d$  signal is sent to the ILC position and PI Controller, while  $\dot{\varphi}_d$  is sent to the ILC velocity and P Controller. Both ILC controllers receive the feedback signals  $\varphi_m$  and  $\dot{\varphi}_m$ , and they also take the input of the Sequence Length,  $N_{seq}$ . The outputs of the four controllers are summed and then fed to the pressure controller. The pressure controller, which has been studied and created by previous bachelors [8], receives the feedback signals ( $\varphi_m$ ,  $\dot{\varphi}_m$ ,  $p_{2,m}$ ,  $p_{4,m}$ ). Then, the control signal is sent to the plant, in this case, the opening and the closing of the piezoelectric valves are controlled. The measured data is then filtered ( $\varphi_m$ ,  $\dot{\varphi}_m$ ) and fed back into the system controllers.

The s-domain transfer function of the PI- and P- Controllers are given by:

$$G(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{1 + T_1 s}. \quad (2.1)$$

In (2.1), where  $k_p$  is the proportional gain,  $k_i$  is the integral gain,  $k_d$  is the derivate gain,  $T_1$  is the time constant of the derivative part.

Moreover, the representation of the P-Type ILC controllers in analog me discrete functions are:

$$u_{j+1}[k] = q[k](u_j[k] + k_p e_j[k + v]). \quad (2.2)$$

In (2.2),where  $u_{j+1}[k]$  is the updated control input for the next iteration,  $q[k]$  represents the learning gain,  $u_j[k]$  is the control input from the previous iteration,  $k_p$  is the proportional gain and  $e_j[k + v]$  represents the error between the desired output and the actual output.

## 2.3. Simulink Model of The Controllers

To implement the ILC controller into the control system, the first step is to construct a Simulink model using the MATLAB software. This model incorporates various blocks from the existing Festo library[6], including the planning filter, PID controller(s), ILC controller(s) developed by a previous Festo bachelor student, Timo Heubach [3], and the pressure controller created by another previous Festo bachelor student, Shruti Sharma [8]. The planning filter block ensures smooth and accurate motion planning, while the PID controller(s) generate control signals based on the error between desired and actual values. The ILC controller(s) play a crucial role in iteratively learning and improving control performance, utilizing feedback information from previous iterations to refine control signals. Basically there are certain functions lie inside each of the simulink block, which has been created by Festo. In this report only the surface structure will be shown, as Figure 2.3 below. Moreover, a dSpace to Simulink block is also added in the Simulink to communicate between hardware and the software.

It is important to have correct and accurate model first in the simulink before it is converted into the dSpace file because it ensures the proper representation and validation of the control system's behavior and performance. An accurate model enables thorough testing and verification of the control system's performance, evaluating stability, robustness, tracking accuracy, and response to disturbances.

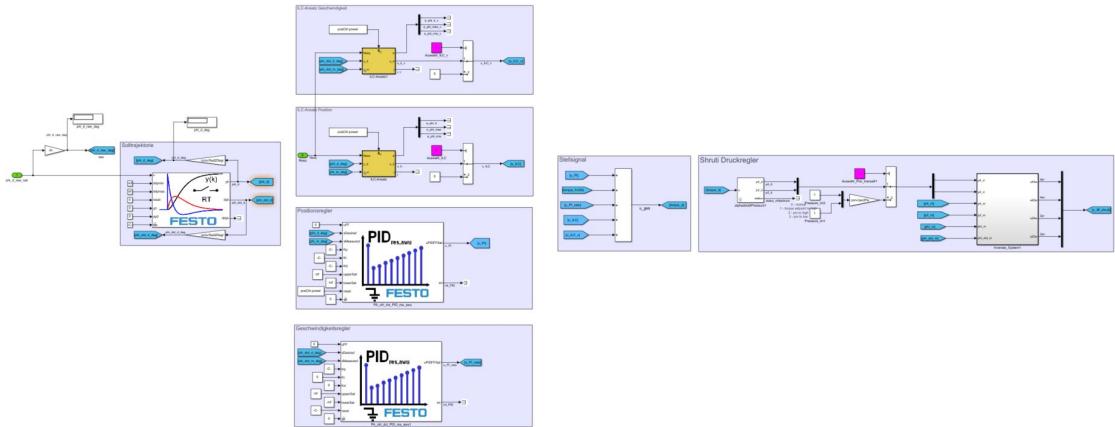


Figure 2.3: Simulink Overview of the controller implementation.

