DEVELOPMENT OF PUMP CONTROLLED DOUBLE ROD ELECTRO-HYDROSTATIC ACTUATOR

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

CUMHUR ÖZBAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

JANUARY 2025

Approval of the thesis:

DEVELOPMENT OF PUMP CONTROLLED DOUBLE ROD ELECTRO-HYDROSTATIC ACTUATOR

submitted by **CUMHUR ÖZBAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Naci Emre Altun
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Serkan Dağ
Head of Department, **Mechanical Engineering** ________________

Assist. Prof. Dr. Hakan Çalışkan
Supervisor, **Mechanical Engineering, METU** ________________

Examining Committee Members:

Assoc. Prof. Dr. Kıvanç Azgın
Mechanical Engineering, METU ________________

Assist. Prof. Dr. Hakan Çalışkan
Mechanical Engineering, METU ________________

Assoc. Prof. Dr. Mehmet Bülent Özer
Mechanical Engineering, METU ________________

Assoc. Prof. Dr. Ulaş Yaman
Mechanical Engineering, METU ________________

Assoc. Prof. Dr. Onur Özcan
Mechanical Engineering, Bilkent University ________________

Date:08.01.2025

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Cumhur Özbaş

Signature :

ABSTRACT

DEVELOPMENT OF PUMP CONTROLLED DOUBLE ROD ELECTRO-HYDROSTATIC ACTUATOR

Özbaş, Cumhur
M.S., Department of Mechanical Engineering
Supervisor: Assist. Prof. Dr. Hakan Çalışkan

January 2025, 173 pages

This thesis addresses the modeling, state estimation, and robust control of Electro-Hydrostatic Actuators (EHA) for high-precision applications. A detailed mathematical model of the EHA system, encompassing hydro-mechanical, electrical, and control subsystems, is developed. Parametric uncertainties due to environmental and operational variations are systematically incorporated to ensure realistic plant behavior across a wide range of conditions.

State estimation techniques, Kalman filters, are implemented for enhanced noise rejection and fault detection. Simulation and experimental results validate the robustness and accuracy of the proposed estimation algorithms.

A Quantitative Feedback Theory (QFT)-based framework is employed for the robust design of velocity and position controllers. Leveraging Particle Swarm Optimization (PSO), the controllers are synthesized to meet stringent performance specifications. The cascade control structure effectively handles the bandwidth separation between inner and outer loops, ensuring high performance and robustness.

The integrated approach, combining detailed modeling, state estimation, and optimized robust control design, achieves significant advancements in actuator reliability, noise rejection, and compliance with aviation standards. This work establishes a comprehensive foundation for the deployment of EHA systems in safety-critical flight control applications.

Keywords: Electro-hydrostatic actuators, Nonlinear modeling, Robust control, Kalman filter, Controller optimization

ÖZ

POMPA KONTROLLÜ ÇİFT ETKİLİ ELEKTRO HİDROLİK EYLEYİCİ GELİŞTİRİLMESİ

Özbaş, Cumhur

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Dr. Öğr. Üyesi. Hakan Çalışkan

Ocak 2025 , 173 sayfa

Bu tez, yüksek hassasiyetli uygulamalar için Elektro-Hidrostatik Eyleyicilerin (EHA) modellenmesi, durum kestirimi ve dayanıklı kontrol tasarımını ele almaktadır. Hidromekanik, elektriksel ve kontrol alt sistemlerini kapsayan ayrıntılı bir matematiksel model geliştirilmiştir. Çevresel ve operasyonel değişkenliklerden kaynaklanan parametrik belirsizlikler, geniş bir çalışma koşulları aralığında gerçekçi sistem davranışı sağlamak için sistematik olarak modele entegre edilmiştir.

Durum kestirimi için Kalman filtreleri kullanılarak gürültü reddi ve hata tespiti sağlanmıştır. Simülasyon ve deneysel sonuçlar, önerilen kestirim algoritmalarının dayanıklılığını ve doğruluğunu doğrulamaktadır.

Hız ve konum denetleyicilerinin dayanıklı tasarımı için Nicel Geri Besleme Teorisi (QFT) tabanlı bir çerçeve kullanılmıştır. Parçacık Sürü Optimizasyonu (PSO) yöntemi ile denetleyiciler, katı performans gereksinimlerini karşılayacak şekilde tasarlanmıştır. Kademeli kontrol yapısı, iç ve dış döngüler arasındaki bant genişliği ayrımını etkin bir şekilde ele alarak yüksek performans ve dayanıklılık sağlamaktadır.

Detaylı modelleme, durum kestirimi ve optimize edilmiş dayanıklı kontrol tasarımını bir araya getiren bu bütüncül yaklaşım, eyleyicinin güvenilirliği, gürültü reddi ve havacılık standartlarına uyumunda önemli ilerlemeler sağlamaktadır. Bu çalışma, EHA sistemlerinin güvenlik açısından kritik uçuş kontrol uygulamalarında kullanımı için kapsamlı bir temel oluşturmaktadır.

Anahtar Kelimeler: Elektro-hidrostatik eyleyiciler, Doğrusal olmayan modelleme, Gürbüz kontrol, Kalman filtresi, Kontrolcü optimizasyonu

To my country...

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my thesis supervisor, Assist. Prof. Dr. Hakan ÇALIŞKAN, for his invaluable support and involvement throughout this journey.

I am deeply indebted to my manager, Ahmet Can AFATSUN, at Repkon Dynamics Inc., for providing me with the opportunity and environment necessary to conduct this research.

Special thanks go to Prof. Dr. Tuna BALKAN for his insightful guidance during the project.

I had the privilege of conducting my research in a stimulating and supportive environment, alongside my friends Furkan BAHÇELİ, Ömer Faruk TURHAN, and Furkan Enes YILDIRIM, who began this project with me. Their sharp minds and unwavering presence enriched my experience immensely. I hope our friendship lasts a lifetime.

I also extend my gratitude to Fırat MERT for his dedication and outstanding technical support, which greatly contributed to the success of this project.

I feel blessed to have Esra Selin EMİRMUSTAFAOĞLU by my side throughout this journey and beyond.

Lastly, I would like to thank my family for their unwavering support and love throughout my life.

This study was supported by TEYDEB under project no. 3210322, and I am grateful for their funding and trust in this research.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

AI	Artificial Intelligence
CAD	Computational Fluid Dynamics
CFD	Computer-Aided Design
DC	Direct Current
EMF	Electromotive Force
EHA	Electro-Hydrostatic Actuator
EMA	Electro-Mechanical Actuator
$F/A-18$	Fighter/Attack Aircraft 18
$F-35$	Fifth-generation multi-role fighter jet
FBW	Fly-By-Wire
FCS	Flight Control Surfaces
FFT	Fast Fourier Transform
GM	Gain Margin
KF	Kalman Filter
MEA	More Electric Aircraft
MSP	Magnetostrictive Position Sensor
PBW	Power-By-Wire
PLC	Programmable Logic Controller
PVA	Position Velocity Acceleration
$PMSM$	Permanent Magnet Synchronous Motor
PM	Phase Margin
PSO	Particle Swarm Optimization

$RMSE$	Root Mean Square Error
QFT	Quantitative Feedback Theory

LIST OF SYMBOLS

SYMBOLS

A_p	Effective piston area $[\mathrm{m}^2]$
b	Viscous friction coefficient of the actuator $[\mathrm{N} \cdot \mathrm{s/m}]$
b_{eq}	Damping coefficient of the pump-motor couple $[\mathrm{N} \cdot \mathrm{s/m}]$
C_c	Leakage coefficient between the chambers of the cylinder $[\mathrm{m}^3/\mathrm{s}/\mathrm{Pa}]$
C_{eq}	Total leakage coefficient $[\mathrm{m}^3/\mathrm{s}/\mathrm{Pa}]$
C_p	Inlet leakage coefficient of the pump $[\mathrm{m}^3/\mathrm{s}/\mathrm{Pa}]$
C_r	Outlet leakage coefficient of the pump $[\mathrm{m}^3/\mathrm{s}/\mathrm{Pa}]$
D_p	Pump displacement $[\mathrm{m}^3/\mathrm{rev}]$
$F(s)$	Prefilter transfer function []
F_c	Coulomb friction force [N]
F_f	Friction force [N]
F_L	External load force [N]
F_s	Static friction force [N]
$G(s)$	Position controller []
$G_v(s)$	Velocity controller []
i	Torque-generating current [A]
i_{ref}	Reference current [A]
J	Cost function for position controller design []
J_v	Cost function for velocity controller design []
J_{eq}	Inertia of the pump-motor couple $[\mathrm{kg} \cdot \mathrm{m}^2]$
k_E	Back-emf constant $[\mathrm{V}/(\mathrm{rad/s})]$
k_T	Torque constant $[\mathrm{Nm/A}]$

L	Motor inductance [H]
$L(s)$	Open-loop transfer function []
$M(s)$	Compliance bound []
m_p	Mass of the piston [kg]
n_p	Number of pole pairs of the motor []
P_0	Initial covariance matrix []
$P_\omega(s)$	Plant transfer function between motor speed reference and piston position []
p_A	Pressure in the A chamber of the hydraulic actuator $[\mathrm{Pa}]$
p_B	Pressure in the B chamber of the hydraulic actuator $[\mathrm{Pa}]$
p_D	$p_A - p_B$ $[\mathrm{Pa}]$
p_r	External pressure of the pump (accumulator circuit pressure) $[\mathrm{Pa}]$
Q	Flowrate $[\mathrm{m}^3/\mathrm{s}]$
R	Armature resistance $[\Omega]$
$T_{\text{low}}(s)$	Lower bound for reference tracking []
$T_{\text{up}}(s)$	Upper bound for reference tracking []
T_L	Torque load of the pump $[\mathrm{N}\cdot\mathrm{m}]$
T_m	Motor torque $[\mathrm{N}\cdot\mathrm{m}]$
T_s	Sample time [s]
V	Motor voltage [V]
V_d	Dead volume $[\mathrm{m}^3]$
W_s	Stability constant []
x	Position of the piston $[\mathrm{m}]$
$\mathbf{F}$	State transition matrix []
$\mathbf{G}$	Input control matrix []
$\mathbf{H}$	Measurement matrix []
$\mathbf{K}_n$	Kalman gain at time step n []

$\mathbf{P}_{n+1\|n}$	Predicted covariance matrix at time step $n+1$ []
$\mathbf{P}_{n\|n}$	Covariance matrix of the state estimate at time step n []
$\mathbf{Q}$	Process noise covariance matrix []
$\mathbf{R}_n$	Measurement noise covariance matrix []
$\mathbf{u}_n$	Control input vector []
$\mathbf{v}_n$	Measurement noise vector []
$\mathbf{w}_n$	Process noise vector []
$\mathbf{x}_n$	State vector at time step n []
$\hat{\mathbf{x}}_{n\|n}$	Estimated state vector at time step n []
$\hat{\mathbf{x}}_{n+1\|n}$	Predicted state vector at time step $n+1$ []
β	Bulk modulus of the fluid $[\mathrm{Pa}]$
ΔP	Pressure difference across the piston $[\mathrm{Pa}]$
$\dot{p}_A$	Pressure dynamics of chamber A $[\mathrm{Pa/s}]$
$\dot{x}$	Velocity of the piston $[\mathrm{m/s}]$
$\ddot{x}$	Acceleration of the piston $[\mathrm{m/s^2}]$
η_v	Volumetric efficiency of the pump $[\%]$
ω	Angular speed of the motor-pump couple $[\mathrm{rad/s}]$
ω_{ref}	Reference angular speed $[\mathrm{rad/s}]$
ω_n	Natural frequency $[\mathrm{rad/s}]$
ζ	Damping ratio []

CHAPTER 1

INTRODUCTION

1.1 What is Electro-Hydrostatic Actuator?

The term "Electro-Hydrostatic Actuator" (EHA) integrates three fundamental concepts: electro, hydrostatic, and actuator. An actuator is a device producing mechanical force by means of pressurized fluid [1]. In the context of EHAs, a cylinder transmits precise translational motion and force. The motion is driven by hydrostatic force, which is generated by the pressure difference in the fluid within the cylinder chambers. An electric motor-pump combination energizes this fluid, hence the term "electro." The movement of the actuator can be controlled by varying the speed of the motor pump unit, adjusting the displacement of the pump, or both. As shown in Figure 1.1, a hydraulic schematic illustrates the configuration of an EHA with a fixed-displacement pump.

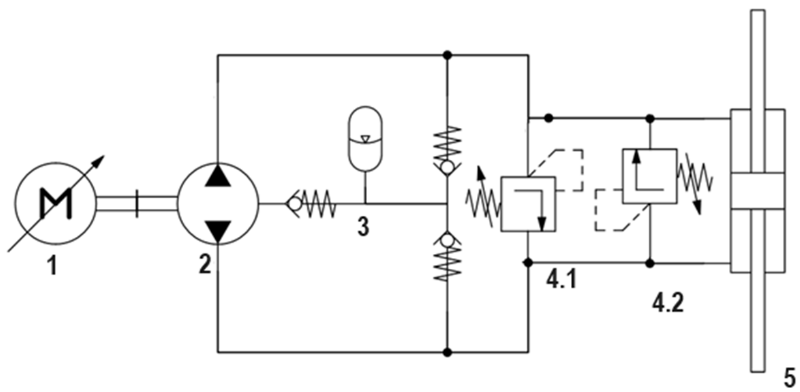

Figure 1.1: Hydraulic circuit of EHA

In this setup, the electric motor (1) converts electrical energy into rotational mechanical energy, while the hydraulic pump (2) pressurizes the hydraulic fluid and supplies the required fluid volume. In the actuator chambers (5), the pressure difference drives mechanical motion. The accumulator (3) compensates for losses due to leakage, and the relief valves (4.1 and 4.2) ensure the circuit is not over-pressurized. To comprehend importance of this hydraulic system, the context of aviation hydraulics is required.

1.2 History of Flight Control Actuators

Hydraulic systems have long been integral to a wide array of industrial, mobile, and aerospace applications, offering unmatched power transmission efficiency and control precision. These systems leverage the incompressibility of fluids to transmit energy, enabling the handling of substantial loads and forces with compact components. One of the key advantages of hydraulics is their high power-to-weight ratio, allowing for efficient operation in space-constrained environments [2] [3].

The first aircraft had no hydraulics. Beginning with the Wright Brothers' design, aircraft have utilized aerodynamic devices called flight control surfaces (FCS) to allow pilots to control flight attitudes. FCS generally possess a rotational degree of freedom facilitated by a hinge structure, enabling the surface angle to influence the direction of aerodynamic forces and maneuver the aircraft. In modern systems, these surfaces are linked to actuators; however, in early aircraft designs, they were controlled manually by the pilot using cables and pulleys, as shown in Figure 1.2. By World War II, aircraft became faster and heavier, necessitating an additional power source. This led to the adoption of hydraulic power systems, which offered a high power-to-weight ratio and met the demands of more advanced aircraft designs. Hydraulic actuation systems were integrated into landing gears, rudders, ailerons, flaps, elevators, and doors. Firstly hydraulic boosters were integrated, where the pilot could still feel mechanical maneuver. Soon complete power-operated control eliminated the pilot feel. Hence role of the pilot was reduced to signaling, no longer transmitting power.

In the 1960s, as electronics became prevalent, mechanical linkages were replaced by electrical control systems, reducing the space and weight occupied by traditional

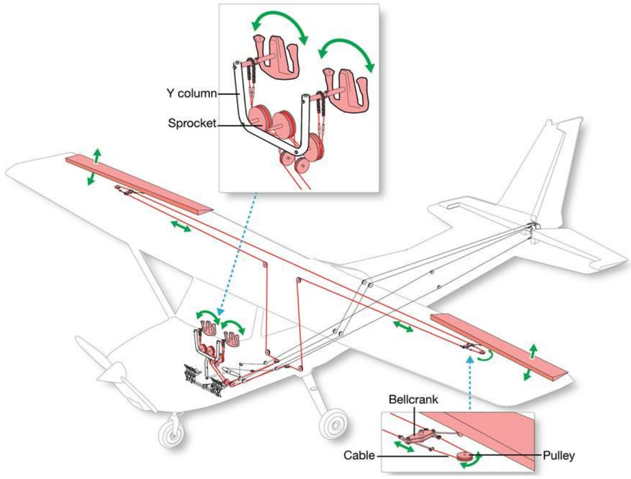

Figure 1.2: Cessna-172N aileron system [4]

mechanisms. This transition introduced the concept of Fly-By-Wire (FBW), where electronics transmit control commands instead of mechanical links. The first civil FBW aircraft was the Airbus A320 [5]. The use of electrical signals enabled the digital computation of control inputs, allowing real-time adjustments based on the aircraft's system characteristics.

Early FBW systems typically integrate a central hydraulic system to provide the necessary power for flight control actuators. These systems rely on a primary pump energized by an engine, with servo valves controlling the pressurized hydraulic fluid supplied to the actuators [6]. A central hydraulic system comprises hydraulic pipes, reservoirs, and other accessories, which significantly contribute to the aircraft's weight and occupy valuable space. Additionally, valve-controlled systems require a constant supply of high-pressure fluid at the valve inlet to ensure rapid response to control commands. This design, however, results in internal leakage within the valves, causing energy losses and thermal management challenges. These challenges are often addressed with larger hydraulic reservoirs or dedicated thermal management systems, as described in [7]. A typical central hydraulic system is illustrated in Figure 1.3 [8]. Hence the concept of replacing central hydraulic systems with More Electric Aircraft (MEA), employing electrical actuators, has gained traction. As advancements in motor technology improve the power-to-weight ratio of electric motors, Electro-Mechanical Actuators (EMAs) have begun to emerge as viable alternatives for flight

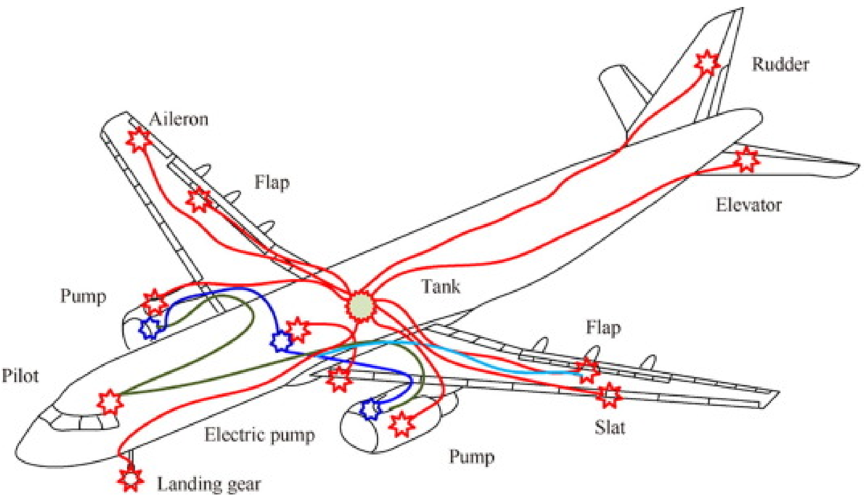

Figure 1.3: Central hydraulic system [8]

control systems. However, EMAs still fall short of hydraulics in terms of power density and face issues such as mechanical wear and jamming [9] [10]. Nevertheless, the idea of integrating electric motors to energize actuators persists. Consequently, the evolving concept of Power-By-Wire (PBW), which involves the use of electrical power distribution and decentralized hydraulic systems, represents a growing trend. In this paradigm, Electro-Hydrostatic Actuators (EHAs) play a key role, combining the reliability of hydraulics with the flexibility of electrical systems to enhance aircraft performance and reduce overall system complexity. The transition to such electrically powered systems is rather smooth. Electrically powered systems initially served as redundant backup systems to conventional hydraulic systems, as illustrated in Figure 1.4 [11].

Now it should be clearer, in the context of MEA trends in aviation why EHAs are advantageous. They are not connected to a central system by pipes, saving space and maintenance costs. Additionally, EHAs function as a single, self-contained subsystem, making them easier to replace and maintain. The integration of the motor, pump, and actuator into a compact system further enhances this modularity. Furthermore, EHAs are more energy-efficient than conventional hydraulic systems, as the pump only operates when required, rather than running constantly [12].

The first operational use of EHAs in military aviation occurred when they replaced the left aileron on the F/A-18, evaluated in the Systems Research Aircraft flight en-

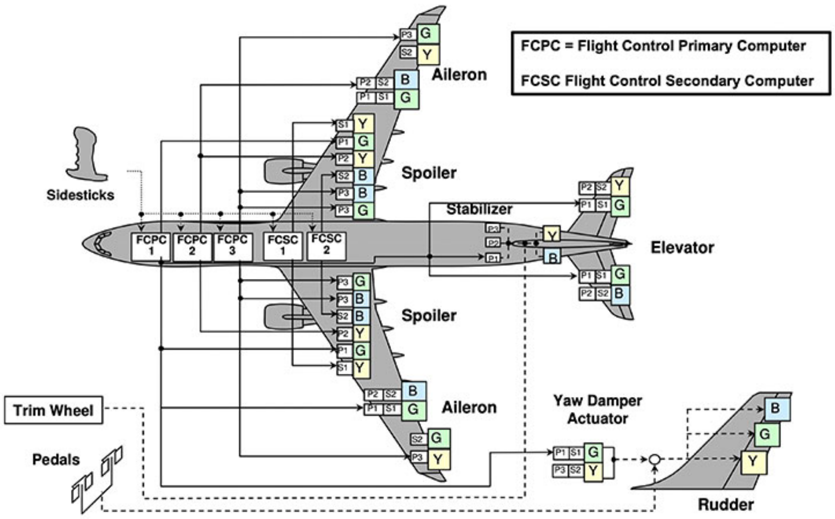

Figure 1.4: Airbus A-380 Control and backup systems [11]

velope [13]. In civil aviation, the Airbus A-380 became the first commercial aircraft to use EHAs, initially as a backup system for its flight control surfaces [14]. More recently, the F-35 fighter jet has integrated EHAs for both primary and secondary control surfaces, reflecting the growing trend towards this technology in modern aircraft [15] [16].

1.3 Motivation Behind This Study

EHAs are a key PBW technology that is increasingly replacing conventional hydraulic systems in aviation. In modern aircraft, fault-tolerant actuator design plays a critical role, as aircraft systems are designed with redundancy to ensure safety and reliability during failures [17]. Consequently, robust control strategies for flight control actuators are essential for maintaining stability and performance under fault conditions.

In addition to robustness, reconfigurable controller designs for actuators are vital in optimizing flight control in the presence of faults. These controllers can dynamically adjust to new fault modes, ensuring operational continuity and safety [18] [19]. Beyond fault tolerance, the integration of autonomy and artificial intelligence (AI) in next-generation aircraft enables the use of adaptive control strategies, where the controller structure evolves based on changing flight conditions or mission requirements

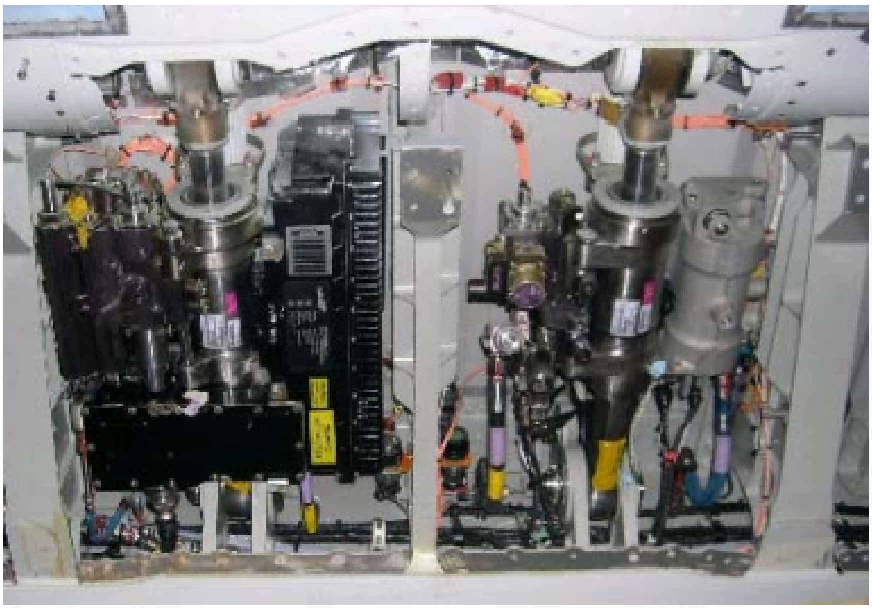

Figure 1.5: Aileron EHA and hydraulic actuator in Airbus A-380 [14]

[20].

Modernization efforts in aircraft systems often involve continuous life-cycle updates, ensuring systems remain relevant and capable of integrating advanced technologies. This trend, coupled with the growing emphasis on fault-tolerant control and adaptive strategies, suggests a paradigm shift in flight control actuator design. Specifically, actuator specifications—such as bandwidth, stiffness, and performance requirements—could become dynamic, adapting in real time to optimize system performance in varying conditions.

To support this evolving landscape, automated robust controller design methodologies are necessary. Hence, this study represents a step toward realizing the future of intelligent flight control systems.

1.4 Thesis Outline

This thesis is organized as follows: In Chapter 1, the working principle of the EHA, along with the historical context of flight control actuators, is presented. In this context, the motivation for the thesis is highlighted.

In Chapter 2, a comprehensive literature review on EHA design and control strategies addressing design limitations is provided. Additionally, background information on the Quantitative Feedback Theory (QFT) control method is included.

In Chapter 3, the design process of the EHA, encompassing component selection and simulation verification, is detailed.

In Chapter 4, the controller design for the EHA is presented. Specifically, a Kalman filter is implemented, and various structural configurations are analyzed. A robust QFT controller, automatically developed through an optimization method, is proposed.

In Chapter 5, the performance results of the proposed EHA system and its controller are summarized and discussed.

