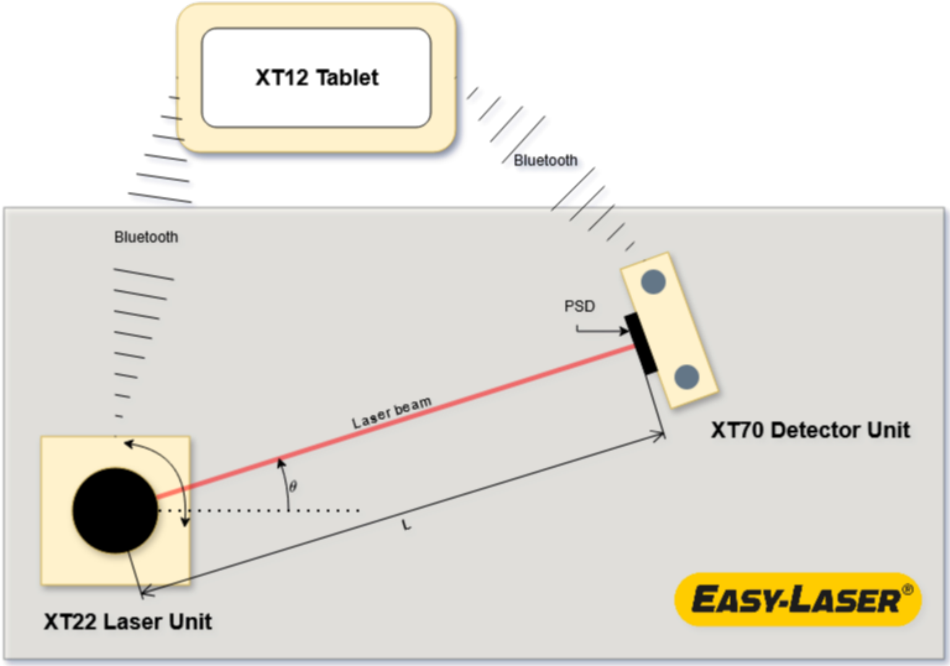
CHALMERS
UNIVERSITY OF TECHNOLOGY



Assessment of Different Control Strategies for Precision Laser Alignment and Distance Estimation

A Case Study in Precision Control for Laser Alignment at Easy-Laser AB

Master's thesis in Systems, Control and Mechatronics

PHILIP FREDRIKSSON & ALEXANDER NORDEMARK

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Assessment of Different Control Strategies for Precision Laser Alignment and Distance Estimation

A Case Study in Precision Control for Laser Alignment at Easy-Laser AB

Philip Fredriksson
Alexander Nordemark

Department of Electrical Engineering
Division of Systems and Control
Automation Unit
Chalmers University of Technology
Gothenburg, Sweden 2025

Assessment of Different Control Strategies for Precision Laser Alignment and Distance Estimation
A Case Study in Precision Control for Laser Alignment at Easy-Laser AB
Philip Fredriksson
Alexander Nordemark

Supervisor: Melina Åkesson, Easy-Laser AB
Examiner: Knut Åkesson, Automation at the Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Division of Automation
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Overview of the system.

Typeset in LaTeX
Gothenburg, Sweden 2025

Abstract

This thesis investigates and evaluates different control strategies for automating the alignment of laser beams in high-precision measurement systems, specifically within Easy-Laser's XT20 platform. The aim is to replace the manual adjustment process with an automated system capable of aligning a laser beam with a Position Sensitive Device (PSD) and estimating the relative distance between the transmitter and detector. A comprehensive system model incorporating a brushed DC motor and its mechanical interface is developed. Both physical modeling and system identification techniques are employed, followed by the design and implementation of two control algorithms: cascade PID and Linear Quadratic Integrator (LQI). The control strategies are compared in terms of response time, overshoot, steady-state error, and robustness. To improve real-world performance, the controllers are further enhanced with signal filtering and a feedforward friction compensator. Experimental results demonstrate that the LQI controller with feedforward compensation performs best, achieving sub-2 second alignment times. The thesis also explores methods for detector localization and distance estimation using the PSD; however, full localization was not achieved. The final solution significantly improves measurement speed and alignment accuracy, enabling automated laser beam positioning within 30 seconds—matching the performance of a skilled human operator, while highlighting key areas for future development.

Contents

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

ESF	Error Shaping Function
LQ	Linear Quadratic
LQI	Linear Quadratic Integrator
LQR	Linear Quadratic Regulator
PID	Proportional Integral Derivative controller
PSD	Position Sensitive Device

Nomenclature

Below is the nomenclature of parameters and variables that have been used throughout this thesis.

Electrical Variables

V_T	Terminal voltage (input voltage to the motor)
i_a	Armature current
R_a	Armature resistance
L	Armature inductance
e_a	Back EMF (electromotive force)
K_T	Torque constant
K_ω	Back EMF constant

Mechanical Variables

T_e	Electromagnetic torque
ω	Angular velocity of the motor
b	Viscous damping coefficient
J	Rotational inertia

Laser Head and Gear Variables

ω_M	Angular velocity of the motor shaft
ω_H	Angular velocity of the laser head
J_M	Motor inertia
J_G	Gear inertia
J_H	Laser head inertia
b_M	Motor damping coefficient
b_G	Gear damping coefficient
b_H	Laser head damping coefficient
n_{PG}	Gear ratio of the planetary gearbox
n_H	Gear ratio to the laser head

State-Space Variables

x	State vector
$x_1 = i$	Current
$x_2 = \theta_H$	Angular position of the laser head
$x_3 = \dot{\theta}_H$	Angular velocity of the laser head
$u = V_T$	Control input (applied voltage)
y	Output vector
$y_1 = i$	Current
$y_2 = \theta_H$	Angular position

State-Space System Matrices

A	System matrix
B	Input matrix
C	Output matrix
D	Direct transmission matrix

Derived Parameters

α	$J_M n_{PG} n_H + J_G n_H + \dfrac{J_H}{n_{PG} n_H}$
β	$b_M n_{PG} n_H + b_G n_H + \dfrac{b_H}{n_{PG} n_H}$

List of Figures

List of Tables

1

Introduction

Bad alignment of equipment is one common cause of machine failures, significantly impacting the environment and interrupting production when repairs or replacements are needed [1]. According to [1], correct laser alignment could reduce energy consumption with 3-10 %, guarantee a machine's longevity and prevent production issues. Easy-Laser supplies the industry with state-of-the-art measuring devices and, as part of this thesis project, aims to enhance its current flatness measurement solution by automating the process of aligning the laser beam with the position sensitive device (PSD).

Figure 1.1: Easy-Laser's product range used for various alignments. XT22 is used for flatness and XT12 provides user-interface functionality.

1.1 Background

Alignment of machinery in industries is of critical importance. The rising demand for high-quality products and the long lifespan of machinery necessitate precise alignment and measurement.

The current system in Figure 1.3 is used for measuring the flatness of a planar surface up to a radius of 40m. Easy-Laser aims to switch from aligning the laser beam manually [2] with the finetuning knob in Figure 1.2. The objective is to automate Easy-Lasers' existing laser-based measuring system Easy-Laser® XT22 in Figure 1.1.

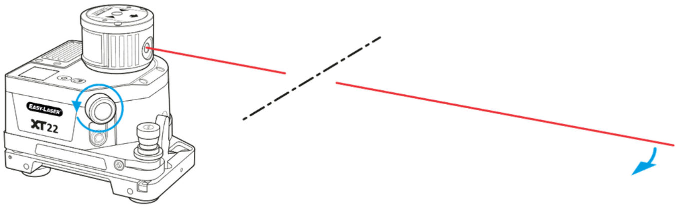


Figure 1.2: Existing angle fine-tuning adjustment knob.

Easy-Laser has identified the main challenge and time-consuming task of their existing system as the manual process of aligning the laser beam with the detector [2]. They plan to solve this by fitting their existing XT20 laser transmitter with an actuator that can automatically and reliably align the laser beam with the position sensitive device (PSD) on the laser detector, thus improving flatness measurement in industrial applications.

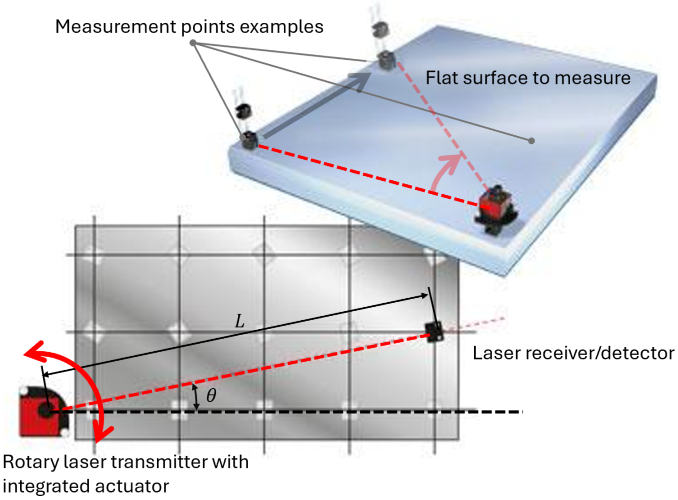


Figure 1.3: Illustration of how the system should be used to measure flatness using the current XT22 hardware. Prone to human errors, the angle θ is currently set manually by rotating the laser beam.

1.2 Aim

The objective of this project is to automatically localize the laser detector and align the laser beam with the Position Sensitive Device (PSD), as illustrated in Figure 1.3. This will be accomplished using appropriate control algorithms. The system is also expected to calculate the relative distance and angular position of the detector unit by using the existing hardware and sensor configuration.

The aim of this thesis is to develop a reliable and high-performing control system that minimizes alignment time, misalignment, and measurement sampling time. As mentioned in 1.1, this will play a critical role in improving both the efficiency and user experience of the company's laser-based measurement systems. The primary focus is to ensure precise alignment of the laser beam with the PSD, while the secondary goal is to calculate the distance between the transmitter and the PSD.

The system should perform as well as a human operator, that is localizing the detector unit, positioning the laser beam on the detector and take a measurement sample no longer than 30[s].

1.3 Limitations, Boundaries and Assumptions

This thesis assumes that the PSD is stationary and oriented orthogonally to the laser beam. This implies that the detector unit is positioned tangentially to the laser's scanning circle, as illustrated in figure 1.4. The development of tracking algorithms for the detector after the initial detection event is outside the scope of this work and will not be considered.

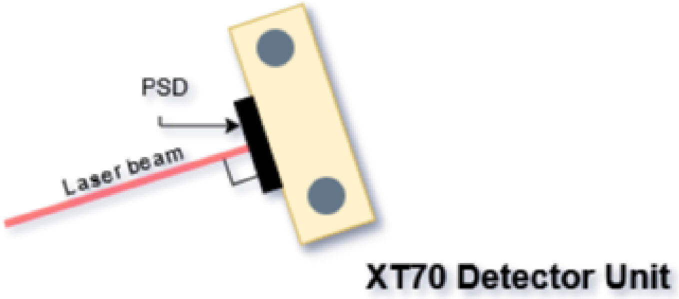


Figure 1.4: The PSD element is assumed to be perfectly orthogonal with the laser beam. This is not the case in the real life application.

The objective will be accomplished utilizing existing sensors and hardware provided by the host company. The sensors that will be used includes a rotary encoder [3], a PSD on the detector unit, and a current sensor for the brushed DC-motor. No other sensor will be implemented in this thesis.

The logging function of the hardware limits the sampling frequency to $100[Hz]$ limiting the possible analysis, since the observation does not include all samples.

1.4 Research Questions

The research questions outlined below summarize the objectives that this thesis seeks to address and resolve.

RQ1. How do the performances of a cascade PID controller and a Linear Quadratic controller compare in positioning the laser beam to a given reference angle?

RQ2. How can the Position Sensitive Device (PSD) on the detector unit be utilized to estimate the relative angle and distance between the detector and the laser transmitter?

Research Question 1 (RQ1) 1.4 seeks to identify the different control strategies for aligning the laser beam within $\pm 2\ mm$ of the Position Sensitive Device (PSD) center up to $40[m]$ which corresponds to an error of $0.05[mm/m]$. To achieve this objective, a mathematical model will be developed to simulate various control strategies. An experimental setup will be designed to evaluate these strategies, where the controller will adjust the motor position to achieve a specified alignment angle.

Research Question 2 (RQ2) 1.4 investigates various methods for estimating the relative angle θ and the distance L. The focus is on comparing the accuracy of angle and distance measurement while also considering computational complexity. RQ2 also explores laser frequency modulation as a method to localize the center of the Position Sensitive Device (PSD). A quantative approach will be used to evaluate the proposed methods outlined in the following.

M1. Modulate the laser frequency based on θ and estimate the laser frequency in the detector unit. Convert the estimated frequency back to the reference angle θ.

M2. Instruct the detector module to communicate with the laser transmitter to log both the initiation and termination angles of the laser beam when it impacts the PSD.

2

System Overview and Requirements

This chapter provides a comprehensive overview of the system architecture and outlines the key functional and performance requirements. It begins by describing the main components and their roles within the system, followed by an explanation of the calibration and measurement procedures. Additionally, the chapter defines the constraints imposed by the existing setup and explains how the improvements proposed in this thesis contribute to achieving its overall objectives. The goal is to establish a clear foundation for the design, modeling, and implementation efforts presented in the subsequent chapters.

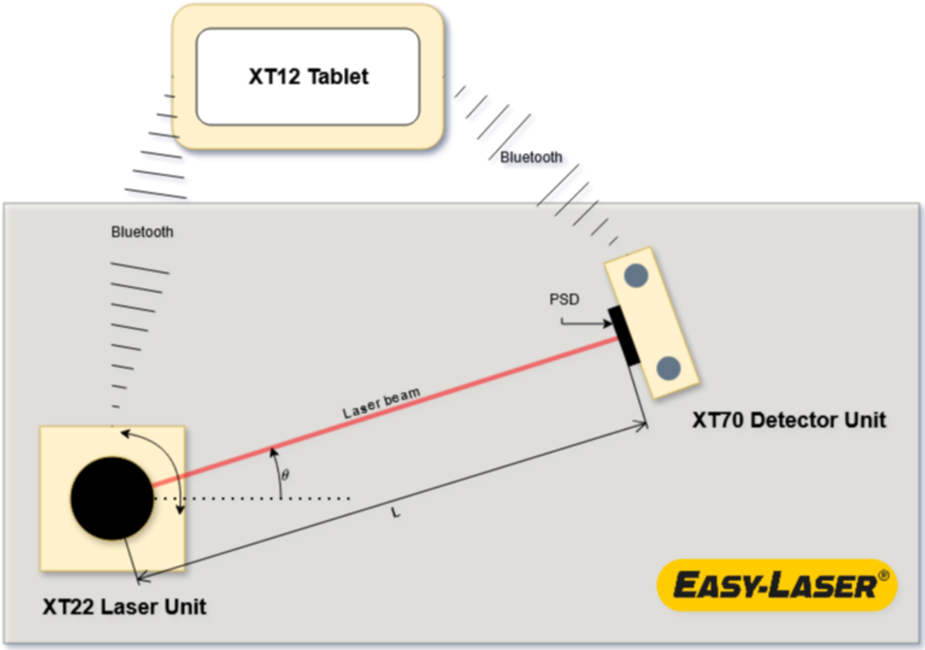


Figure 2.1: Overview of the system showing angle θ to be controlled and distance L to estimate.

The system is essentially a measuring tool used to ensure flatness of a planar surface. It could be an industrial floor on which to install a large machine like a press, or a coordinate table for a CNC-machine or the flange of a wind turbine. The applications are endless and the need to speed up this process is a major challenge. The flatness measuring process consists of two steps, configuration and measuring, [4], simplified in Figure 2.2.

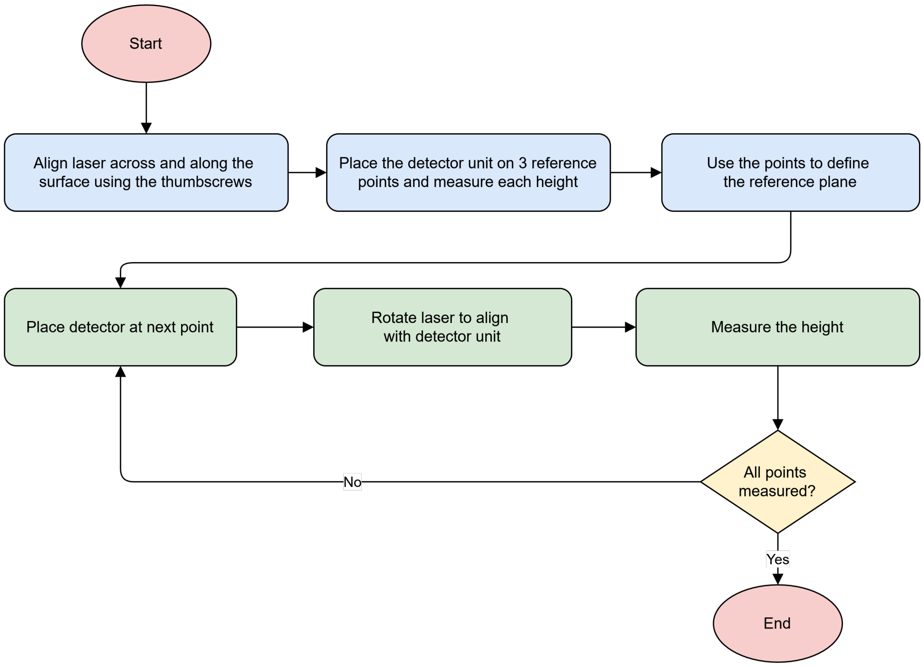


Figure 2.2: Diagram showing the calibration and measurement steps.