## 2.4. dSpace Control Desk

The Simulink Model is then built based on the topology and then converted into a dSpace software readable file namely .sdf. With this, the parameters can be changed manually from time to time to study the accuracy and stability of the control system. The parameters are adjusted, and the system's response in real time is observed. The responses are then observed by using the control desk, with the feedback of the sensors and plotted into real-time time plotter. Figure 2.4 shows the interface of the dSpace Control desk, and one can simply tune the parameters for a suitable outcome. Where Figure 2.5 illustrates the timeplotter graphs, where one can observe the real-time output signals.

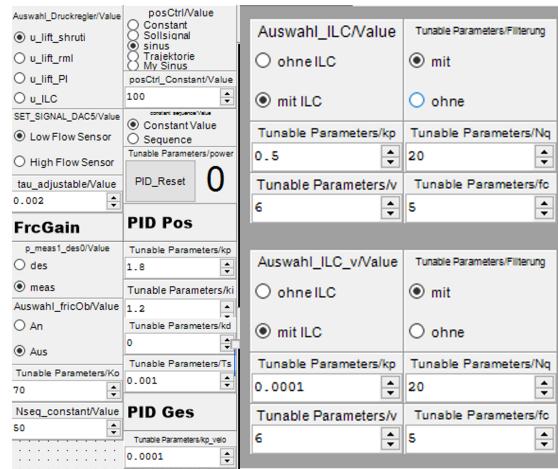


Figure 2.4: Interface for parameters tuning in dSpace Control Desk



Figure 2.5: Time Plotter in dSpace Control Desk

### 3. Results

In this chapter, the results from the testing on a pJoint will be communicated. The focus of the testing is on slow trajectory motion control, which involves controlling the movement of the pJoint at a slow and precise pace. The purpose of this testing is to evaluate the performance of the pJoint in executing slow trajectory motions and to assess its accuracy and stability with the implementation of ILC.

The parameters are initialised according to the data as shown as Table 3.1 below. These parameters are tuned to a stable, and efficient system by easily using the trial and error method. The gains of the controllers play important roles to achieve a stable system. These parameters are kept constant throughout the whole testing process. The control parameters for the controllers were determined experimentally. It is possible that they could be further optimized, but for this case, the controller's performance is sufficient. The gain for the ILC velocity controller is set to 0 because in this test, it was found that it is not necessary to implement.

Table 3.1: Parameter Table.

Parameter	Value	Unit
<b>PI Controller</b>		
$k_P$	1.8	$\frac{\text{Nm}}{\text{s}^\circ}$
$k_I$	0.9	$\frac{\text{Nm}}{\text{s}^\circ}$
<b>P Controller</b>		
$k_P$	1e-4	$\frac{\text{Nm}\cdot\text{s}}{\text{s}^\circ}$
<b>ILC Position Controller</b>		
$k_P$	0.8	$\frac{\text{Nm}}{\text{s}^\circ}$
<b>ILC Velocity Controller</b>		
$k_P$	0	$\frac{\text{Nm}\cdot\text{s}}{\text{s}^\circ}$
<b>Pressure Controller</b>		
$P_{\text{mean}}$	4.5	bar
$k_{\text{pressure}}$	35	$\frac{1}{\text{s}}$

### 3.1. Low Amplitude with Slow Trajectory Testing

As explained in Chapter 1.3, one can understand the importance of investigating controlling results with a slow trajectory motion. However, by testing with low amplitudes, it is possible to observe and characterize the nonlinear behavior more accurately. The results for both the 3rd Iteration and the 33rd Iteration will be illustrated in the Figures 3.1-3.5 below. Before the results are shown, it is expected that as the number of iterations increases, the measured signals ( $\varphi_m$  and  $\dot{\varphi}_m$ ) will closely follow the planned trajectory signals, indicating improved system performance. Moreover, as the number of iterations increases, the errors between the measured and planned trajectory signals are expected to become smaller. The planned trajectory (desired angle) is subjected to,  $\varphi_d = 3^\circ \sin(0.2\pi t)$ .