CHAPTER 2

LITERATURE SURVEY AND BACKGROUND

2.1 EHA Design

As the trend of MEA emerged in the industry, the application of electric motors in flight control actuator systems was studied. One of the pioneering works on EHA focused on electric motors, suggesting the use of permanent magnet motors instead of brushless DC motors [21]. Anderson demonstrates component selection based on actuator specifications and points out that motor size is a significant factor for compactness, hence suggesting variable displacement pumps that require smaller torque power. He investigates parameters such as piston area, motor speed, and pump displacement in terms of the stiffness of the actuator, recommending small displacement and high-speed motors. A detailed examination of the actuator's stiffness is presented in his paper [22]. For the design specifications, Frischemeier [23] describes the functional requirements of EHA, analyzing design components to reduce the weight of the EHA. Habibi describes a methodology for EHA design [24]. The evaluation of EHA under flight conditions and according to flight specifications serves as a benchmark for the verification and design specifications of EHA [13].

Novel approaches to hydraulic circuit design were also applied to EHA. Kim developed a force-controlled system with an additional sliding mode controller regulating a bypass valve to enhance force tracking. An external position disturbance was introduced, and the EHA with the bypass valve was compared to the EHA without it, showing significant improvements in force tracking [25]. A similar approach using a hydraulic sub-circuit was adopted, where Rongie proposed a power regulator to enhance the dynamic performance of the EHA. An additional accumulator was designed to refeed the circuit via a proportional valve based on different control modes. Furthermore, a hydraulic lock valve was integrated to significantly improve stiffness when the system is locked. Test results demonstrated that the dynamic performance of the EHA was enhanced in terms of position tracking and frequency response. Additionally, when an external load was applied, the system exhibited improved stiffness [26]. The suggested hydraulic circuit is illustrated in Figure 2.1. Determining EHA

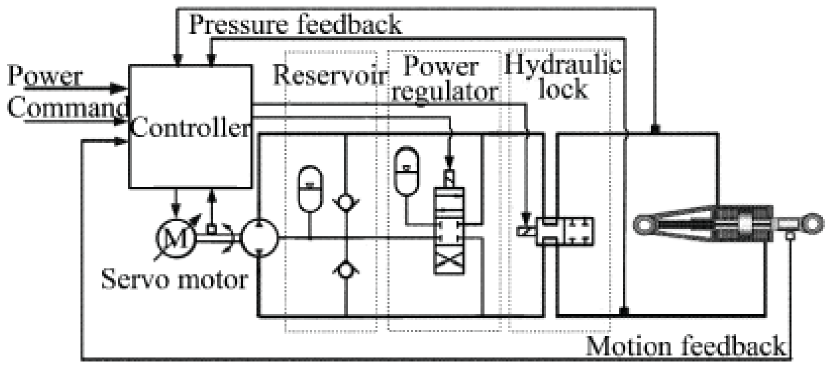

Figure 2.1: Power regulator sub-circuit and hydraulic lock integration on EHA [26]

components involves consideration of design specifications as well as the reduction of weight, heat generation, and efficiency of the EHA. Consequently, the optimization of EHA design is a common topic in the literature. One of the challenges is parameterizing components; hence, estimation methods have been developed to estimate parameters such as a motor's torque constant, resistance, or a pump's displacement based on the mass or other qualities of the components [27]. Multi-objective optimizations involving simulations are frequently conducted in the literature to propose optimal EHA designs [28] [29].

2.2 EHA Control

While performance specifications of EHA depend on sub-component properties, control methods also try to improve the tracking, stiffness, and dynamic performance of EHA. Control methods on EHA try to propose robust solutions to the nonlinearity of EHA, sensor-noise, and parametric uncertainties. One of the nonlinearities of EHA is dead zone, caused by static friction and leakage on the system [30]. Inner-loop control strategies are proposed to overcome the effects of dead zone [31]. Hu further developed an inner-loop control strategy with a sliding mode controller, for a fault-tolerant system. The results are compared with Kalman Filter estimations for fault as well as tracking performances [32]. Estimation methods are frequently used for fault detection [33], as well as noise rejection along with robust control strategies [34]. Machine learning techniques such as few-shot data augmentation are also used in fault detection of EHA [35]. Hence, it is common to integrate state estimation methods such as Kalman Filter either for improved controller response or fault detection algorithms. Various robust control algorithms are suggested for the EHA control in order to improve dynamic performance characteristics.

Cho suggested a Simple Adaptive Control in order to improve the tracking response of the system [36]. Lee suggested an adaptive anti-windup strategy to improve tracking response and disturbance load rejection [37]. Sliding mode controllers are also widely researched [38], in [39] optimal controller for a sliding mode controller is studied. In sliding mode controllers, obtaining full system information, disturbances, and inherent chattering due to sliding mode technique is a problem. In order to obtain full system information, observers and estimators are frequently utilized [40]. Yang proposed a solution to disturbance by adapting reaching law, demonstrating better performance results through simulation [41]. However, his study lacks experimental verification.

Another robust control method in the literature is Quantitative Feedback Theory (QFT) based control, involving modeling uncertainty of the plant. In the context of hydraulic flight control actuators, Thompson suggested a well-structured approach to improve the stiffness, and bandwidth of the actuator [42]. Work of Kang is the first publication of QFT methodology on EHA, emphasizing sensitivity constraint on sensor noise [43]. Self-tuning QFT controller was also proposed on hydraulic load simulator [44]. In recent works, fault-tolerant QFT controller on leakage of EHA was suggested [45].

2.3 QFT Control

Quantitative Feedback Theory (QFT) is a robust control design methodology that emphasizes achieving desired system performance in the presence of uncertainties and disturbances. QFT framework was first introduced by Isaac Horowitz, in 1962 [46]. QFT leverages frequency-domain techniques to explicitly account for plant parameter variations and external disturbances. The central idea of QFT is to design a controller that meets specified performance criteria, such as tracking, stability, and disturbance rejection, over a range of uncertain plant models. By shaping the open-loop transfer function to satisfy robust stability and performance requirements, QFT provides a systematic framework for balancing robustness and performance. While commercial tools are available for designing QFT controllers, this thesis develops custom algorithms in MATLAB to achieve the same goals. A great source for the application of QFT is *Robust Control Engineeringg* by Mario Garcia-Sanz [47].

One disadvantage of QFT method is loop shaping process, while boundaries for the controller specifications are clear and precise; there is no exact method available to design frequency-based controller. Open loop plant $L(s) = P(s)G(s)$, is shaped in Nichols chart often requiring manual intervention and expertise. Essentially, a lead-lag compensator is designed however its complexity is dependent on designer. In this thesis, an optimization method is suggested in order to synthesize controller respecting boundaries generated by QFT. Utilizing Particle Swarm Optimization (PSO), an automatic loop shaping process is suggested. Such approach is also common in literature [48, 49, 50]. Another advantage of optimization is that, while there are hard constraints for flight control actuators, it is almost always desirable to minimize controller effort. Hence, performance specifications are treated as strict requirements. Once these specifications are satisfied, the optimization algorithm reduces control effort, reflecting an engineering approach to the problem.

.

CHAPTER 3

EHA DESIGN AND SIMULATION

In this section, the design steps of the EHA and considerations during this process will be explained. Subsequently, the mathematical model of the EHA and relevant simulations based on the mathematical model will be analyzed alongside experimental data. A similar design procedure proposed in [24] will be followed. As a pre-concept phase, the availability of EHA components that are required in the market is investigated together with the literature survey. The conceptual design step of EHA is to determine key component constants based on specifications and derive a basic model of EHA in order to validate these specifications and iterate for component selection. Later, after components are selected and integrated into the physical EHA design, based on experimental data more detailed simulations will be followed.

3.1 Preliminary Design

During the preliminary design process, the system requirements and the sub-component selection process are explained. In aviation applications, compact and specially integrated parts are often produced to increase efficiency and reduce weight costs. Since this study is conducted in a laboratory environment, commercial products for pump, motor, and other auxiliary hydraulic elements should be selected.

3.1.1 System Specifications

The design specifications are based on the EHA evaluated for the F-18 program [13]. There is also an example study on specification determination of EHA [23]. The system specifications for the EHA are provided in Table 3.1. A dual redundant design will be conducted. Additionally, a compact design criterion is selected for the dimensions of the EHA; however, a detailed discussion of this criterion is beyond the scope of this thesis.

Table 3.1: Design specifications of redundant EHA

Criterion	**Value**
Maximum output force	59 kN
Maximum velocity	190 mm/s
Bandwidth	7 Hz
Stroke	120 mm
Linearity	0.2% command

3.1.2 Sub-Component Determination

First, the maximum system pressure is determined, as it directly influences the selection of hydraulic components such as pumps and auxiliary components. Since most commercial hydraulic components operate around 210 bar as rated pressure, the maximum system pressure is determined to be 240 bar, incorporating general safety margins. Beyond this value, relief valves should be activated to protect the system.

To meet the maximum output force requirement, the effective piston area A_p can be calculated using the pressure formula for force on a stationary piston:

$$A_p = \frac{F_{\text{load}}}{\Delta p} = \frac{59}{240}\,\frac{\text{kN}}{\text{bar}} = 2458\,\text{mm}^2 \tag{3.1}$$

The effective piston area is the difference between the piston and rod areas. The actuator design should be symmetrical to ensure that the in-flow and out-flow are equal, as suggested in [30]. While single-rod EHAs are commonly used in industries such as presses due to their efficiency, symmetrical actuators are preferred in aviation for better performance. For the actuator, the piston stroke should be designed as 120 mm.

For the motor-pump couple, the pump should work bi-directionally. For a compact design, higher speed motors are preferred since as motor torque increases motor dimensions increase for the same amount of power [51]. From the maximum velocity requirement, the maximum flow-rate of the system can be determined:

$$Q_{max} = A_p \cdot 190 \frac{\mathrm{mm}^3}{s} \simeq 28 \text{ L/min} \tag{3.2}$$

For the solution of equation 3.2, flow-rate dependence on pump displacement D_p and speed ω is required. For a fixed displacement pump disregarding volumetric efficiency:

$$Q_{max} = D_p \cdot \omega_{max} \tag{3.3}$$

A faster pump speed is better since the torque required to hold 60 kN at maximum pressure decreases. Disregarding mechanical efficiency simple relation can be shown:

$$\tau_{max} = \Delta p_{max} \cdot D_p \tag{3.4}$$

In aviation, EHAs commonly employ fixed-displacement axial-piston pumps due to their efficiency and reliability, operating at speeds of up to 22,500 rpm [52]. High-speed bi-directional pumps are desirable; however, such axial-piston pumps are not available for commercial use. While there are EHA designs utilizing other types of pumps, such as gear pumps [53], their application is typically specific to industrial systems. Therefore, for laboratory conditions, an external gear pump is selected as the pump type for the bi-directional pump due to its relative efficiency and reliability. Similarly, the study in [30] also employs a gear pump for an EHA.

Since the pump's speed is naturally limited by the manufacturer, the pump displacement is selected for an 8,000 rpm range, according to Equation 3.2. The motor's maximum torque is determined using Equation 3.3, ensuring an 8,000 rpm rotational speed. An accumulator is selected to compensate for expected leakage losses from the pump and cylinder. A detailed analysis is conducted to address potential pump cavitation. Additionally, buckling analysis of the hydraulic cylinder and the material type for load forces determine the rod diameter, thus piston diameter based on the required effective piston area [54].

Based on preliminary analysis of equations 3.1, 3.2, 3.3 and 3.4 sub-components are selected according to Table 3.2.

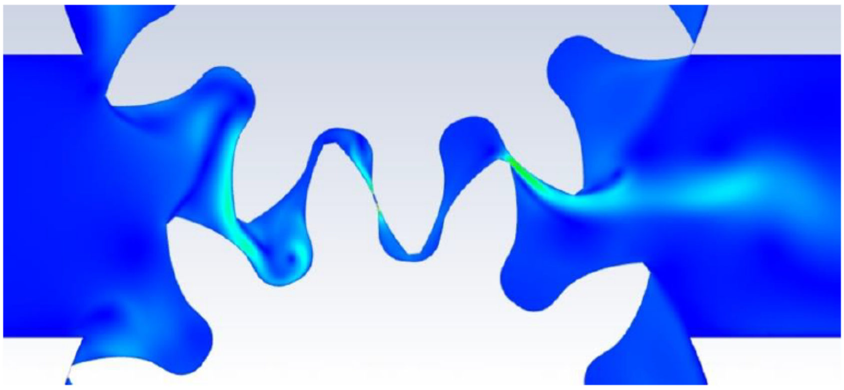

Figure 3.1: Velocity gradient of CFD analysis of external gear pump [54]

Table 3.2: Equipment specifications

Equipment	**Model**	**Description**
EHA pump	VIVOIL X1R2725FJJE	4.3 cc/rev displacement
EHA motor	BECKHOFF AM8053-1N10	15.6 A standstill current
Servo Drive	BECKHOFF AX5118	18 A nominal current
PLC	Beckhoff IPC C6015	Intel Atom® E3815, 1.46 GHz
Relief valve	DANFOSS CP211-1-B-6B-K-A	255 bar cracking pressure
Pressure sensor	HYDAC HDA4446-A-400-000	400 bar maximum range
Accumulator	HYDAC SB330-6A1/112U-330A	6 L nominal volume
MPS	OPKON 150-S152023-201	150 mm stroke
Actuator	-	2572 mm^2 effective piston area

3.2 Mathematical Model of EHA

EHA's mathematical model can be derived using the following relations, which are in parallel with the hydraulic models in the literature.

3.2.1 Electric motor model

A permanent magnet synchronous motor (PMSM) is used for the EHA. The dynamic equations are expressed in d,q rotor reference frame, and the driver settings of the manufacturer utilize this reference frame. The voltage equations expressed in the d-q frame are: [51]:

$$V_d = R_s i_d + \dot{\lambda_d} - \omega_e \lambda_q \tag{3.5}$$

$$V_q = R_s i_q + \dot{\lambda_q} - \omega_e \lambda_d \tag{3.6}$$

where:

- R_s: Stator resistance, in ohms (Ω).
- ω_e: Electrical angular speed of the rotor, in $\mathrm{rad/s}$.
- V_d, V_q: Direct and quadrature axis voltages, in volts (V).
- i_d, i_q: Direct and quadrature axis currents, in amperes (A).
- λ_d, λ_q: Direct and quadrature axis flux linkages, representing the magnetic flux linked with the rotor winding along the respective axes, in weber-turns (Wb).

The flux linkages in the d- and q-axes for a PMSM are given as:

$$\lambda_q = L_q i_q \tag{3.7}$$

$$\lambda_d = L_d i_d + \lambda_m \tag{3.8}$$

where:

- λ_m: Flux linkage from the permanent magnet, in Webers (Wb).
- L_d, L_q: Inductance along the direct and quadrature axes, respectively, in henries (H).

The electrical angular speed of the rotor has a relation with mechanical speed ω :

$$\omega_e = n_p \cdot \omega \tag{3.9}$$

where n_p is number of pole pairs.

Notice that 3.7 and 3.8 can be inserted into 3.6 and 3.5 respectively as time derivatives, and 3.9 can be integrated:

$$V_d = R_s i_d + L_d \frac{di_d}{dt} - n_p \omega (L_q i_q) \tag{3.10}$$

$$V_q = R_s i_q + L_q \frac{di_q}{dt} - n_p \omega (L_d i_d + \lambda_m) \tag{3.11}$$

The electromagnetic torque of the PMSM can be expressed in terms of the d-q currents as:

$$T_m = \frac{3}{2} n_p \left(\lambda_m i_q + (L_d - L_q) i_d i_q \right) \tag{3.12}$$

As explained later in this chapter, the currents i_d and i_q are controlled by the driver and its implemented control method. Field-Oriented Control (FOC) is one such method, implemented by the manufacturer Beckhoff. In this type of FOC, setting the i_d reference to zero allows for the decoupling and simplification of control. This can be observed from the governing equations (3.10), (3.11), and (3.12). Assuming i_d is zero results in a model similar to a DC motor, characterized by the torque constant k_T and the back-EMF constant k_E. In fact, the manufacturer publishes these constants in the technical data of the selected motor. Hence a simplified torque equation can be shown:

$$T_m = k_T i \tag{3.13}$$

Similarly, constants such as resistance and inductance can be simplified since only the dynamics of the q-axis is relevant:

$$V - iR - L\frac{di}{dt} - k_E \omega = 0 \tag{3.14}$$

Where V is the motor voltage, i is the torque-generating current, R is the armature resistance, L is the motor inductance, and ω is the angular speed of the motor-pump couple.

3.2.2 External gear pump model

The hydraulic pump is the main component of EHA. Coupled with the motor shaft, it transmits energy to excite the actuator. For the hydraulic pump, an external gear pump is preferred since it is bi-directional and has relatively high efficiency. The displacement is constant hence only drive speed varies for the controlling of the actuator position. Figure 3.2 illustrates the operating principle of an external gear pump.

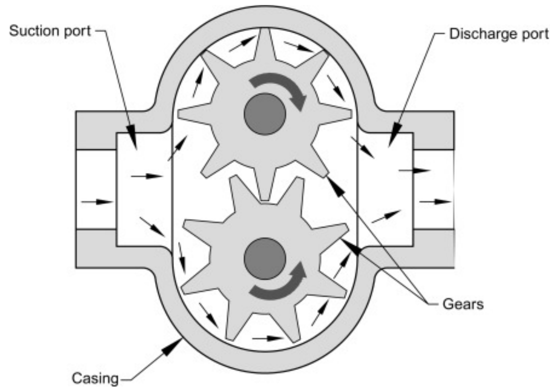

Figure 3.2: External gear pump [55]

For a fixed displacement pump, the flow equation of the pump, assuming laminar leakage losses, is written as follows:

$$\begin{aligned} Q_{\text{pump out}} &= D_p\omega - C_p(p_A - p_B) - C_r(p_A + p_B - 2p_r), \\ Q_{\text{pump in}} &= -D_p\omega + C_p(p_A - p_B) - C_r(p_A + p_B - 2p_r). \end{aligned} \tag{3.15}$$

Where $Q_{\text{pump out}}$ and $Q_{\text{pump in}}$ are outlet and inlet flow-rates of the external gear pump, D_p is the pump displacement, C_p and C_r are the inlet and outlet leakage coefficients of the pump, p_r is the external pressure of the pump corresponding to the pressure in the accumulator circuit, and p_A and p_B are the pressures in the A and B chambers of the hydraulic actuator, respectively. The torque requirement of the pump is given by:

$$T_L = D_p(p_A - p_B) \tag{3.16}$$

The equation of motion for the pump-motor couple is given by:

$$J_{eq}\dot{\omega} + b_{eq}\omega = T_m - T_L \tag{3.17}$$

Where J_{eq} is the inertia of the pump-motor couple, and b_{eq} is the damping coefficient of the pump-motor couple. Notice that friction is only modeled as viscous friction where coulomb friction of the motor-pump couple is neglected in this case.

3.2.3 Hydraulic accumulator model

Hydraulic accumulators are commonly operated as pulsation dampeners to manage pressure changes, serve as emergency sources, and function as energy storage equipment. In the EHA sub-circuit, the accumulator acts as a secondary energy provider to compensate for leakage losses from the pump and cylinder, allowing smooth flow continuity. Additionally, it operates as a hydraulic reservoir, where external leakage from the pump is returned to the accumulator sub-circuit. A gas-charged bladder accumulator is selected during the preliminary design stage due to its lighter weight compared to other types of accumulators.

The accumulator stores hydraulic energy resulting from pressure changes by adjusting its gas volume. It features an elastic bladder that acts as a membrane between the pressurized hydraulic fluid and the gas in the accumulator. The change in gas volume within the accumulator is modeled as a polytropic process, neglecting thermal effects:

$$P_g V_g = P_0 V_0 \tag{3.18}$$

where:

- P_g is the instantaneous gas pressure inside the accumulator (Pa),
- P_0 is the pre-charge gas pressure (Pa),
- V_g is the instantaneous gas volume inside the accumulator (m^3),
- V_0 is the pre-charge gas volume (m^3).

The gas volume V_g adjusts dynamically to changes in pressure P_g, maintaining the energy storage capability of the accumulator. Hence,

- When the hydraulic pressure increases, the bladder compresses, reducing the gas volume V_g and increasing the gas pressure P_g.
- When the hydraulic pressure decreases, the bladder expands, increasing V_g and reducing P_g.

While a more detailed thermal model can be utilized to account for the complex thermodynamic behavior of the gas inside the accumulator, the current non-linear behavior sufficiently achieves the primary objective of the accumulator: compensating for leakage losses in the system. To simplify the analysis, a linearized model can be adopted under the assumption that the accumulator pressure remains in equilibrium during operation. In this linear model, the accumulator effectively acts as an external reservoir with a constant pressure p_r, simplifying the mathematical representation of the accumulator's interaction with the system.

3.2.4 Actuator model

The actuator converts hydraulic energy into mechanical energy. There are two symmetric dead volumes, A and B, representing the total volumes of the hydraulic cavities and chambers of the cylinder. The hydraulic pump exerts energy into the fluid, resulting in compressed pressure and flow rate. The pressure change depends on the compressibility of the fluid. The hydraulic pump's rotation generates fluid flow, but leakages may decrease or increase the fluid flow into a cylinder chamber. The motion of the piston also shifts hydraulic fluid, resulting in an equivalent flow. The compressibility equation can be expressed as:

$$\beta \frac{dP}{dt} = \frac{Q_{\text{in}} - Q_{\text{out}}}{V} \tag{3.19}$$

where:

- β: Bulk modulus of the hydraulic fluid, representing fluid compressibility (Pa),

- P: Pressure in the hydraulic chamber (Pa),
- Q_{in}: Inflow rate of hydraulic fluid into the chamber (m^3/s),
- Q_{out}: Outflow rate of hydraulic fluid from the chamber (m^3/s),
- V_d: Volume of the hydraulic chamber (m^3).

This equation captures the dynamic relationship between pressure, flow rates, and the compressibility of the hydraulic fluid. Hence, for two symmetrical chambers A and B, compressibility relation in equation 3.19 can be formed. The pressure dynamics of chamber A is given by:

$$\dot{p}_A \frac{V_d + A_p x}{\beta} = Q_{\text{pump in}} - C_c(p_A - p_B) - A_p \dot{x} \tag{3.20}$$

Here, the left-hand side is the compressed flow, where capacitance is determined by the bulk modulus β. V_d is the volume of one of the chambers when the piston is in the middle position. Notice that the volume of the fluid changes with the piston position. A_p is the area of the piston, x is the position of the piston, and C_c is the leakage coefficient between the chambers of the cylinder. For reverse directions, the pressure dynamics of chamber B is given by:

$$\dot{p}_B \frac{V_d - A_p x}{\beta} = Q_{\text{pump out}} + C_c(p_A - p_B) + A_p \dot{x} \tag{3.21}$$

Notice that equations for pressure changes in chambers A and B are equivalent. Using equations 3.20 and 3.21, a load pressure state p_D can be defined by neglecting the swept volume by actuator position $A_p x$. The following simplification is obtained:

$$\dot{p}_D = \dot{p}_A - \dot{p}_B = \frac{2\beta}{V_d} \left[D_p \omega - \frac{C_r + 2C_c + 2C_p}{2} p_D - A_p \dot{x} \right] \tag{3.22}$$

The equation of motion for the hydraulic actuator is given by:

$$A_p(p_A - p_B) - F_L = m_p \ddot{x} + b\dot{x} \tag{3.23}$$

Where F_L is the external load, m_p is the piston mass, and b is the viscous friction. Notice that similar to the hydraulic pump equation, only a linear part of the friction, viscous friction is modeled. The coulomb friction may be considered inside the external load term. In a detailed non-linear model, a friction model of the actuator will be investigated. The reference of directions in the mathematical model of EHA is summarized in Figure 3.3, q_i, and q_e refer to internal and external leakage flow rates.

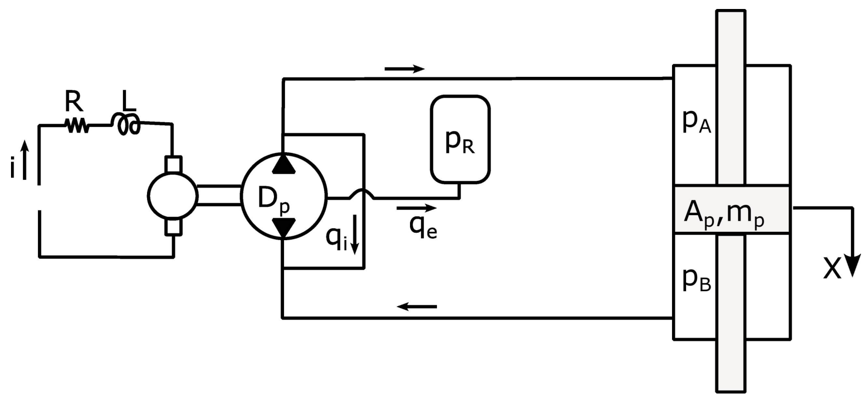

Figure 3.3: Physical directions for the mathematical model of EHA

Hence from the above equations, defining $C_{eq} = \frac{C_r+2C_c+2C_p}{2}$, a simple state space model for EHA can be derived as follows:

$$\begin{bmatrix} \frac{di}{dt} \\ \dot{\omega} \\ \dot{x} \\ \ddot{x} \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{k_E}{L} & 0 & 0 & 0 \\ \frac{k_T}{J_{eq}} & -\frac{b_{eq}}{J_{eq}} & 0 & 0 & -\frac{D_p}{J_{eq}} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{b}{m_p} & \frac{A_p}{m_p} \\ 0 & \frac{2\beta D_p}{V_d} & 0 & -\frac{2\beta A_p}{V_d} & -\frac{\beta C_{eq}}{V_d} \end{bmatrix} \begin{bmatrix} i \\ \omega \\ x \\ \dot{x} \\ p_d \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} V \tag{3.24}$$

Notice that in this state space model voltage is input, which is common in the literature as well. However, in practical application, a cascade loop structure for the motor speed is required.

3.3 Preliminary EHA Simulation

During this design process, the preliminary EHA simulation is studied in order to validate component selection and iterate if there is an unexpected issue. Hence for a more detailed simulation, a nonlinear simulation using the Simscape library in Simulink is generated compliant with the literature [56]. Nonlinear system characteristics such as changing volume of cylinder chambers, friction, and accumulator effects are observed. For cylinder friction, generic values are estimated later to be updated with respect to experimental verification. Linear model simulation of EHA is illustrated in Figure 3.4.