A typical flatness assessment involves numerous measurement points, often ranging from 20 to 40 or more. Conducting these measurements is a time-consuming process that usually requires two operators: one managing the laser transmitter and the other moving the detector unit between points. As a result, substantial efficiency gains can be achieved by automating or speeding up the measuring procedure.

The central challenge lies in accurately determining the detector unit's position without making extensive modifications to the existing hardware, beyond integrating DC-motor control 1.4 and processing the Position Sensitive Detector (PSD) data 1.4. As documented in [5], a Position Sensitive Detector (PSD) is a type of sensor that is capable of measuring positions in either one or two dimensions on a designated surface. The PSD can be configured in various configurations, one of which includes continuous transmission of positional data alongside estimating the modulation frequency of the laser at both the leading and trailing boundaries of the PSD element when the laser

Figure 2.3: Picture shows the deviation of the measured coordinate table with respect to the calibrated reference plane.

sweeps past it. Additionally, it is feasible to establish a detection trigger by utilizing the sampled outputs from the PSD.

3

System Modeling

A model of the system is required for two primary purposes: (1) to simulate the system's behavior under various scenarios, and (2) to utilize model-based optimal control. According to [6], Chapter 8.1, a model need not—and cannot—be exact; feedback control is forgiving if the stability margins are sufficiently large.

The system model comprises a brushed DC motor, a planetary gearbox, and a helical gear that transmits motion from the planetary gearbox output to the laser head. The model incorporates both the electrical dynamics of the brushed DC motor and the mechanical dynamics of the motor and gear system. As illustrated in Figure 3.1, the electrical subsystem of the motor is coupled to the mechanical components, including the planetary gearbox, and helical gear.

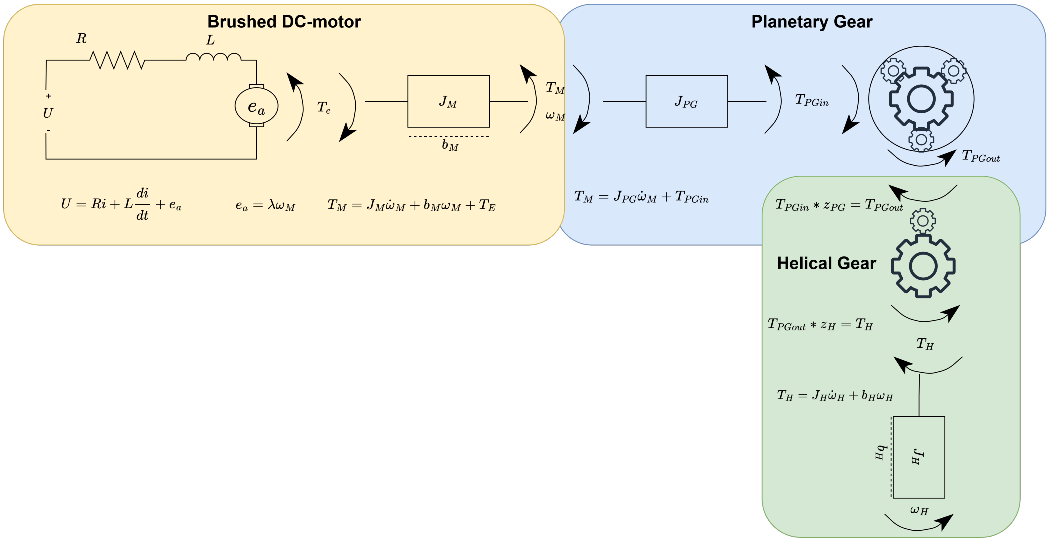


Figure 3.1: Mathematical model of drive system including subsystems.

Since the controllers in the system will be implemented in discrete time while the model is formulated in continuous time, a standard continuous-time analysis cannot be applied directly. According to [6], there are two methods to address this issue. The first approach involves designing both the controller and the plant in continuous time and relying on a sufficiently high sampling frequency—assuming the hardware can support it. The second approach is to discretize the continuous-time model. As noted by [6], for a linear system with piecewise constant inputs between samples, a corresponding linear discrete-time model can be derived. This method properly captures the effects of time discretization. However, it only describes the system behavior at discrete sampling instances, not in between them. Since the control system in this thesis is executed on a microcontroller with limited sampling capabilities, the first approach may not provide sufficient accuracy. Therefore, the discretization approach is chosen to more accurately account for the effects of sampling in the control design.

To accurately model the system's transient and steady-state behavior, several parameters must be either measured or estimated. This challenge was addressed using instruments such as oscilloscopes, multimeters, and a DC power supply. The parameters are listed in table 3.1.

Table 3.1: System parameter values

Parameter	**Value**	**Unit**	**Method**
R_a	2.616	Ω	LCR-meter
L	0.0336	mH	LCR-meter
K_T	1.96	mNm/A	Datasheet
J_M	0.288e-7	Kg $\cdot m^2$	Datasheet
b_M	4.5e-8	-	Provided by the company
J_{PG}	0.015e-7	Kg $\cdot m^2$	CAD drawing
b_{PG}	2e-5	-	Provided by the company
n_{PG}	185193/2744	-	Datasheet
J_G	85.623e-9	Kg $\cdot m^2$	CAD drawing
b_G	13e-2	-	Provided by the company
n_G	132/30	-	Provided by the company

3.1 Physical Modeling

There are two approaches to model a system according to [7]: physical modeling and system identification. Physical modeling relies on known physical principles and parameters to describe the system, while system identification uses observed input-output data to fit a model that captures the system's behavior. Chapter 3.1 focuses on the physical modeling approach, whereas Chapter 3.3 addresses the identification method.

The physical model of the system can be derived by combining the electrical and mechanical equations of a brushed DC motor. This approach is outlined in [8], where the system is described using the differential equations shown in Equations 3.1–3.4:

$$V_T = R_a i_a + L\frac{di_a}{dt} + e_a \tag{3.1}$$

$$T_e = K_T i_a \tag{3.2}$$

$$e_a = K_\omega \omega \tag{3.3}$$

$$T_e = b\omega + J\frac{d\omega}{dt} \tag{3.4}$$

Figure 3.2 provides an overview of the brushed DC motor, while Figure 3.3 illustrates the implementation of the equations in a block diagram. By applying block diagram manipulation, the transfer function of the brushed DC motor—from the input voltage V_T to the angular velocity ω_M can be derived, as shown in equation 3.5.

$$\frac{\omega}{V_T} = \frac{K_T}{JLs^2 + (JR + Lb)s + K_T K_\omega} \tag{3.5}$$

Using torquebalance of the system yields the following dynamics, relating T_e to with angular velocity for the motor.

$$T_e = \dot{\omega}_M \left(J_M + \frac{J_G}{n_{PG}} + \frac{J_H}{n_{PG}^2 n_H^2} \right) + \omega_M \left(b_M + \frac{b_G}{n_{PG}} + \frac{b_H}{n_{PG}^2 n_H^2} \right) \tag{3.6}$$

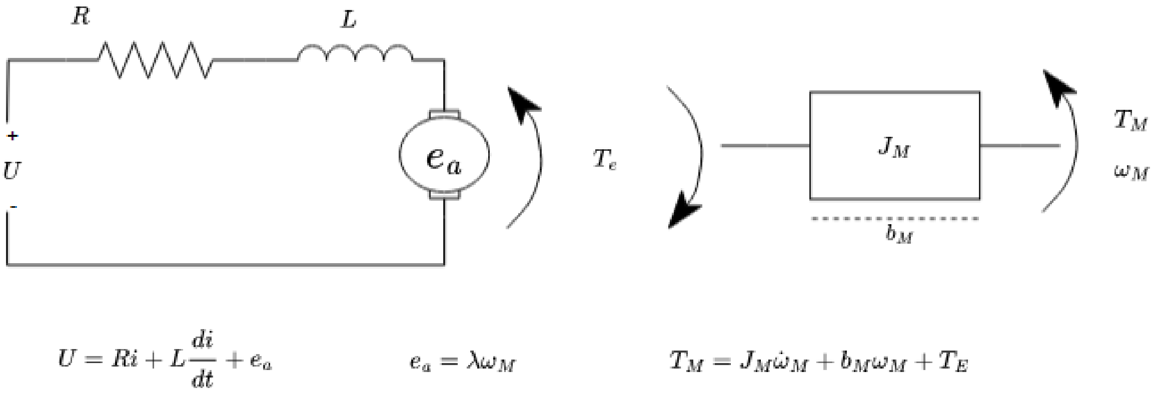


Figure 3.2: Electric and mechanical model of the DC-motor.

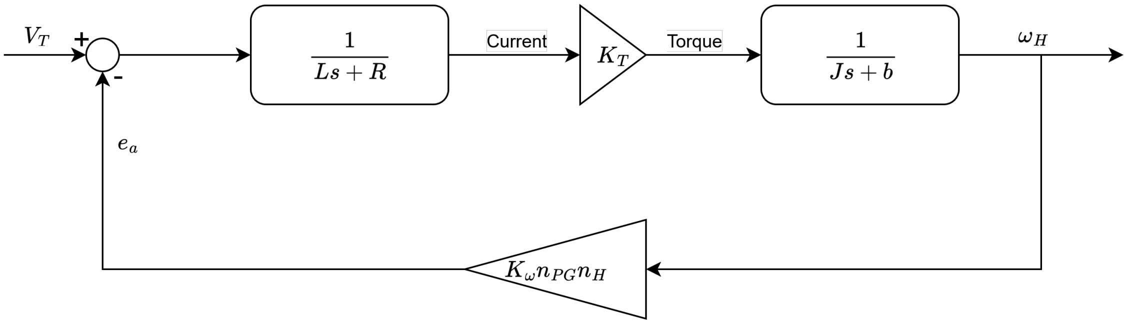


Figure 3.3: Block diagram for DC motor.

Transfer function:

$$\frac{\omega_M}{T_e} = \frac{1}{s\left(J_M + \frac{J_G}{n_{PG}} + \frac{J_H}{n_{PG}^2 n_H^2}\right) + \left(b_M + \frac{b_G}{n_{PG}} + \frac{b_H}{n_{PG}^2 n_H^2}\right)} \tag{3.7}$$

Or from electromagnetic torque to angular velocity of the laser head.

$$\frac{\omega_H}{T_e} = \frac{1}{s\left(J_M n_{PG} n_H + J_G n_H + \frac{J_H}{n_{PG} n_H}\right) + \left(b_M n_{PG} n_H + b_G n_H + \frac{b_H}{n_{PG} n_H}\right)} \tag{3.8}$$

With both the electrical and mechanical transfer functions established, a closed-loop system can now be designed. As outlined in 1.4, three different control systems will be designed, all of which will be implemented in discrete time.

The discretization method employed in this thesis is the Zero-Order Hold (ZOH). According to [9], ZOH is a straightforward discretization technique that accounts for the dead time between samples. In practice, this means that the system retains the sampled value for a brief period before acquiring a new sample.

3.2 Friction Modeling

The friction within the system was approximated through an empirical evaluation using both current and angle measurements. Other advanced methods was investigated based on a friction observer approach, as outlined by [10]. Although several methodologies exist for modeling friction, a linear friction model was preferred to simplify the analysis and design of the control system. For modeling friction at low velocities, the LuGre and Elasto-Plastic models offer excellent capabilities [11]. This will discussed further in chapter 4.2.5

3.3 Model Identification

From chapter 3.1, the physical model of the system is complex. This raises the question of whether a simpler modeling approach could be used. According to [12], a brushed DC motor typically behaves as an overdamped second-order system, which can be approximated by a first-order transfer function.

As outlined in [12], the angular velocity of a DC motor can be approximated using a first-order system model as seen in figure 3.4. Consequently, the motor's angular position can be represented through a second-order function, according to equation 3.9. This characterization arises from the interaction of a rapid electrical system with a comparatively slower mechanical system.

$$\frac{\omega}{V} = \frac{K}{\tau s + 1} \Leftrightarrow \frac{\theta}{V} = \frac{K}{s(\tau s + 1)} \tag{3.9}$$

According to [13], the parameters of a first-order system are defined as follows: the steady-state gain $K = lim_{t\to\infty}(y)$, and the time constant $\tau = y_{63\%}$ in response to a unit step input. Based on this approach, an experimental step response was conducted, resulting in figure 3.4. The figure shows both the actual system response and the equivalent first-order approximation derived from the identified parameters.

It is worth noting that more advanced methods exist for estimating parameters in linear models. [7] discusses black-box identification techniques such as Box-Jenkins (BJ), Output Error (OE), AutoRegressive Moving Average with eXogenous inputs (ARMAX), and AutoRegressive with eXogenous inputs (ARX) models. Once a model structure is selected, linear regression can be used to fit the model to the system's input-output data.

For simplicity, the basic first-order approximation was chosen in this thesis, and the results are presented in figure 3.4.

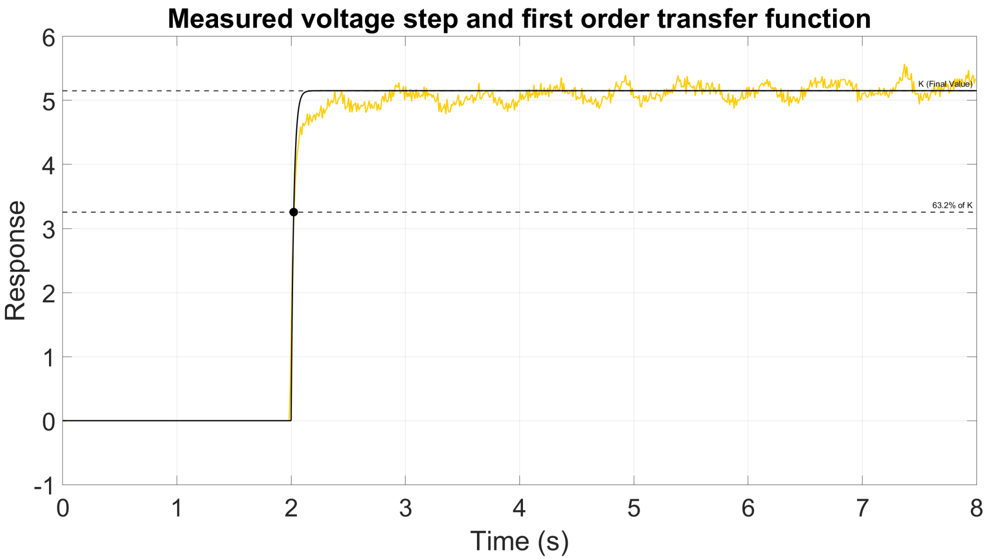


Figure 3.4: Voltage step response used for model identification.

3.4 LTI State-Space Form

The differential equations 3.1–3.4 can be reformulated into the standard state-space representation. By expressing these equations in state-space form, the resulting model is a modified version of the one presented in [6]. The key difference is that this formulation includes the angular position as a state variable, whereas the model in [6] considers only the angular velocity.

$$\dot{x} = \begin{bmatrix} \dot{i}_a & \dot{\theta}_H & \ddot{\theta}_H \end{bmatrix}^T, \quad x = \begin{bmatrix} i_a & \theta_H & \dot{\theta}_H \end{bmatrix}^T, \quad u = V_T, \quad y = [i_a \quad \theta_H]^T$$

$$A = \begin{bmatrix} \frac{-R_a}{L} & 0 & \frac{-K_\omega n_{PG} n_H}{L} \\ 0 & 0 & 1 \\ \frac{-K_T}{\alpha} & 0 & \frac{-\beta}{-\alpha} \end{bmatrix} \quad B = \begin{bmatrix} 1/L \\ 0 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad D = 0 \tag{3.10}$$

Where the constants α and β are defined as follows.

$$\alpha = J_M n_{PG} n_H + J_G n_H + \frac{J_H}{n_{PG} n_H} \tag{3.11}$$

$$\beta = b_M n_{PG} n_H + b_G n_H + \frac{b_H}{n_{PG} n_H} \tag{3.12}$$

3.5 Model Verification

To minimize the gap between the developed model and the physical system, it is essential to verify that the model accurately captures the system's key dynamics. As emphasized in [14], uncertainties in the model, such as unmodeled dynamics or parameter variations, can lead to degraded performance or even instability when the system is placed in closed-loop operation. Reliable controller tuning therefore depends critically on how well the model represents the real system.

In this context, a test rig was developed to experimentally assess the model's accuracy. By comparing simulated and measured system responses under controlled conditions, the reliability of the model for control design purposes could be evaluated.

The purpose of the test rig, shown in Figure 3.5b, was to evaluate the behavior of the real system in comparison to the simulated model. It was designed to measure the angular velocity of the helical gear, the motor current, and the applied voltage.

The experiment focused on observing the steady-state angular velocity and the corresponding motor current for various applied voltages. The voltage was incremented in steps of 0.1 V.

Motor current was measured using a multimeter connected in series with the motor. To measure angular velocity, a photo-interrupter sensor, consisting of an LED and a phototransistor was monitored using an Arduino. By counting the number of pulses generated as the gear rotated, the Arduino was able to compute the angular velocity. Given a known pulses-per-revolution (PPR) value, the angular velocity ω_{PG} was calculated using equation 3.13.