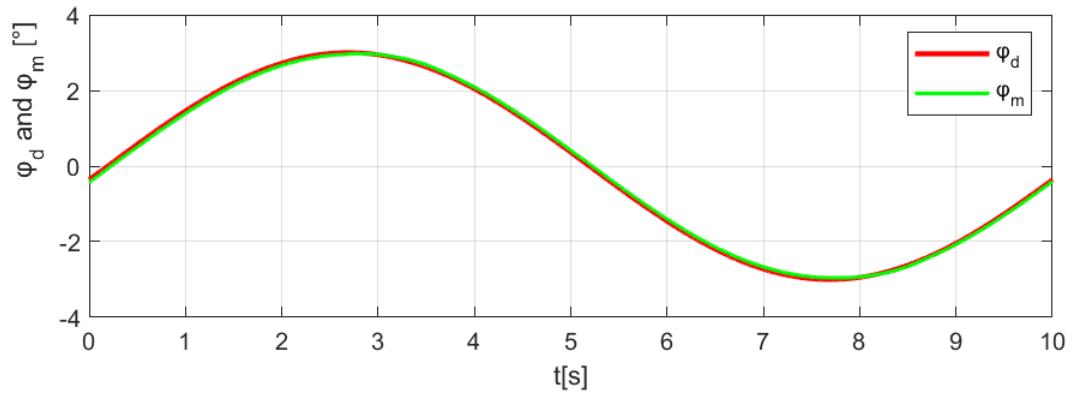


Figure 3.1:  $\varphi_d$  and  $\varphi_m$  at 3rd Iteration.

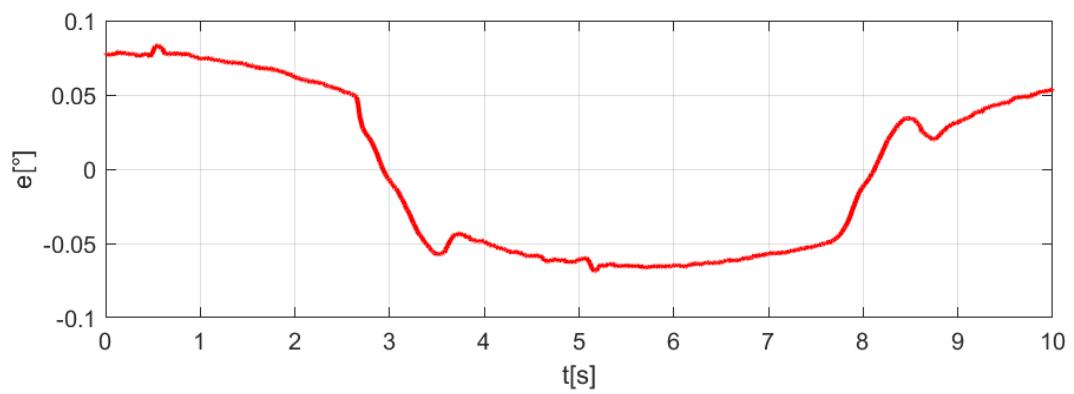


Figure 3.2: measured error at 3rd Iteration  $e_{\max} = 0.1081^\circ$ ,  $e_{\text{rms}} = 0.07158^\circ$ .

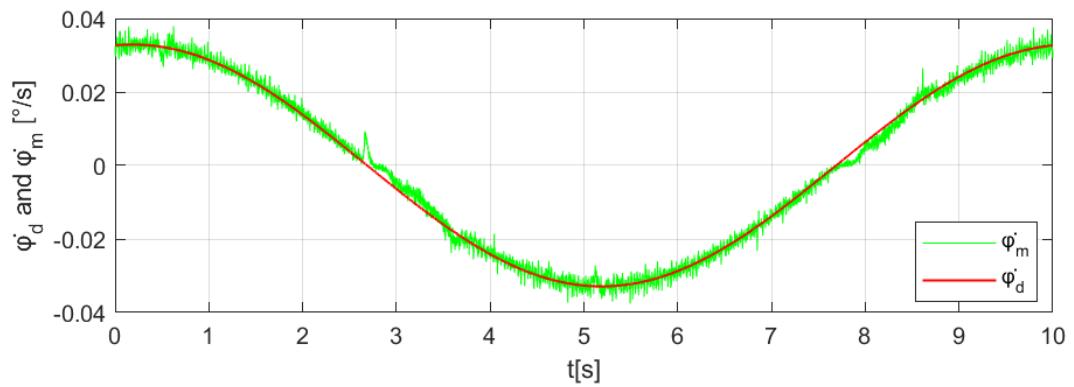


Figure 3.3:  $\dot{\varphi}_d$  and  $\dot{\varphi}_m$  at 3rd Iteration.