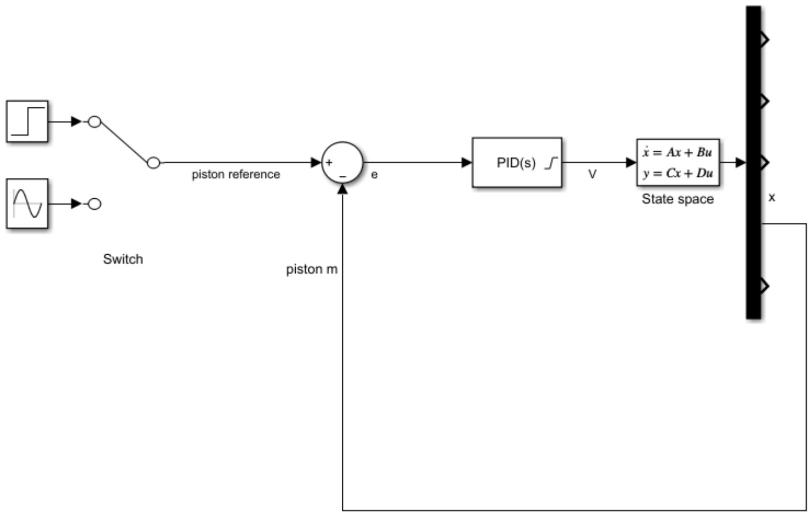

Figure 3.4: EHA linear model simulation in Simulink

For the nonlinear Simscape model, EHA subsystem with its components is shown in Figure 3.5.

The motor and pump couple is modeled as demonstrated in Figure 3.6. In order to model the internal and external leakage of the pump, hydraulic resistance blocks are integrated on the simulation. For the electric motor, a DC motor equivalent for the PMSM motor of the manufacturer is modeled as in the state-space model. As in the state-space model, motor inertia is not calculated separately for the pump and motor. Pump inertia is neglected since it is much lower than the inertia of the motor.

Hydraulic actuator is modeled as demonstrated in Figure 3.7. For the actuator, friction

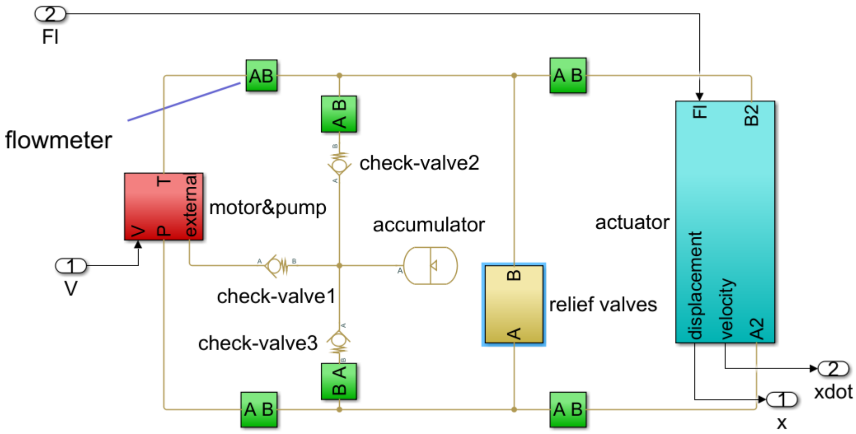

Figure 3.5: EHA nonlinear model simulation in Simscape

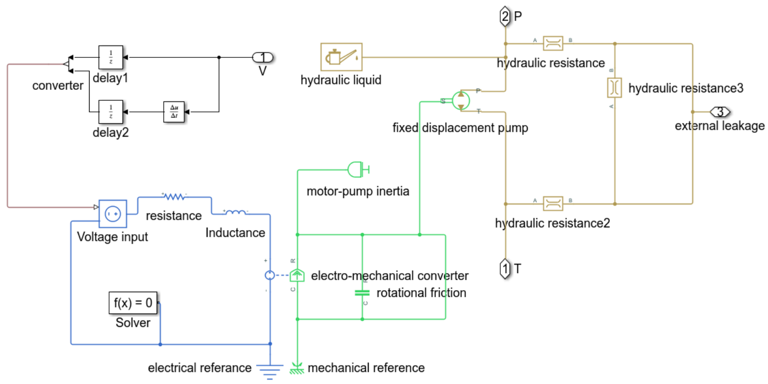

Figure 3.6: EHA electric motor-pump subsystem in Simscape

is modeled with translational friction block which uses Stribeck friction model [57]. An external load signal is integrated in order to observe performance of EHA under static and dynamic load conditions.

Similarly, two directional relief valve sub-system is shown in Figure 3.8.

Open-loop response of both models is investigated. For the comparison of linear and nonlinear EHA models, a chirp signal for voltage input is generated from 0.1 Hz to 30 Hz as demonstrated in Figure 3.9. State responses of piston displacement and current are illustrated in Figure 3.10 and Figure 3.11 respectively.

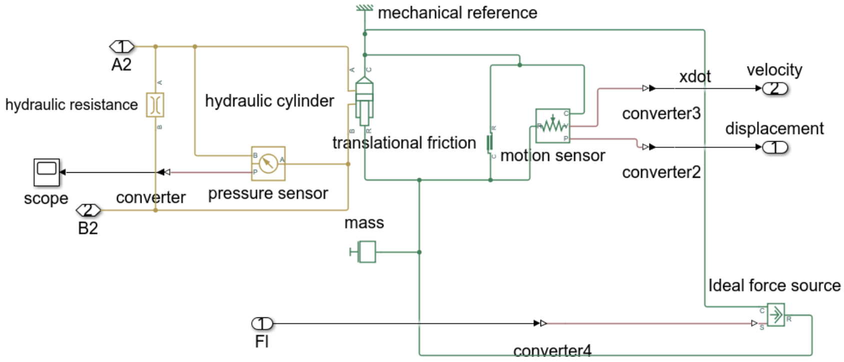

Figure 3.7: EHA actuator subsystem in Simscape

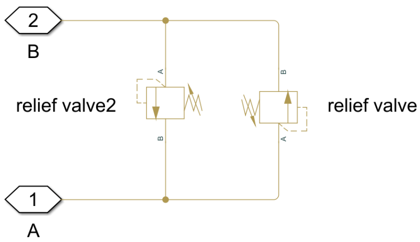

Figure 3.8: EHA relief valve subsystem in Simscape

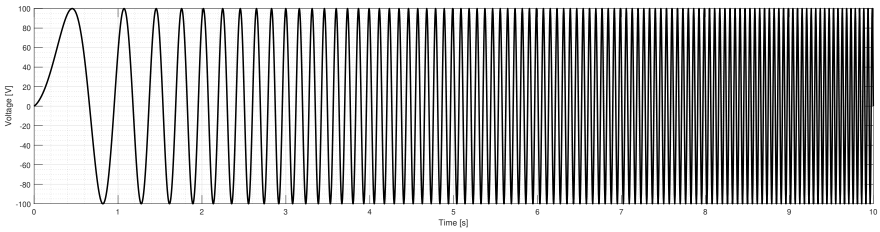

Figure 3.9: Chirp signal for voltage input

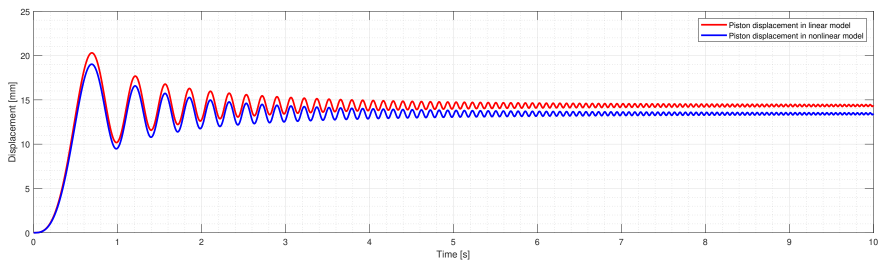

Figure 3.10: Piston displacement response for the given chirp signal

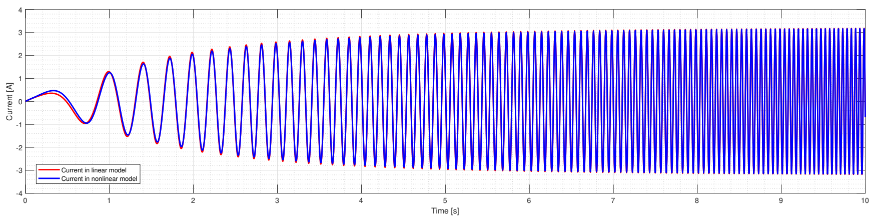

Figure 3.11: Electric current response for the given chirp signal

Inspecting figures, nonlinear and linear models act very similarly, especially for current change. However, due to friction and cylinder dead-volume changes in the nonlinear model, there is a shift in piston displacements which should be expected.

Friction values were estimated according to experience and common literature values for the hydraulic actuators, also leakage coefficients and pump efficiencies were determined with respect to common hydraulic values. The model is significant in order to determine EHA equipment, however, simulation is not a complete simulation. With the preliminary simulation, redundant designs of EHA will compared in terms of their performance, and the hydraulic structure of redundancy will be determined.

3.4 Redundant EHA design

One of the specifications of the EHA is having a redundant structure such that, in the event of a fault, it can switch to its secondary mode to continue operation. As discussed in the introduction, redundancy is often required for flight control actuators. Moog's EHA product for aviation, for example, possesses a dual-redundant property [58]. Therefore, before the mechanical design phase, preliminary simulation and analysis are conducted to determine the appropriate redundant design.

Two different designs are considered: a single cylinder with double pump-motor couplings in a single circuit, and two cylinders with their separate circuits. The simulation behavior of the two designs is analyzed to contribute to the mechanical design process of the EHA structure. While single-piston design is more conventional in the literature, there are also studies [37] and patents on two-piston [59] cylinder configuration.

In the single-piston configuration, it is important to observe that there are two motor-pump couplings incorporated into the system. The primary purpose of this dual-coupling arrangement is to provide redundancy. In the event that one of the motor-pump couplings fails or becomes non-functional, the other one is designed to automatically activate to ensure uninterrupted operation. This activation is achieved through the control of signaling on-off valves, which are strategically positioned, as clearly depicted in Figure 3.12. Furthermore, it is crucial to highlight that the hydraulic circuit includes check valves located on the outer circuit, as well as an on-off valve situated between ports A and B of the cylinder. These components are indispensable for the proper filling of hydraulic oil within the system.

Additionally, it is worth noting that there are on-off valves installed at both the inlet and outlet of the pumps. These valves play a critical role in controlling the activation or deactivation of the respective motor pump couplings. When these on-off valves are configured appropriately, it can be observed that the overall hydraulic circuit closely resembles the configuration depicted in Figure 1.1. This design ensures operational flexibility and enhances the robustness of the system.

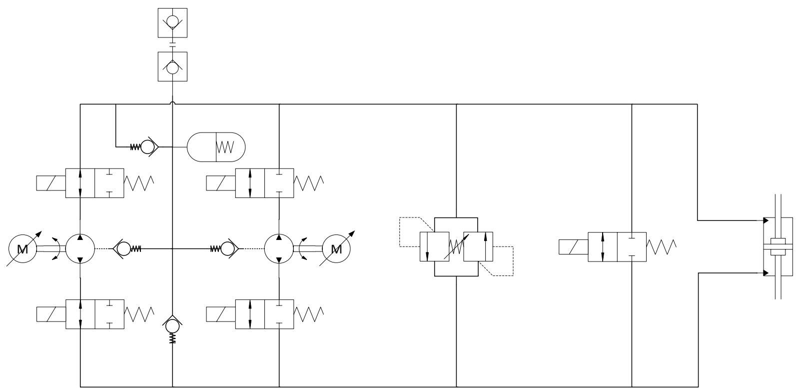

Figure 3.12: Illustration of the single piston configuration for a redundant EHA setup.

In the double-piston configuration, the hydraulic system is characterized by the presence of two distinct motor-pump couplings. Each of these couplings is dedicated to driving a specific piston, thereby ensuring efficient operation. This setup employs tandem pistons, which are mechanically connected by a shared rod to facilitate synchronized motion.

An interesting aspect of this design is that the motion of the rod, which is driven by the active hydraulic circuit, does not induce any pressure difference in the passive circuit. This is because the oil in the passive circuit can flow freely without restriction. This operational principle significantly enhances the reliability and functionality of the system. The hydraulic schematic that demonstrates this configuration in greater detail is presented in Figure 3.13.

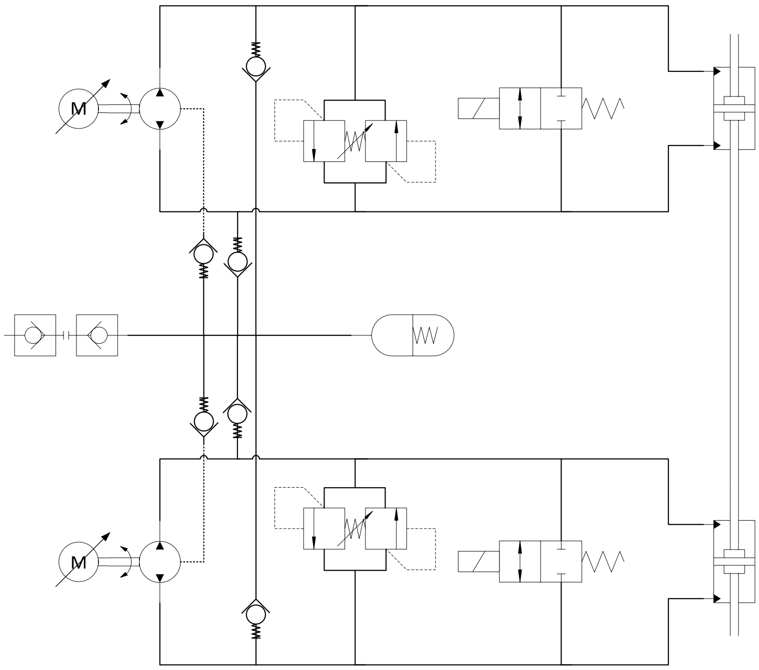

Figure 3.13: Double piston configuration for redundant EHA

3.4.1 Preliminary controller design for closed-loop simulation

In order to compare two models in the simulation, their closed-loop response in case of failure-switching action is investigated. Hence a controller is required. For the given state space model in 3.24, open-loop transfer function between input voltage (V) and piston displacement (x) can be obtained using matrix relation $\mathbf{P_{voltage}}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}$ where

- $\mathbf{A}$: system matrix defined in 3.24.

- **B**: input matrix defined in 3.24.

- **C**: output matrix corresponding to piston displacement.

- **D**: null.

Hence resulting transfer function is shown in equation 3.25 and the open-loop frequency response is illustrated in Figure 3.14.

$$\mathbf{P_{voltage}}(s) = \frac{1.5475 \times 10^{10}}{s(s + 1.46 \times 10^4)(s + 29.32)\,(s^2 + 109.5s + 6.221 \times 10^7)} \tag{3.25}$$

,

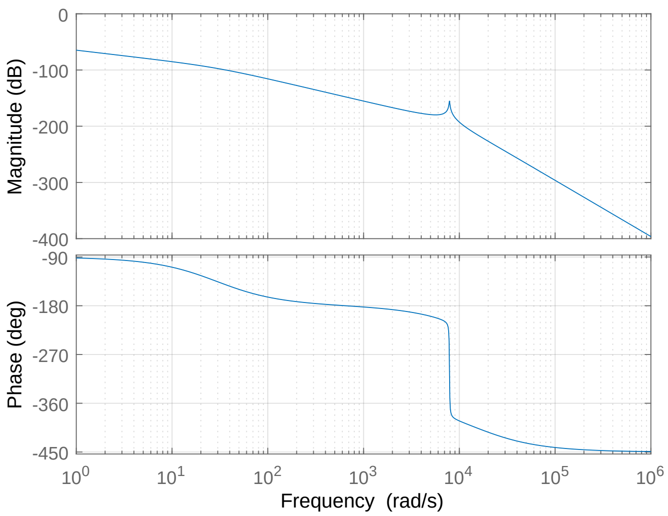

Figure 3.14: Open-loop frequency response of $\mathbf{P_{voltage}}(s)$

A PID controller is designed in order to match design specifications using PID tuning methods with parameters given in Table 3.3.

The resulting frequency response of the closed loop with the PID the controller is illustrated in Figure 3.15. Since a controller is designed, simulation for two redundant conditions shall be studied.

Table 3.3: PID controller parameters for $\mathbf{P_{voltage}}(s)$

Parameter	**Value**	**Description**
K_P	$130\ \frac{\text{V}}{\text{mm}}$	Proportional gain coefficient
K_I	$8.51\ \frac{\text{V}}{\text{mm}\cdot\text{s}}$	Integral gain coefficient
K_D	$0.84\ \frac{\text{V}\cdot\text{s}}{\text{mm}}$	Derivative gain coefficient

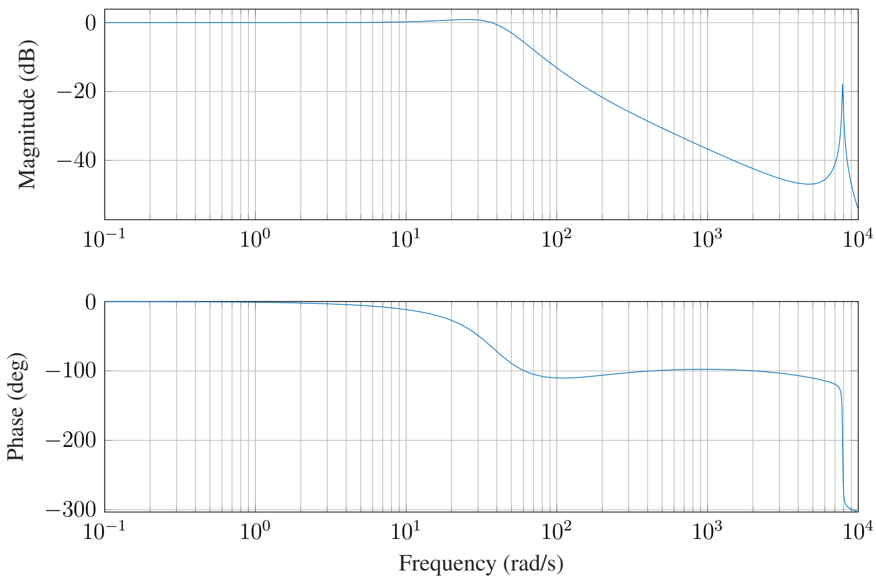

Figure 3.15: Closed-loop frequency response of $\mathbf{P_{voltage}}(s)$

3.4.2 Simulation of redundant EHA models

The simulation of redundant systems is analyzed to evaluate their performance. For the simulation, the physical system shown in Figure 3.5, along with the subsystem blocks introduced in Figures 3.7, 3.6, and 3.8, is used. An on-off valve subsystem block is introduced for the redundant models, as shown in Figure 3.16.

Additionally, for the simulation of switching a switching logic controller is integrated, which is used in both single-piston and double-piston redundant systems in order to switch between modes. The switch logic excites by the switch motor input, which for the application is given as a step signal, switching once during simulation. The

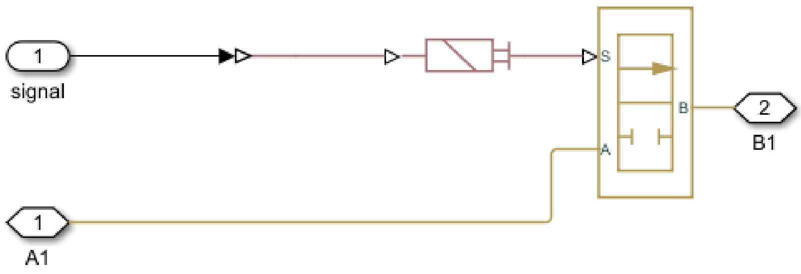

Figure 3.16: On-off valve block integrated in redundant nonlinear EHA models

switch logic then actives or de-actives motor-pump couples by sending voltage signal and closing or opening corresponding on-off valves as shown in Figure 3.17.

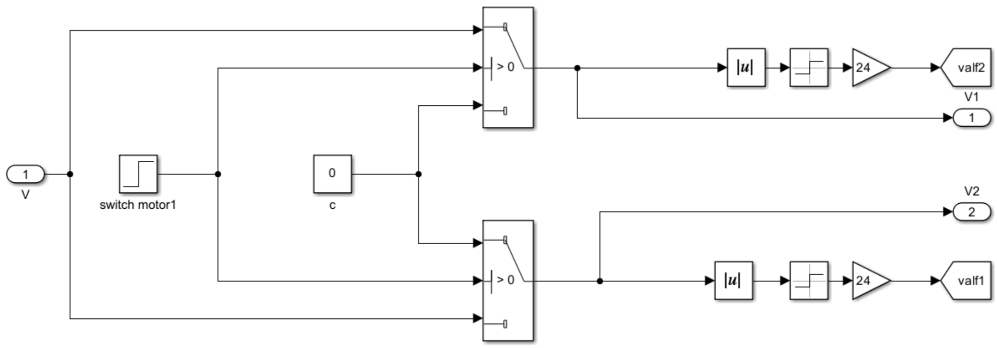

Figure 3.17: Switch logic block integrated in redundant nonlinear EHA models

Apart from the blocks introduced in this section, no modification to the system in Figure 3.5 is implemented. The configuration for the single piston is modeled according to the hydraulic schematic in 3.12, where the respective Simulink model is demonstrated in 3.18.

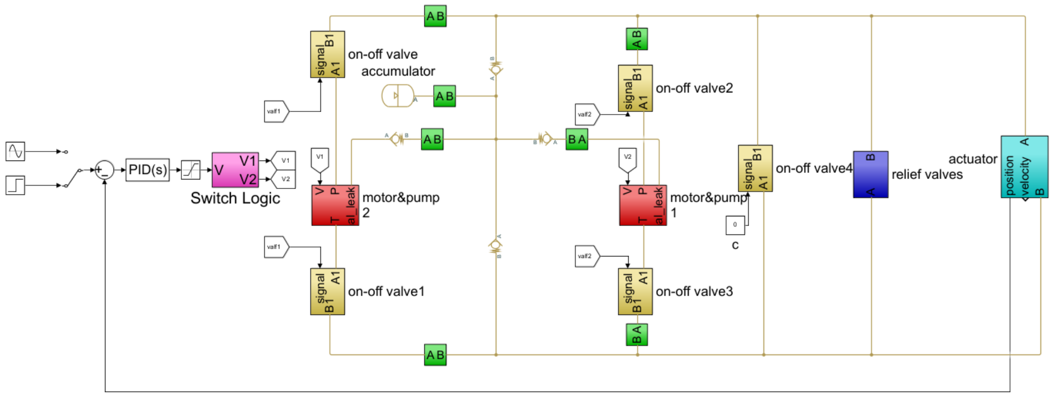

Figure 3.18: Single piston nonlinear EHA model in Simulink

The configuration for the single piston is modeled according to the hydraulic schematic in 3.13, where the respective Simulink model is demonstrated in 3.19. Notice that actuators are connected my mechanically in this setup. Switch logic controls which motor-pump couple to activate and respective on-off valves in EHA circuits.

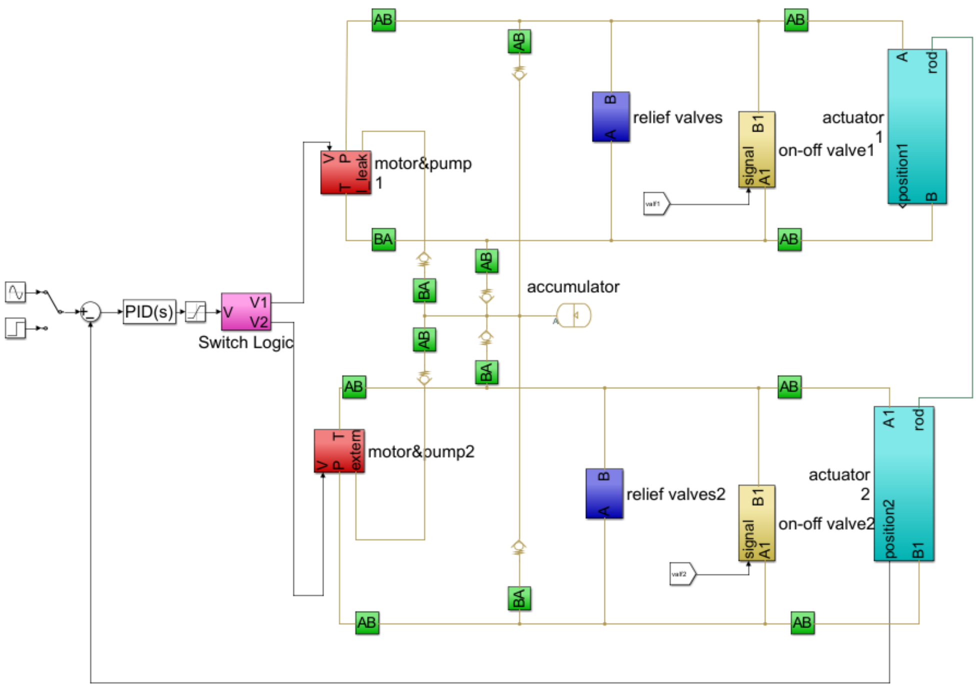

Figure 3.19: Double piston nonlinear EHA model in Simulink

Both configurations have the same PID controllers, which were previously designed based on the linear model with a saturation value of $\pm 480\,\mathrm{V}$. A performance evaluation simulation is conducted for both redundant systems to observe their robustness during sudden changes, such as activating the redundant sub-system or applying external step loads. The simulation parameters are summarized in Table 3.4.

Table 3.4: Simulation parameters for redundant EHA configurations

Parameter	**Value**
External load	10 kN
External load step time	1 s
Switching step time	3 s
Reference frequency	3 Hz
Reference amplitude	3 mm
Solver	ode23t

The piston displacements of the resulting test are illustrated in Figure 3.20. Notice that both configurations have the same displacement response which validates the redundancy design equivalency to EHA circuit introduced. However, their difference in response is observed during critical load step time and switch step time of 1 s and 3 s at the simulation, respectively. The difference in the piston displacements during the load step is zoomed in Figure 3.21. Similarly difference of the piston displacements during the switching action is zoomed in Figure 3.22

From the figures, it is clear that double piston configuration is less robust to step changes, due to relatively high (1 mm) sudden step changes in displacement during operation. While in single piston configuration, effects are damped smoother. On the other hand, it can be argued that both configurations maintain operation successfully in a short time interval of 50 ms.