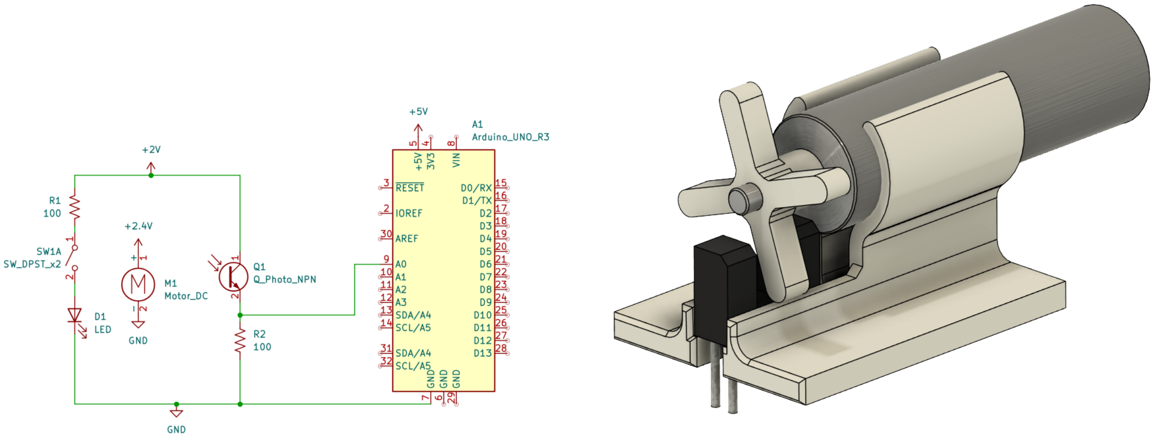
$$RPM = \frac{60 \cdot N}{PPR \cdot \Delta t} \tag{3.13}$$

where N is the pulse count within the time interval Δt. Using the measurements, the motor and gearbox parameters are separated.

The pulses that the Arduino will register is generated by the photo-interrupter, as previously mentioned, it consist of a LED and a phototransistor. The LED is connected to V_{cc} and ground, which means that it always is lit. If the LED is lit, the phototransistor will register this and start leading current, which will be registered by the Arduino as a rising edge. When the LED turns off, the phototransistor will also turn off, therefore leading no current, generating a falling edge to the Arduino. From the phototransistor's perspective, the LED will act as unlit while a spoke from the rig is passing by. This is illustrated in figure 3.5a, note that the switch in the schematic represents when a spoke from the output shaft passes through the photointerrupter. In reality, the LED remains on, and the phototransistor detects interruptions in light, generating signal transitions monitored by the Arduino.

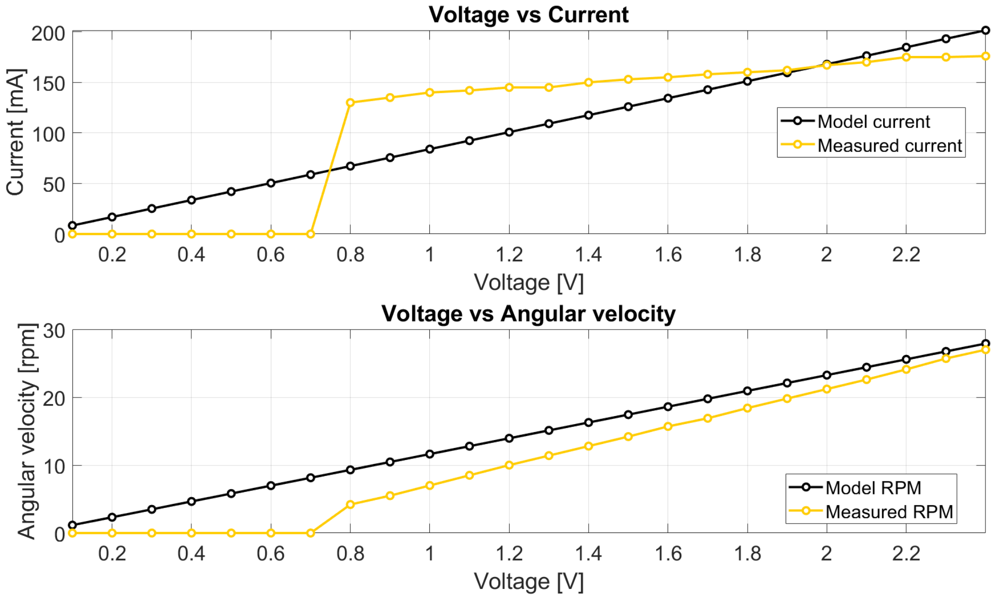
To enhance the accuracy of angular velocity estimation, the motor's output shaft is equipped with eight spokes. This approach improves measurement consistency and enables faster convergence towards the correct angular velocity. Given the rise time of the sensor [15], it is fast enough to detect even 8 spokes.

The simulation and test rig's results are presented in figure 3.5c. From the figure, it is clear that the system behaves linearly from 0.8 V and higher. This means that for relatively low voltages, the system behaves non-linearly. This will be discussed further in chapter 4.



(a) Schematic of test rig.

(b) Motor test-rig.



(c) Verification of the model showing the simulated steady-state current and angular velocity of the laser head.

Figure 3.5: Overview of the test rig and validation setup. Top: mechanical schematics of the test rig and motor holder. Bottom: model verification showing simulated laser head performance.

4

Controller Design

To enable the transmitter to accurately align the laser with the detector, a precise controller with low steady-state error and high robustness is essential. This chapter presents two types of controllers: (1) a cascade PID controller and (2) a Linear Quadratic Integrator (LQI) and explores methods to improve their performance and robustness.

4.1 Controller Requirements

The host company supplied a requirements list of the overall performance of the system. To generate the innovation in a business case the settling time is set to a maximum of 30 seconds since that is approximately the time it takes for a seasoned measurement technician to perform one measurement sample. No other requirements were imposed by the company, but from a control system perspective, the oscillations around the setpoint and overshoot/undershoot should be reduced as much as possible.

Furthermore, in this application, the input signal is typically a step function with a maximum amplitude of π, as dictated by equation 4.21. To determine the required system bandwidth, a step input with an amplitude of π was analyzed. The step signal and its corresponding Fast Fourier Transform (FFT) are shown in Figure 4.1.

As seen in 4.1, the frequency content of the step reference is concentrated at low frequencies, particularly below 10 Hz. This implies that the closed-loop system does not require a high bandwidth to accurately track such inputs.

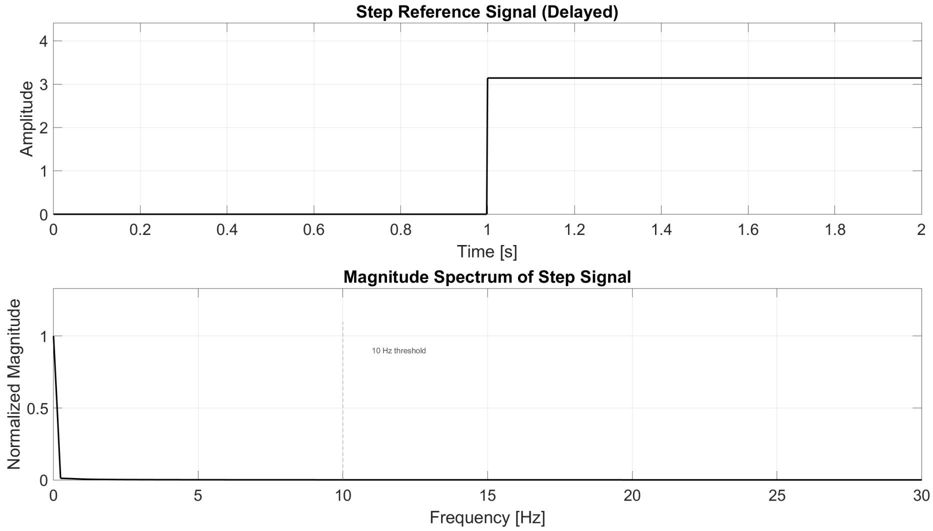


Figure 4.1: A step input and its corresponding frequency spectrum. The signal's energy is concentrated in low frequencies, with most content below 10 Hz, indicating the system does not require high bandwidth for accurate tracking.

4.2 Cascade PID

A study [16] proposes a cascade PID control scheme for controlling the angular velocity of a motor. Leveraging the fact that position is the integral of velocity, the same control principle can be extended to regulate the motor's position, and consequently, the laser head's position. Furthermore, as noted in [17], a cascade control structure offers improved robustness and enhanced disturbance rejection compared to a single-loop configuration.

Using the zero-order hold (ZOH) discretization method with a sampling frequency of 500 Hz, the cascade PID control loop, illustrated in Figure 4.2, is implemented.

4.2.1 PID Tuning

To improve the performance and robustness of the PID controller, two different tuning methods were investigated: 1) the Ziegler-nichols method and 2) integral of time-weighted absolute error (ITAE). According to [18] and [19], the ITAE method has less overshoot and faster settling and rise time. Figure 7 in [19] shows that ITAE is performing more robustly regarding parameter variation compared to Ziegler-Nichols, however, implementing ITAE is more complex compared to Ziegler-Nichols because it requires minimizing a cost function by numerical integration.

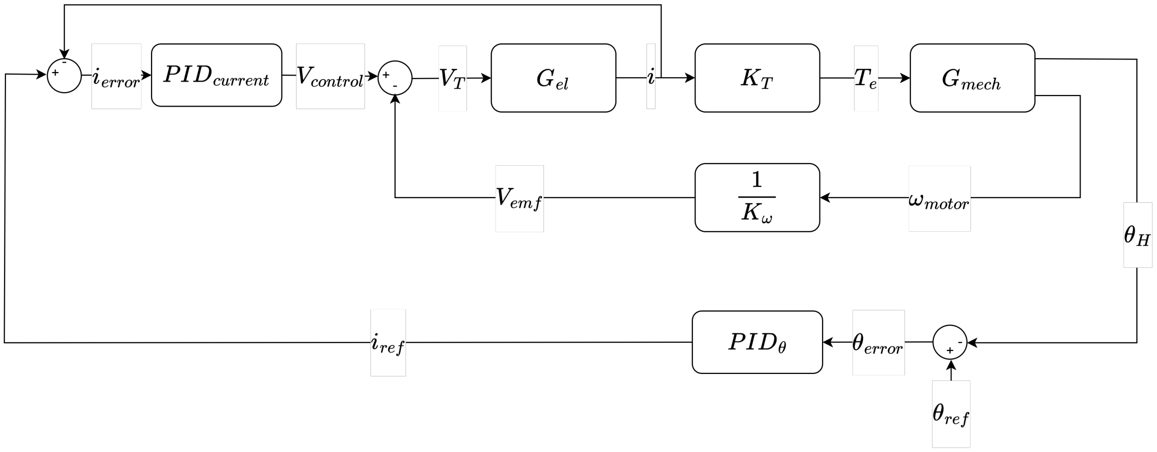


Figure 4.2: Closed loop cascade PID

Ziegler-Nichols

The cascade PID controller was tuned using inspiration from the Ziegler–Nichols method described in [9].

The final parameters are described in Table 4.1 below.

Table 4.1: Ziegler–Nichols tuned parameters for cascade PID.

Internal PID		**External PID**	
Parameter	Value	Parameter	Value
`I_P`	0.9	`THETA_P`	120.0
`I_I`	0.01	`THETA_I`	40.0
`I_D`	0.0	`THETA_D`	0.046

Integral of time-weighted absolute error

According to [19], the Integral of Time-weighted Absolute Error (ITAE) is often considered a robust performance index when designing PID controllers. This method is also employed in [18]. The ITAE criterion is defined by the following cost function:

$$ITAE = \int_0^T t|e(t)|dt \tag{4.1}$$

Minimizing this performance index typically leads to an improved transient response. The approach adopted in this thesis follows a similar methodology to that described in [18].

To compute the ITAE, Simpson's 1/3 rule — a well-known numerical integration technique — is utilized. Simpson's rule approximates a definite integral as:

$$\int_a^b f(x)dx \approx \frac{b-a}{6}(f(a)+4f(\frac{a+b}{2})+f(b)) = \frac{1}{3}h(f(a)+4f(a+h)+f(b)) \tag{4.2}$$

Where:

$$h = \frac{b-a}{n}$$

Here, a and b are two points on the function and h is the step size for $n = 2$.

This expression can be derived for $n = 2$ over the subinterval $[a, b]$. Using Lagrange polynomial interpolation, which states that a function can be approximated as:

$$P(x) = f(a)\frac{(x-a-h)(x-b)}{-h(a-b)} + f(a+h)\frac{(x-a)(x-b)}{h(a+h-b)} + f(b)\frac{(x-a)(x-a-h)}{(b-a)(b-a-h)}$$

If one were to center the integral over a step rather than directly from a to b, make the substitution $u = x - a - h \Leftrightarrow x = u + a + h$ and realize that $h = (b-a)/2$, the integral could be written as:

$$
\begin{aligned}
& f(a)\left(\int_{-h}^{h} \frac{u(u+a+h-b)}{-h(a-b)}\,du\right) + f(a+h)\left(\int_{-h}^{h} \frac{(u+h)(u+a+h-b)}{h(a+h-b)}\,du\right) \\
& + f(b)\left(\int_{-h}^{h} \frac{u(u+h)}{(b-a)(b-a-h)}\,du\right) \\
& = \frac{1}{3}h\left[f(a) + 4f(a+h) + f(b)\right]
\end{aligned}
$$

Note that this formulation applies specifically to $n = 2$, i.e., two subintervals. To improve accuracy, the interval $[a, b]$ can be divided into more subintervals. Applying Simpson's 1/3 rule to each pair of points yields:

$$
\begin{aligned}
\int_{a}^{b} f(x)\,dx &\approx \frac{1}{3}h\sum_{i=1}^{n/2}\left[f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})\right] \\
&= \frac{1}{3}h\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + ... + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right] \\
&= \frac{1}{3}h\left[f(x_0) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + f(x_n)\right]
\end{aligned}
$$

Once the ITAE index is evaluated, the goal is to minimize it. In this thesis, MATLAB's `fminsearch` function is used for optimization. `fminsearch` is a nonlinear programming solver that searches for a local minimum based on an initial guess. The algorithm adjusts the PID parameters to minimize the ITAE index and thereby optimize system performance. The best performing tuning obtained from ITAE is shown in Table 4.2.

Table 4.2: Best-performing ITAE tuned parameters for cascade PID.

Internal PID		**External PID**	
Parameter	Value	Parameter	Value
`I_P`	1.0	`THETA_P`	35.6024
`I_I`	0.0	`THETA_I`	157.6069
`I_D`	0.0	`THETA_D`	24.9886

4.2.2 Optimal ITAE

To improve the system's robustness, optimal Integral of Time-weighted Absolute Error (ITAE) was researched. This thesis follow a similar methodology described in [19], where the focus lie in designing a robust PID controller.

Table 2 in [19] presents the optimal coefficients of $T_{ITAE}(s)$ based on the Integral of Time-weighted Absolute Error (ITAE) criterion for a step input. For convenience, the table is reproduced below:

Table 4.3: The optimum coefficients of $T_{ITAE}(s)$

Order	**Characteristic equation**
1	$s+\omega_n$
2	$s^2+1.4\omega_n s+\omega_n^2$
3	$s^3+1.75\omega_n s^2+2.15\omega_n^2 s+\omega_n^3$
4	$s^4+2.1\omega_n s^3+3.4\omega_n^2 s^2+2.7\omega_n^3 s+\omega_n^4$

Here, $T_{ITAE}(s)$ is expressed in equation 4.3. Note that the numerator must be a constant to satisfy the ITAE criterion:

$$T_{ITAE}=\frac{Y(s)}{R(s)}=\frac{a_0}{s^n+b_{n-1}s^{n-1}+...+b_1 s+b_0} \tag{4.3}$$

To rewrite the transfer function found in chapter 3.1 as a closed loop with a PID is very complex and time demanding. Therefore, for simplicity, the first order system identified in chapter 3.3 was used instead. The closed loop function is given by equations 4.4 - 4.6.

$$G(s) = \frac{K}{s(\tau s + 1)} = \frac{K}{\tau s^2 + s} \tag{4.4}$$

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s} \tag{4.5}$$

$$T(s) = \frac{G(s)G_c(s)}{1 + G(s)G_c(s)} \tag{4.6}$$

To achieve optimality, the denominator of the closed-loop transfer function in equation 4.6 must match one of the characteristic equations from Table 4.3. Thus, the characteristic equation becomes:

$$1 + G(s)G_c(s) = 1 + \frac{K}{\tau s^2 + s} \frac{K_d s^2 + K_p s + K_i}{s} = \tag{4.7}$$

$$= s^3 + \frac{1 + KK_d}{\tau} s^2 + \frac{KK_p}{\tau} s + \frac{KK_i}{\tau}$$

From the equation above, the third order system is chosen from table 4.3. The natural frequency, ω_n, is chosen to be the same as in [19], i.e. $\omega_n = 8$. From the model identification, $K = 5.3498$ and $\tau = 0.0200$. By observation, the following result is obtained:

$$\frac{1 + KK_d}{\tau} = 1.75\omega_n \Leftrightarrow K_d = \frac{1.75\omega_n \tau - 1}{K} \tag{4.8}$$

$$\frac{KK_p}{\tau} = 2.15\omega_n^2 \Leftrightarrow K_p = \frac{2.15\omega_n^2 \tau}{K} \tag{4.9}$$

$$\frac{KK_i}{\tau} = \omega_n^3 \Leftrightarrow K_i = \frac{\omega_n^3 \tau}{K} \tag{4.10}$$

With the PID parameters now determined, it is important to note that the ITAE transfer function requires a constant numerator to ensure optimality. However, the transfer function in equation 4.6 does not satisfy this condition. A simple and effective solution is to introduce a pre-filter, thereby creating a two-degree-of-freedom (2DOF) control structure. In [19] a zero-pole cancellation pre-filter was employed, where the pre-filter includes poles matching the zeros of the closed-loop transfer function:

$$T_{closed\ loop} = \frac{-36(s-6.139)(s+2.317)}{(s+5.665)(s^2+8.335s+90.38)} \tag{4.11}$$

$$F_{pre-filter\ non-normalized} = \frac{1}{(s-6.139)(s+2.317)} \tag{4.12}$$

$$F_{pre-filter\ normalized} = \frac{-14.222}{(s-6.139)(s+2.317)} \tag{4.13}$$

Using the non-normalized pre-filter results in significant DC gain attenuation, as shown in Figure 4.3. To correct this, the filter is normalized so that $|F(0)| = 1$, as illustrated in Figure 4.4.