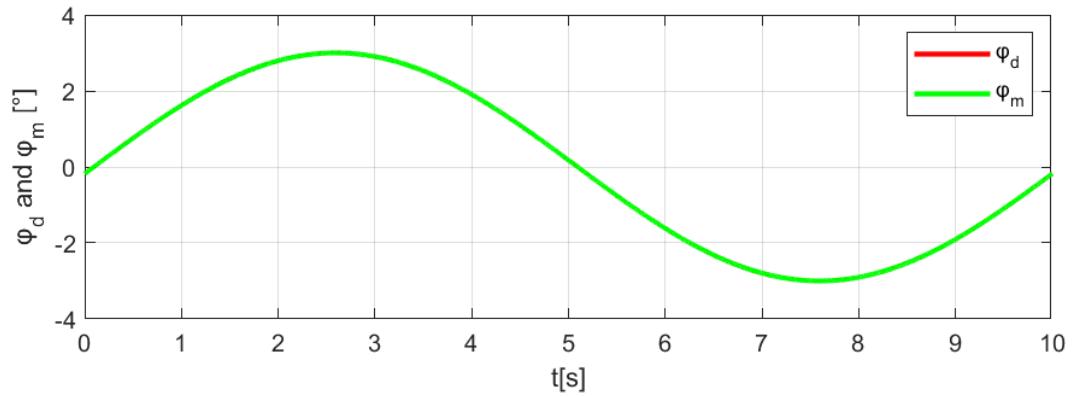


Figure 3.4:  $\varphi_d$  and  $\varphi_m$  at 33rd Iteration.

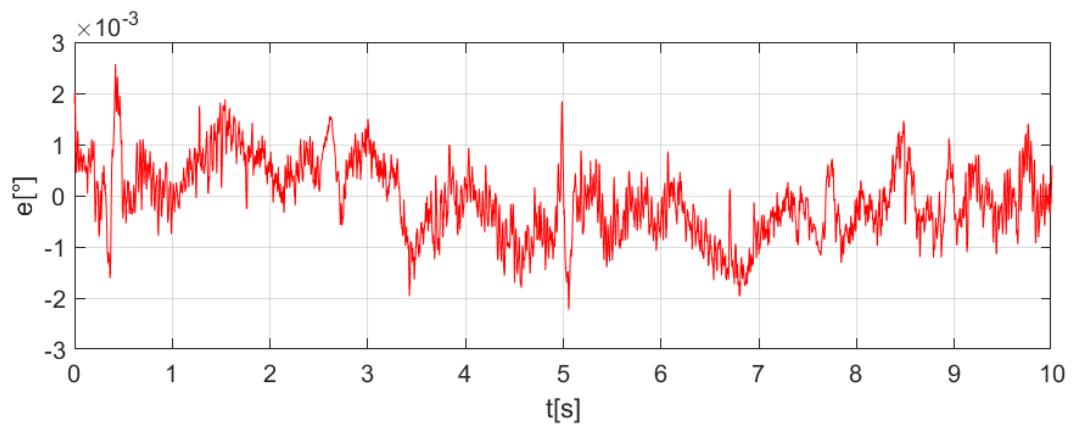


Figure 3.5: measured error at 33rd Iteration  $e_{\max} = 2.866 \cdot 10^{-3}$   $^{\circ}$ ,  $e_{\text{rms}} = 7.187 \cdot 10^{-4}$   $^{\circ}$ .