The impacts of step signals and switching action can be observed from the motor speeds of the configurations, which are illustrated in Figure 3.23 and Figure 3.24.

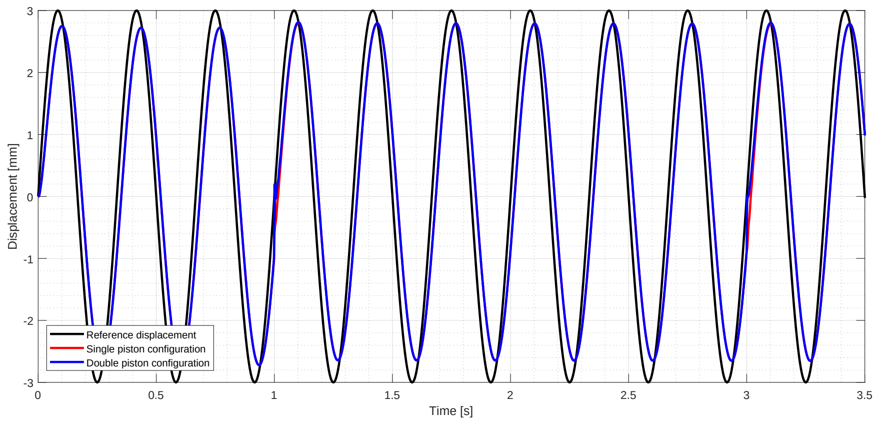

Figure 3.20: Piston displacement during performance evaluation simulation of redundant EHA configurations

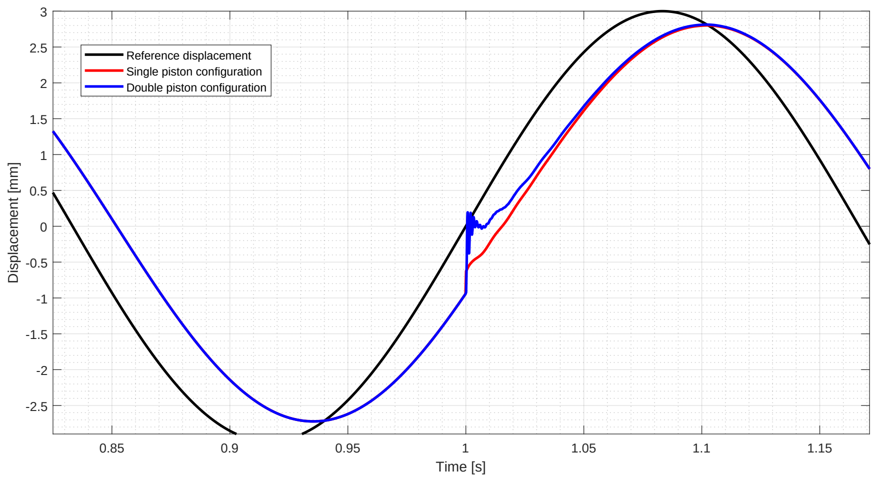

Figure 3.21: Piston displacements while load step affects

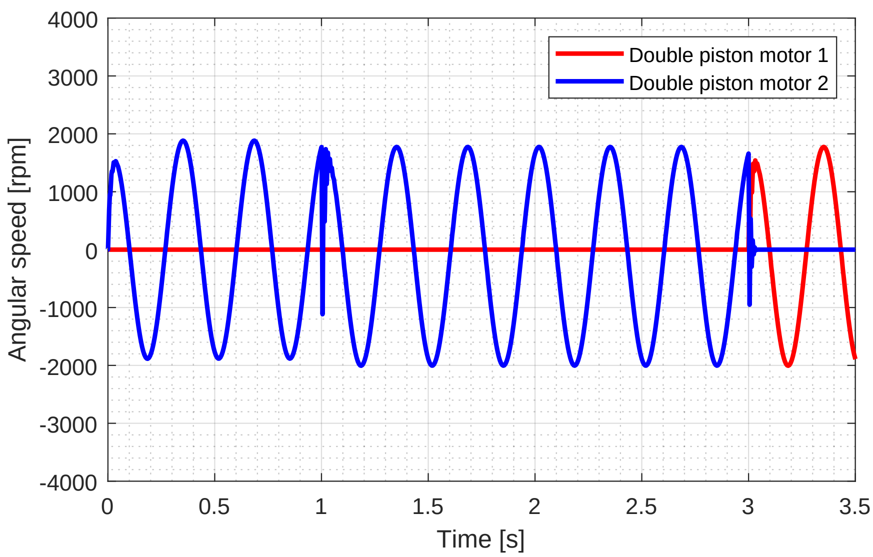

Figure 3.23: Double piston configuration motor speeds during switch and load steps

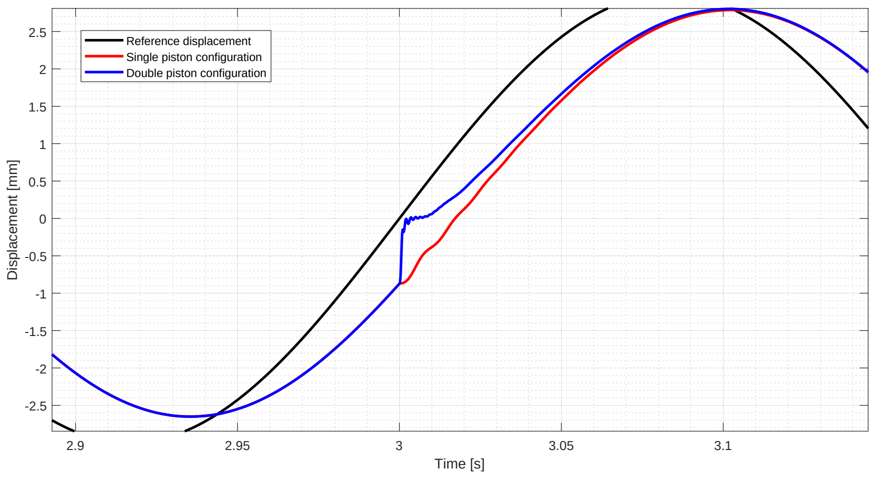

Figure 3.22: Piston displacements while switch activates

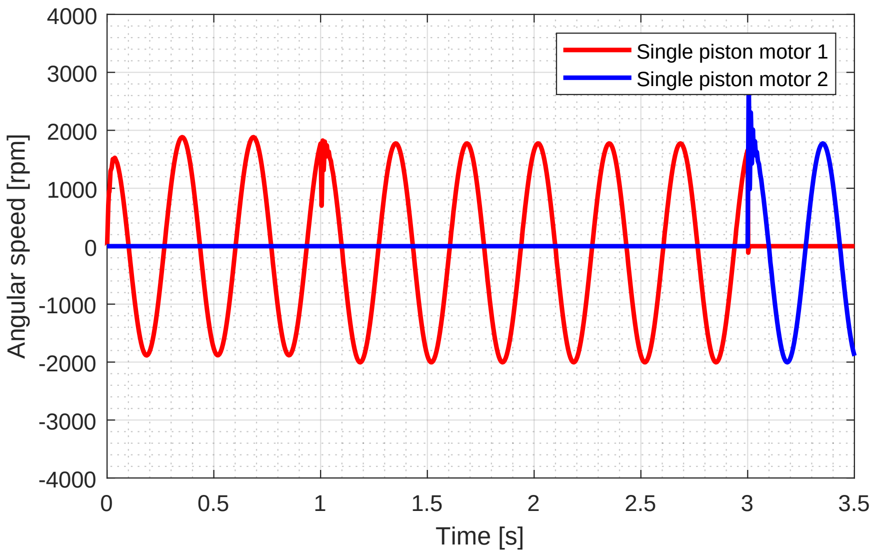

Figure 3.24: Single piston configuration motor speeds during switch and load steps

From a redundancy perspective, the double-piston configuration has an advantage because the pistons themselves are redundant. However, while the performance of both redundant systems is nearly identical under nominal conditions, the double-piston configuration performs worse during sudden load changes or redundancy switches. Additionally, in terms of compactness, the additional piston increases the cylinder's length, resulting in larger overall dimensions. Hence, considering the simulation results and dimensional concerns, an EHA design with a single-piston redundant configuration is preferred.

3.5 Mechanical design of EHA

Compactness is a key criterion for the mechanical structure of the EHA. In specific flight control applications, manifolds are designed specifically designed together with the sub-components such as sensors, pumps, etc. [60]. However, for this study, industrial equipment is assembled together with a manifold where equipment is specified in Table 3.2. Equipment is selected and updated through iterative designs according to simulation results, such as accumulator volume and cracking pressure of check valves. In order to prevent cavitation, based on fluid dynamics analysis on the external gear pump, EHA's idle pressure is set to 10 bar. Rod and piston dimensions are determined according to performance specifications as well as buckling analysis. Therefore material and dimensions are determined, considering safety factors for maximum load and stress on material. Another factor to consider is the total mass of the assembly since mass is also critical for aerospace applications along with space occupied. As illustrated in Appendix A, the final hydraulic circuit is designed considering the mechanical structure along with the sensor placements, which affects manifold geometry and structure. Since there are 6 lines between two piston chambers and two motor-pump couples sideways, a symmetrical octagon manifold architecture is designed and illustrated in Figure 3.25.

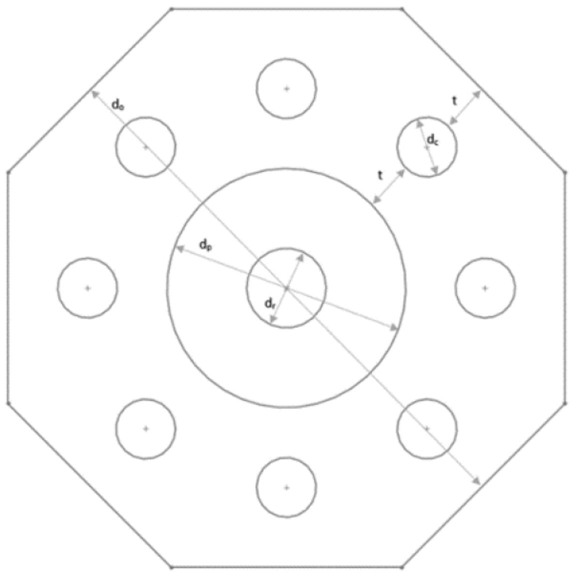

Figure 3.25: Dimensions of octagon manifold [54]

Dimensions of the octagon manifold are presented in Table 3.5. These dimensions are critical to the simulation parameters of the piston mass and dead volume of the hydraulic chambers, together with the mentioned safety factors on load and buckling.

Table 3.5: Fundamental dimensions of octagon manifold

Basic Dimension	**Symbol**	**Value**	**Unit**
Diameter of Rod	d_r	18	mm
Diameter of Piston	d_p	60	mm
Length of Rods (Two Sides)	l_r	230	mm
Length of Piston	l_p	40	mm
Minimum Thickness of Wall	t_{wall}	26	mm
Diameter of Hydraulic Channels	d_c	8	mm
Diameter of Octagon Manifold	d_o	190	mm
Length of Octagon Manifold	l_{om}	280	mm

Hence the mass of the rod, piston, and octagon manifold can be calculated according to the equations 3.26, 3.27, 3.28 and 3.29 respectively.

$$d_o = d_p + 2d_c + 4t_{\text{wall}} \tag{3.26}$$

$$m_r = 2\rho_{sr}\left(\frac{\pi d_r^2}{4} l_r\right) \tag{3.27}$$

$$m_p = \rho_{sp}\left(\frac{\pi d_p^2}{4} l_p\right) \tag{3.28}$$

$$m_{om} = \rho_{aom}\left(\frac{d_o^2}{1+\sqrt{2}} l_{om} - \frac{\pi s d_p^2}{4} - 2\pi s d_c^2\right) \tag{3.29}$$

The resulting mechanical design is illustrated in Figure 3.26. There is a single manifold structure connected to the supplied equipment. There are pump-motor couples connected to sideways of the manifold and the accumulator on the bottom is connected to the hydraulic circuit. The motor drivers and controller are not integrated into the manifold structure.

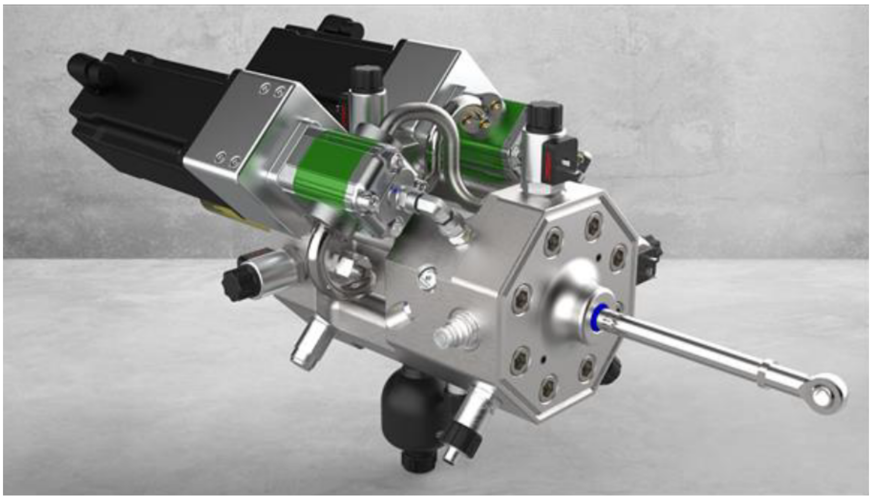

Figure 3.26: Mechanical model of EHA with selected equipment

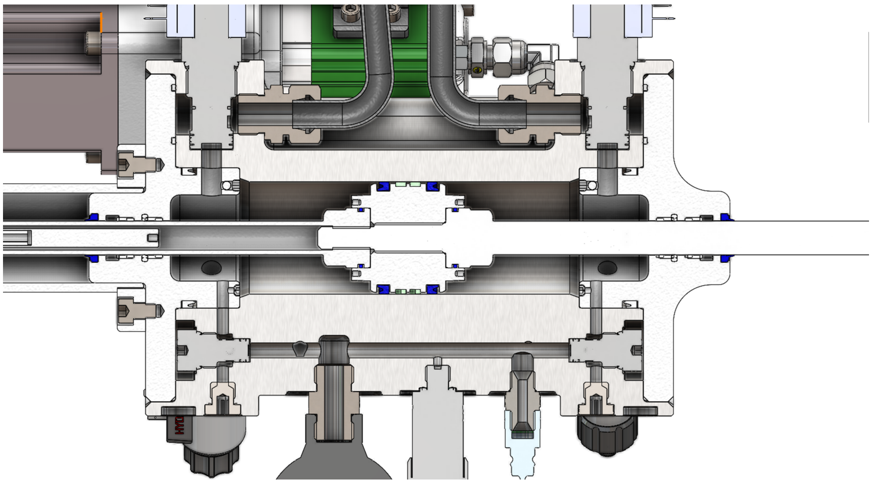

Figure 3.27: Cross section of EHA manifold displaying hydraulic lines

In Figure 3.27, the cross-section of the EHA model is illustrated. The dead volume, V_d, of one cylinder chamber, including the hydraulic cavities, is approximately $4.0123 \times 10^{-4}, \mathrm{m}^3$, where the piston chamber occupies approximately 80% of the dead volume.

3.6 System identification of EHA

Prior to the manufacturing process of EHA, selected component is procured and identification tests on them are conducted at Repkon Dynamics Laboratory as shown in Figure 3.28. For the motor model, technical data of motor characteristics are validated by comparison of simulations and experiments. Driver control of the motor is also modeled based on experimental data. For the external gear pump, the volumetric efficiency of the pump depending on pressure and rotational speed is mapped. After the assembly process, characterization of friction on the actuator in particular and general EHA system is conducted.

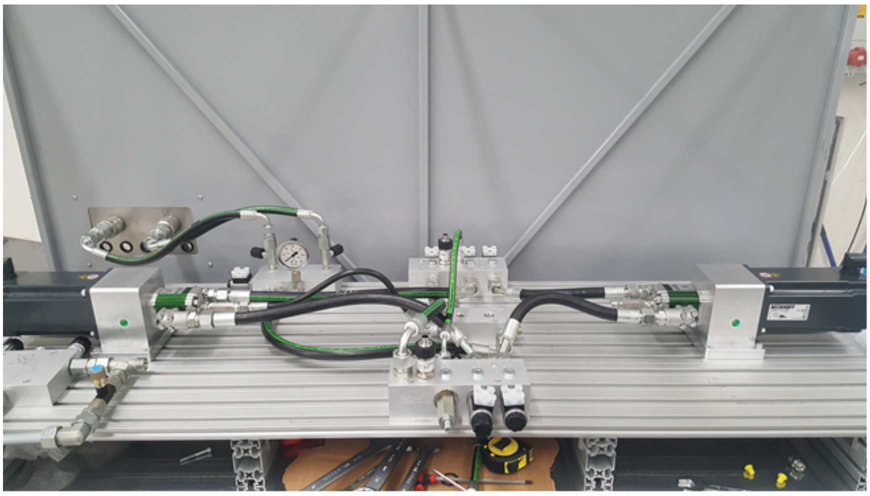

Figure 3.28: Hydraulic test bench at Repkon Dynamics Laboratory

In Figure 3.29, the assembly of EHA is demonstrated which is displayed at the 24th National Conference on Automatic Control.

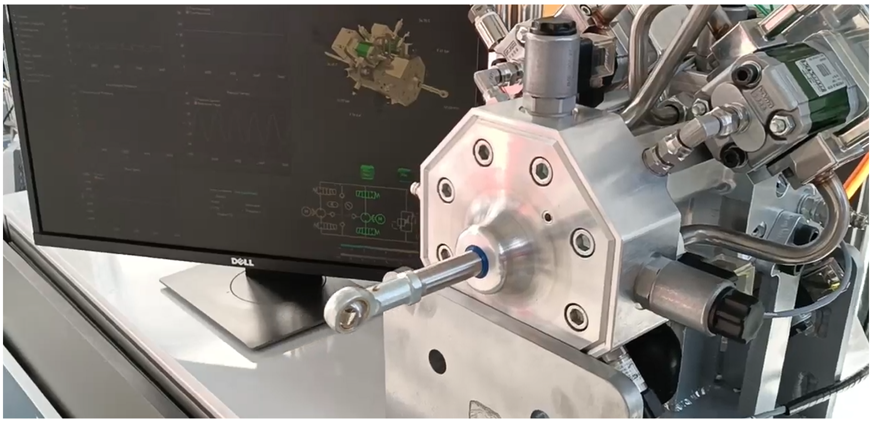

Figure 3.29: EHA demonstration at the 24th National Conference on Automatic Control [61]

3.6.1 Identification of PMSM and motor driver

After selecting the electric motor and conducting experiments in a laboratory environment, it was realized that, while the state-space model in Equation 3.24 precisely describes the hydro-mechanical dynamics of the actuator, it is incomplete in modeling the connection between the electric motor and the driver. Therefore, an update to the preliminary simulation is required, taking into account the experimental data and the selected components. Although it is correct that voltage is the input that excites the system, there is a cascade loop structure in the motor driver, as illustrated in Figure 3.30.

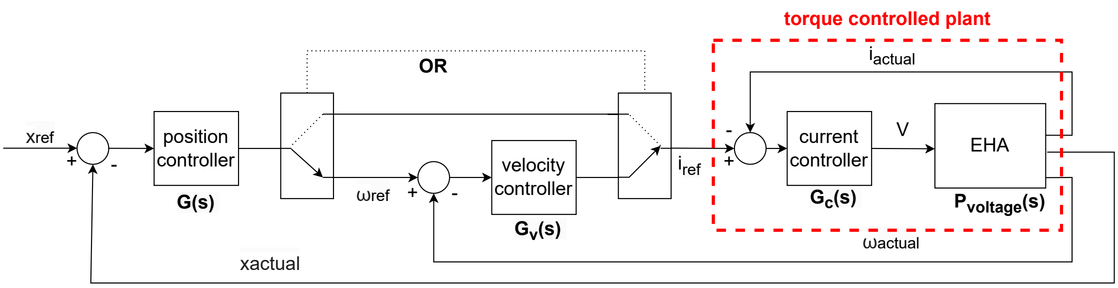

Figure 3.30: Cascade loop structure of the motor control structure integrated into EHA control

The cascade loop structure illustrated is a simplification that will be used for the linearized EHA models, neglecting flux-generating current. Selected driver and motor control voltage with the current controller. Since torque is related to torque generating current by torque constant, this control method is also called torque mode, or torque control. In section 3.2, the mathematical model of EHA was discussed including PMSM motor model that was simplified to DC motor model. The equations 3.10 and 3.11, can be modeled in Simulink considering nonlinearity such as saturation of current values and driver delays. Additionally, current and velocity controllers that are integrated into the motor driver can be modeled in parallel with the experimental results. According to voltage dynamics and torque generated by the motor, the following Simulink diagram is formed in Figure 3.31.

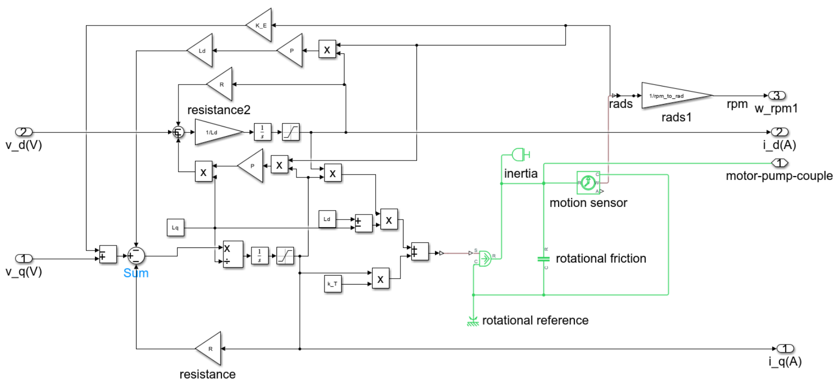

Figure 3.31: PMSM model in Simulink

The parameters of inductance, resistance, number of poles, rotor inertia, torque, and back-emf constants are shared by the manufacturer on the technical data sheet of AM8053N. However, damping and friction information is not available, has tests are required to tune these parameters. Additionally, it is known that these constants are approximations, and constants actually derived to represent a linear relation. In practical applications, constants may change with respect to torque and factors such as temperature. Controllers on the driver AX5118, are in a cascade loop structure as described in Figure 3.30. The current controller structure in Simulink is illustrated in Figure 3.32.

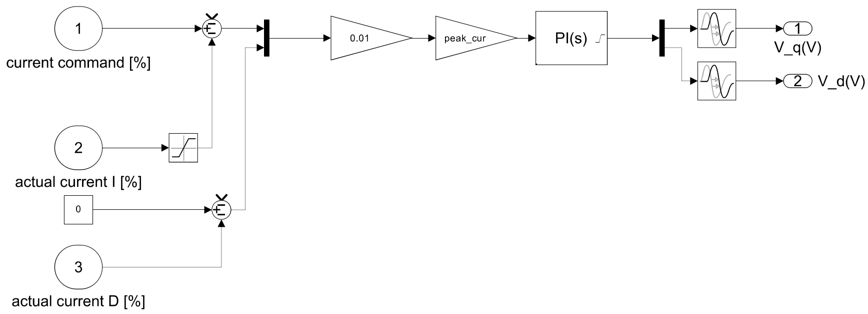

Figure 3.32: Current controller of the driver in Simulink

Since the set point for the flux-generating current is zero, an approximation can be made to neglect the flux-generating current in the linear model. To validate the controller model and the PMSM motor model, an open-loop test is conducted in simulation for motor velocity. During the experiment, an angular speed command of 1000 rpm is applied at increasing frequencies. Current command output from the velocity controller loop as well as motor velocity and current are measured. In the simulation, verification of the current controller and PMSM model is investigated by applying the current command. In Figure 3.33, current command measurement and resulting torque generating current compassion of test data and simulation is illustrated. The current command is followed smoothly in the simulation. However, the test data exhibits significant noise, which is likely due to torque disturbance compensation, as the experiment is conducted in the velocity loop, and due to measurement noise.

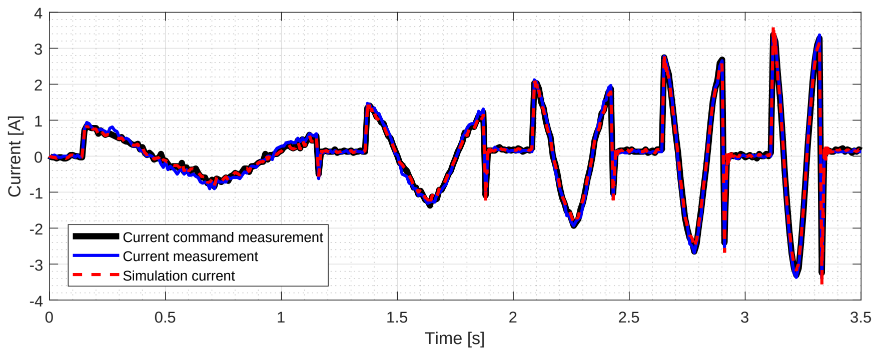

Figure 3.33: Torque generating currents

The effects of such disturbances and noise are also reflected in the open-loop velocity responses. In Figure 3.34. While velocity amplitudes are quite coherent, simulation has a steady state error due to having no closed-loop feedback of motor velocity. Hence, after verification of the current controller, the velocity controller response shall be investigated.