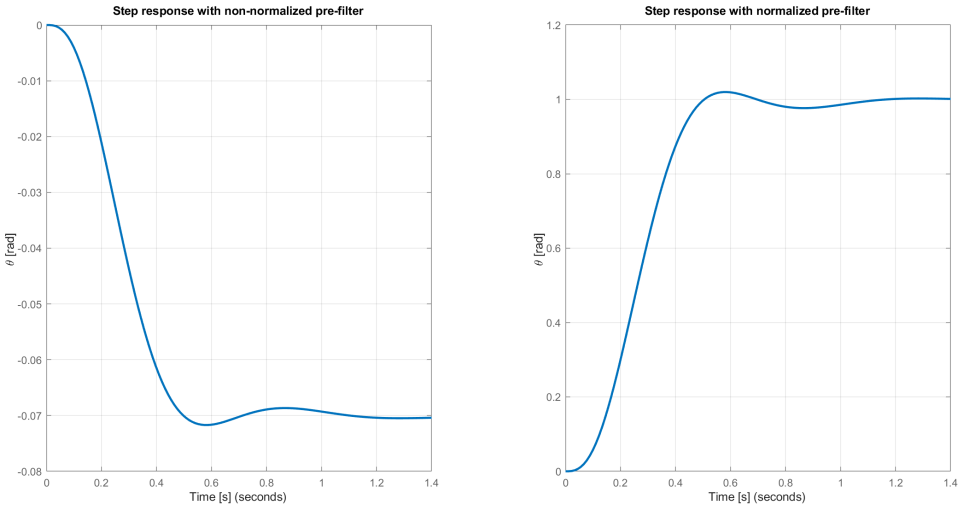


Figure 4.3: Step response for non-normalized and normalized pre-filter

The result of using this zero-pole cancellation pre-filter is shown in the middle graph of Figure 4.5, achieving the same performance as the optimal ITAE response. However, the normalized pre-filter introduces an unstable pole at $s = 6.139$ as shown in equations 4.12 and 4.13, due to the non-minimum phase nature of the closed-loop system (i.e., a right-half plane zero).

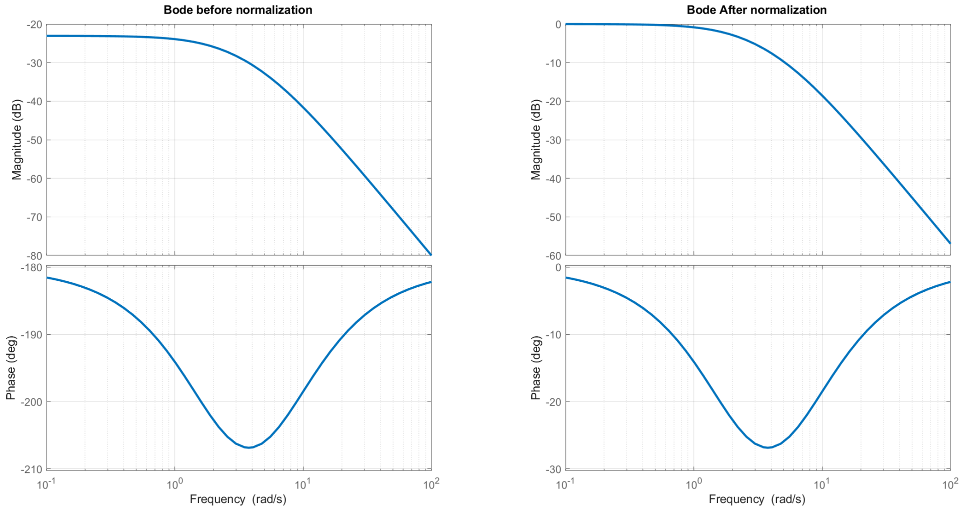


Figure 4.4: Bode plot for non-normalized and normalized pre-filter

Zero-pole cancellation is generally discouraged in non-minimum phase systems, as any deviation from exact cancellation can lead to instability.

An alternative approach is to treat the pre-filter as a low-pass filter with carefully selected poles. Instead of fully cancelling the right-half plane zero, the filter partially compensates for it, maintaining system stability while reducing the adverse effects. One such filter is given by:

$$T_{LP} = \frac{2.5}{s + 2.5} \tag{4.14}$$

The above discussion is valid for the continuous-time case. The discrete-time implementation follows a similar approach. However, for the discrete case, a new pre-filter was developed to further reduce the initial undershoot in the step response. A more aggressive three-pole Chebyshev Type 2 filter was implemented. The resulting discrete-time step response is shown in Figure 4.6. The figure shows that the initial undershoot is negligible, at the cost of increased overshoot. The tuning obtained from optimal ITAE is shown in Table 4.4.

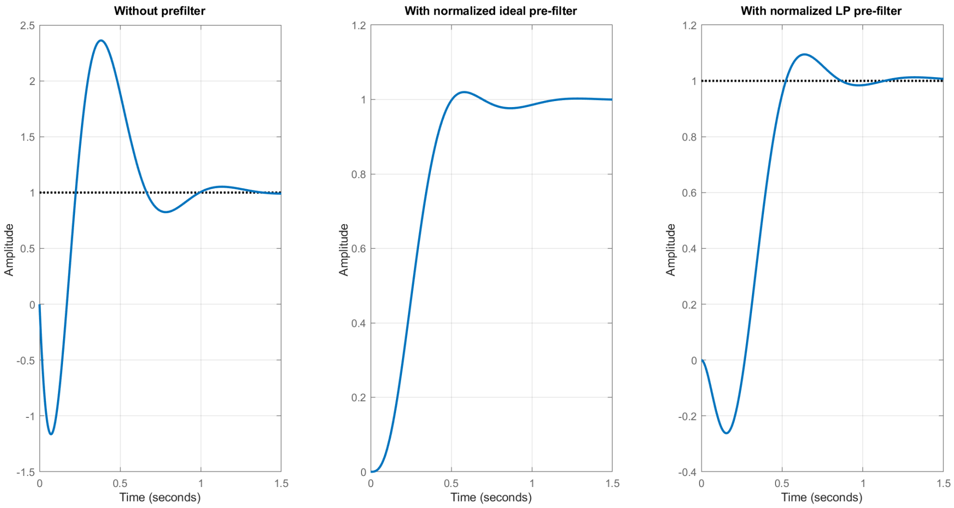


Figure 4.5: Comparison of different pre-filters

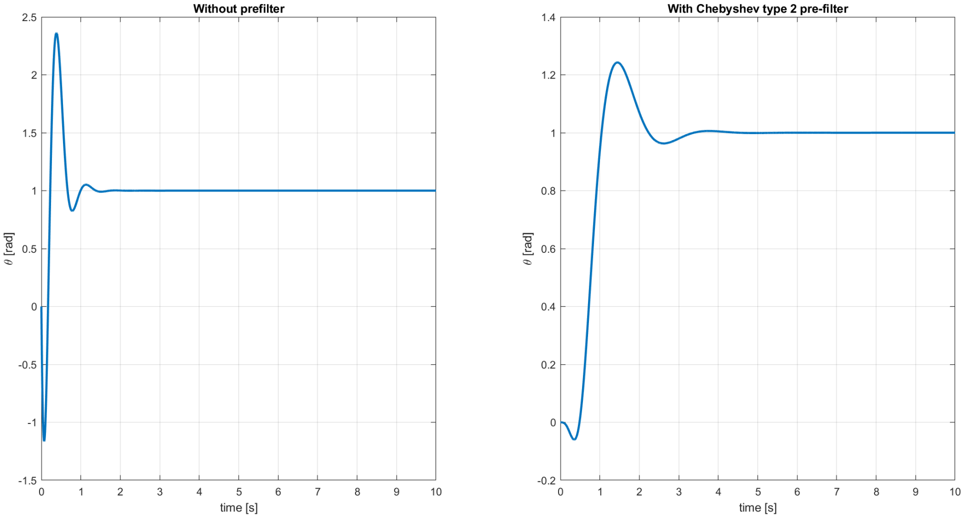


Figure 4.6: Discrete time optimal ITAE

Table 4.4: Optimal ITAE tuned parameters for cascade PID.

Internal PID		**External PID**	
Parameter	Value	Parameter	Value
`I_P`	1.0	`THETA_P`	0.5144
`I_I`	0.0	`THETA_I`	1.9141
`I_D`	0.0	`THETA_D`	-0.1346

4.2.3 Final tuning

Using the PID parameters determined through the Integral of Time-weighted Absolute Error (ITAE) criterion, presented in table 4.2, the closed-loop system achieves a bandwidth of $64.38[rad/s]$. Since the system's bandwidth is significantly higher than the frequency content of the step input, illustrated in figure 4.1, it will be capable of accurately track a step response. The relatively high bandwidth of the cascade PID tuning makes it possible to actively track the detector while an operator is moving it, this will be discussed further in chapter 7.

From figure 4.2, the following closed loop transfer function from θ_{ref} to θ_H is obtained:

$$\frac{\theta_H}{\theta_{ref}} = \frac{0.019933(z-0.8816)(z^2-1.994z+0.9936)}{(z+0.3148)(z-0.8816)(z^2-1.993z+0.9934)(z^2-1.819z+0.8331)} \tag{4.15}$$

The corresponding system poles are listed in Table 4.5 and illustrated in Figure 4.7. According to [6], a discrete-time closed-loop system is considered stable if all poles lie within the unit circle in the complex plane. As shown in both the table and the figure, this condition is met, thereby confirming the stability of the designed system.

Table 4.5: Closed loop system's poles

Pole	**Value**
1	0.1152 + 0.4529i
2	0.1152 - 0.4529i
3	0.4744 + 0.0000i
4	0.9807 + 0.0249i
5	0.9807 - 0.0249i
6	0.9842 + 0.0000i
7	0.5327 + 0.0000i

The Bode plot of the closed-loop system, shown in Figure 4.8, demonstrates a gain margin of 16.2 dB and a phase margin of 124 degrees. These values suggest that the system is highly robust and capable of withstanding significant disturbances, an essential characteristic for high-precision control applications, as emphasized in [6]. The same figure also includes the systems' step response, which confirms that the system accurately follows a step input with minimal deviation.

In theory, this system performs excellently. However, in practice, the real system has

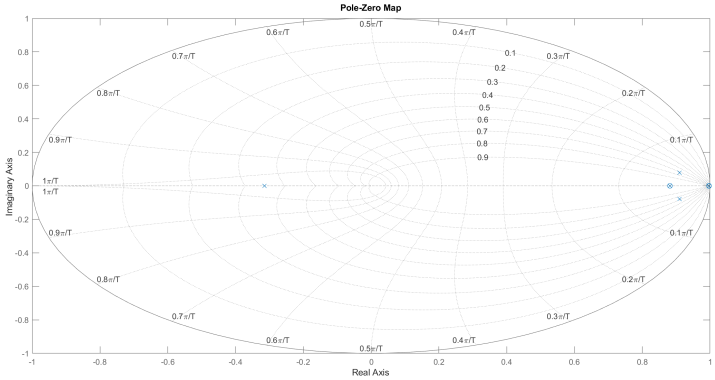


Figure 4.7: Pole-Zero map for the cascade PID closed loop.

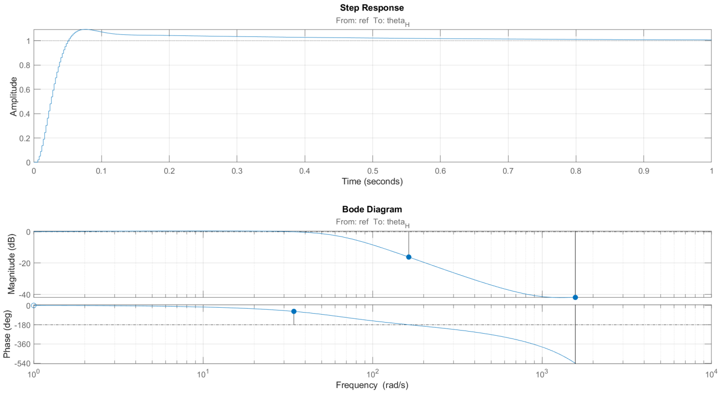


Figure 4.8: Step response and Bode plot of the closed loop system.

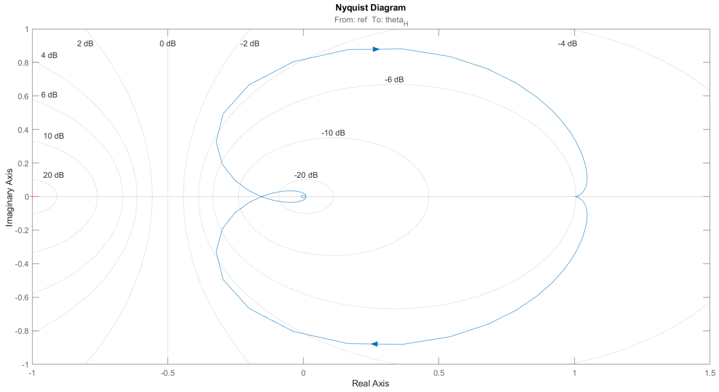


Figure 4.9: Nyquist plot of the closed loop system.

significantly longer settling time, hence, the system does not perform similarly to the simulator as seen in figure 4.10. From figure 4.11, it is clear that in practice the system does not move until the voltage reaches around 1.95 V as noted in chapter 3.5. The lower graph in figure 4.11 illustrates that even if the system does not rotate, the motor's torque keeps increasing until the system finally starts rotating. From this it can be concluded that, to initiate a rotation, a certain threshold voltage must be overcome, which depends on the static friction in the system. This will be discussed further in chapter 3.2. Figure 4.11 also shows that this static friction is significant since the motor needs a relatively high voltage to start rotating. For this purpose, as suggested in [20], a feedforward control algorithm was developed.

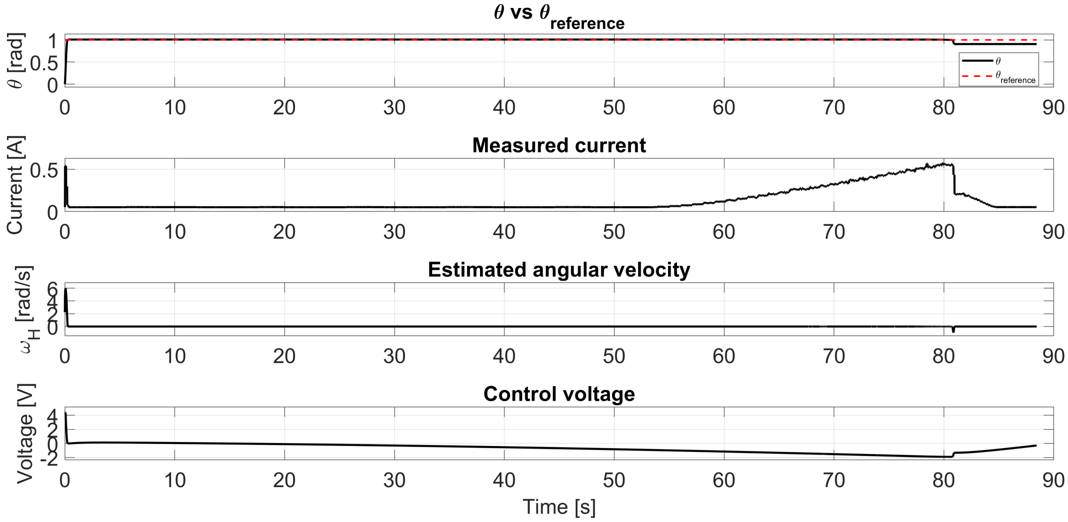


Figure 4.10: Initial performance of the Cascade-PID controller showing difficulties in accurately controlling small movements.

4.2.4 Feedforward Friction Compensator

It is worth mentioning that beyond the threshold voltage, the system behaves linearly as mentioned in 3.5. However, for low angular velocities, the non-linearity in the system dominates as mention in [20]. To enhance the controller's performance, a friction compensator is integrated into the system. This thesis adopts a methodology similar to that proposed in [21], specifically by evaluating various friction models and developing a feedforward compensator based on the identified model.

Following the implementation of the first friction model, as illustrated in figure 5 of [22], the results were satisfactory, as evidenced in figure 4.12 and 4.13.

From these figures, it is clear that the system exhibits less overshoot without the

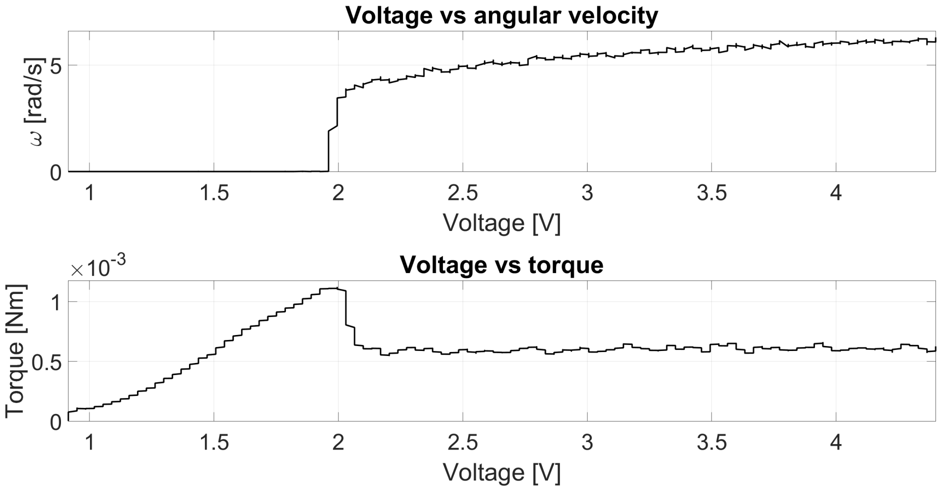


Figure 4.11: Estimated angular velocity and torque when stepping the duty cycle from 0% to 100%.