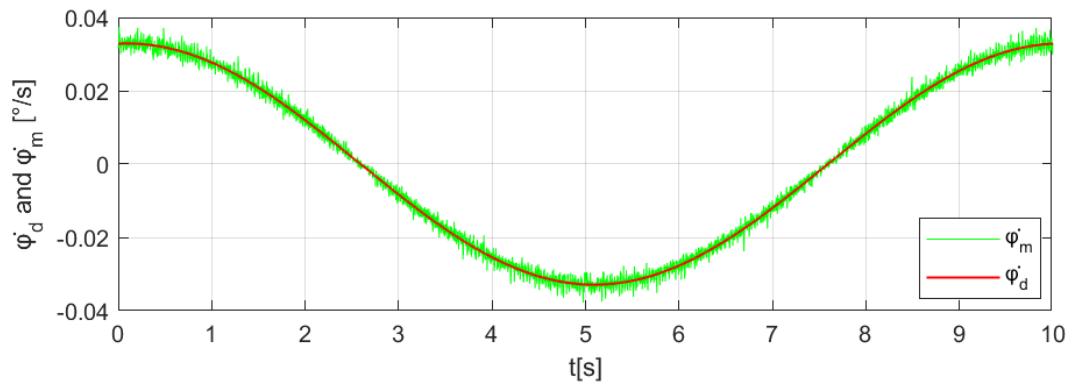


Figure 3.6:  $\dot{\varphi}_d$  and  $\dot{\varphi}_m$  at 33rd Iteration.

Figure 3.1 and Figure 3.4 show the  $\varphi_d$  and  $\varphi_m$  on 3rd Iteration and 33rd Iteration respectively. The red line represent  $\varphi_d$  and the green line represent the  $\varphi_m$ . It is noticeable that the  $\varphi_m$  in 33rd iteration follows the planned trajectory perfectly compared to the one measured in 3rd iteration.

Figure 3.2 and Figure 3.5 depict the measured error in degrees for the 3rd Iteration and 33rd Iteration, respectively. It is evident that the error in the 33rd iteration is smaller compared to the error observed in the 3rd Iteration. This indicates that as the number of iterations increases, the system's performance improves, resulting in a reduction in tracking errors. The smaller error in the 33rd iteration signifies the effectiveness of the ILC algorithm in learning from previous iterations and refining the control input to minimize deviations from the desired trajectory. This observation further supports the conclusion that the ILC approach enhances the system's tracking accuracy and demonstrates its potential for achieving improved performance.

Figure 3.3 and Figure 3.6 shows the  $\dot{\varphi}_d$  and  $\dot{\varphi}_m$  on 3rd Iteration and 33rd Iteration respectively. The red lines represent  $\dot{\varphi}_d$  and the green line represent the  $\dot{\varphi}_m$ . It illustrates that the  $\dot{\varphi}_m$  in 33rd iteration follows closer to the planned trajectory compared to the one measured in 3rd iteration.

Table 3.2: Low Amplitude Test Result.

<b>Error \ Iteration</b>	<b>1st Iteration</b>	<b>3rd Iteration</b>	<b>33rd Iteration</b>
$e_{\max} [^\circ]$	0.2249	0.1081	$2.866 \cdot 10^{-3}$
$e_{\text{rms}} [^\circ]$	0.1121	0.07158	$7.187 \cdot 10^{-4}$

## 3.2. High Amplitude with Slow Trajectory Testing

In Appendix A, the measured results of the test with the planned trajectory (desired angle),  $\varphi_d = 30^\circ \sin(0.2\pi t)$ , are presented. It was observed that the better results were achieved, as expected, with the ILC approach. This test is to test the robustness of our controllers.

## 4. Discussion

In this chapter, the results shown in the previous chapter will be further analysed and discussed. Based on the previous chapter, it is then found that the results with implementation has become better with the increase of iteration. This proves that the ILC concept works perfectly in this slow sinus repetitive trajectory control.

The results of both tests indicate a significant improvement in the measured error when utilizing the ILC algorithm. Notably, the ILC algorithm demonstrates effective performance within the range of  $\pm 3^\circ$  to  $\pm 30^\circ$ , even with constant controller parameters. This serves as evidence of the robustness of the ILC control. The observed improvements in the measured error validate the capability of the ILC algorithm to effectively track the desired trajectory or reference signal. The fact that the algorithm consistently performs well within the specified amplitude range, while keeping the controller parameters constant, further supports its robustness.

The specific amplitude range mentioned  $\pm 3^\circ$  to  $\pm 30^\circ$  may vary depending on the specific system and context. However, based on the test results, it can be concluded that the ILC is robust within the tested parameters. The test results complies to the concept of Iterative Learning Control (ILC) involves using feedback information from previous iterations to improve the performance of a system over multiple iterations as the equation shown at (1.1).