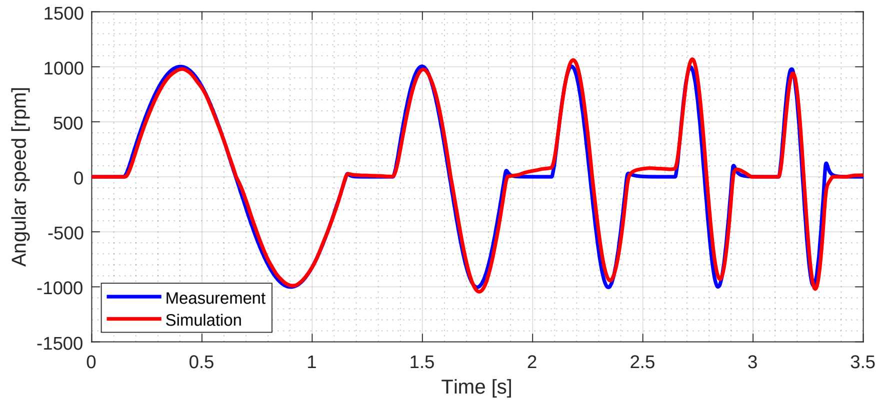

Figure 3.34: Motor velocities of simulation with no reference velocity

In Figure 3.35, the velocity controller in Simulink is demonstrated. For the given velocity command, a torque/current command is generated in the velocity loop to the current controller which was previously validated. In Figure 3.36, for the same experimental data however applying velocity reference instead of the current command, velocity response is shown. The frequencies are between 1 Hz to 15 Hz. The program of the sine sweep test, as well FFT analysis of the results are shared in Appendix B.

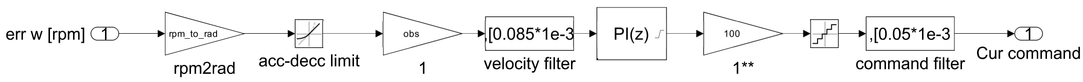

Figure 3.35: Velocity controller of the driver in Simulink

The simulation and experimental data demonstrate a high correlation in their frequency responses. An initial 8 ms delay in the driver was observed, attributed to the delayed activation of three buffers between the driver and logic controller, as specified by the manufacturer's technical support team. This delay was reduced in later experiments by adjusting the controller's sample time, and the revised delay effects were integrated into the simulation.

The Root Mean Square Error (RMSE) between the simulation and experimental results is 27 rpm, with a maximum error of 117 rpm, primarily due to phase shifts ob-

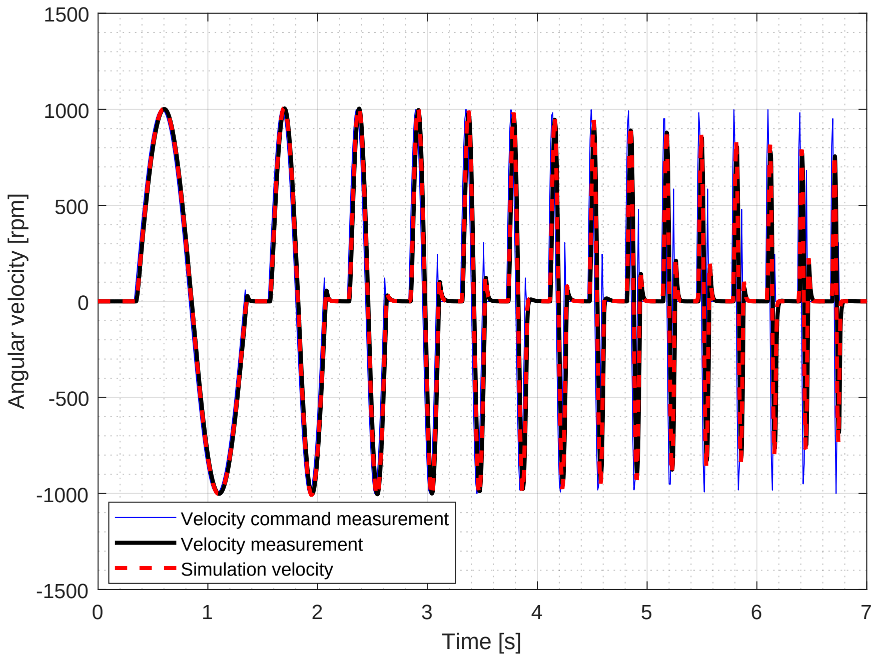

Figure 3.36: Motor velocities in velocity controller loop

served at higher frequencies. These findings validate the accuracy of the simulation model while emphasizing the influence of controller timing on system performance. Default PI parameters for the controllers were applied during the experiments, further optimization of controller parameters will be investigated.

3.6.2 Identification of external gear pump model

For the modeling of the external gear pump VIVOIL X1R2725FJJE, an experiment is conducted at different pressures and rotation speeds in order to map the relation between flow rate. While there are many factors affecting the volumetric efficiency of the pump, such as properties of the hydraulic oil, temperature, wear, motor speed, and pressure, two main states that are measurable — motor speed and pressure — are the primary factors influencing efficiency. In terms of simulating EHA, motor speed and pressure are also fundamental variables, unlike temperature or oil cleanliness which depend on many external factors. The hydraulic schematic of the experiment is illustrated in Figure 3.37.

For the experiment, a pressure-compensated variable displacement pump operating

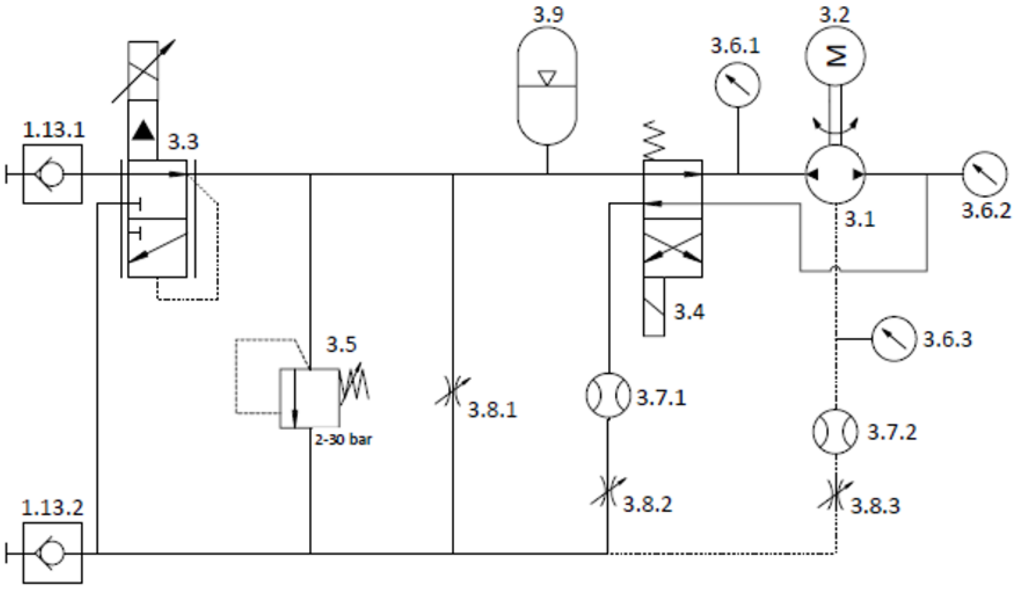

Figure 3.37: Hydraulic pump test setup schematic

as a pressure source connected to the hydraulic schematic is demonstrated. The pressure source is set at low pressures (2 bar), supplying flow rate as external gear pump requires. The main operating principle can be summarized as:

- 1.13.1 coded connector supplies pressure from the pressure source through proportional valve 3.3.
- 1.13.2 is the return connector to the hydraulic reservoir.
- 3.5 is pressure relief valve, that protects inlet pressure of the external gear pump, against potential pressure fluctuations.
- 3.4 is directional valve supplies hydraulic oil at decided line of the hydraulic pump.
- 3.8.2 is needle valve, manually adjusted in order to increase pressure at the pump outlet, where 3.7.1 flow-sensor records outlet flow rate of the pump.
- 3.7.2 measures external leakage of the hydraulic pump at 3.1, which is rotated at different speeds by electric motor 3.2. From the motor, rotational speed data is also recorded.
- Accumulator at 3.9 prevents pressure fluctuations on the inlet line of the pump.

- inlet, outlet and leakage pressures are recorded by pressure sensor 3.6.1, 3.6.2 and 3.6.3 respectively.

The equipment of the hydraulic circuit is demonstrated in Table 3.6.

Table 3.6: Pump test circuit equipment

No	Description	Brand	Product Code
1.13.1	Poppet type quick release coupling, 100 L/min	DNP	PVV3.2013.112
1.13.2	Poppet type quick release coupling, 100 L/min	DNP	PVV3.2013.113
3.1	EHA pump	Vivoil Oleodinamica Vivolo	X1R2725FJJE
3.2	Test motor	BECKHOFF	AM8063-18A0-0000
3.3	2/2 pressure reducing valve, solenoid controlled, 100 L/min	HYDAC	PDR08P-01 M-C-N-087-0
3.4	4/2 valve, solenoid controlled, 100 L/min	HYDAC	4WE 6 D A01-24 D G /N
3.5	Safety valve, 2–30 bar	PARKER	A04B2PZN
3.6.1	Pressure sensor, 0–20 bar	HYDAC	HDA4446-A-016-000
3.6.2	Pressure sensor, 0–300 bar	HYDAC	HDA4446-A-400-000
3.6.3	Pressure sensor, 0–20 bar	HYDAC	HDA4446-A-016-000
3.7.1	Flow meter, 100 L/min, single direction	VSE	VS 1/1 GP012V-42R11/5
3.7.2	Flow meter, 10 L/min, single direction	VSE	VS 0.1/6 GP012V-42R11/5
3.8.1	Needle valve, <5 L/min	HYDAC	DV-06-01.X/0

3.8.2	Needle valve, 10 L/min	HYDAC	DV-16-01.X/0
3.8.3	Needle valve, <10 L/min	HYDAC	DV-06-01.X/0
3.9	Accumulators	HYDAC	SB330-6A1/112U-330A

In the experiment, the needle valve was adjusted at every 200 rpm increment from 200 rpm to 6000 rpm to increase the system pressure for each specific speed. As the needle valve was progressively closed, a corresponding decrease in flow rate was observed. This relationship between flow rate and pressure is exemplified in Figure 3.38.

It is important to note that no active heat management was applied during the experiment. The initial temperature of the system was recorded at 19 °C, gradually rising to 23.2 °C by the end of the experiment. This temperature increase may have contributed to minor variations in the observed results.

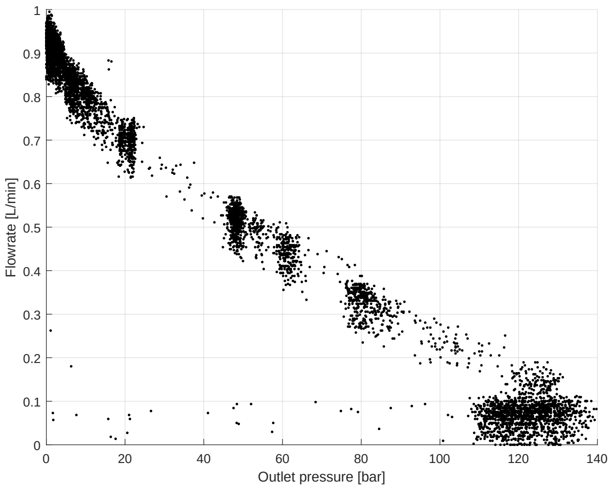

Figure 3.38: Flow-rate measured from the pump at varying outlet pressures at 200 rpm

Notice that there are sudden flow-rate drops at pressures while the needle valve is adjusted, a transient behavior is observed. Such flow-rate data is discarded from efficiency calculation. The volumetric efficiency plot obtained is illustrated in Figure 3.39.

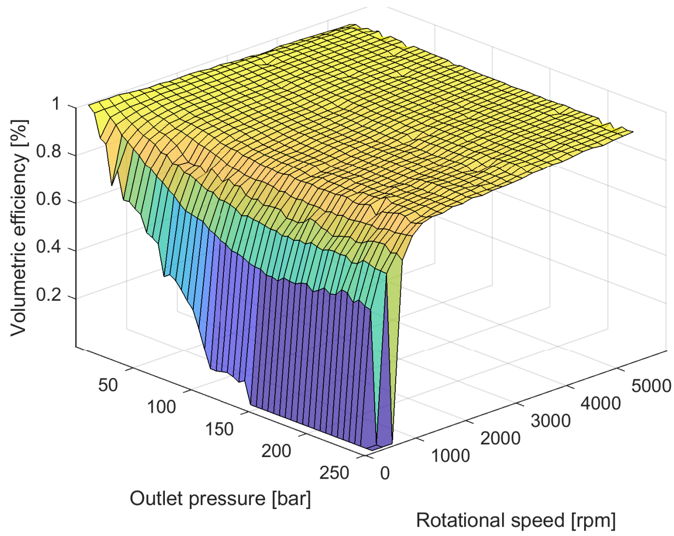

Figure 3.39: Volumetric efficiency plot of external gear pump

Outlet pressure is equivalent to the pressure difference between the inlet and the outlet since inlet pressure is connected to the hydraulic tank. From the volumetric efficiency map, leakage coefficients of the pump can be linearized for pressures between 50 - 250 bar. Notice in equation 3.15, internal and external leakage coefficients C_p and C_r are also related to inlet and accumulator pressures which may be neglected, hence equation can be modeled as in equation 3.31:

$$Q = D_p\omega - (C_p + C_r)p_D \tag{3.30}$$

Assuming inlet and outlet leakages are equivalent, a map for the total leakage coefficient is derived from the volumetric efficiency relation:

$$Q = D_p\omega \cdot \eta_v(\omega, p_D) \simeq D_p\omega - (C_p + C_r)p_D \tag{3.31}$$

From the obtained leakage coefficient map, average inlet and outlet leakage coefficients are calculated as $2 \times 10^{-12}\,\mathrm{m^3/(s \cdot Pa)}$, where these coefficients are useful for the linear plant model. The non-linear volumetric efficiency plot is integrated as a look-up table to Simscape simulation for the nonlinear simulation analysis.

3.6.3 Identification of actuator model

For the identification of the actuator model, dead volume in the compact EHA is calculated from the mechanical CAD model. Hence V_d in equation 3.20 is derived, symmetrical for both chambers.

In equation 3.23, part of the external load F_L and viscous friction b can be decomposed to form cylinder friction during motion. Hence the equation can be written as:

$$A_p(p_A - p_B) - F_f - F_L = m_p\ddot{x} \tag{3.32}$$

Notice that external load does not include friction in this mathematical model. In terms of friction, equation 3.32 is a nonlinear dynamic model while the previous equation 3.23 is a linear model of the friction only including viscous effects. There are many models of friction in the context of hydraulic friction [62], Stribeck friction model is a common model that mostly other models compared on. The actuator friction model is modeled as Stribeck friction due to its wide-range acceptance, and compliance to simulation solvers. Additionally, Simscape translational friction applies Stribeck model. Stribeck friction model is represented in equation 3.33:

$$F_f = \sqrt{2e} \cdot (F_s - F_c) \cdot e^{-\left(\frac{v}{v_s\sqrt{2}}\right)^2} \cdot \left(\frac{v}{v_s\sqrt{2}}\right) + F_c \cdot \tanh\left(\frac{v}{v_s/10}\right) + \mu v \tag{3.33}$$

Where:

- F_f: Total friction force
- F_c: Coulomb friction force
- F_s: Static friction force
- v: Relative velocity
- v_s: Stribeck velocity threshold
- μ: Viscous friction coefficient

Notice that F_f depends on v and other variables are constants. As illustrated in Figure 3.40, a closed-loop displacement test is conducted in order to identify the friction model. Sinusoidal reference was commanded at increasing frequencies, resulting in higher actuator velocities. Pressure from chambers A and B of the cylinder are recorded, under no external load.

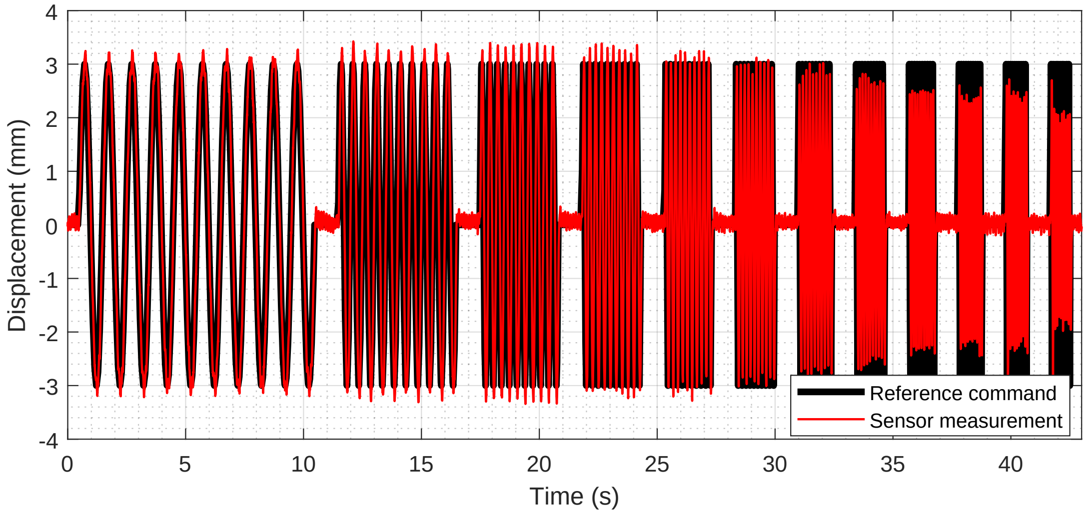

Figure 3.40: Sample test for the friction modeling

Hence dynamic equation in equation 3.32 is mostly constructed, however inertia of the moving piston mass should be considered too. In the experiment, it is observed that the pressure difference during the test is around 1000 N magnitude. The effect of inertial force due to acceleration of piston mass is negligible, even at higher frequencies. From the displacement reference amplitude and frequencies, given piston mass, peak force resulting from the reference command is illustrated in Figure 3.41.

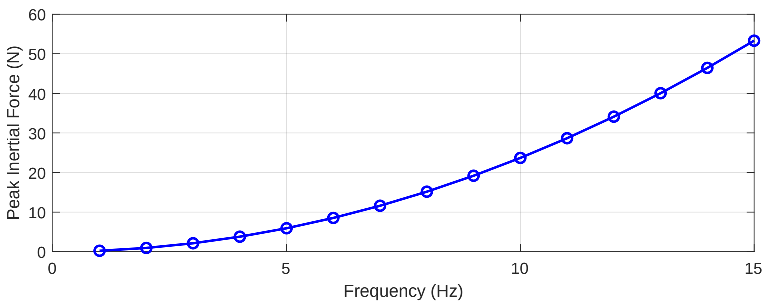

Figure 3.41: Peak inertial force with respect to reference command frequency

Hence, it can be concluded that the effect of friction force is significantly higher, particularly at lower frequencies. From the experimental data, by neglecting the inertial force, force derived from the pressure differences is plotted with respect to piston speed, as shown in Figure 3.42. Based on this data, the Stribeck friction model is fitted, using velocity as the input.

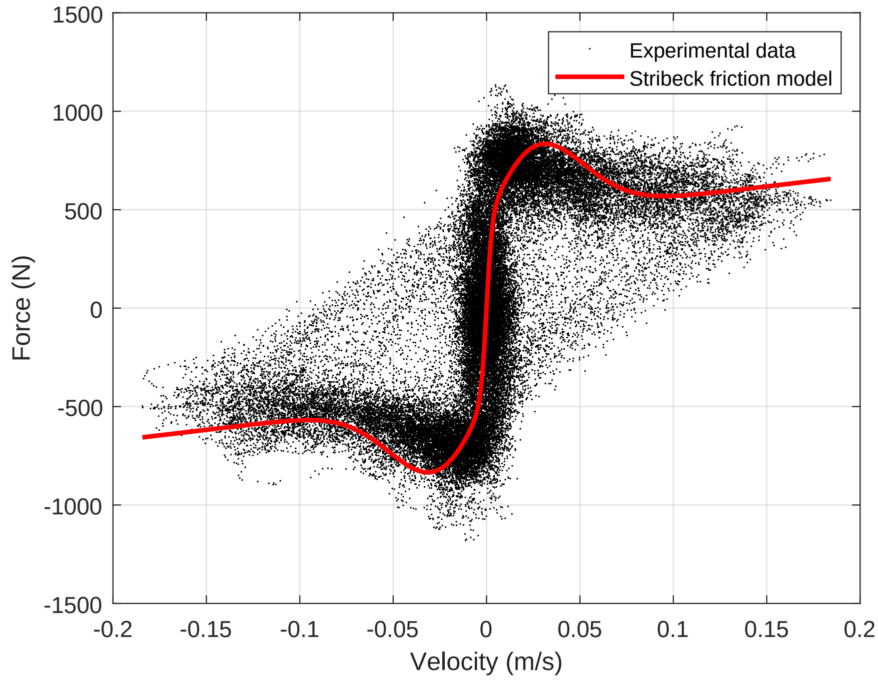

Figure 3.42: Stribeck fit to experimental friction force data

The friction model is fitted symmetrical in both directions and around the operating point of the piston. The resulting Stribeck constants are resented in Table 3.7.

Table 3.7: Stribeck coefficients of the friction model

Coefficient	**Value**	**Description**
F_c	450	Coulomb friction force (N)
F_s	800	Static friction force (N)
v_s	0.03	Stribeck velocity threshold (m/s)
μ	1120	Viscous friction coefficient (Ns/m)

Additionally, a load test is conducted on EHA system in order to observe the holding capacity of load, thermal response, and leakage characteristics. Heat is generated on EHA system due to leakage and inefficiencies of the pump-motor couple. As the temperature of the hydraulic oil increases, the viscosity of the oil drops significantly hence leakage of the system increases. The load test setup is demonstrated in Figure

3.43.

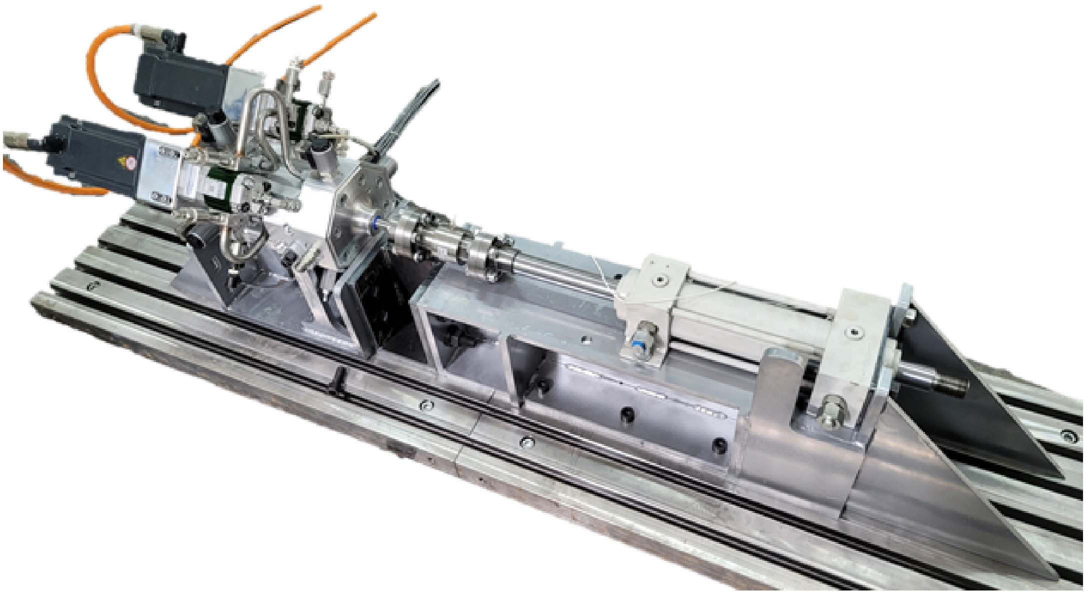

Figure 3.43: Test setup for the load tests

An external load of 23 kN is applied on EHA while the position controller is active. Thermo-couples are placed on the inlet and outlet of the active pump and temperature measurements from the motor's own sensors are recorded during the experiment. The effect of temperature increase on leakage is followed by the rotational speed of the motor, compensating for the leakage in the system. Hence, since the displacement of the pump is known and the piston is stationary, it is possible to derive the leakage coefficient at the given pressure difference on EHA. In Figure 3.44, the temperature is highest at the outlet of the gear pump. There is a clear correlation as the temperature increases, motor speed also increases in order to compensate for leakages. From a direct ratio of pressure and flow rate supplied by motor speed, the dynamic leakage coefficient is plotted with respect to time. It is clear high temperatures on EHA increase the leakage coefficient of the system. However, the exact relation between temperature and leakage is out of the scope of this thesis. The leakage coefficient average from the low-temperature range is calculated for system identification since most performance tests result in short time intervals without significant temperature increases.