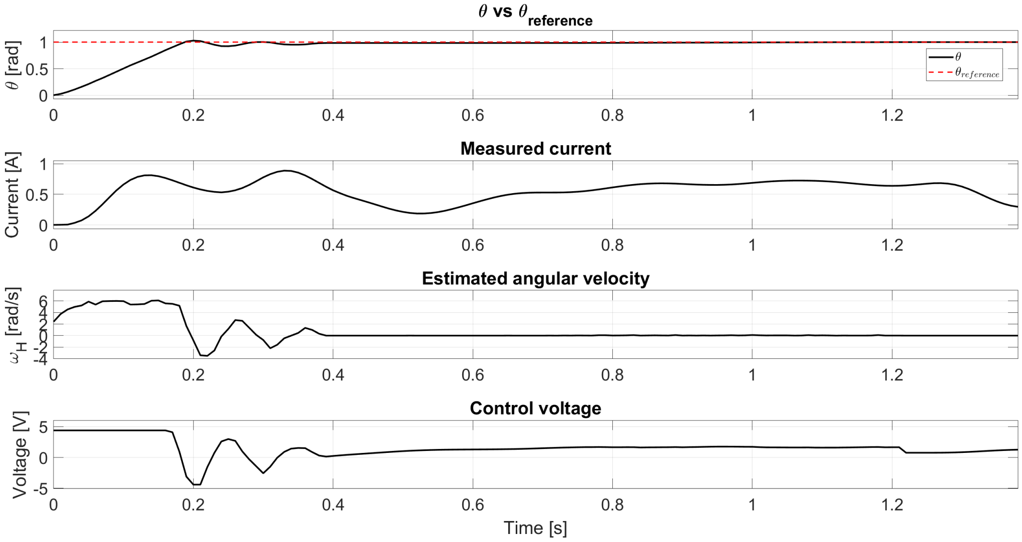


Figure 4.12: Step response with integrated compensator

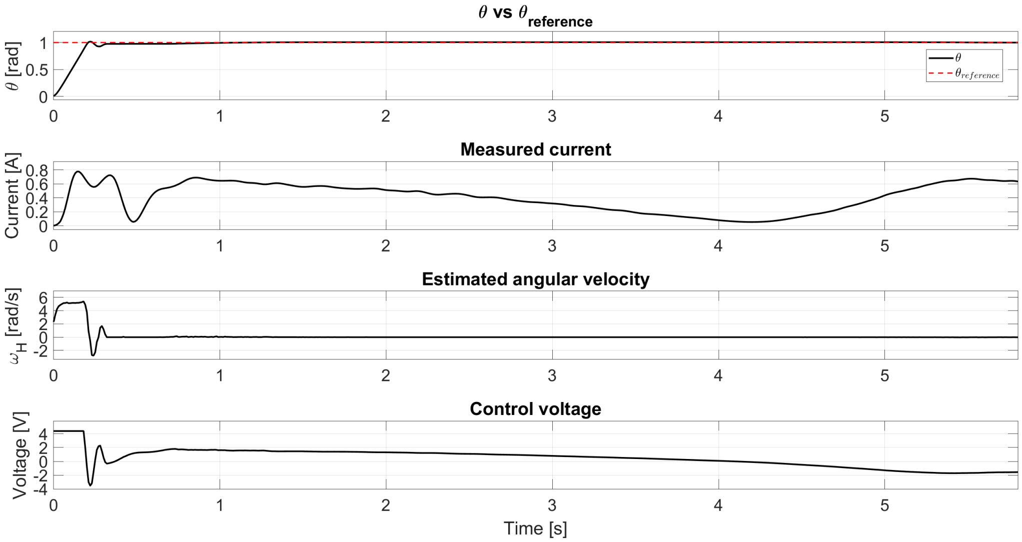


Figure 4.13: Step response without integrated compensator

compensator than when the compensator is integrated. However, the settling time is significantly shorter with the compensator. The results are summarized in table 4.6.

Case	**Overshoot [%]**	**Settling time [s]**	**Rise time [s]**	**SS-error [mm/m]**
With compensator	2.7687	1.3900	0.1209	0.0750
Without compensator	1.7908	5.8100	0.1302	0.3090

Table 4.6: Comparison between with and without a friction compensator integrated to the system.

4.2.5 Nonlinear Feed-Forward Friction Compensator

The model is defined as:

$$M_{dyna}(\omega) = \left(f_k + (f_{max} - f_k)e^{-(\omega/\omega_s)^2} + m_v|\omega|\right)\operatorname{sgn}(\omega) \tag{4.16}$$

Where:

- f_k is the Coulomb friction moment,

- f_{max} is the maximum static friction,
- ω_s is the Stribeck velocity, and
- m_v is the viscous friction coefficient.

The parameters can be estimated as follows:

- f_{max} is measured as the maximum torque before the system starts rotating.
- f_k and m_v are estimated from the linear approximation when $\omega \gg \omega_s$, using:

$$M_{dyna}(\omega) = f_k + m_v\omega \tag{4.17}$$

- ω_s is estimated by setting $\omega = \omega_s$ in the original model:

$$M_{dyna}(\omega) = f_k + (f_{max} - f_k)e^{-1} + m_v\omega_s \tag{4.18}$$

Following this estimation process, the resulting friction model curve is shown in Figure 4.14. To convert the friction torque into a voltage signal for feedforward compensation, the following relations are used:

$$\tau = K_T i_a \Rightarrow i_a = \frac{\tau}{K_T} \tag{4.19}$$

$$V_T = R_a i_a = \frac{R_a}{K_T}\tau \tag{4.20}$$

Where τ is the dynamic friction torque $M_{dyna}(\omega)$. The final voltage output is shown in Figure 4.14.

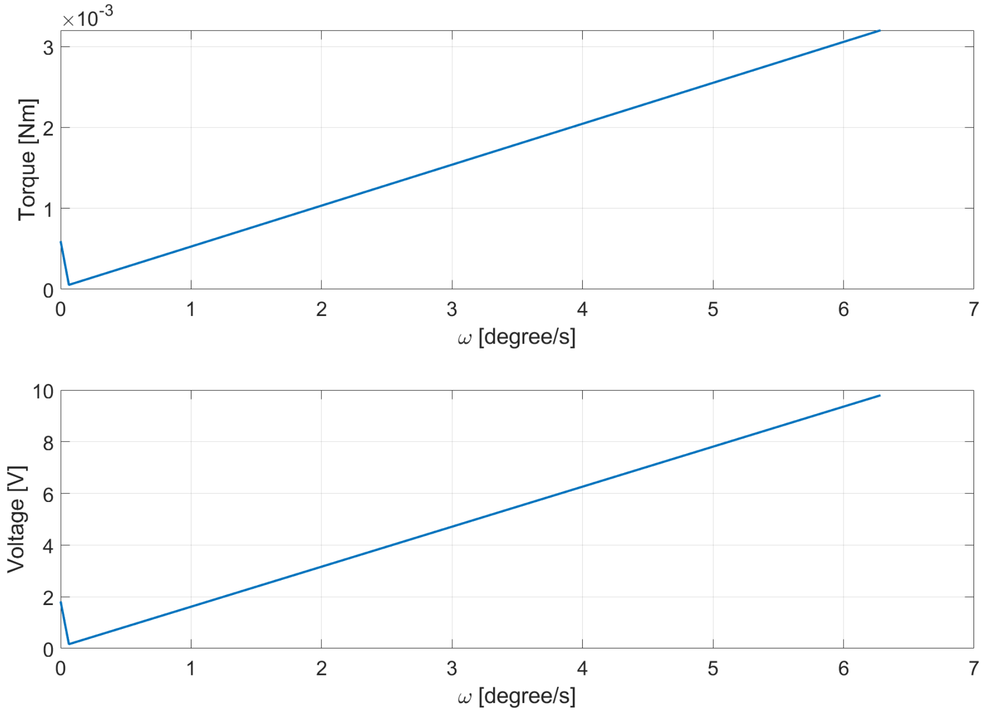


Figure 4.14: Dynamic feedforward friction compensator.

4.2.6 Shortest Path Reference Error

When comparing a desired reference angle θ_{ref} with an encoder's measured angle θ_{meas}, a naive difference

$$\theta_{error} = \theta_{\text{reference}} - \theta_{\text{measured}}$$

can yield misleadingly large values whenever the true angle passes the 0–2π boundary. For instance, if $\theta_{\text{ref}} = 0.01$ rad while $\theta_{\text{meas}} = 2\pi$rad, the raw subtraction produces $\Delta\theta = -1.99\pi$, despite the fact that the actual misalignment is merely 0.01 rad. To avoid such boundary artifacts and always compute the minimal signed rotation from the measured to the reference angle, the two-argument arctangent of the sine and cosine of the difference was employed. Specifically, the error is defined as below in 4.21.

$$\theta_{\text{error}} = \text{atan2}\Big(\sin(\theta_{\text{ref}} - \theta_{\text{meas}}),\, \cos(\theta_{\text{ref}} - \theta_{\text{meas}})\Big). \tag{4.21}$$

The elegance of this formulation lies in its projection of the raw angle difference onto the unit circle: by computing both $\sin(\Delta\theta)$ and $\cos(\Delta\theta)$, any multiple of 2π is moved to the same point on the circle, and the use of atan2 guarantees a unique result in the principal range $(-\pi, \pi]$. Thus, (4.21) always returns the signed shortest-path error, free from discontinuities at the $0/2\pi$ border.

This approach eliminates any manual boundary checks or special-case logic, providing a robust, concise computation of the encoder's shortest-path reference error.

4.2.7 Filtering the Current Signal

From figure 4.15, it is evident that the current measurements exhibit fluctuations. These fluctuations can negatively impact the performance of the cascade PID controller, as the inner control loop relies on accurate current feedback, this dependency is illustrated in figure 4.2. The observed fluctuations in the current signal are primarily caused by the back electromotive force (EMF) generated by the DC motor.

While such small variations may often be negligible, they become problematic when aiming for precise control of the DC motor's position. Therefore, filtering the measurement signals becomes essential to ensure stability and accuracy in the control response. According to [23] signal processing is essential to handle noise interference, accurately interpret sensor data and deliver reliable outputs.

The filter selected for this purpose is a Chebyshev Type 2 filter. As with all filter design, there is a trade-off between noise attenuation and time delay. Figure 4.15 presents a current measurement, while Figure 4.16 shows the corresponding signal's Fast Fourier Transform (FFT). These figures form the basis for the filter design considerations.

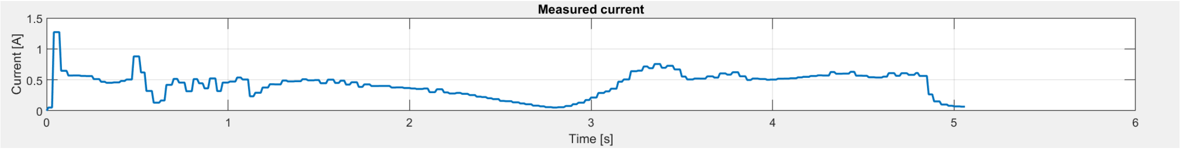


Figure 4.15: Current measurement in time domain.

Figure 4.17 presents a comparison of different filters evaluated for the purpose of smoothing the current measurement signal. As shown in figure 4.16, the signal primarily contains low-frequency components. Consequently, filters with passband ripple—such as the Chebyshev Type I and elliptic filters—were deemed unsuitable and thus rejected.

The comparison then focused on the Butterworth and Chebyshev Type II filters. The main difference between the two lies in their roll-off characteristics: the Chebyshev Type

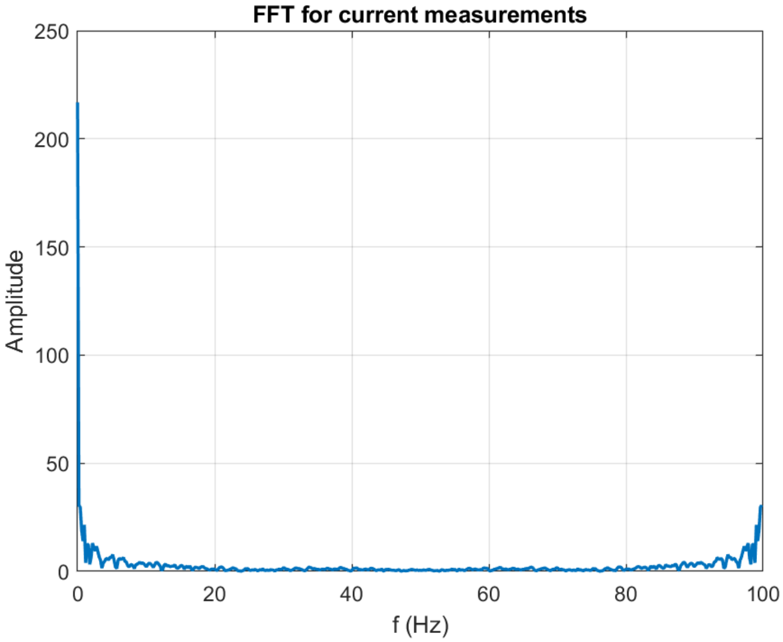


Figure 4.16: Current FFT in frequency domain.

II filter offers a steeper roll-off compared to the Butterworth filter. To more effectively attenuate high-frequency noise while preserving the low-frequency signal content, the Chebyshev Type II filter was selected. However, the Butterworth filter could also have been a reasonable alternative.

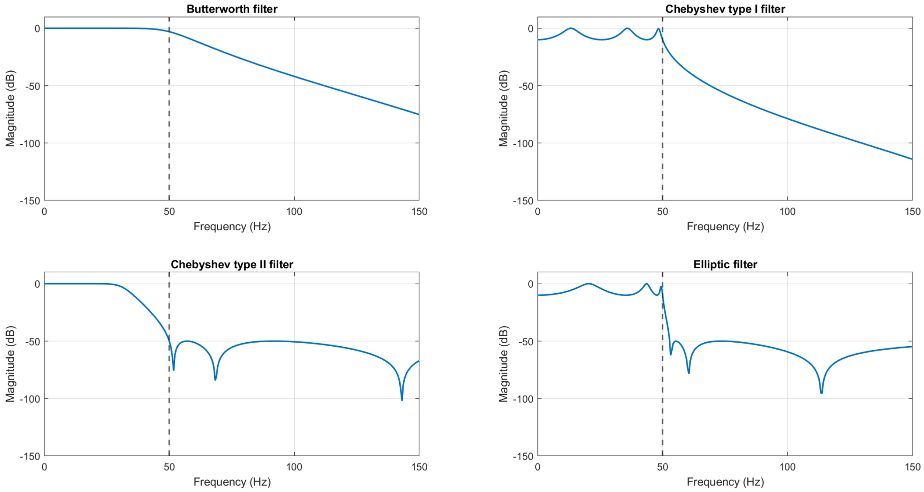


Figure 4.17: Filter comparison for current noise attenuation.

Based on the results presented in Figure 4.18, four key observations can be made. First, the application of the filter significantly reduces the noise in the measured current signal. Second, the measured voltage signal exhibits improved smoothness, particularly in scenarios where the control error is small. Third, under low control error conditions, the variable θ transitions more smoothly toward the reference angle, in contrast to the discrete steps observed prior to filtering. Finally, the implementation of the filter leads to a reduction in the settling time of the step response.

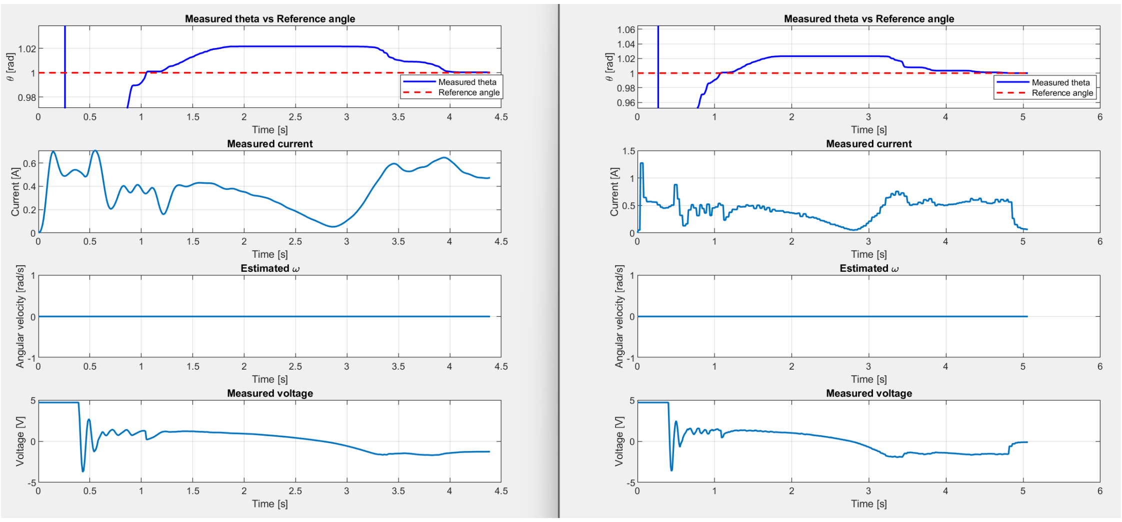


Figure 4.18: Comparison of the same step response with current filtering to the left and without current filtering to the right.