## 5. Conclusion

In this chapter , the conclusion of the overall testing will be concluded. Based on the results obtained from the experiments, it can be concluded that the iterative learning control (ILC) algorithm significantly improved the performance of the system. As the number of iterations increased, the measured signals, including  $\varphi_m$  and  $\dot{\varphi}_m$ , closely followed the planned trajectory signals. This indicates that the ILC algorithm effectively learned from previous iterations and adjusted the control input to minimize tracking errors. The system demonstrated enhanced tracking accuracy and better alignment with the desired trajectory, showcasing the effectiveness of the ILC approach in achieving improved system performance.

Furthermore, the observed reduction in errors between the measured and planned trajectory signals as the number of iterations increased highlights the adaptability and robustness of the ILC algorithm. By iteratively adjusting the control input based on the observed errors, the system effectively compensated for uncertainties, disturbances, and modeling errors. This ability to adapt and improve performance over time is a valuable characteristic of the ILC approach. However, it is important to consider factors such as convergence speed, computational complexity, and sensitivity to initial conditions when implementing ILC in practical applications. Overall, the results demonstrate the potential of ILC in enhancing tracking performance and provide a foundation for further research and optimization in real-world scenarios.

## **Appendix A.**

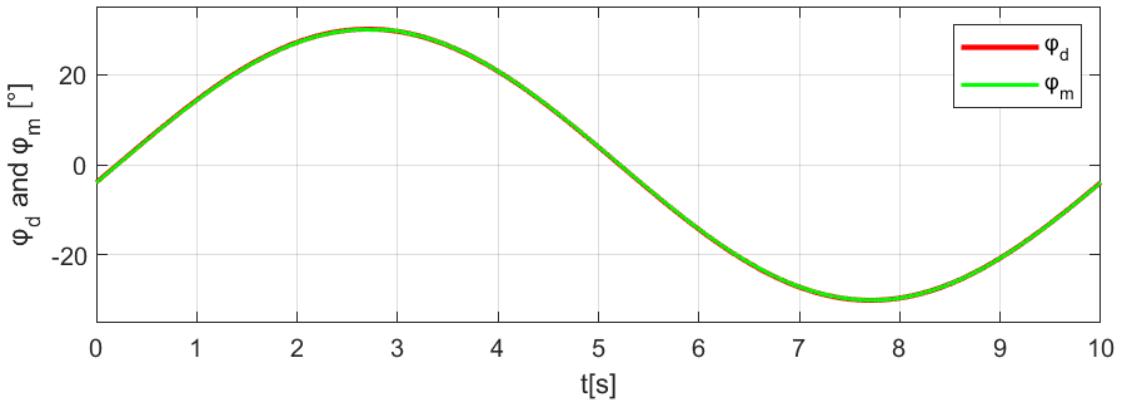


Figure A.1:  $\varphi_d$  and  $\varphi_m$  at 3rd Iteration.

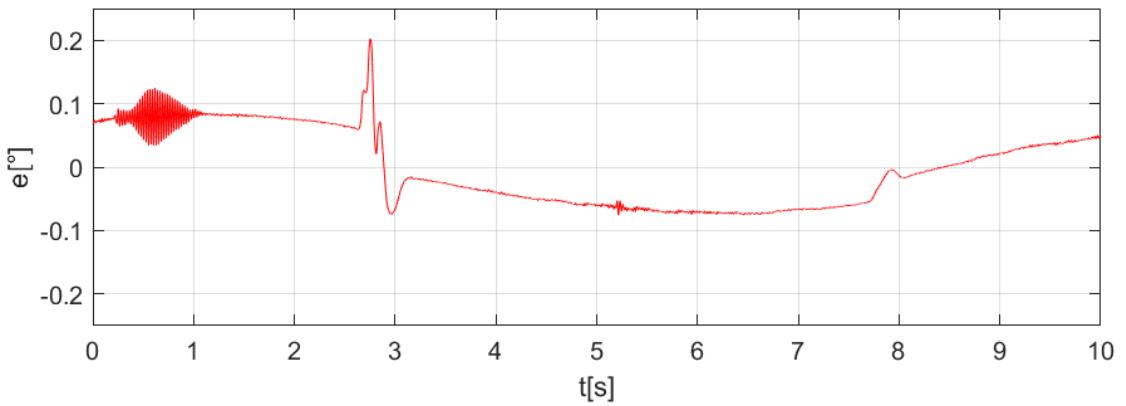


Figure A.2: Measured error at 3rd Iteration  $e_{\max} = 0.3680^\circ$ ,  $e_{\text{rms}} = 0.07636^\circ$ .