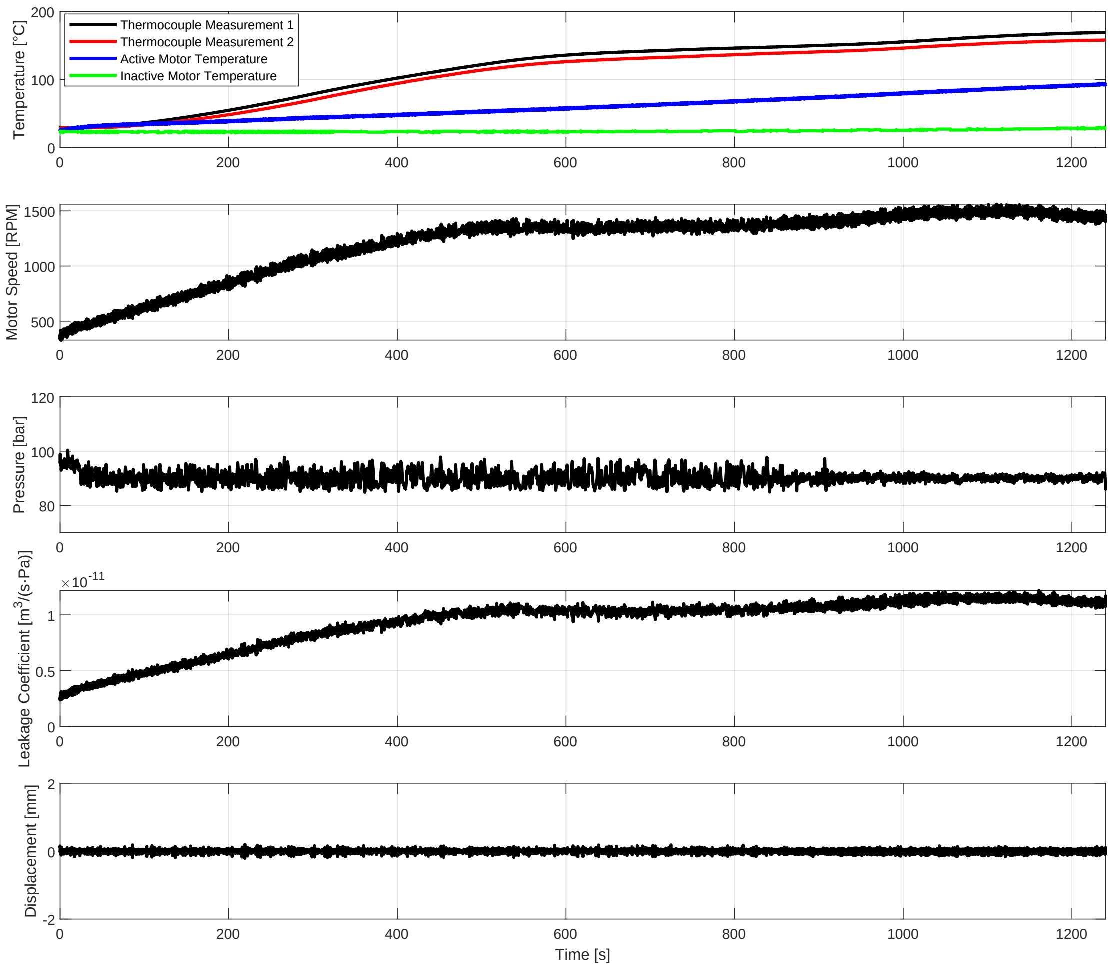

Figure 3.44: Load test conducted to observe leakage coefficient of EHA

3.6.4 Validation of EHA model

As a result of identification tests, detailed parameterization of the EHA sub-components has been achieved. This enables the creation of accurate non-linear simulations to evaluate controller performance under specific conditions and assess overall system behavior. Furthermore, the availability of a detailed system model allows for confident linearization, as it can be rigorously compared against the non-linear model. Voltage in equation 3.14 is generated via the current controller of the driver, which has a fixed PI structure. Hence PI controller equation is given as:

$$V = K_p^i i_e + K_I^i \int i_e \, dt \text{ where } \quad i_e = i_{\text{ref}} - i \tag{3.34}$$

Here, K_p^i and K_I^i are the proportional and integral constants of the PI controller, i_e is the current loop error, and i_{ref} is the reference current. Hence voltage input based state space model in equation 3.24 can be updated with the instruction of $\int i_e \, dt$ as state:

$$\begin{bmatrix} \frac{di}{dt} \\ \dot{\omega} \\ \dot{x} \\ \ddot{x} \\ \dot{p}_d \\ i_e \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{R+K_p^i}{L} & -\frac{k_E}{L} & 0 & 0 & 0 & \frac{K_I^i}{L} \\ \frac{k_T}{J_{eq}} & -\frac{b_{eq}}{J_{eq}} & 0 & 0 & -\frac{D_p}{J_{eq}} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{b}{m_p} & \frac{A_p}{m_p} & 0 \\ 0 & \frac{2\beta D_p}{V_d} & 0 & -\frac{2\beta A_p}{V_d} & -\frac{\beta C_{eq}}{V_d} & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A^i}} \begin{bmatrix} i \\ \omega \\ x \\ \dot{x} \\ p_d \\ \int i_e \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{K_p^i}{L} \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{\mathbf{B^i}} i_{ref} \tag{3.35}$$

The state space model in equation 3.35 is the torque-controlled EHA plant, similar to the transfer function generation in equation 3.24, a transfer function between i_{ref} and x can be generated.

$$P_i(s) = \frac{x(s)}{i_{\text{ref}}(s)} = \mathbf{C}^i \left(s\mathbf{I} - \mathbf{A}^i\right)^{-1} \mathbf{B}^i \tag{3.36}$$

$$\mathbf{C}^i = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.37}$$

For the integration of velocity controller to the state space, the same principle can be followed in the PI loop. The current error i_e is given as:

$$i_e = -i + \frac{1}{k_T}\left(K_p^\omega(\omega_{\text{ref}} - \omega) + K_I^\omega \int \omega_e\right) \tag{3.38}$$

Notice $\frac{1}{k_T}$ term directly merges from the driver settings. The current derivative $\frac{di}{dt}$ can be expressed as:

$$\frac{di}{dt} = \frac{K_p^i i_e + K_I^i \int i_e - iR - k_E\omega}{L} \tag{3.39}$$

Substituting i_e into the current derivative:

$$\frac{di}{dt} = -i\frac{(K_p^i + R)}{L} - \omega\frac{k_E + K_p^\omega K_p^i \frac{1}{k_T}}{L} + \int i_e \frac{K_I^i}{L} + \int \omega_e \frac{K_I^\omega K_p^i}{k_T L} + \omega_{\text{ref}} \frac{K_p^\omega K_p^i}{k_T L} \tag{3.40}$$

Hence giving equation 3.40, introducing $\int \omega_{ref}$ as a state, a state space model including current and velocity controllers of the driver can be generated:

$$\begin{bmatrix} \frac{di}{dt} \\ \dot{\omega} \\ \dot{x} \\ \ddot{x} \\ \dot{p}_d \\ i_e \\ \omega_e \end{bmatrix} = \underbrace{\begin{bmatrix} -\frac{R+K_p^i}{L} & -\frac{k_E+K_p^\omega K_p^i \frac{1}{k_T}}{L} & 0 & 0 & 0 & \frac{K_I^i}{L} & \frac{K_I^\omega K_p^i}{k_T L} \\ \frac{k_T}{J_{eq}} & -\frac{b_{eq}}{J_{eq}} & 0 & 0 & -\frac{D_p}{J_{eq}} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{b}{m_p} & \frac{A_p}{m_p} & 0 & 0 \\ 0 & \frac{2\beta D_p}{V_d} & 0 & -\frac{2\beta A_p}{V_d} & -\frac{\beta C_{eq}}{V_d} & 0 & 0 \\ -1 & -\frac{K_p^\omega}{L} & 0 & 0 & 0 & 0 & \frac{K_I^\omega}{L} \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}^\omega} \begin{bmatrix} i \\ \omega \\ x \\ \dot{x} \\ p_d \\ \int i_e \\ \int \omega_e \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{K_p^\omega K_p^i}{k_T L} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\frac{1}{m_p} \\ 0 & 0 \\ \frac{K_p^\omega}{k_T} & 0 \\ 1 & 0 \end{bmatrix}}_{\mathbf{B}^\omega} \begin{bmatrix} \omega_{\text{ref}} \\ F_L \end{bmatrix} \tag{3.41}$$

Notice that state space in 3.41 has also an external load as an input which will be analyzed for controller design in the next section. Similar to the plant derivation of the torque controller EHA plant, transfer functions between external force and position $P_d(s) = \frac{x(s)}{F_L(s)}$ and transfer function between reference motor speed and position $P_\omega(s) = \frac{x(s)}{\omega_{ref}(s)}$ can be obtained:

$$\begin{bmatrix} P_\omega(s) & P_d(s) \end{bmatrix} = \mathbf{C}^\omega \left(s\mathbf{I} - \mathbf{A}^\omega\right)^{-1} \mathbf{B}^\omega \tag{3.42}$$

$$\mathbf{C}^\omega = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.43}$$

Hence a detailed linear plant of EHA is obtained for different control modes (torque and motor speed). The linear model can be compared with the nonlinear model that is constructed according to the identified sub-components. In nonlinear model, current and voltage controllers are integrated as well as pump and actuator characteristics updated on the preliminary EHA model introduced in Figure 3.5.

For the closed loop experiment, a PID controller is designed with tuning methods. The simulation parameters are demonstrated in Table 3.8. Resulting closed loop response of non-nonlinear and linear simulations compared with the experimental result is illustrated in Figure 3.45. The linear model has no delays nor friction in the model hence it has a smoother start while nonlinear model matches experimental result better due to friction and driver delays in the nonlinear model.

It is important to note that the model is far from ideal. For instance, it is well-known and observed that above 50°C, hydraulic oil viscosity decreases rapidly, and oil addi-

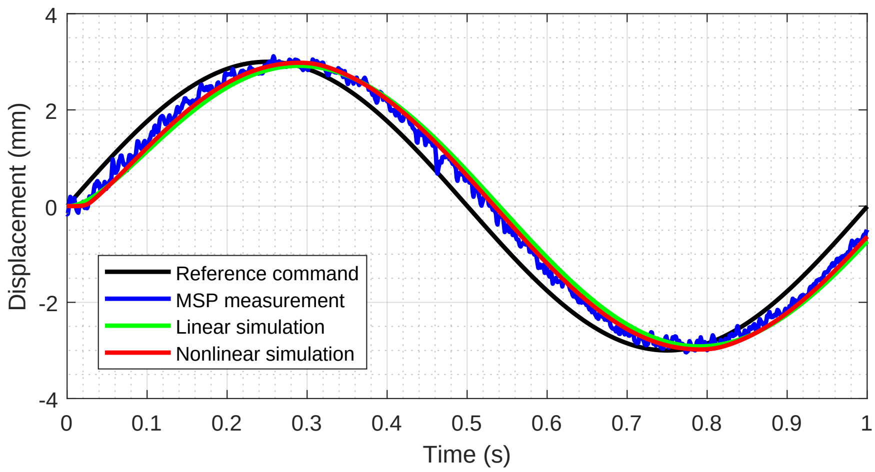

Figure 3.45: Comparison of simulation results with experimental data

tives begin to degrade. Even more critically, due to the compact design of the EHA, there are no hydraulic filters, which poses a risk to long-term performance. Hydraulic oil cleanliness may influence oil viscosity and cylinder friction, particularly over extended operational periods. Reduced viscosity can also lead to increased wear in the external gear pump, as the oil's lubrication properties degrade. While measures have been taken to prevent cavitation in the gear pump, such phenomena may still occur, leading to a reduction in the pump's volumetric efficiency over prolonged use.

Additionally, it should be noted that the torque and back-EMF constants used in the model are approximations. These constants are known to vary with factors such as current, temperature, and other external conditions, which introduces further uncertainty into the model's accuracy. Hence robust control designs which accounts uncertain plant behavior should be considered.

Table 3.8: Nominal values for simulation

Parameter	**Nominal value**
Leakage coefficient (C_{eq})	3.8×10^{-12} $\left[\frac{\text{m}^3/\text{s}}{\text{Pa}}\right]$
Pump displacement (D_p)	6.684×10^{-7} $\left[\frac{\text{m}^3}{\text{rad}}\right]$
Bulk modulus (β)	1.379×10^9 [Pa] (MIL-PRF-5606)
Torque constant (k_T)	0.73 $\left[\frac{\text{Nm}}{\text{A}}\right]$
Motor resistance (R)	0.45 [Ω]
Motor inductance (L)	2.1×10^{-3} [H]
Pump-motor inertia (J_{eq})	5.93×10^{-4} [kg.m^2]
Actuator area (A_p)	2572×10^{-6} [m^2]
Chamber volume (V_d)	4.01×10^{-4} [m^3]
Piston mass (m_p)	2 [kg]
Viscous damping of actuator (b)	1120 $\left[\frac{\text{N}}{\text{(m/s)}}\right]$
Viscous damping of motor-pump couple (b)	100 $\left[\frac{\text{N}}{\text{(rad/s)}}\right]$
Velocity loop integral gain (K_I^ω)	56 $\left[\frac{\text{Nm}}{\text{(rad/s)}}\right]$
Velocity loop P gain (K_P^ω)	0.45 $\left[\frac{\text{Nm}}{\text{rad/s}}\right]$
Current loop P gain (K_P^i)	11.3 $\left[\frac{\text{V}}{\text{A}}\right]$
Current loop integral gain (K_I^i)	14125 $\left[\frac{\text{V}}{\text{A.s}}\right]$
Position control loop P gain (K_P^{pos})	900 $\left[\frac{\text{rpm}}{\text{mm}}\right]$
Position control loop I gain (K_I^{pos})	105.88 $\left[\frac{\text{rpm}}{\text{mm.s}}\right]$
Position control loop D gain (K_D^{pos})	0.9 $\left[\frac{\text{rpm.s}}{\text{mm}}\right]$

CHAPTER 4

CONTROLLER DESIGN

In this chapter, techniques for state estimation aimed at mitigating noise will be explored. The EHA plant will be evaluated, and parametric uncertainties will be defined. These will then inform the design of a robust controller.

4.1 Kalman Filter Implementation

In EHAs and flight control systems, state estimation is crucial for improving measurements and diagnosing faults. Some states cannot be directly measured, while others may be prohibitively expensive to sense. State estimation methods address these challenges by offering a cost-effective alternative. Among the foundational contributions to the state estimation of EHAs is Chinniah's work, which implemented the Kalman filter for friction modeling and fault diagnosis [33]. This section presents the theoretical basis of the Kalman filter, an example from existing literature, and its application to a faulty position sensor.

4.1.1 Background

The Kalman filter, developed by Rudolf E. Kalman in 1960, is a statistically optimal algorithm used to estimate the states of a dynamic system using measurements and a mathematical model of the system. Its foundation lies in linear systems theory and statistical principles, making it indispensable in areas such as control systems, robotics, and signal processing.A great source for Kalman filtering is the works of Alex Becker, explaining Kalman filter in his website and book[63].

4.1.1.1 State Prediction Equation

The next state of the system is predicted using the system dynamics:

$$\hat{\mathbf{x}}_{n+1|n} = \mathbf{F}\hat{\mathbf{x}}_{n|n} + \mathbf{G}\mathbf{u}_n + \mathbf{w}_n, \tag{4.1}$$

where:

- $\hat{\mathbf{x}}_{n+1|n}$: Predicted state vector at time $n + 1$.
- $\hat{\mathbf{x}}_{n|n}$: Estimated state vector at time n.
- $\mathbf{u}_n$: Control input vector.
- $\mathbf{w}_n$: Process noise (assumed zero-mean Gaussian).
- $\mathbf{F}$: State transition matrix.
- $\mathbf{G}$: Input control matrix.

The *state prediction equation* is a state-space equation derived from the system dynamics. It models how the system evolves over time based on the previous state, control inputs, and inherent uncertainties.

It is important to note that the process noise $\mathbf{w}_n$ is not directly measurable. It represents the deviation between the ideal state-space model and the actual system dynamics, which are influenced by unmodeled dynamics or disturbances. This deviation is quantified by the process noise covariance matrix $\mathbf{Q}$, which may vary with time.

In practice, $\mathbf{Q}$ reflects how uncertain the system dynamics are, helping to account for non-idealities in the state-space model.

4.1.1.2 Covariance Prediction Equation

The uncertainty in the predicted state is propagated as:

$$\mathbf{P}_{n+1|n} = \mathbf{F}\mathbf{P}_{n|n}\mathbf{F}^T + \mathbf{Q}, \tag{4.2}$$

where:

- $\mathbf{P}_{n+1|n}$: Predicted covariance matrix.

- $\mathbf{P}_{n|n}$: State estimate covariance matrix at time n.

- $\mathbf{Q}$: Process noise covariance matrix.

4.1.1.3 Measurement Update Equation

The measurement model updates the state estimate using:

$$\mathbf{z}_n = \mathbf{H}\mathbf{x}_n + \mathbf{v}_n, \tag{4.3}$$

where:

- $\mathbf{z}_n$: Measurement vector.

- $\mathbf{x}_n$: True system state (hidden state).

- $\mathbf{H}$: Measurement matrix.

- $\mathbf{v}_n$: Measurement noise (zero-mean Gaussian).

Here measurement update equation may be regarded as the equivalent output equation for the state space representation of the system. States may not be measurable themselves, hence there is $\mathbf{H}$ defined, a linear transformation between states and measurement. Similar to the state prediction, measurement noise $\mathbf{v}_n$ is not measured however its covariance $\mathbf{R}_n$ informs Kalman gain on how much to rely on sensor measurements.

4.1.1.4 Kalman Gain Calculation

The Kalman gain determines the weight given to the new measurement:

$$\mathbf{K}_n = \mathbf{P}_{n|n-1}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_{n|n-1}\mathbf{H}^T + \mathbf{R}_n\right)^{-1}, \tag{4.4}$$

where $\mathbf{R}_n$ is the measurement noise covariance matrix.

4.1.1.5 State Update Equation

Using the Kalman gain, the state estimate is updated as:

$$\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + \mathbf{K}_n \left(\mathbf{z}_n - \mathbf{H}\hat{\mathbf{x}}_{n|n-1}\right), \tag{4.5}$$

4.1.1.6 Covariance Update Equation

Finally, the estimate covariance is updated:

$$\mathbf{P}_{n|n} = \left(\mathbf{I} - \mathbf{K}_n\mathbf{H}\right)\mathbf{P}_{n|n-1}\left(\mathbf{I} - \mathbf{K}_n\mathbf{H}\right)^T + \mathbf{K}_n\mathbf{R}_n\mathbf{K}_n^T. \tag{4.6}$$

Notice that equation 4.6 has a simplified from by inserting Kalman gain equation 4.4:

$$\mathbf{P}_{n|n} = (\mathbf{I} - \mathbf{K}_n\mathbf{H})\mathbf{P}_{n|n-1} \tag{4.7}$$

While equation 4.7 is a more elegant representation and may perform well, it is a numerically unstable equation.

4.1.1.7 Kalman Filter Workflow

5 main Kalman Filter equations-measurement equation as an auxiliary- 4.1, 4.2, 4.4, 4.5 and 4.6, operates in predict and correct cycle. During initialization, the system state is predicted for the next step (4.1), also the uncertainty of the prediction is extrapolated (4.2). Based on the measurement signal, uncertainty (4.6) and states (4.5) are updated according to the Kalman gain (4.4), balancing measurement noise and prediction uncertainty.

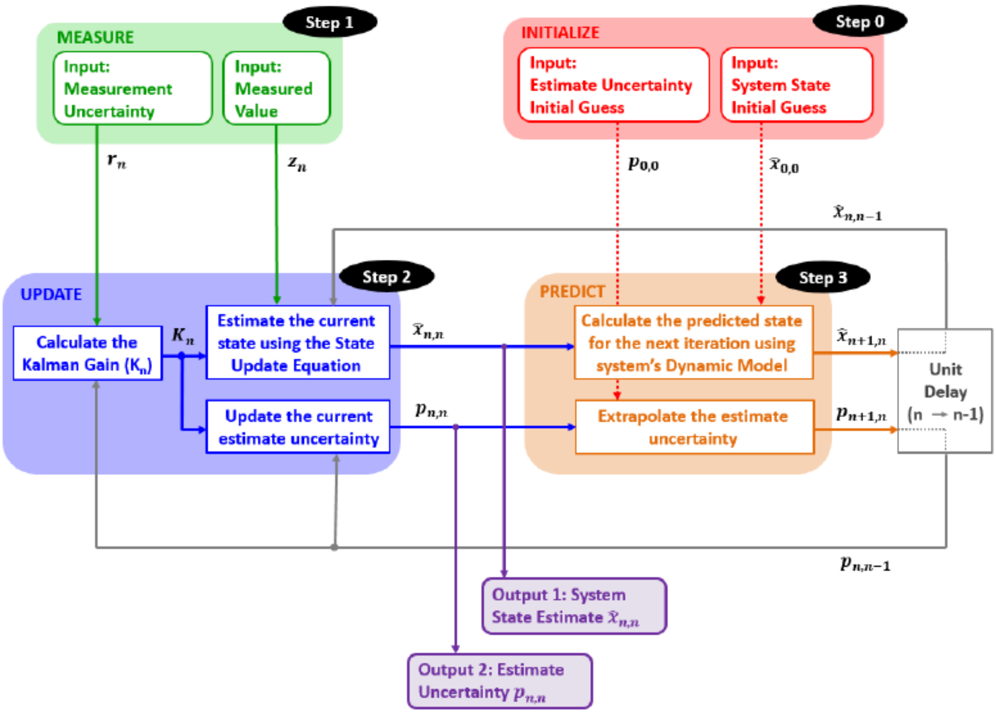

Figure 4.1: Description of Kalman filtering steps [63]

The entire Kalman filter process can be visualized as in Figure 4.1, comprising the following steps:

1. **Initialization (Step 0):** Set the initial state $\hat{\mathrm{x}}_{0|0}$ and initial covariance $\mathbf{P}_{0|0}$.
2. **Measurement (Step 1):** Provide measured state $\mathbf{z}_n$ and measurement variance $\mathbf{R}_n$.
3. **State Update (Step 2):** Update states and prediction uncertainty by calculating Kalman gain.
4. **Prediction (Step 3):** Extrapolate system state estimation and its variance to update state and estimation uncertainty on the next step.

4.1.2 Implementation from literature

A friction-estimation algorithm proposed by Chinniah will be investigated in this section [64]. This algorithm is also explained in greater detail in his doctoral thesis [65].

He proposes an experimental friction model described as follows:

$$F_f = \alpha_1 \,\text{sign}(\dot{x})\dot{x}^2 + \alpha_2\dot{x} + \alpha_3 \,\text{sign}(\dot{x}) \tag{4.8}$$

where:

- α_1, α_2, and α_3 are model parameters of friction,
- $\dot{x}$ represents the piston velocity,
- sign($\dot{x}$) is the sign function indicating the direction of motion.

The friction coefficient parameters are demonstrated in Table 4.1.

Parameter	**Value**	**Unit**
α_1	2.1×10^4	Ns^2/m^2
α_2	-1450	Ns/m
α_3	46	N

Table 4.1: Friction model parameters

The obtained friction model is relevant for the Newton's second law of motion for the force balance equation that is previously mentioned in EHA's mathematical model in equation 3.23.

$$A_p(p_A - p_B) - F_f = m_p\ddot{x} \tag{4.9}$$

Hence, a nonlinear state space model can be generated based on equations 4.8 and 4.9:

$$\dot{X}_1 = X_2 + w_1 \tag{4.10}$$

$$\dot{X}_2 = \frac{A_p\Delta P}{m_p} - \frac{X_3X_2^2}{m_p}\text{sign}(X_2) - \frac{X_4X_2}{m_p} - \frac{X_5}{m_p}\text{sign}(X_2) + w_2 \tag{4.11}$$

$$\dot{X}_3 = w_3 \tag{4.12}$$

$$\dot{X}_4 = w_4 \tag{4.13}$$

$$\dot{X}_5 = w_5 \tag{4.14}$$

Equation 4.11 assumes that the friction behavior is unknown and models the noise of the system as friction based on the relationship between pressure and piston acceleration. X_3, X_4, X_5 correspond to the friction coefficients in equation 4.8. Here, X_2 represents the piston velocity, and X_1 denotes the piston position. For the observability problem, an iterative Kalman filter structure is proposed by Chinniah, calculating α_3 with a particular Kalman filter, then calculating α_1 and α_1 in a second Kalman filter. Hence, α_3 can be extracted from equation 4.8:

$$F_f^{\alpha_3} = \alpha_3 \,\mathrm{sign}(\dot{x}) \tag{4.15}$$

For the discrete model, continuous equations are converted by sample time T_s. The system transition matrix in relation to equation 4.9 and 4.15 can be represented discretely as:

$$F_k = \phi_k = \begin{bmatrix} 1 & T_s & 0 \\ 0 & 1 - \frac{X_3 k T_s}{m_p} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad T_s = 1 \times 10^{-4}\,\mathrm{s} \tag{4.16}$$

The first state represents the piston position, the second state represents the piston velocity, and the third state corresponds to friction generated by α_3 which is modeled as a process noise. T_s is the simulation step size, and the corresponding discrete-time derivatives are used for this value. Equation 4.1 can be represented as follows:

$$\hat{x}_{k+1} = \phi_k \hat{x}_k + G u_n \tag{4.17}$$

$$G = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \tag{4.18}$$

$$u_n = \frac{\Delta P T_s A_p}{m_p} \tag{4.19}$$

Notice in equation 4.19 pressure difference is directly integrated to the input vector, without any measurement noise, which will be discussed later in the section. For the measurement matrix $\mathbf{H}$ defined in equation 4.3, piston position and velocity are measured. Hence matrix initializations for the 3-state Kalman filter are defined as:

$$R_k = \begin{bmatrix} 10^{-9} & 0 \\ 0 & 10^{-3} \end{bmatrix} \tag{4.20}$$

$$P_0 = \begin{bmatrix} 10^{9} & 0 & 0 \\ 0 & 10^{9} & 0 \\ 0 & 0 & 10^{9} \end{bmatrix} \tag{4.21}$$

$$Q_k = \begin{bmatrix} 10^{-9} & 0 & 0 \\ 0 & 10^{-12} & 0 \\ 0 & 0 & 10^{-4} \end{bmatrix} \tag{4.22}$$

Thus first part of the 3-state Kalman filter is defined completely. The second Kalman filter follows 4 state system matrices approach to estimate α_1 and α_2 coefficients, keeping the same $\mathbf{H}$ to measure piston position and velocity. The Simulink implementation of the two-filter iterative structure is illustrated in Figure 4.2. White noise

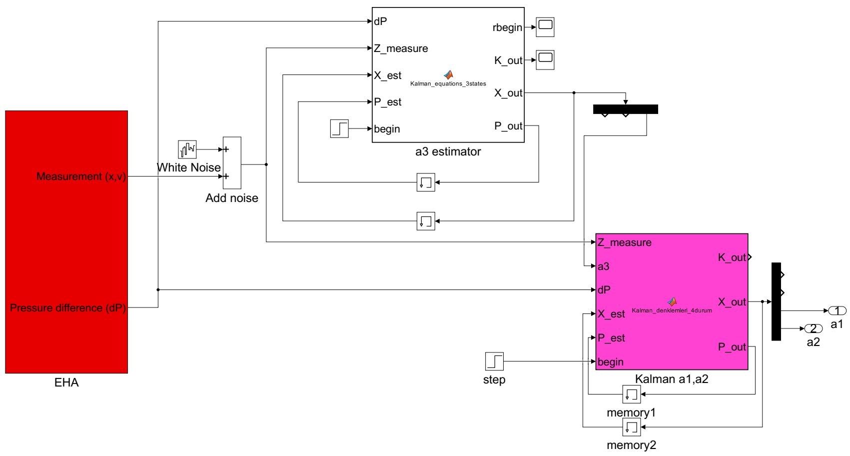

Figure 4.2: Kalman filter structure proposed by Chinniah constructed in Simulink

is added to the position and velocity measurements in the system. For the position measurement, the noise variance is $1 \times 10^{-10}\,\mathrm{m}^2$, and for the velocity measurement, the noise variance is $0.01\,(\mathrm{m/s})^2$. The Matlab function codes for the friction model, 3-state Kalman filter and 4-state Kalman filter are shared in the Appendix C. For the 4-state Kalman Filter, α_3 value estimated on the previous filter is inserted as an input.