4.3 Linear Quadratic Control

Based on RQ1, the second control strategy investigated is the Linear Quadratic (LQ) controller. According to [6], a continuous-time optimal linear controller can be designed by minimizing both the control error and the control effort. This objective is expressed through the cost function in Equation 4.22:

$$\min\left(|\mathbf{e}|_Q^2 + |\mathbf{u}|_R^2\right) = \min\int \mathbf{e}^T(t)Q\mathbf{e}(t) + \mathbf{u}^T(t)R\mathbf{u}(t)dt \tag{4.22}$$

4.3.1 LQR

Since this thesis considers a discrete-time system, a discrete-time equivalent of the cost function must be used. As also described in [6], the corresponding discrete formulation is given in Equation 4.23:

$$\min\left(||\mathbf{e}||_Q^2 + ||\mathbf{u}||_R^2\right) = \min\sum_{t=1}^{\infty} \mathbf{e}^T(t)Q\mathbf{e}(t) + \mathbf{u}^T(t)R\mathbf{u}(t) \tag{4.23}$$

A formal solution to the discrete-time LQ problem exists if the system is stabilizable and detectable. The matrix Q is positive semi-definite, and R is positive definite. Under these conditions, the optimal control law that minimizes Equation 4.23 is derived from the solution to the Discrete Algebraic Riccati Equation (DARE), shown in equation 4.24. The resulting optimal feedback gain is given in equation 4.25:

$$P = A^T PA - A^T PB(B^T PB + R)^{-1}B^T PA \tag{4.24}$$

$$K = (B^T PB + R)^{-1}B^T PA \tag{4.25}$$

The initial idea was to use a linear quadratic regulator for controlling the laser. However, the result form [12] shows that the system experience steady state error. Steady-state error has to be minimized in this thesis, hence a Linear Quadratic Regulator is not enough.

4.3.2 LQI

To decrease the steady state error to an acceptable level, a Linear Quadratic Integrator controller was implemented instead. The difference between an LQI and LQR is the integral state. This can be achieved by complementing the state space with an integral state, which is done in Equation 4.26 below.

$$x_i(t) = \int_0^t (r(\tau) - y(\tau))\, d\tau \tag{4.26}$$

$$\dot{x}_i(t) = r(t) - y(t) = r(t) - Cx(t) \tag{4.27}$$

$$\tilde{x}(t) = \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} \tag{4.28}$$

$$\dot{\tilde{x}}(t) = \underbrace{\begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}}_{A_{aug}} \tilde{x}(t) + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{B_{aug}} u(t) + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_{E_{aug}} r(t) \tag{4.29}$$

$$y(t) = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{C_{aug}} \tilde{x}(t) \tag{4.30}$$

Since only the last state is an integral state, the LQI gains must be divided as shown in Figure 4.19. This describes the closed loop where the states are chosen to be as in Equation 3.10.

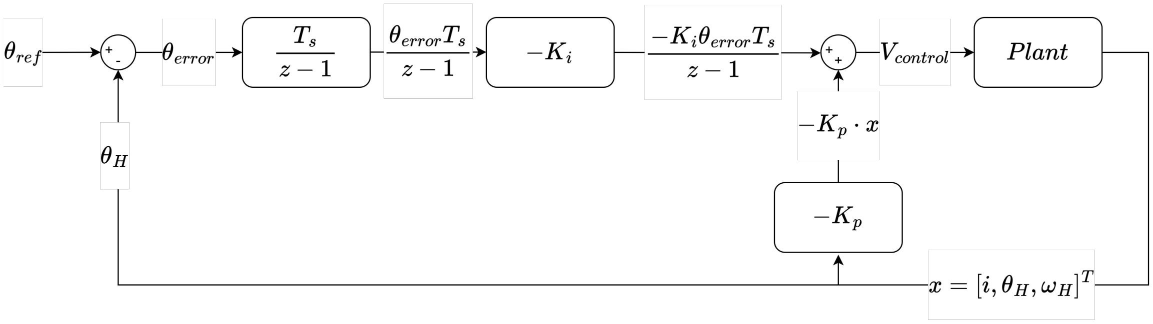


Figure 4.19: Block diagram of the LQI-controller.

By adjusting the Q and R matrices, the control error and control effort will be penalized. The weighting matrices in this thesis was selected by empirical studies based on the control requirement presented in chapter 4.1. The Q and R matrices is represented in equation 4.31 and the corresponding step response is illustrated in figure 4.20.

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.5 \end{bmatrix} \qquad R = 0.1 \tag{4.31}$$

The calculated gains from the solution to the discrete Ricatti equation are given by equation 4.32, where the first three elements are the proportional gains and the last element is the integral gain.

$$K = [0.00016559 \quad 6.80435 \quad 0.21521 \quad -12.1871]^T \tag{4.32}$$

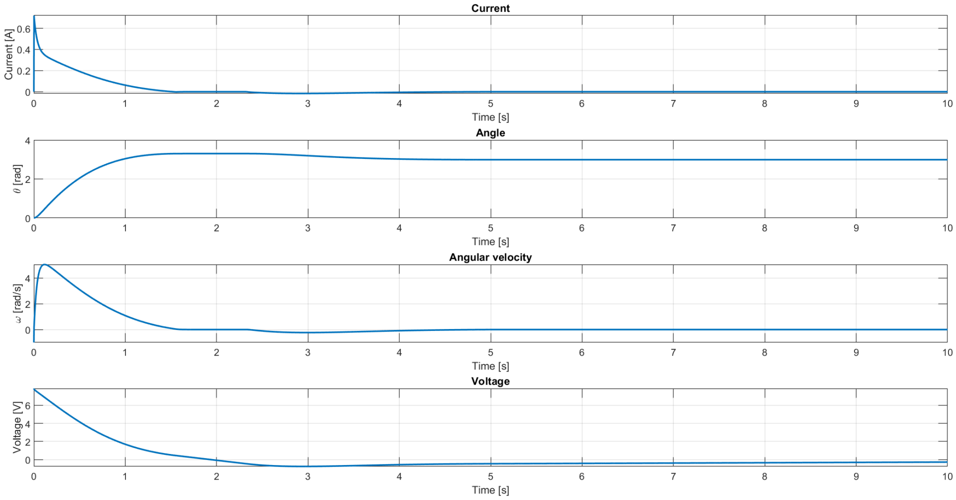


Figure 4.20: Simulation results of the LQI controller

The closed loop system's poles are represented in Figure 4.21. According to [6], the solution to the discrete-time Riccati equation always leads to a stable solution. Therefore, all the poles presented in figure 4.21 are within the unit circle.

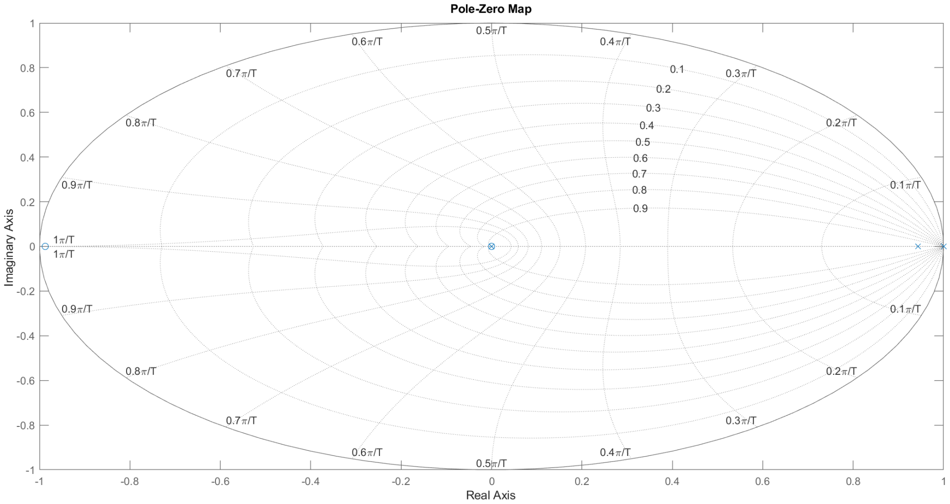


Figure 4.21: Pole zero map for closed loop system with LQI controller

As previously mentioned, the system worked excellently in simulation because a linear model was used; however, at low angular velocity, the non-linearity dominates for the friction slowing down the controller when close to the reference angle, as shown in Figure 4.22.

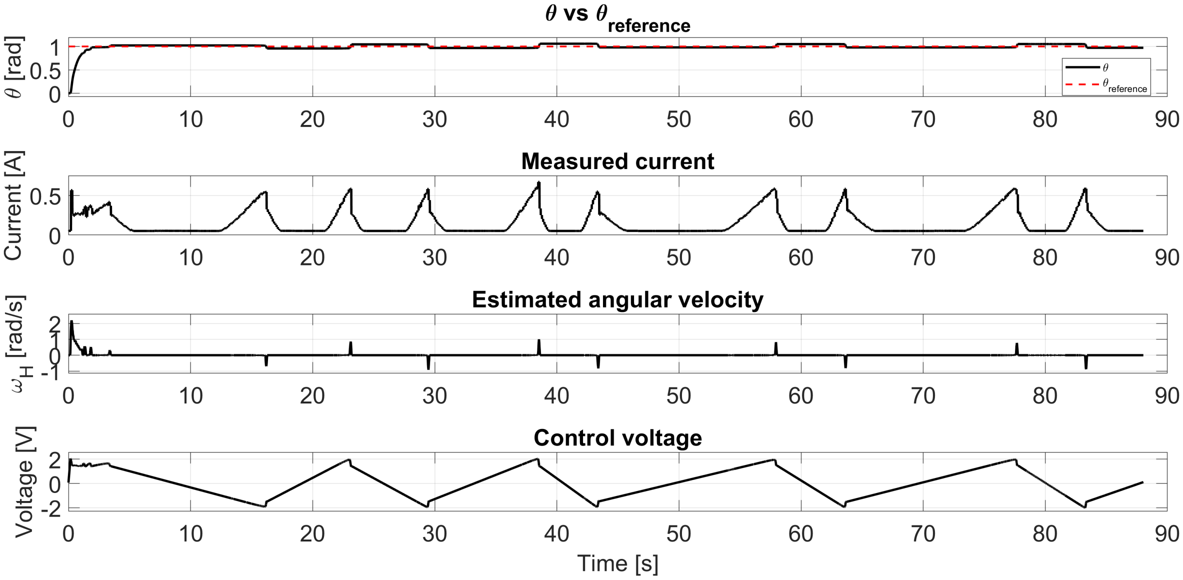


Figure 4.22: Regular LQI without friction compensator does not reach the setpoint and oscillates/overshoots the setpoint.

Because of the low gain for the current state, i.e. the first state x_1, the LQI controller is not sensitive to current fluctuations which arise from sensor noise and back-emf. Therefore, the current filter has a limited effect on the system's performance.

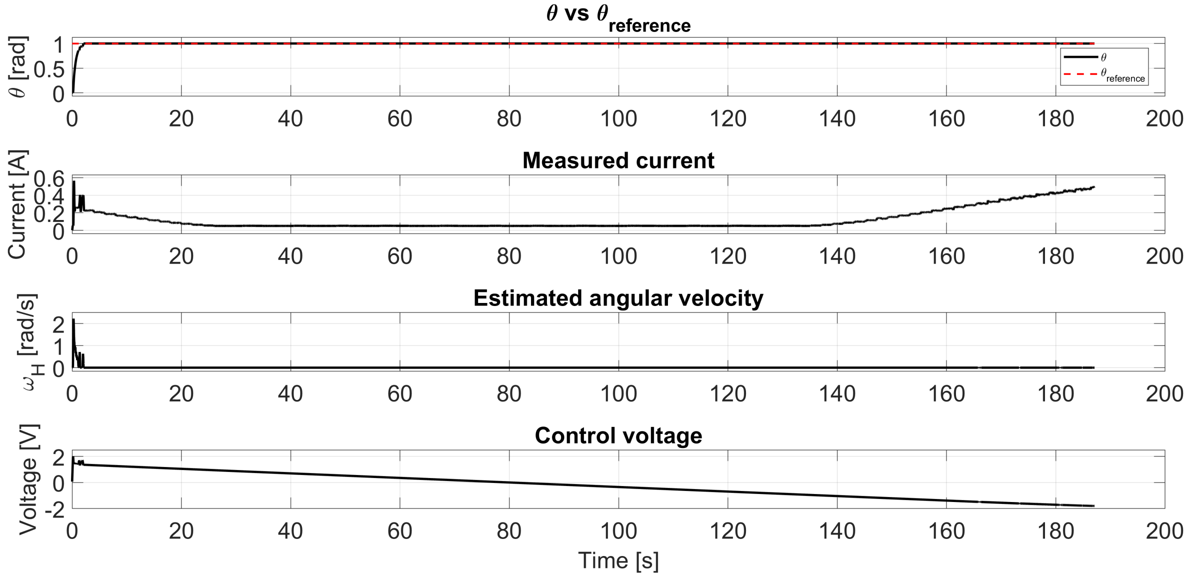


Figure 4.23: Limited effect using the current filter. By coincidence the laser head ended up closer to the setpoint than in 4.22 and therefore the integral builds up slower.

To account for the nonlinear friction, a compensator was also needed for the LQI. Figure 4.24 shows the effect of the compensator on the system.

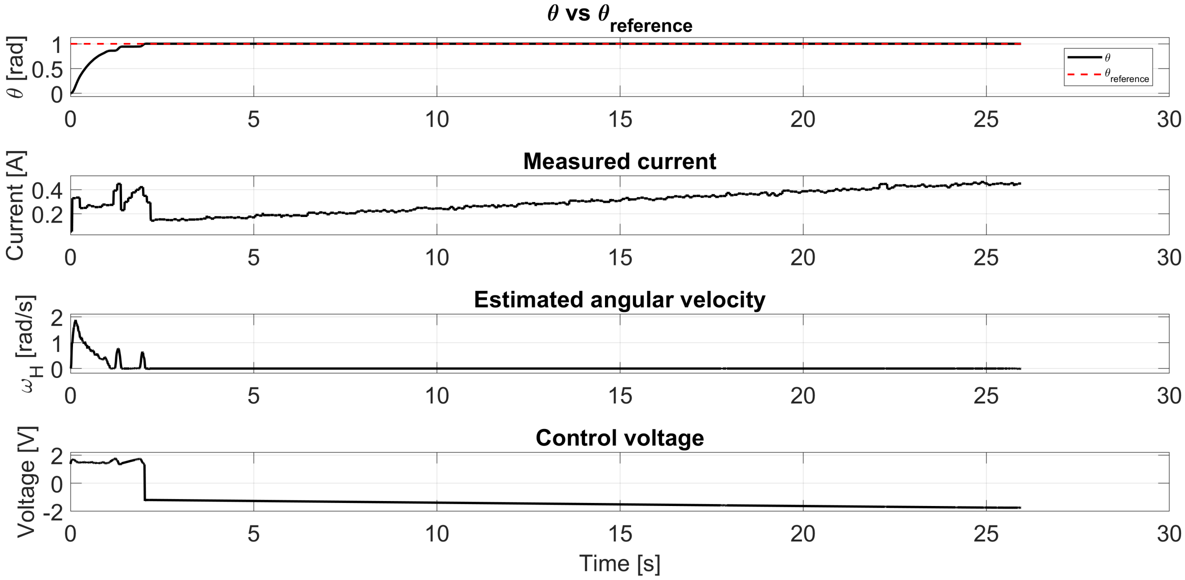


Figure 4.24: LQI with friction compensator reaches setpoint in 25 seconds. At timestamp 2 [s] the compensator help the controller to switch direction.

The strategies combined resulted in the following performance in Figure 4.25.

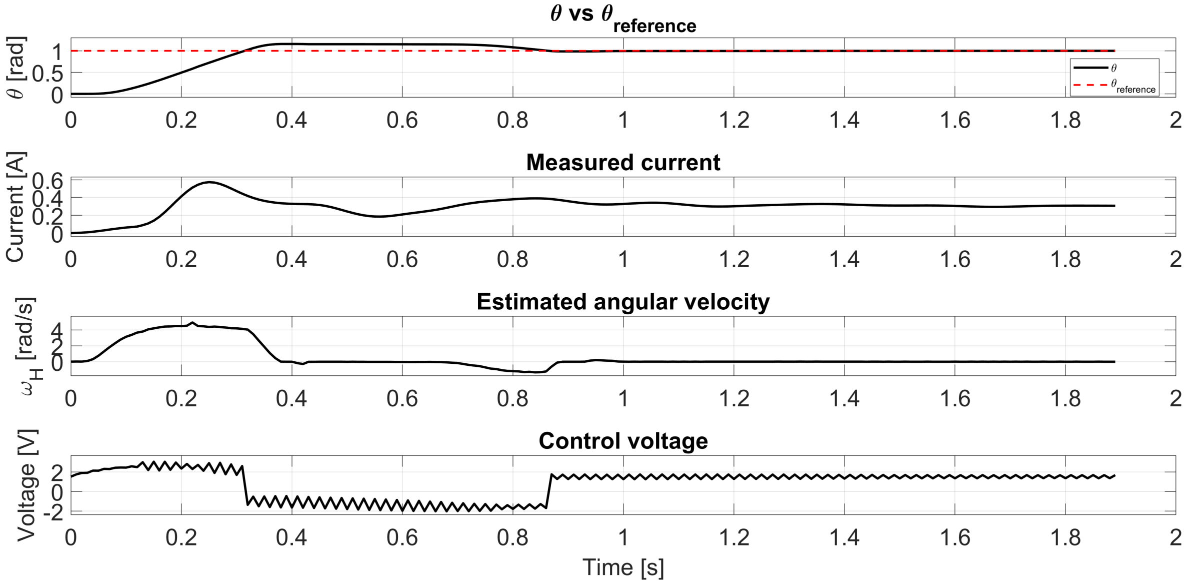


Figure 4.25: Final LQI reaches setpoint in 1.8 seconds.

4.3.3 Angular Velocity Estimation

Ultimately signals are perfect without disturbances, but in practice they often need to be preprocessed to help the control system perform as intended. Noise in sensor signals and outlying samples can make the system unstable [24]. This section describes the signals involved in the control system and how they were processed to improve noise rejection and increase robustness.