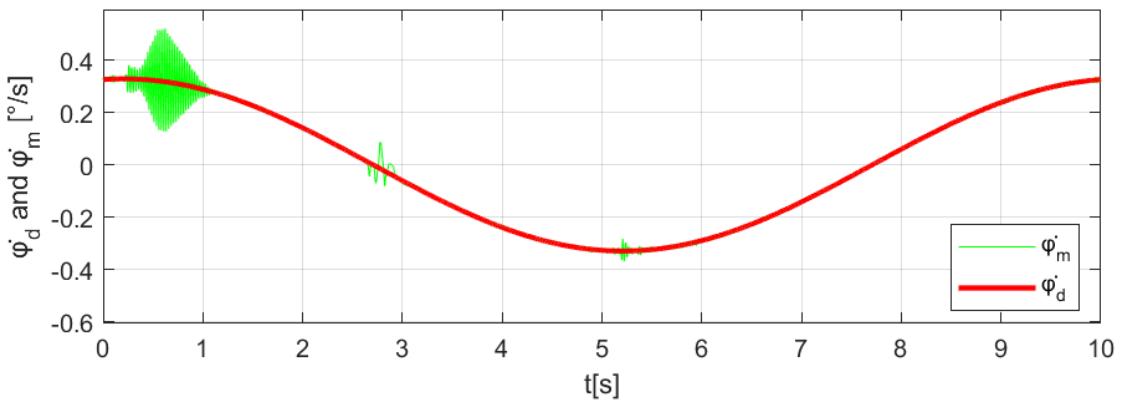


Figure A.3:  $\dot{\varphi}_d$  and  $\dot{\varphi}_m$  at 3rd Iteration.

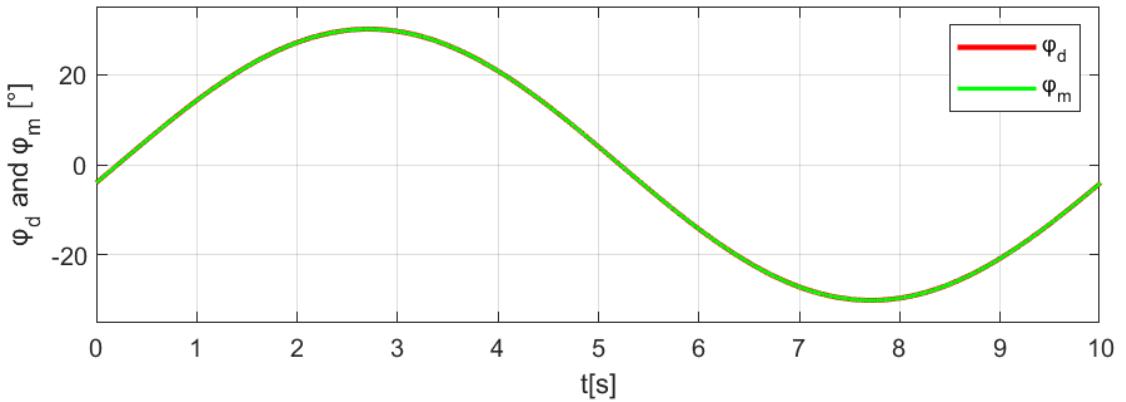


Figure A.4:  $\varphi_d$  and  $\varphi_m$  at 33rd Iteration.