The state transition matrix in the 4-state Kalman filter is derived from equations 4.10-4.14 however α_3 and ΔP terms are defined as inputs to the system equations:

$$\Phi_k = \begin{bmatrix} 1 & T_s & 0 & 0 \\ 0 & 1 & -\text{sign}(\hat{x}_{2k})\frac{T_s\hat{x}_{2k}^2}{m_p} & -\frac{T_s\hat{x}_{2k}}{m_p} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

Accordingly, the system dynamics can be modeled as follows:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + G u_n \tag{4.24}$$

where:

$$G = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.25}$$

$$u_n = \frac{T_s}{m_p}\left(A_p \Delta P + \hat{\alpha_3}\text{sign}(\hat{x}_{2k})\right) \tag{4.26}$$

Hence, complete modeling of the system is achieved by state prediction equations. Matrix initializations for 4-state Kalman filter are designed as:

$$R_k = \begin{bmatrix} 10^{-12} & 0 \\ 0 & 10^{-3} \end{bmatrix} \tag{4.27}$$

$$P_0 = \begin{bmatrix} 10^9 & 0 & 0 & 0 \\ 0 & 10^9 & 0 & 0 \\ 0 & 0 & 10^9 & 0 \\ 0 & 0 & 0 & 10^9 \end{bmatrix} \tag{4.28}$$

$$Q_k = \begin{bmatrix} 10^{-12} & 0 & 0 & 0 \\ 0 & 10^{-7} & 0 & 0 \\ 0 & 0 & 10^{-7} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{bmatrix} \tag{4.29}$$

A closed-loop simulation is conducted on EHA with 4 Hz sin input with an amplitude of 3 mm. The resulting friction estimations of Kalman filter are illustrated in Figure 4.3. The resulting algorithm estimates friction coefficients successfully, where such algorithm may be used for fault-detection methods.

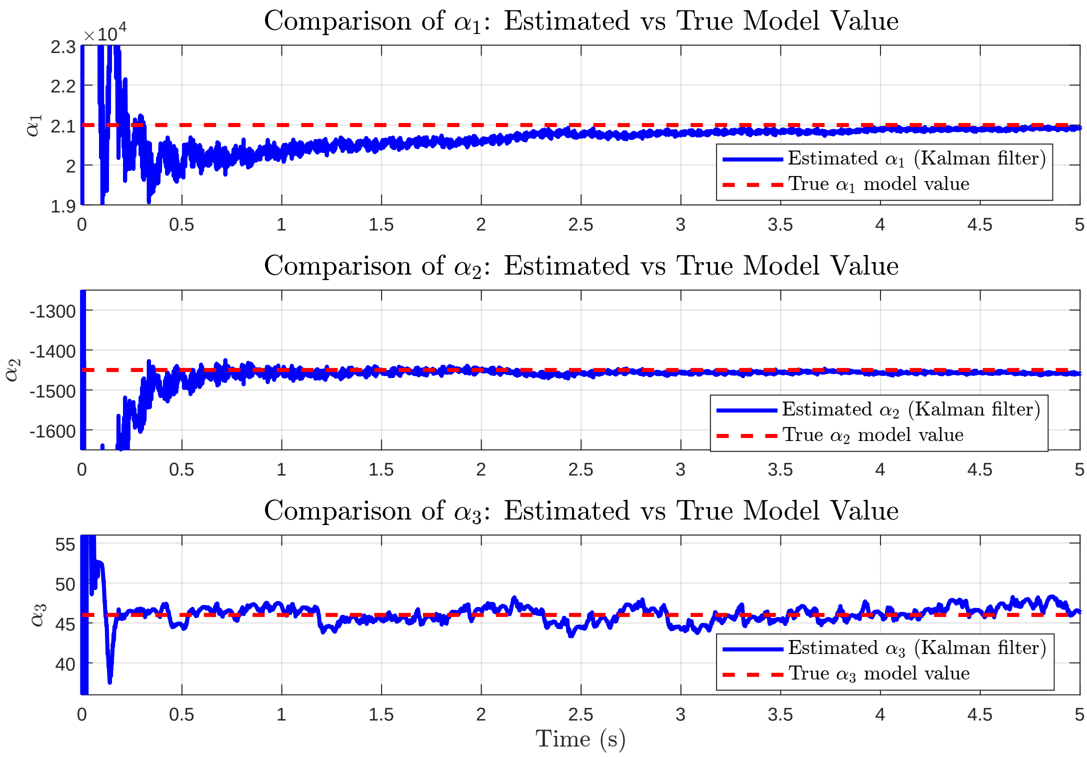

Figure 4.3: Convergence of friction parameter estimations during simulation

To investigate the robustness of the algorithm, two questions arise, pressure values directly being fed as an input and whether the algorithm is robust to friction model changes. For this purpose a white noise with a variance of $0.1\,\text{bar}^2$ is added to the pressure in simulation to model measurement noise. This corresponds to a standard deviation of $\sqrt{0.1} \approx 0.316\,\text{bar}$. The added noise simulates real-world inaccuracies in pressure measurements and helps evaluate the system's robustness. The success of the filtering approach can be observed in Figure 4.4, where piston velocity is estimated given the noise added signal. Similarly, friction coefficients are estimated successfully.

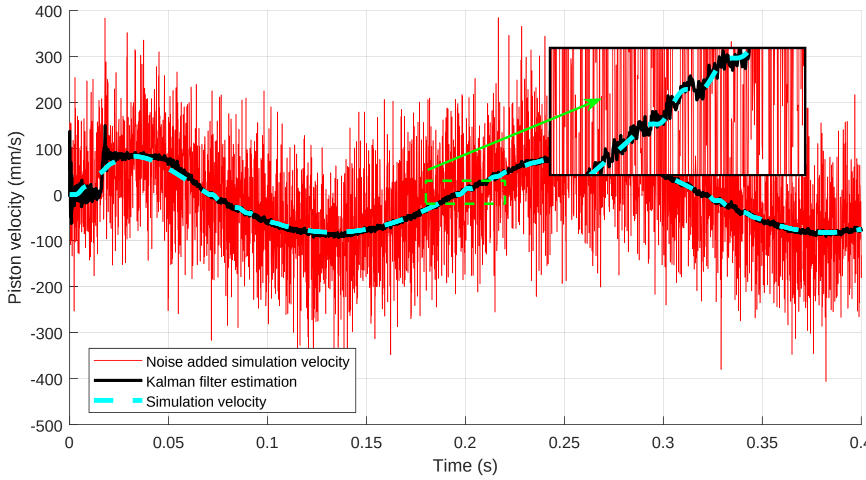

Figure 4.4: Piston velocity estimation with pressure noise addition

Notice that the Kalman filter has a settling time for the estimation of the friction values. One question arises: since the friction model is an approximation, does the filter detect dynamic changes in the coefficients? This is crucial, as the friction of the piston may vary due to factors such as heat, cleanliness, and wear.

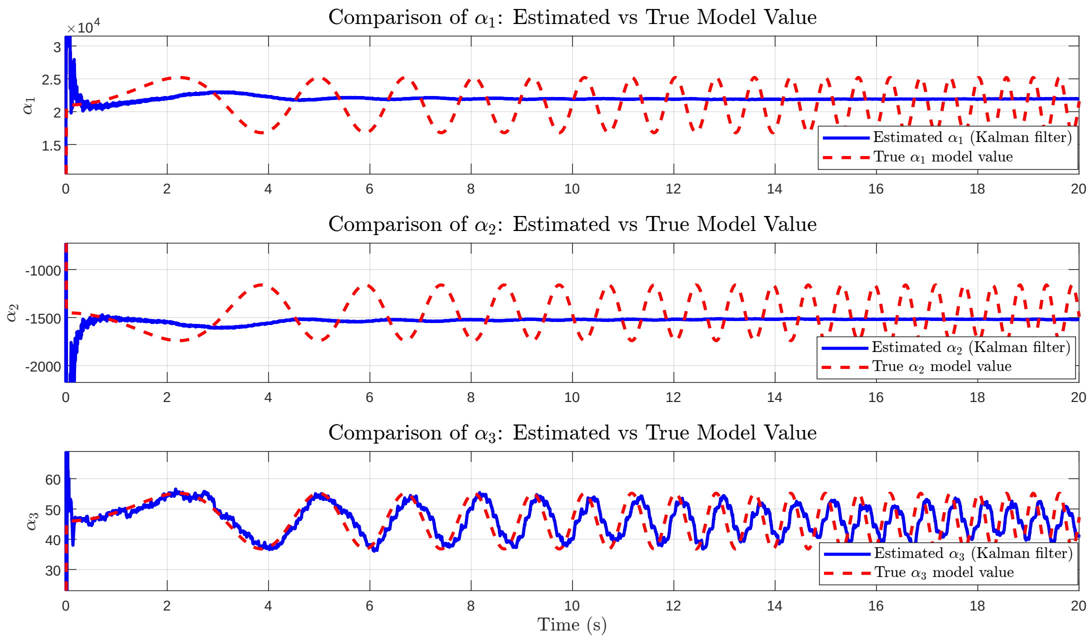

Figure 4.5: Friction parameter estimations under dynamic changes

To investigate this, a chirp signal distortion for the friction parameters is generated, varying the coefficients by $\pm 20\%$ of their original value. The resulting estimations of friction parameters are demonstrated in Figure 4.5. While the friction coefficient α_3 is estimated close to the simulated value especially at low frequencies, the coefficients α_1 and α_2 fail to track the dynamic changes of the values. However, they settle close to the average value of the friction coefficient.

4.1.3 Position estimation

The position sensor used for displacement measurement has a high noise amplitude with a low frequency behavior, which can be considered under faulty mode, as illustrated in Figure 4.6.

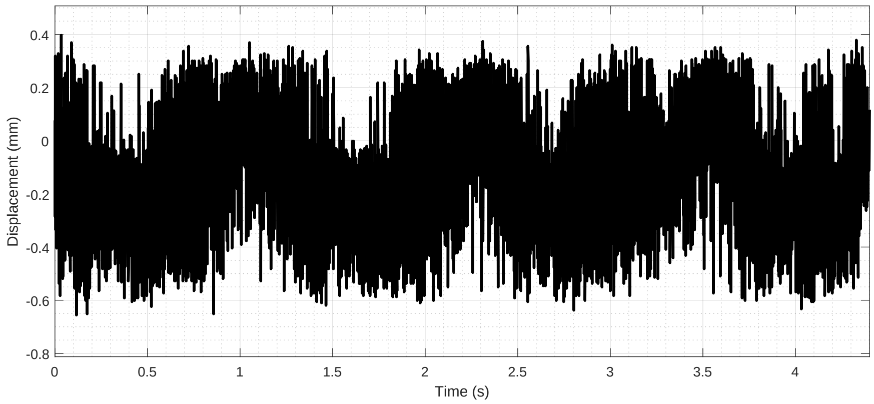

Figure 4.6: MSP measurement on standalone configuration

To estimate piston displacement effectively, different approaches are followed and discussed in this section.

The Position-Velocity-Acceleration (PVA) Kalman Filter is a state estimation method that models an object's motion using its position measurement, modeling process noise as jerk. The algorithm estimates the state X, which includes position, velocity, and acceleration, based on noisy measurements Z. According to [66, page 167], the Kalman filter implementation on PVA for the discrete time step can be shown as:

$$F_k = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix}, \tag{4.30}$$

where T_s is the sampling time.

The measurement model is:

$$H = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{4.31}$$

The process noise covariance Q is defined as:

$$Q_k = \begin{bmatrix} \frac{W}{20}T_s^5 & \frac{W}{8}T_s^4 & \frac{W}{6}T_s^3 \\ \frac{W}{8}T_s^4 & \frac{W}{3}T_s^3 & \frac{W}{2}T_s^2 \\ \frac{W}{6}T_s^3 & \frac{W}{2}T_s^2 & WT_s \end{bmatrix}, \tag{4.32}$$

where W is the jerk noise intensity.

The measurement noise covariance R_k is defined according to the experimental measurement noise of the position sensor:

$$R_k = 2.5 \times 10^{-8} \tag{4.33}$$

The initial state covariance P_0 is:

$$P_0 = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 100 \end{bmatrix} \tag{4.34}$$

The resulting Kalman filter is integrated into Simulink, as demonstrated in previous examples shown in Figure 4.2, with a clock timer added to run the block at a different sample time from the simulation. The code structure of matlab and TWINCAT are provided in Appendix C. A closed-loop simulation with a noisy position signal, a step position reference, and external force disturbance is executed to observe the effectiveness of the PVA Kalman filter. The resulting displacement estimation is illustrated in Figure 4.7.

However, on experimental test setup one disadvantage of the PVA filter it either can not dismiss low frequency noise or has a phase lag on its frequency response. Implementation of the filter on EHA's measurement sensor is illustrated in Figure 4.8.

Hence a model-based approach in order to eliminate low frequency noise is followed. A key approach here is that motor encoder is very reliable in terms of measurement,

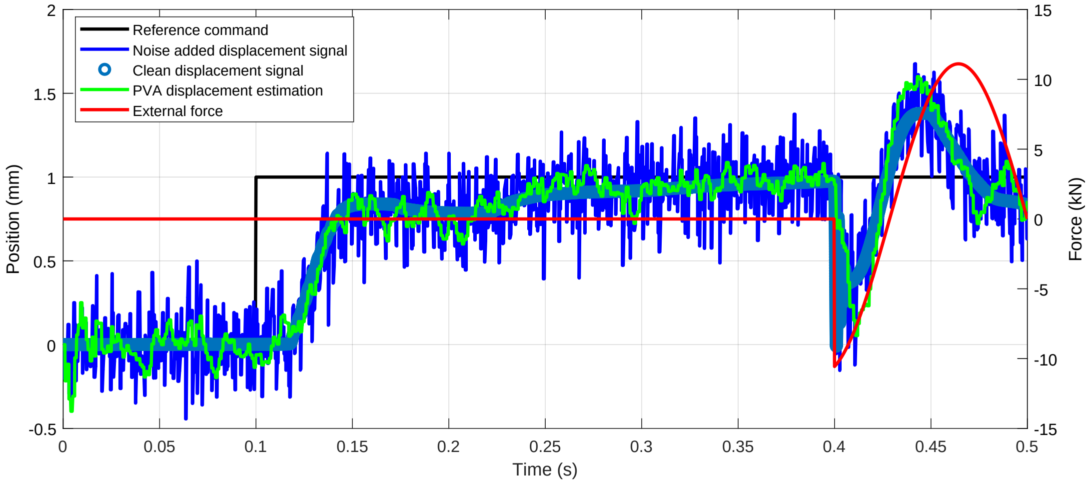

Figure 4.7: Displacement estimation of PVA Kalman filter

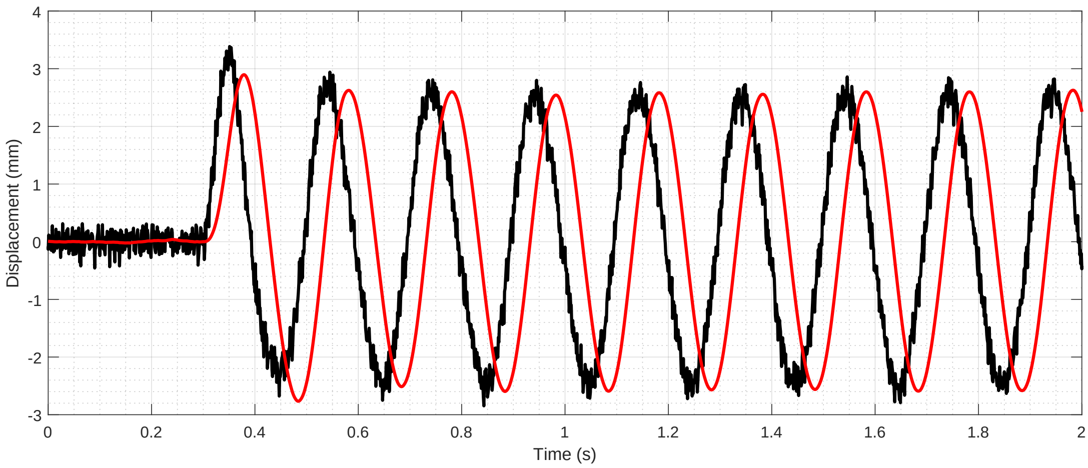

Figure 4.8: Displacement estimation of PVA filter during experiment

hence it can be modeled as an input into the system as in Chinniah's friction estimation with pressure difference as an input to the system example in section 4.1.2. Hence a state prediction equation is suggested in the form:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{p}_D \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{m_p} & -\frac{A_p}{m_p} \\ 0 & -\frac{2A_p\beta}{V_d} & -\frac{C_{eq}\beta}{V_d} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ p_D \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{1}{m_p} \\ \frac{2D_p\beta}{V_d} & 0 \end{bmatrix} \begin{bmatrix} \omega \\ F_L \end{bmatrix} \tag{4.35}$$

State-space form can be discretized with zero-order hold with a given sample time. ω is measured accurately from the motor driver. For the disturbance input F_L, in the no-load case, the friction value can be estimated by neglecting acceleration:

$$F_L = -\text{sign}(\dot{x}) \min(p_D A_p, F_s) \tag{4.36}$$

F_s is utilized as Coulomb friction of magnitude 800 N. For the loaded case, force measurement from a load sensor and a force estimator may be integrated with friction force to estimate external force. The measurement uncertainty matrix and system noise matrix are tuned with trial and error.

The controller has similar code structure with PVA with its implementation on matlab function block and TWINCAT program. For the TWINCAT program matrix blocks, a conversion code is written in matlab in order to obtain numerically accurate values in PLC program. The simulation results of the model based Kalman filter is illustrated in Figure 4.9.

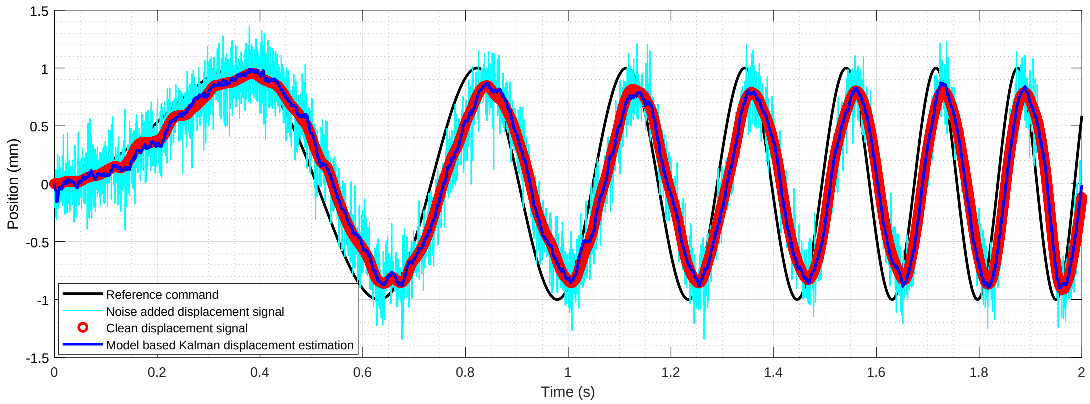

Figure 4.9: Simulation results on model-based Kalman Filter

The update matrices are selected as follows:

$$R = \begin{bmatrix} 2.5 \times 10^{-10} & 0 & 0 \\ 0 & 160.0 & 0 \\ 0 & 0 & 100000.0 \end{bmatrix} \tag{4.37}$$

$$P_0 = \begin{bmatrix} 1.0 \times 10^{-9} & 0 & 0 \\ 0 & 1.0 \times 10^{-9} & 0 \\ 0 & 0 & 1.0 \times 10^{-10} \end{bmatrix} \tag{4.38}$$

$$Q = \begin{bmatrix} 1.0 \times 10^{-12} & 0 & 0 \\ 0 & 1.0 \times 10^{-7} & 0 \\ 0 & 0 & 1.0 \times 10^{-11} \end{bmatrix} \tag{4.39}$$

$$H = \begin{bmatrix} 1.0 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.0 \end{bmatrix} \tag{4.40}$$

$$F = \begin{bmatrix} 1.0 & 0.0000197 & -1.533 \times 10^{-10} \\ 0 & -0.987 & -1.311 \times 10^{-8} \\ 0 & 231200.0 & -0.9869 \end{bmatrix} \tag{4.41}$$

$$G = \begin{bmatrix} 5.928 \times 10^{-8} & -2.54 \times 10^{-7} \\ 5.628 \times 10^{-6} & -0.0005163 \\ 772.4 & 70.13 \end{bmatrix} \tag{4.42}$$

$$T_s = 1 \times 10^{-3} s \tag{4.43}$$

The process covariance values due to noises are determined via tuning while measurement covariance values are determined with respect to amplitudes of sensor measurement noise. On Figure 4.10, model-based Kalman Filter implementation on MSP measurement is illustrated. The filter successfully eliminates sensor noise, however now a phase lead effect is observed along with gain difference.

The frequency response of the developed Kalman Filter, generated via FFT of estimation with respect to measurement signal, is shown in Figure 4.11. A gain drop is observed around the bandwidth of the EHA, along with a phase lead in the frequency response. The filter estimates position more rapidly due to its linearity, as it does not account for driver delay or static friction effects. While an artificial delay could be added to eliminate the phase lead, it has been observed that the phase lead does not degrade the performance of the EHA.

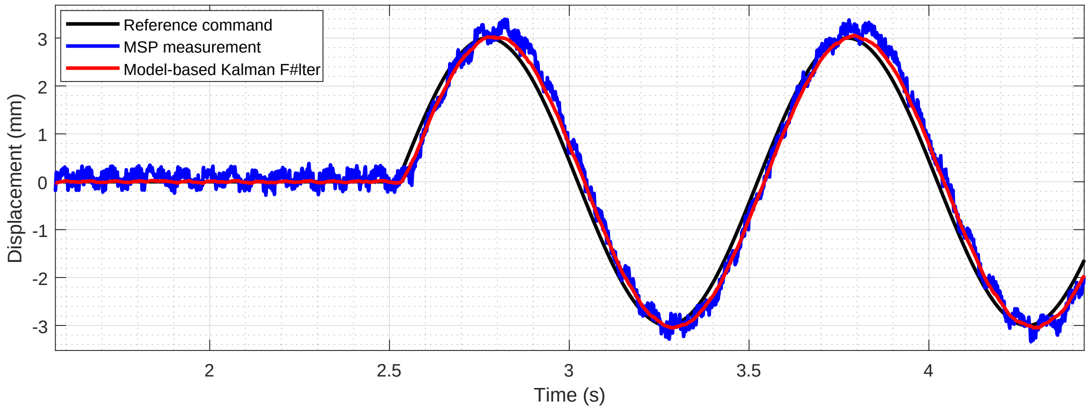

Figure 4.10: Model-based Kalman Filter on position measurement

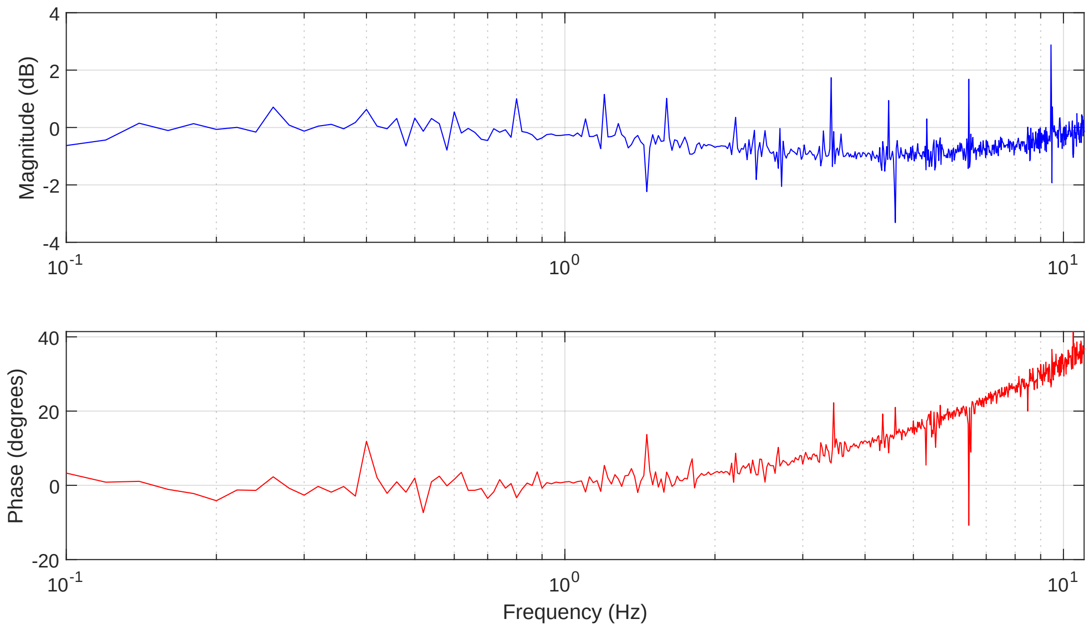

Figure 4.11: Frequency response of model-based Kalman Filter position estimation with respect to position measurement

As a consequence, a framework for fault detection in EHAs and a Kalman Filter for position estimation is developed to reject measurement noise.

4.2 Analysis of EHA plant

EHA was identified in the previous section with respect to experimental data. Various state-space models of EHA were derived, which will be discussed in this section. These are:

1. **Hydro-mechanical state-space model** in equation 4.35, which involves dynamics between the electric motor's rotational speed and the hydraulic actuator.
2. **Voltage input state-space model** in equation 3.24, which additionally integrates the electrical behavior of the motor, where voltage affects motor current and speed.
3. **Torque mode state-space model** in equation 3.35, which additionally integrates the motor driver's PI controller of torque mode.
4. **Velocity mode state-space model** in equation 3.41, which additionally integrates the motor driver's PI controller of velocity mode.