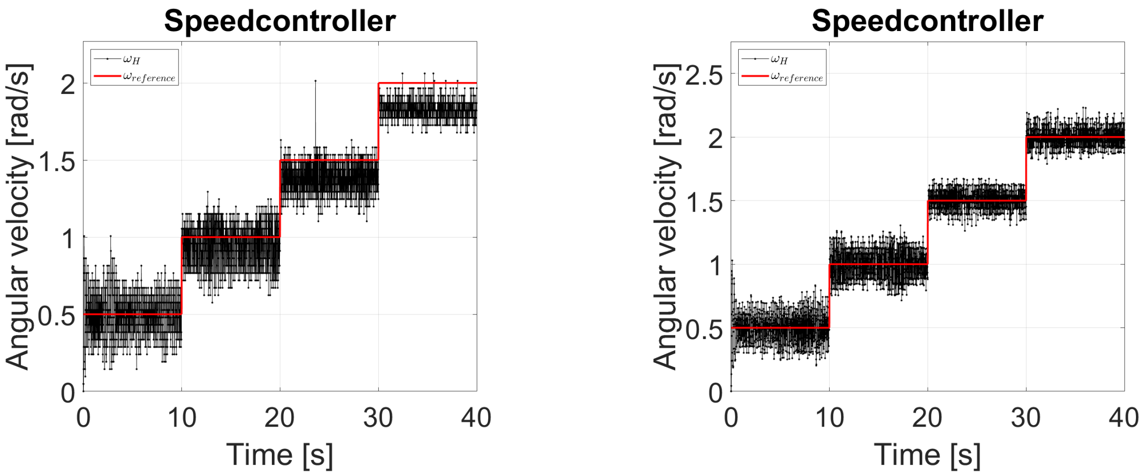
The logged signal are at 100Hz even though the real signal is sampled at 500Hz. Therefore, the logging functionality is limited to catch unwanted signal at more than roughly 50Hz as per Nyqvist samplings theorem in [13].

To avoid outliers in the discrete time derivative of the angular velocity estimate an exponential moving average filter was implemented with and α coefficient of 0.5. The moving average filter is often used to smooth signal in a simple way, however more complex filter could be employed as with the current filter. Since the encoder measurement is not noisy by itself, the moving average is merely used to reduce the noise of the discrete derivative. The filter coefficient was determined using a recorded angle signal and adjusting the coefficient. A lower alpha results in a slower filter and a larger alpha results in a more responsive filter. The resulting filter is a compromise between speed and noise rejection.

The coefficient of a moving average filter depends on the quality of the signal, however, as a rule of thumb lower α results in a more responsive filter and higher *alpha* smoothens the signal more.

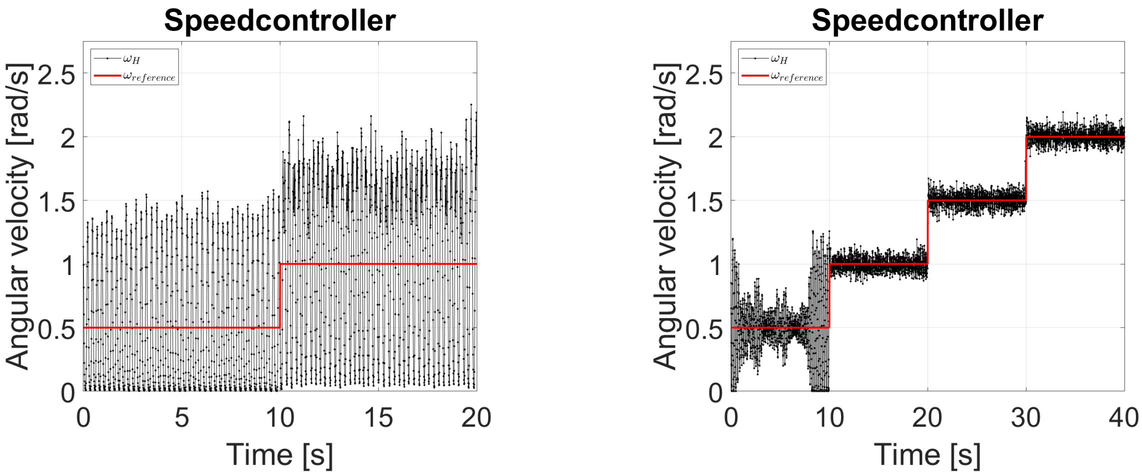
$$\omega_{est_k} = \alpha \cdot \omega_{est_{k-1}} + (1 - \alpha) \cdot \omega_{measured_k} \tag{4.33}$$

Evaluating the obtained velocity by applying a constant voltage showed that it was fluctuating, and at low speed coming to a halt similar to a stepping behavior. When searching for the detector it is necessary to keep the angular velocity under a certain threshold enabling the detector to detect the laser beam, hence a simple PID velocity controller was developed using the model derived in Chapter 3.



(a) Speedcontroller without velocity filter results in a growing offset as reference velocity gets larger.

(b) With velocity filter ($\alpha = 0.1$). Fairly even performance across all tested references.



(c) With velocity filter ($\alpha = 0.9$). The filter is too slow for the controller to react.

(d) With velocity filter ($\alpha = 0.5$). Less noisy at higher velocities but unstable at $0.5[rad/s]$

Figure 4.26: Comparison of speed controller responses with and without velocity filtering for different tuning of the filter.

4.3.4 Nonlinear Error Shaping

As the reference error approaches zero, the control effort naturally diminishes. This leads to a slower convergence process, as depicted in Figure 4.22, where the integral error builds up more slowly when getting close to the setpoint. If the system's overall performance remains satisfactory, characterized by an acceptable rise time, minimal overshoot, and low oscillations, an alternative approach involves implementing a nonlinear error shaping function to scale the reference error, as suggested in [25]. Employing such a function results in a dynamic yet controlled response movement as the system converges to the setpoint. The work by [25] explores several of these functions, each having its distinct advantages and disadvantages, however, the strategy was employed for a single-input single-output PID controller, and hence the performance and stability have not been mathematically verified for LQI.

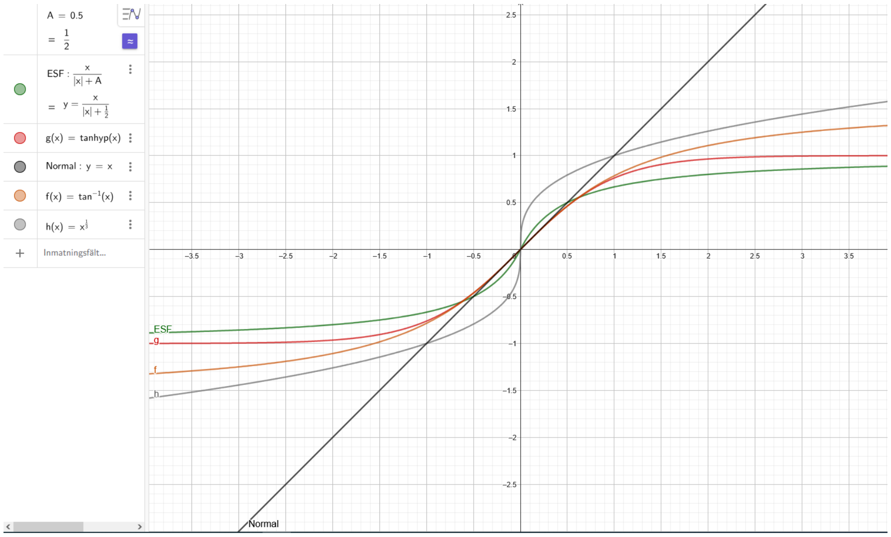


Figure 4.27: Nonlinear error shaping functions of error to boost the control effort when the reference error is small [25] compared to normal error calculation.

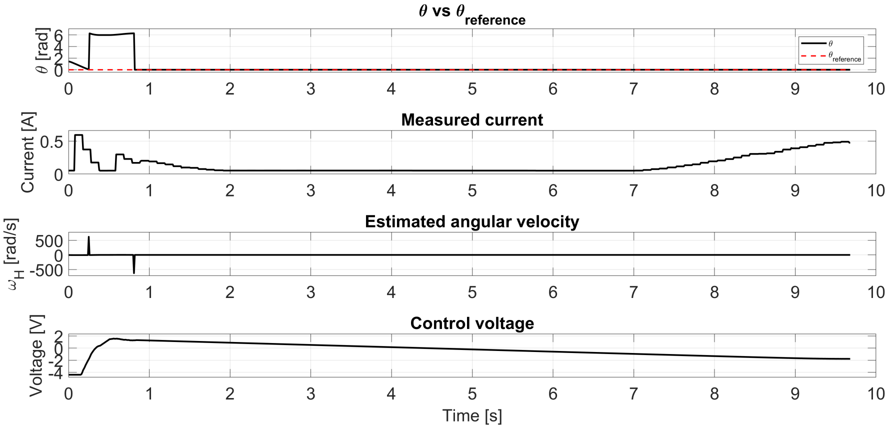


Figure 4.28: Performance of regular LQI with added velocity filter, current filter, and nonlinear scaling of the reference error.

5

Techniques for Localizing the Detector Unit

This chapter describes and evaluates several methods for enabling communication between the transmitter and detector unit to estimate the distance between them and to obtain a reference angle. The goal is to achieve accurate and reliable localization of the detector using the constraints of the current hardware and software.

5.1 Obtaining the Reference Angle

Two main approaches were investigated for estimating the reference angle between the transmitter and the detector unit. The first is based on detecting a laser modulation frequency change using the PSD, and the second relies on detecting a voltage spike when the laser beam passes over the PSD.

The least complex of these is the detection-trigger method, referred to as (M2). It involves recording the start and stop angles of the laser sweep when the detector sends a signal upon detecting the laser beam. This method assumes low communication latency and fast signal processing. Its accuracy depends on both the timing of the trigger signal and the angular velocity of the laser head. In particular, if the laser rotates too quickly, the angle recorded at the time of signal reception may differ significantly from the actual detection angle unless compensated for.

The method of estimating the reference angle using detection trigger consists of sweeping the laser beam in one direction and record both entry and exit trigger when the laser spot hits the PSD. By assuming constant angular velocity, the midpoint of those two angles are taken as the reference angle. The maximum allowable angular velocity $\omega_{H_{\mathrm{MAX}}}$ of the laser head to ensure the beam remains detectable across the 20 mm wide PSD is given by Equation 5.1. This depends on the sweep time t_{sweep} and the distance L between the laser and the detector.

$$v_{\text{PSD}} = \omega_H \cdot L$$
$$b_{\text{PSD}} = v_{\text{PSD}} \cdot t_{\text{sweep}} \approx 20.0 \text{ mm}$$

$$\omega_{H_{\text{MAX}}} = \frac{v_{\text{PSD}}}{L} = \frac{b_{\text{PSD}}}{t_{\text{sweep}} \cdot L} \tag{5.1}$$

Table 5.1: Parameter Definitions for Equation 5.1

Symbol	**Description**
ω_H	Angular velocity of the laser head [rad/s]
v_{PSD}	Tangential (linear) velocity of the laser spot across the PSD [mm/s]
b_{PSD}	Effective width of the PSD (area over which the laser is detected) [mm]
t_{sweep}	Time for the laser spot to cross the PSD [s]
L	Distance between the laser head and the detector [mm]
$\omega_{H_{\text{MAX}}}$	Maximum allowable angular velocity of the laser head [rad/s]

The second method is based on frequency detection. When the laser beam sweeps over the PSD, the modulation frequency of the laser beam is changed. If the frequency modulation can be measured accurately, this provides an alternative way to detect the angle of the laser beam without relying on fast communication. In theory, this method eliminates the need for timestamp-based synchronization and is less sensitive to communication latency. However, it introduces new challenges, such as measurement noise and aliasing. The current system's ADC sampling rate of 95 kHz supports frequencies up to 47.5 kHz according to the Nyquist sampling theorem [13]. Additionally, buffer overflow at higher speeds ADC sampling rates further limits usable precision, constraining the measurement resolution.

5.2 Distance Estimation

While the reference angle provides angular alignment between the transmitter and detector, the distance L between them is equally important for achieving full 2D-localization. Unfortunately, the current hardware imposes signal delay, which restrict the accuracy and reliability of distance estimation.

A triangulation-based approach, inspired by [26], was evaluated. This method uses the geometric relationship between the angular span $\Delta\theta$ of the laser beam and the known width of the PSD, denoted b_{PSD}. Assuming the laser rotates about a fixed center point and the beam sweeps across the full width of the PSD, the following relationship can be used to compute the distance:

$$\frac{\Delta\theta}{2} = \tan^{-1}\left(\frac{2b_{\mathrm{PSD}}}{L}\right) \tag{5.2}$$

$$L = \frac{2b_{\mathrm{PSD}}}{\tan\left(\Delta\theta/2\right)} \tag{5.3}$$

Here, b_{PSD} is a fixed parameter based on the physical PSD dimensions, while $\Delta\theta$ must be measured. The precision of L therefore relies directly on how accurately $\Delta\theta$ can be determined.

Due to the nonlinearity of the *arctan* function, small errors in $\Delta\theta$ cause increasingly large errors in L as the distance grows. This sensitivity is visualized in Figures 5.1 and 5.2, which show how both absolute and relative distance errors scale with increasing range.

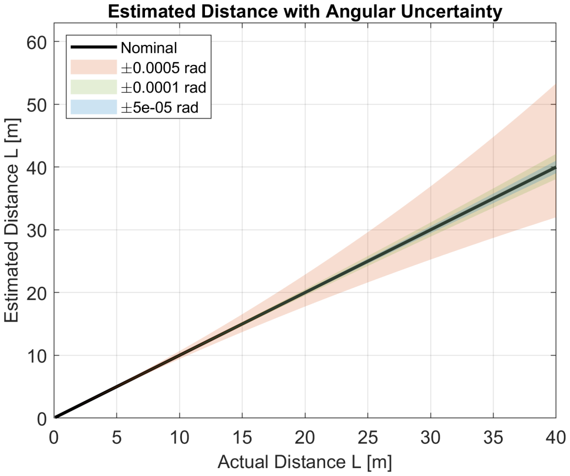


Figure 5.1: Estimated distance L with uncertainty bands due to angle inaccuracies.

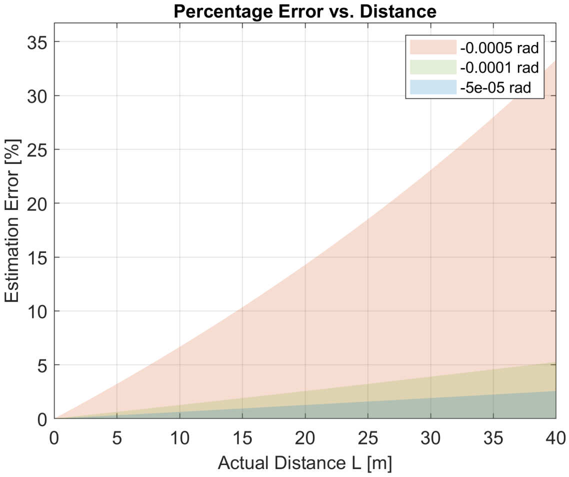


Figure 5.2: Percentage error in estimated distance L.

There are two general strategies for estimating $\Delta\theta$ in this context:

- **Trigger-based estimation:** As discussed in Section 5.1, the laser beam's entry and exit from the PSD can be detected via voltage spikes. By recording the angular position of the laser head at both events, $\Delta\theta$ is computed as their difference. While simple in theory, this method is highly sensitive to communication latency and detection delays, which limits its effectiveness at high angular velocities.

- **Sweep-time and velocity estimation:** In contrast to trigger-based methods, this approach measures the total time the laser beam is detected by the PSD (t_{sweep}) and multiplies it by the estimated angular velocity ω_H of the laser head:

  $$\Delta\theta = \omega_H \cdot t_{\text{sweep}}.$$

  This method avoids the need for real time communication and may be less affected by individual timing delays. However, it relies on the assumption that angular velocity remains constant during the sweep and that both the time and velocity estimates are accurate.

5.3 Software and Hardware Implementation

The host company had already established certain functionalities, including communication protocols, PWM control of the integrated motor, measurement of the laser's sweep duration over the PSD, and the measuring of the laser beam's starting and ending frequencies while sweeping over the PSD. In order to implement the controllers formulated in this paper, Simulink's code generation tool was deployed to translate it into C-code. Leveraging codegen, it was essential to update the existing code infrastructure by incorporating a finite state machine dedicated to managing the controller module, which would invoke the generated controller file to compute the control voltage based on the laser's current and angle measurements.

The starting and ending frequencies, measured by the detector, are sent to the transmitter unit via the console application which acts as the Tablet mentioned in Figure 2. The transmitter unit is responsible for computing the distance and reference angle. Given that the laser transmitter handles the laser beam modulation, the measured laser frequency can subsequently be mapped back to an absolute reference angle for use by the controller.

The hardware limits the update frequency to $1000[Hz]$ but to reduce computation power, $500[Hz]$ is used for the finite state machine to reduce the risk of race conditions with other processes.

6

Results

This chapter presents the key findings obtained from both simulations and experimental validation. The performance of the two implemented control strategies—cascade PID and Linear Quadratic Integrator (LQI)—is evaluated based on metrics such as alignment time, overshoot, and steady-state error. Additional focus is placed on the effectiveness of signal filtering and friction compensation techniques. Furthermore, results related to the feasibility of laser detector localization and distance estimation using the Position Sensitive Device (PSD) are presented. Each subsection addresses one of the research questions.

6.1 RQ1 – Controller Comparison

[RQ1.] How do the performances of a cascade PID controller and a Linear Quadratic controller compare in positioning the laser beam to a given reference angle?

The goal of RQ1 is to evaluate the performance of two feedback control strategies for aligning the laser beam with the detector:

1. A cascade Proportional-Integral-Derivative (PID) controller
2. A Linear Quadratic Integral (LQI) controller

Both controllers were implemented in software and tested in a hardware-in-the-loop environment to ensure practical viability. The performance criteria included rise time, settling time, overshoot, and steady-state error. Additional enhancements, described in Chapter 4, were implemented to improve real-world robustness and accuracy.