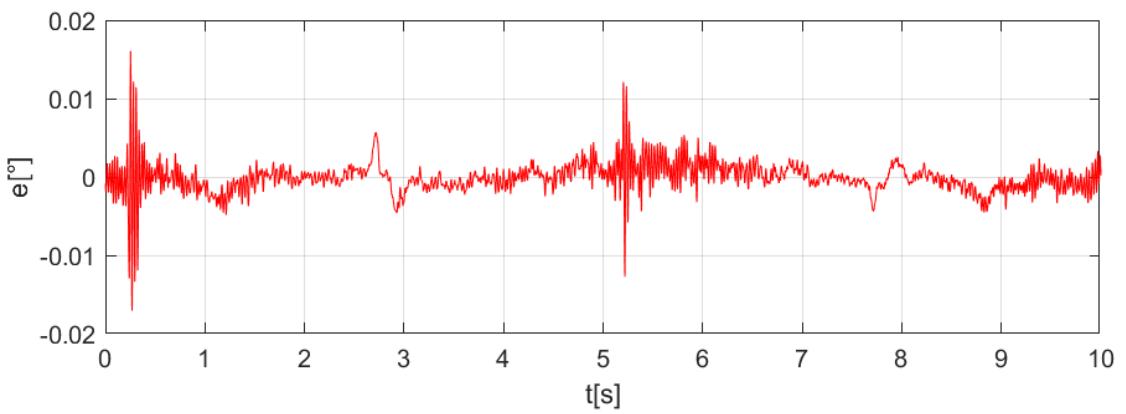


Figure A.5: measured error at 33rd Iteration  $e_{\max} = 0.01708^\circ$ ,  $e_{\text{rms}} = 2.025 \cdot 10^{-3}^\circ$ .

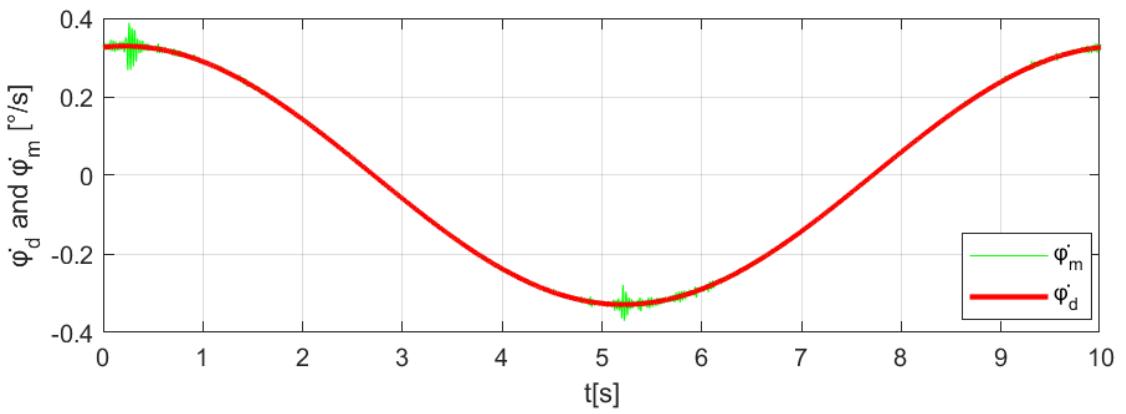


Figure A.6:  $\dot{\varphi}_d$  and  $\dot{\varphi}_m$  at 33rd Iteration.

Table A.1: High Amplitude Test Result.

Error	Iteration	1st Iteration	3rd Iteration	33rd Iteration
	$e_{\max}$ [°]	0.3013	0.3680	0.01708
$e_{\text{rms}}$ [°]	0.1434	0.07636	$2.025 \cdot 10^{-3}$	

# References

- [1] Daniel Bergmann. Design and Experimental Validation of a Robust Sliding Mode Position Control for a pneumatic Robot Joint. Bachelor thesis, Universität Stuttgart, 2021.
- [2] Adler Fonseca de Castro and Leonardo Antônio Borges Tôrres. Iterative learning control applied to a recently proposed mechanical ventilator topology. *IFAC-PapersOnLine*, 52(1):154–159, 2019.
- [3] Timo Heubach. Entwurf und Erprobung einer Regelung für ein pneumatisches Robotergelenk für sich wiederholende Bewegungen. Bachelor thesis, Hochschule Aalen, 2023.
- [4] Richard W. Longman. Iterative learning control and repetitive control for engineering practice. *International Journal of Control*, 73(10):930–954, 2000.
- [5] Tingting Meng and Wei He. Iterative learning control of a robotic arm experiment platform with input constraint. *IEEE Transactions on Industrial Electronics*, 65(1):664–672, 2018.
- [6] Rainer Nitsche and Christian Dieterich. Components Subset for Simulink. Process Automation Standard Library (PASL) - Documentation. Technical report, Festo AG & Co. KG , Dept. BA-DEE, 2022.
- [7] Samer Said Saab, Dong Shen, Mohamad Orabi, David Kors, and Rayana H. Jaafar. Iterative learning control: Practical implementation and automation. *IEEE Transactions on Industrial Electronics*, 69(2):1858–1866, 2022.
- [8] Shruti Sharma. Model-Based Design and Control of a Pneumatic Robot Joint. Bachelor thesis, Birla Institute of Technology and Science, Pilani, 2019.