The key features integrated into each controller are summarized in Table 6.1.

Table 6.1: Controller performance enhancements

Controller	Shortest path	Feedforward compensator	Current filter	ITAE tuning	Nonlinear integration
PID	Yes	Yes	Yes	No	No
LQI	Yes	Yes	Yes	No	Yes

The final measured performance of both controllers is shown in Table 6.2.

Table 6.2: Closed-loop performance comparison of PID and LQI controllers

Controller	Overshoot [%]	Settling time [s]	Rise time [s]	Steady-state error [mm/m]
PID	2.77	1.39	0.12	0.075
LQI	0.19	1.78	0.18	0.075

The results show that both controllers successfully align the laser beam within the target precision of ±2 mm. The PID controller provides a faster rise time and shorter settling time but at the cost of slightly higher overshoot. In contrast, the LQI controller offers minimal overshoot and equal steady-state accuracy, albeit with a slightly longer response time. Given the application's requirement for stability and repeatability, the LQI controller is preferred when smooth, low-overshoot behavior is prioritized.

6.2 RQ2 – Detector Localization Feasibility

[RQ2.]How can the Position Sensitive Device (PSD) on the detector unit be utilized to estimate the relative angle and distance between the detector and the laser transmitter?

To explore detector localization, a quantitative analysis was conducted to evaluate the feasibility of estimating the reference angle using only PSD signal timing and encoder feedback. Two methods were proposed (Method M1 and M2) and partially tested in hardware.

Empirical testing revealed the following observations:

- The Position-Sensitive Detector (PSD) consistently responds to incident laser light within approximately 50 milliseconds.
- The communication latency between the detector and the transmitter is more difficult to quantify directly, but is estimated to be approximately 100 milliseconds.

Due to the PSD's 20 mm width, the angular velocity of the laser sweep must be limited to ensure reliable detection. Table 6.3 shows the physical velocity constraints at various distances.

Table 6.3: Maximum allowable angular velocity for detection based on distance

Distance L [m]	**Max velocity [rad/s]**	**Angle detection error [rad]**	**Full turn sweep time [s]**
0.5	0.800	0.080	7.9
1.0	0.400	0.040	15.7
5.0	0.080	0.008	78.5
10.0	0.040	0.004	157.1
20.0	0.020	0.002	314.2
30.0	0.013	0.001	471.2
40.0	0.010	0.001	628.3

To match or outperform a skilled human operator, the system must localize the detector within a 90° range (a quarter turn) in under 30 seconds. Table 6.4 outlines the required detection and communication latency at various sweep speeds, assuming a fixed PSD width and a distance of 40 m.

$$t_c = \frac{e_{\text{PSD}}}{L \cdot \omega_H} \tag{6.1}$$

Table 6.4: Detection and communication latency requirements at 40 m for various sweep times

Search time (90° sweep) [s]	Angular velocity ω_H [rad/s]	Linear velocity across PSD [m/s]	Max detection time [ms]	Max communication delay [ms]
2.0	0.79	31.42	0.64	0.06
5.0	0.31	12.57	1.59	0.16
10.0	0.16	6.28	3.18	0.32
15.0	0.10	4.19	4.77	0.48
20.0	0.08	3.14	6.37	0.64
25.0	0.06	2.51	7.96	0.80
30.0	0.05	2.09	9.55	0.96

The calculations indicate that at greater distances and slower sweep rates, the system exhibits increased tolerance to detection and communication latency. However, these results deviate significantly from the actual performance observed in the real detector unit. As sweep speed increases, the permissible response time decreases further, falling well below the system's measured delay of approximately 150 ms. This constraint poses challenges for detector localization unless further optimization is implemented.

Frequency estimation was performed using the STM32's internal timer. The system counts rising edges of the voltage signal from the PSD and averages across samples to reduce noise. Sweep time was measured using the microcontroller's hardware timer, which is triggered by a sharp voltage spike from the PSD.

In practice, both detection trigger and frequency measuring face limitations under the current hardware constraints, particularly due to limited resolution and latency. However, they offer a useful framework for future development, especially if real-time processing capabilities are improved.

7

Discussion

This chapter reflects on the results presented in Chapter 6 and examines their implications in the context of the project goals and research questions. The comparative performance of the cascade PID and LQI controllers is analyzed, with particular attention to their real-world applicability, limitations, and potential for further refinement. The discussion also considers the partial success in distance estimation using the PSD and explores the challenges encountered in full detector localization. Broader implications such as engineering ethics, system robustness, and ideas for future development are also addressed.

RQ1

The PID controller performs reasonably well, but it requires significant control effort and shows noticeable oscillations around the setpoint. In contrast, the LQI controller provides smoother control but shows an initial undershoot unless a state prefilter is applied. The LQI controller is theoretically guaranteed to be stable, assuming the system model is accurate enough—though this assumption may be questionable and should be examined further.

Tuning has a strong impact on the robustness of both controllers. Aggressive tuning can lead to instability, while more moderate tuning results in better robustness and reliability.

Compared to LQI, the PID controller shows more oscillations, likely due to tuning limitations. The PID was tuned using the ITAE (Integral of Time multiplied by Absolute Error) criterion, which gave good simulation results. However, in the real system, the performance differs—especially at lower angular velocities.

The simulation model should be improved by including dynamic friction effects, as measured in Section 4.2. This could make the simulation more accurate and improve real-world performance. One approach is to use Bayesian optimization in combination with ITAE to tune the controller while accounting for disturbances. For example, MATLAB's `bayesopt` could be used to optimize the control gains. This could help reduce the oscillations caused by unmodeled nonlinear friction in the simulation.

Some simple but effective methods were tested. For example, using a constant friction compensator with the PID controller reduced the settling time from about 6 seconds to 2 seconds. This method is also computationally efficient.

Filter use introduces extra computation. The best update frequency depends on the controller. For example, using 1000 Hz may improve PID performance, while LQI may not benefit due to how the discrete derivative is calculated. This area needs further investigation.

Both control strategies perform poorly at low angular velocities. As mentioned earlier, a feedforward friction compensator is essential for improving overall control performance. Several friction models exist that can be used for this purpose. This thesis focuses on the Stribeck friction model, which accounts for the transition between static and dynamic friction but the nonlinear friction compensator was not implemented. However, the constant friction compensator was effective and easy to implement and tune, which is useful during production and online tuning.

Several ideas for future improvements are listed below:

- Create a nonlinear model using collected data.
- Reduce system friction to make behavior more linear at low angular velocities.
- Use a nonlinear friction observer to improve controller robustness.
- Investigate nonlinear control methods such as backstepping or multi-step nonlinear controllers.
- Add a physical angular velocity sensor to reduce error in the calculated derivative.
- Explore sensorless position control to lower cost and complexity by using a cheaper encoder with a Kalman filter. In this setup, short-term tracking could rely on ripple detection, while long-term tracking uses encoder data to avoid drift.

RQ2

As seen in Chapter 5, several PSD-based methods offer trade-offs between speed and delay tolerance. However, Section 6.2 confirms that none are fully feasible with the current hardware setup. As explained in Section 6.2, both methods M1 and M2 were implemented and tested. However, with the current hardware and software, it is not possible to accurately localize the detector as required in Chapter 2. The main issue is high latency in both the detector unit and the communication channel.

Analysis shows that the total delay must be under 10 ms in the slowest case to meet system requirements. This is not achievable with the current setup, so new hardware and software implementations are necessary for the detector unit.

To maintain constant angular velocity when rotating the final head, a better velocity controller is needed (see Section 4.3.3). Changing the motor or increasing the gear ratio could potentially help increase torque, improving performance at low speeds where friction has the greatest impact.

Although not analyzed in the report, it may be possible to use the measured sweep time (from the detector trigger) with a particle filter to estimate the position. The sweep time calculation is implemented, but the software trigger is too slow to react reliably. If the system response could be improved, this method could be viable for estimating overshoot after a trigger is received. Further work is needed to evaluate this approach.

To reduce communication delay, one suggestion is to allow direct communication between the detector and transmitter, avoiding intermediate devices. This would reduce latency, hence improving performance.

Another idea, not yet tested, is to use the PSD's position data while the laser passes over it. However, this still requires detecting the PSD's location in the first place—bringing us back to the original challenge.

To improve the trigger-based method, several alternative strategies are suggested below. These rely on detecting the voltage spike generated when the beam crosses the PSD and use different assumptions to refine the angle estimate.

- **S1** Delay Compensation: Similar to the implemented trigger based method, but accounts for system delay using an estimated angular velocity to adjust the midpoint accordingly.

- **S2** Bidirectional Averaging: Perform sweeps from both directions and average the two midpoint estimates. This cancels out symmetrical delays under the assumption of consistent timing and accurate velocity estimation.

- **S3** Slow Sweep: Reduce the angular velocity to minimize delay impact and avoid the need for compensation. This method trades accuracy for speed and is most likely hard to achieve because of the nonlinear friction presented in Section 3.2.

Each of these suggestions presents a different balance between speed, computational complexity, and robustness to delay. A qualitative comparison is provided in Table 7.1.

Table 7.1: Comparison of suggested localization strategies for estimating $\theta_{reference}$

Suggestion	**Latency Tolerance**	**Speed**	**Complexity**
Delay Compensated	Medium	Medium	High
Bidirectional Average	High	Medium	Medium
Slow Sweep	Very High	Low	High

These ratings are based from estimations and should be interpreted as approximations. Further investigation, including simulation or experimental validation, is recommended to assess their practical applicability and effectiveness.

In conclusion, while reference angle detection using modulation frequency measuring and trigger-based techniques are technically possible, accurate distance estimation requires very high angular resolution and careful compensation for delay and signal limitations. Further research and hardware improvements, particularly in frequency measurement and low-latency communication, are needed to enable robust and accurate localization.

Ethical considerations

This project was developed exclusively for civilian and industrial use, specifically targeting applications in laser-based precision measurement and alignment. While technologies of this type may be repurposed for military contexts, such as aiding in the installation of defense equipment, this was neither the intention nor the scope of the thesis.

The system uses a visible laser for alignment, which introduces safety concerns, particularly regarding potential harm to eyesight. To handle this, all development and testing adhered to established laser safety standards imposed by the host company.

Beyond physical safety, the ethical implications of automation were also taken into consideration. Automating previously manual alignment tasks may impact certain roles in industrial settings. However, the system was designed to assist human operators rather than replace them, improving measurement precision and reducing human error in repetitive tasks.

Summary

- **RQ1:** Both the PID and LQI controllers met the alignment precision requirement of ±2 mm at 40 m distance. The LQI controller demonstrated lower overshoot and comparable steady-state performance, making it more suitable for smooth, stable operation.

- **RQ2:** The localization approach using PSD timing and encoder data is theoretically feasible but limited by the combined system delay. Without further optimization of signal processing and communication, full real-time localization is not achievable at long distances or high sweep speeds.

- **Ethics:** The project adhered to responsible engineering practices, with particular attention to automation impacting human operators, and the intended civilian application of the results.

Bibliography

[1] Select Laser Alignment. *The Importance of Precision Alignment in Manufacturing Industries*. Accessed: 2025-01-21. May 2024. URL: `https://selectlaseralignment.com/the-importance-of-precision-alignment-in-manufacturing-industries/`.

[2] Easy-Laser. *High Precision and Long Distance with XT22*. Accessed: 2025-01-21. June 2023. URL: `https://easylaser.com/en-us/about-easy-laser/blog/high-precision-and-long-distance-with-xt22`.

[3] iC-Haus GmbH. *iC-PZ Series - High-Resolution Reflective Absolute Encoder*. Accessed: 2025-01-29. 2025. URL: `https://www.ichaus.de/product/ic-pz-series/#documents`.

[4] Easy-Laser AB. *Download Easy-Laser manuals*. `https://easylaser.com/en-us/support/manuals`. Accessed: 2025-02-03. 2025.

[5] Shaher Dwik and Natarajan Somasundaram. “Modeling and Simulation of Two-Dimensional Position Sensitive Detector (PSD) Sensor”. In: *International Journal of Innovative Technology and Exploring Engineering* 9 (Nov. 2019), pp. 744–753. DOI: `10.35940/ijitee.A4226.119119`.

[6] Torkel Glad and Lennart Ljung. *Control Theory: Multivariable and Nonlinear Methods*. 1st ed. CRC Press Inc., 2000. ISBN: 9780748408788.

[7] Lennart Ljung, Torkel Glad, and Anders Hansson. *Modeling and Identification of Dynamic Systems*. 2nd ed. Lund: Studentlitteratur, 2021. ISBN: 978-91-44-15345-2.

[8] “An Optimized Cascade PI-D-P Controller for Speed Control in DC Motors”. In: *Proceedings of the IEEE Conference*. IEEE, 2022. URL: `https://ieeexplore.ieee.org/document/9730640`.

[9] George Ellis. “Chapter 6 - Four Types of Controllers”. In: *Control System Design Guide (Fourth Edition)*. Ed. by George Ellis. Fourth Edition. Boston: Butterworth-Heinemann, 2012, pp. 97–119. ISBN: 978-0-12-385920-4. DOI: `https://doi.org/10.1016/B978-0-12-385920-4.00006-0`. URL: `https://www.sciencedirect.com/science/article/pii/B9780123859204000060`.

[10] Carlos Canudas de Wit, K. Braun, and Karl Johan Åström. “Adaptive Friction Compensation in DC-Motor Drives”. In: *IEEE Transactions on Robotics and Automation* RA-3 (1987), pp. 681–685. URL: `https://portal.research.lu.se/en/publications/adaptive-friction-compensation-in-dc-motor-drives-2`.

[11] Joanna Piasek et al. “Identification of a Dynamic Friction Model and Its Application in a Precise Tracking Control”. In: *Acta Polytechnica Hungarica* 16 (Jan. 2019), pp. 83–99. DOI: `10.12700/APH.16.10.2019.10.6`.

[12] Janki Chotai and Krupa Narwekar. “Modelling and position control of brushed DC motor”. In: *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*. 2017, pp. 1–5. DOI: `10.1109/ICAC3.2017.8318792`.

[13] Bengt Lennartson. *Regelerteknikens grunder*. Swedish. 4:9. Revised in 2002. Studentlitteratur AB, 2000, p. 528. ISBN: 9789144024165.

[14] Karl Johan Åström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008. ISBN: 978-0691135762. URL: `https://fbsbook.org`.

[15] Sharp Corporation. *GP1S53V/GP1S58V Compact Photointerrupter Datasheet*. URL: `https://www.sharp-world.com/products/device/`.

[16] Bhavina Rathod, Nitesh Jamliya, and Keerti Vashishtha. “Cascade Control of DC Motor with Advance Controller”. In: *International Journal of Electrical, Electronics and Data Communication* 2.3 (2014), pp. 132–134. ISSN: 2320-2084. URL: `http://www.iraj.in/journal/journal_file/journal_pdf/1-42-1395665830132-134.pdf`.

[17] MathWorks. *Designing Cascade Control System with PI Controllers*. Accessed: 2025-04-23. 2025. URL: `https://se.mathworks.com/help/control/ug/designing-cascade-control-system-with-pi-controllers.html`.

[18] Fernando G. Martins. “Tuning PID Controllers using the ITAE Criterion”. In: *International Journal of Engineering Education* 21.5 (2005). Accepted 30 May 2005, pp. 867–873. URL: `https://www.ijee.ie/articles/Vol21-5/Ijee1673.pdf`.

[19] Sharon P. Simon, Lillie Dewan, and M. P. R. Prasad. “Design and Analysis of ITAE Tuned Robust PID Controller for Brushed DC Motor”. In: *2023 International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2023, pp. 1–6. DOI: `10.1109/ISCV57856.2023.10028938`. URL: `https://ieeexplore.ieee.org/document/10028938`.

[20] Hassan K. Khalil. *Nonlinear Systems: Pearson New International Edition*. 3rd. Pearson, 2013. ISBN: 9781292053851.

[21] Pu Yang et al. “Improved PID Friction Feed-forward Compensation Control Based on Segment Friction Model”. In: *The Open Automation and Control Systems Journal* 6 (2014), pp. 1620–1628. URL: `https://www.researchgate.net/publication/288016751_Improved_PID_Friction_Feed-forward_Compensation_Control_Based_on_Segment_Friction_Model`.

[22] H. Olsson et al. “Friction Models and Friction Compensation”. In: *European Journal of Control* 4.3 (1998), pp. 176–195. ISSN: 0947-3580. DOI: `https://doi.org/10.1016/S0947-3580(98)70113-X`. URL: `https://www.sciencedirect.com/science/article/pii/S094735809870113X`.

[23] Case Western Reserve University. *Signal Processing in Control Systems: Techniques and Trends*. Accessed: 2025-04-22. Aug. 2024. URL: `https://online-engineering.case.edu/blog/signal-processing-control-systems-techniques-trends`.

[24] Qiang Li et al. “Kalman Filter and Its Application”. In: *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. 2015, pp. 74–77. DOI: `10.1109/ICINIS.2015.35`.

[25] Sang-Choel Lee and Ju H. Park. “Performance improvement of PI controller with nonlinear error shaping function: IDA-PBC approach”. In: *Applied Mathematics and Computation* 215.10 (2010), pp. 3620–3630. ISSN: 0096-3003. DOI: `https://doi.org/10.1016/j.amc.2009.10.058`. URL: `https://www.sciencedirect.com/science/article/pii/S0096300309009746`.

[26] Henrik Andersson. “Position Sensitive Detectors – Device Technology and Applications in Spectroscopy”. Supervisors: Prof. Hans-Erik Nilsson, Dr. Göran Thungström. Doctoral thesis. Sundsvall, Sweden: Mid Sweden University, 2008. ISBN: 978-91-85317-91-2. URL: `https://www.diva-portal.org/smash/get/diva2:1939/FULLTEXT01.pdf`.