4.2.1 Plant poles

The hydro-mechanical system defined in equation 4.35, the transfer function between motor speed and piston position $\frac{X(s)}{\omega(s)}$ can be generated:

$$\frac{X(s)}{\omega(s)} = \frac{6663.1}{s\left(s^2 + 263.7s + 2.564 \times 10^7\right)} \tag{4.44}$$

The poles of transfer function in equation 4.44 are illustrated in Table 4.2.

Table 4.2: Poles of the transfer function $\frac{X(s)}{\omega(s)}$

Pole	**Natural Frequency (ω_n) [rad/s]**	**Damping Ratio (ζ)**
$-131.85 + j5061.88$	5063.6	0.026
$-131.85 - j5061.88$	5063.6	0.026
0	Integrator	N/A

The integrator term arises due to transformation of velocity to position. The high frequency poles arises due to dynamics of hydro-mechanical system, where the compressibility of the hydraulic fluid generated a high frequency dynamics, a common theme in hydraulic systems. If the integrator is neglected, the gain between pump speed and piston speed is calculated as $2.59 \cdot 10^{-4}$. This gain corresponds to transformer ratio between pump speed and actuator speed in physical ratio of $\frac{D_p}{A_p}$, in equivalent value of 0.26 $\frac{mm}{rad}$. This ratio is also intuitive in the sense at low frequencies, neglecting leakage and compressibility of the fluid, flow delivered by the pump displacement translates the piston. It is possible to observe effect of high frequency oscillation and piston speed to actuator speed transformation in Figure 4.12 given impulse response of transfer function $\frac{X(s)}{\omega(s)}$.

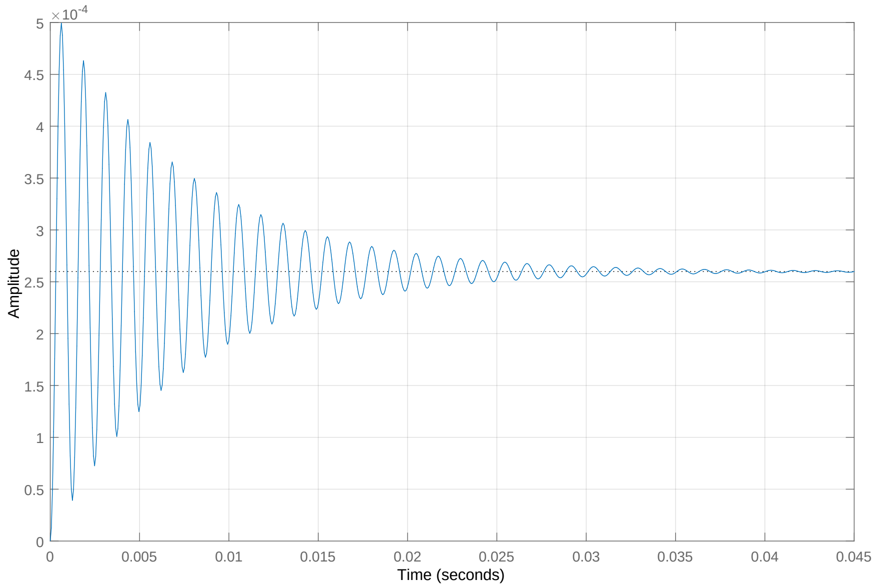

Figure 4.12: Impulse response of $\frac{X(s)}{\omega(s)}$

In equation 3.24, a state space model including dynamics of the electrical motor is introduced. Eigenvalues of system matrix A, corresponding to poles of the transfer function of the system are illustrated in Figure 4.13, along with the eigenvalues of the hydro-mechanical EHA model of previous analysis.

Notice that poles of fluid compressibility dynamics are same together with integrator, now a lower frequency dynamics is observed with the integration of simple electric motor dynamics. Resulting poles are shown in Table 4.3. In the previous sections it is

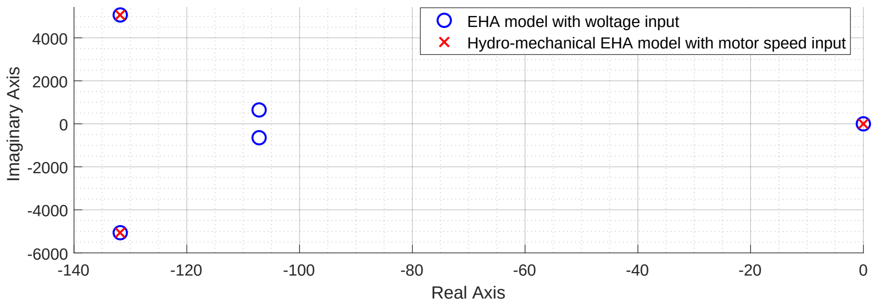

Figure 4.13: Poles of system models of EHA

demonstrated that linear simulation results comply with nonlinear and experimental results, however there is a delay occurring due to friction and driver time delay. In torque and velocity mode plant models, process noises such as motor friction and measurement noises of current and motor encoder are compensated by the controllers.

Table 4.3: Poles of the voltage-input modeled EHA system

Pole	**Natural Frequency (ω_n) [rad/s]**	**Damping Ratio (ζ)**
$-107.15 + j654.68$	662.3	0.162
$-107.15 - j654.68$	662.3	0.162
$-131.8 + j5062$	5064	0.026
$-131.8 - j5062$	5064	0.026
0	Integrator	N/A

4.2.2 EHA stiffness

The stiffness of the actuator without the controller also can be investigated. External load can be integrated into equation 3.24 such that:

$$\begin{bmatrix} \frac{di}{dt} \\ \dot{\omega} \\ \dot{x} \\ \ddot{x} \\ \dot{p}_d \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{k_E}{L} & 0 & 0 & 0 \\ \frac{k_T}{J_{eq}} & -\frac{b_{eq}}{J_{eq}} & 0 & 0 & -\frac{D_p}{J_{eq}} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{b}{m_p} & \frac{A_p}{m_p} \\ 0 & \frac{2\beta D_p}{V_d} & 0 & -\frac{2\beta A_p}{V_d} & -\frac{\beta C_{eq}}{V_d} \end{bmatrix} \begin{bmatrix} i \\ \omega \\ x \\ \dot{x} \\ p_d \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} V + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{m_p} \\ 0 \end{bmatrix} F_L \tag{4.45}$$

From equation 4.2.2, transfer function between position and external force $\frac{X(s)}{F_L(s)}$ can be generated:

$$\frac{X(s)}{F_L(s)} = \frac{0.5\,(s+16.38)\,(s^2+211.6s+4.332\times 10^5)}{s\,(s^2+214.3s+4.278\times 10^5)\,(s^2+263.6s+2.565\times 10^7)} \tag{4.46}$$

Equation 4.46 corresponds to compliance of the EHA without position controller. Stiffness of the actuator can be derived along with frequency of force, if inverse of compliance transfer function is considered. An analogy here is stiffness coefficient between force and position, however the coefficient varies with respect to frequency. A high stiffness is desired for the flight control actuator. In Figure 4.14, the frequency response of dynamic stiffness of EHA plant is plotted. Notice at high frequencies, due to inertia of force, stiffness increases. In fact, due to s^2 derivative term of $(\frac{X(s)}{F_L(s)})^{-1}$, at high frequencies it increases at a rate of $40dB/decade$. At frequency around 800 Hz, a resonance with the compressibility of the fluid occurs which corresponds to high frequency poles in Figure 4.13, hence stiffness decreases.

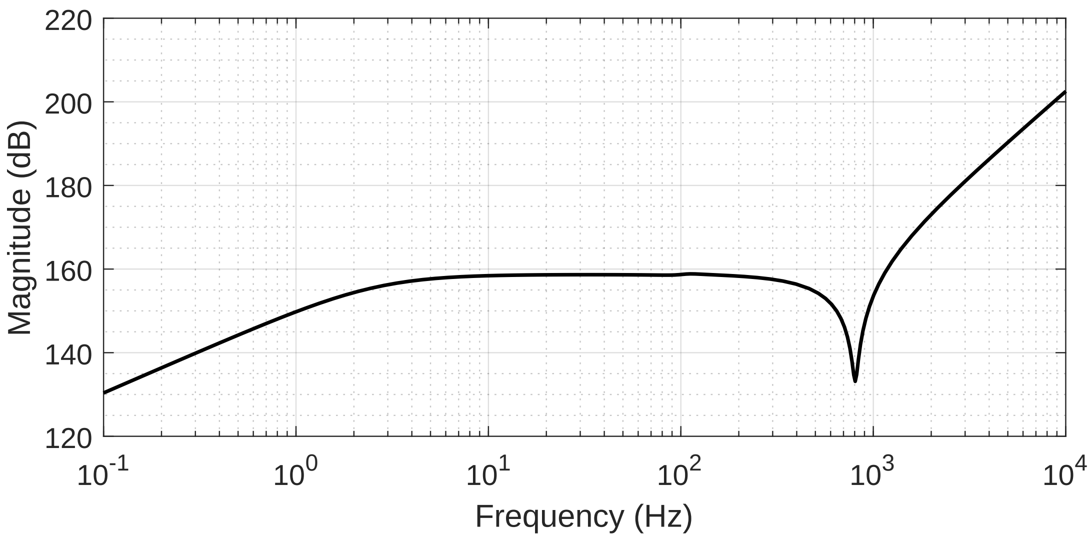

Figure 4.14: Frequency response of $(\frac{X(s)}{F_L(s)})^{-1}$

With the PID controller designed for the position loop on Table 3.4, linear model simulation and nonlinear simulations are conducted to observe the effects of non-linear terms. An external load of 5 kN is applied at EHA, in Simscape 'Frequency

Response' application is operated between an external load signal and actuator position for different non-linear configurations. Simulation configuration is demonstrated in Table 4.4.

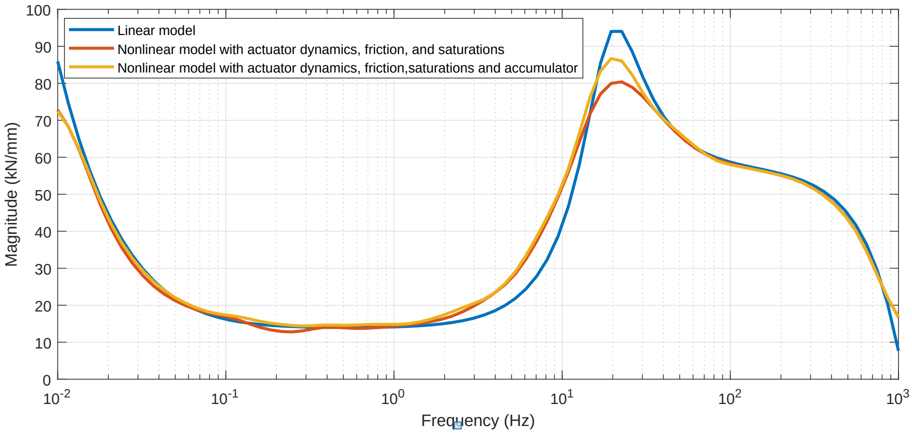

Figure 4.15: Closed loop simulation's frequency response of dynamic stiffness

In Figure 4.15, it is observed that for both linear and nonlinear models, the actual operating region of EHA, around 1 Hz, has low stiffness. The controller integrator reacts to force disturbances at low frequencies, hence increasing the stiffness of the actuator. Also as illustrated in Figure 4.14, due to hydro-mechanical structure of the actuator, it has already a stiffness increasing starting from 10 Hz. Hence, the controller should specially compensate low stiffness around 1 Hz. From a design perspective, notice that integration of the accumulator improves stiffness overall.

Table 4.4: Simulation configuration of frequency response test in Simulink

Parameter	**Value**
Frequency response parameters	
Amplitude	5000 N
Number of Periods	12
Setting Periods	4
Ramp Periods	0
Number of Samples at Each Period	40
Frequency Range	10^{-2} Hz to 10^{3} Hz
Solver Configuration	
Solver Type	Variable-step
Solver	ode23s (stiff/Modified Rosenbrock)
Maximum Step Size	Auto
Minimum Step Size	Auto
Initial Step Size	Auto
Relative Tolerance	1×10^{-3}
Absolute Tolerance	Auto
Zero-Crossing Control	Use Local Settings
Time Tolerance	10×128 eps
Number of Consecutive Zero Crossings	1000

4.2.3 Inner loop controllers

In order to observe noise rejection properties, a state space model including a disturbance can be formed from the equation 3.24 and torque controller $G_i(s)$.

$$P_{\frac{i}{V}}(s) = \frac{476.2s^3 + 1.256 \times 10^5 s^2 + 1.221 \times 10^{10} s + 7.159 \times 10^8}{s^4 + 477.9s^3 + 2.613 \times 10^7 s^2 + 5.61 \times 10^9 s + 1.097 \times 10^{13}} \tag{4.47}$$

$$G_i(s) = \frac{1}{11.3 + \frac{1.4 \times 10^4}{s}} \tag{4.48}$$

The sensitivity function of external disturbances such as measurement noise for the current is formed as:

$$S_i(s) = \frac{1}{1 + G_i(s)P_{\frac{i}{V}}(s)} = \frac{s\left(s^2 + 214.3s + 4.278 \times 10^5\right)}{(s + 3618)\,(s + 1977)\,(s + 0.05511)} \tag{4.49}$$

From the resulting bode diagram illustrated in Figure 4.16. It is observed that the current controller rejects noise until 1 kHz.

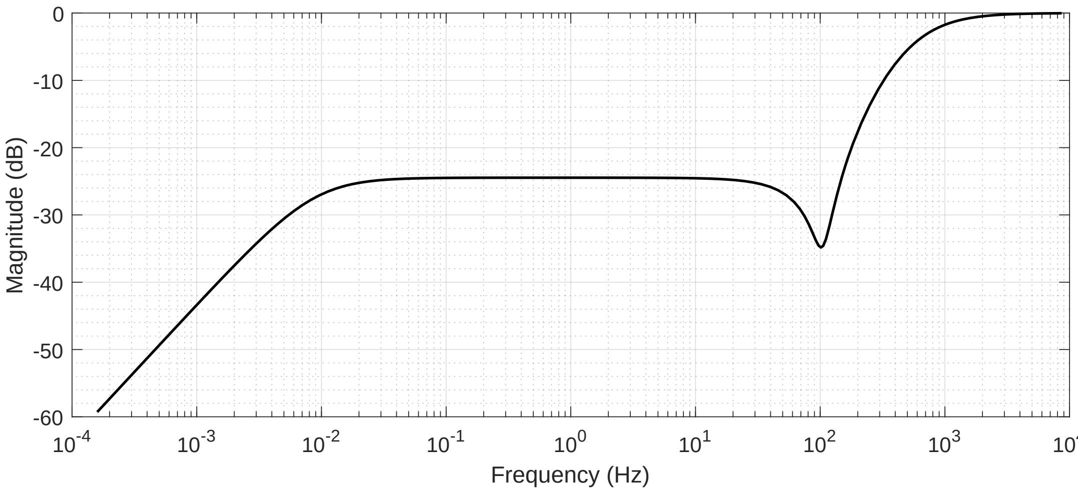

Figure 4.16: Sensitivity plot of current disturbance rejection

Equation 4.16 also informs about the bandwidth of the control loop, since the sum of complementary sensitivity and sensitivity functions is unity. From torque mode state space model in equation 3.35, the frequency response of the controller can be investigated. Hence transfer function between motor current and reference current $\frac{i(s)}{i_{ref}(s)}$ is derived from the state equation:

$$\frac{i(s)}{i_{\text{ref}}(s)} = \frac{5381(s + 1250)(s + 7.188)}{(s + 3818)(s + 1777)(s + 7.125)} \tag{4.50}$$

The frequency response of the resulting transfer function in equation 4.50 is plotted in Figure 4.17, where the bandwidth of the current controller is calculated around 1 kHz. Additionally, the torque controller state space model is validated since the sum of sensitivity in equation 4.49 and complementary sensitivity in equation 4.50 equals to 1. The disturbance rejection on the current controller demonstrates an advantage of the cascade control structure: each inner loop rejects relevant noise in its loop by operating at the higher bandwidth. This is also relevant for the EHA system, as a requirement position control bandwidth is on the magnitude of 10 Hz, while velocity control loop bandwidth is around 100 Hz and the current controller bandwidth is calculated as 1000 Hz.

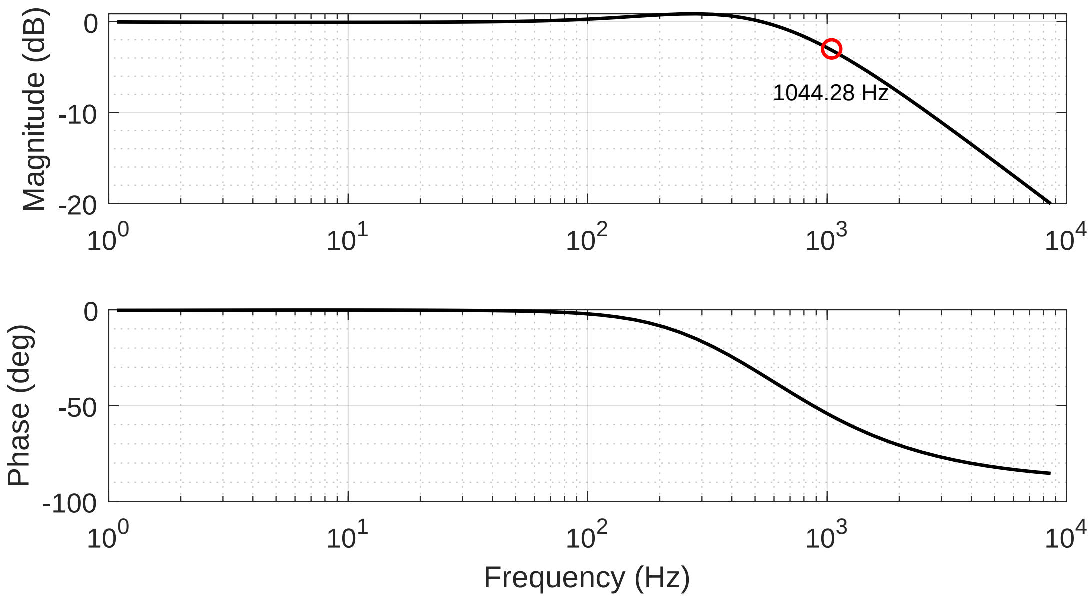

Figure 4.17: Bandwidth of the current controller

Hence inspecting the current control loop frequency response, it is concluded default controller setting of the driver has satisfying characteristics. Tuning PI structure of the torque controller can not change plant characteristics significantly, as illustrated in Figure 4.18. From equation 3.41, $P_\omega(s)$ is obtained which plant from motor velocity reference command and actuator position. Increasing the integral and proportional coefficients of the torque controller by double does not affect the plant $P_\omega(s)$ until high frequencies, since the velocity controller performs dominant characteristics at lower frequencies.

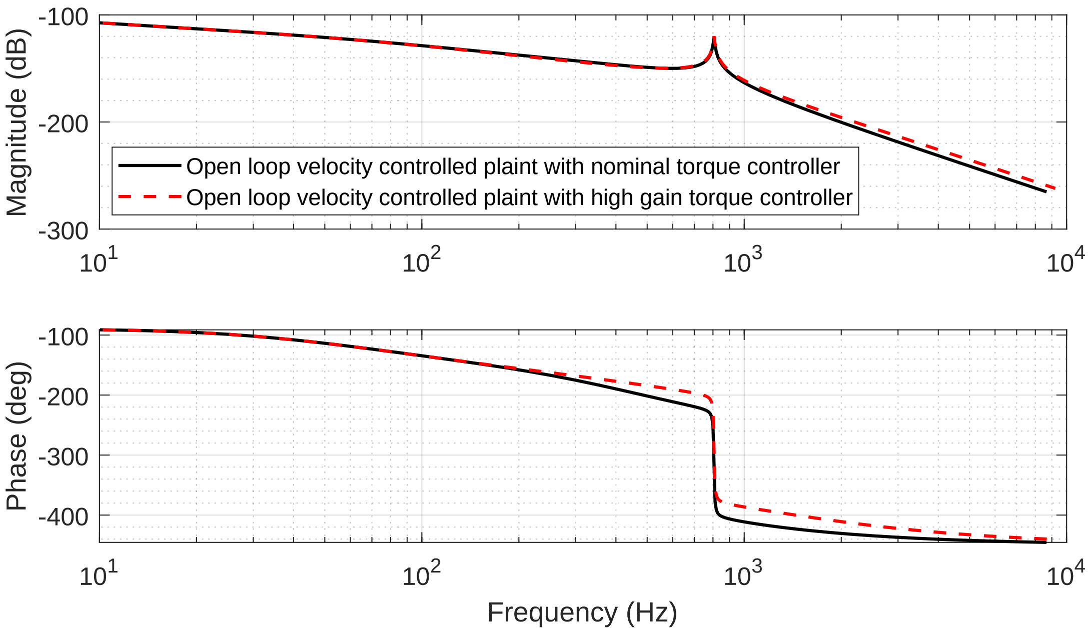

Figure 4.18: Open loop responses of velocity controlled plants

4.3 QFT controller design

Quantitative Feedback Theory (QFT) is a robust control engineering design methodology that allows for the systematic design of controllers considering plant uncertainties. In the QFT framework, relevant design specifications are integrated into uncertain plant models, where boundary conditions for the controller requirements are derived. This feedback-based approach compensates for the effects of plant uncertainty while ensuring that controller requirements are satisfied.

One of the key advantages of QFT is its ability to systematically and quantitatively determine the scope of robustness against uncertainties and faults in the plant. This makes the robustness numerically clear to the controller designer, unlike some other nonlinear robust control methods. Since the controller is designed in the frequency domain, typically using lead and lag compensators, the resulting controller is deterministic and predictable. This contrasts with other nonlinear control methods, such as sliding mode controllers and backstepping controllers, where it is challenging to evaluate robustness comprehensively or predict unexpected behaviors under certain fault conditions. While sliding mode and backstepping controllers demonstrate good tracking performance in the presence of nonlinearities like friction, their behavior under unexpected conditions, such as faults, is less predictable.

Another advantage of QFT is its similarity to PID controllers in the context of being frequency-based controllers for the flight control actuators. Flight control actuators are often treated as subsystems of a flight vehicle and are typically modeled as transfer functions or intervals of transfer functions. From a system-level perspective, actuator performance specifications are essential, and the frequency-domain approach of QFT provides a deterministic and evaluable framework for meeting these specifications. This makes QFT particularly well-suited for flight technologies, as noted in [67], and for the controller design of Electro-Hydrostatic Actuators (EHA), as discussed in [43]. In that regard, it is also possible design PID controllers with QFT methodology, suggesting a systematic approach to design controllers based on the parametric uncertainties of the plants.

For the design of EHA controllers, the cascade loop structure suggested on Figure 3.30 will be used to design velocity and position controllers respectively. For the system integration, model-based Kalman Filter for the position estimation will be used. In Figure 4.19,the schematic of control system is illustrated. Initially, a velocity controller $G_v(s)$ will be designed that is compatible with the PI controller structure of the driver system. Then position controller $G(s)$ will be designed to meet system specifications.

Prefilter in the schematic is also a design step of QFT methodology, equivalent to feed-forward compensation of the system. In Figure 4.20, steps through QFT design is shared.

The step 2 of selecting nominal plant is already achieved from the identification of EHA, as relevant parameters are shared in Table 3.4. From Figure 3.44, change of total leakage coefficient with recept to temperature is also recorded. As for other parameters, such as bulk modulus, motor constants may also vary with respect to environmental factors. These values are also considered uncertain parameters in the applications of the literature, in the context of hydraulic systems and electric motors. The effective piston area, dead volume, motor inertia and piston mass are more deterministic and certain parameters that are unlikely to significantly change with respect to environmental changes. Further steps includes determining of performance

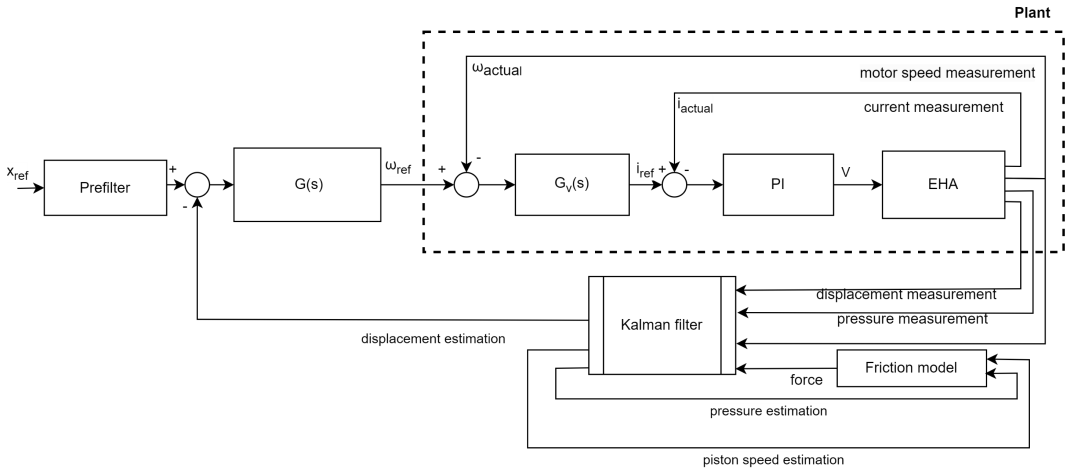

Figure 4.19: Cascade controller structure for EHA

specifications, and converting such speciations in polar form. The specifications in polar form allows to calculate maximum and minimum boundary requirements with respect to performance specifications, as well as allowing loop shaping. During loop shaping, controller is generated with a systematic trial-and-error, or in this work, via optimization. After the feedback controller design, pre-filter is designed to meet reference tracking requirement, afterwards controller can be validated trough frequency analysis, linear time domain analysis and nonlinear simulation process. Finally, an experimental procedure is followed to validate the performance of the controller. Since identification and model of EHA is derived, controller design process is accurate during design process, with less requirement of tuning during the experimental validation process.

4.3.1 Parametric uncertainties

For the plant model in equations 3.35 and 3.41, parametric uncertainties are introduced as given in Table 4.5. Bulk modulus and leakage coefficients are chosen to consider environmental changes such as temperature and hydraulic oil quality. Uncertainty in pump displacement is specified to reflect variations in pump efficiency,

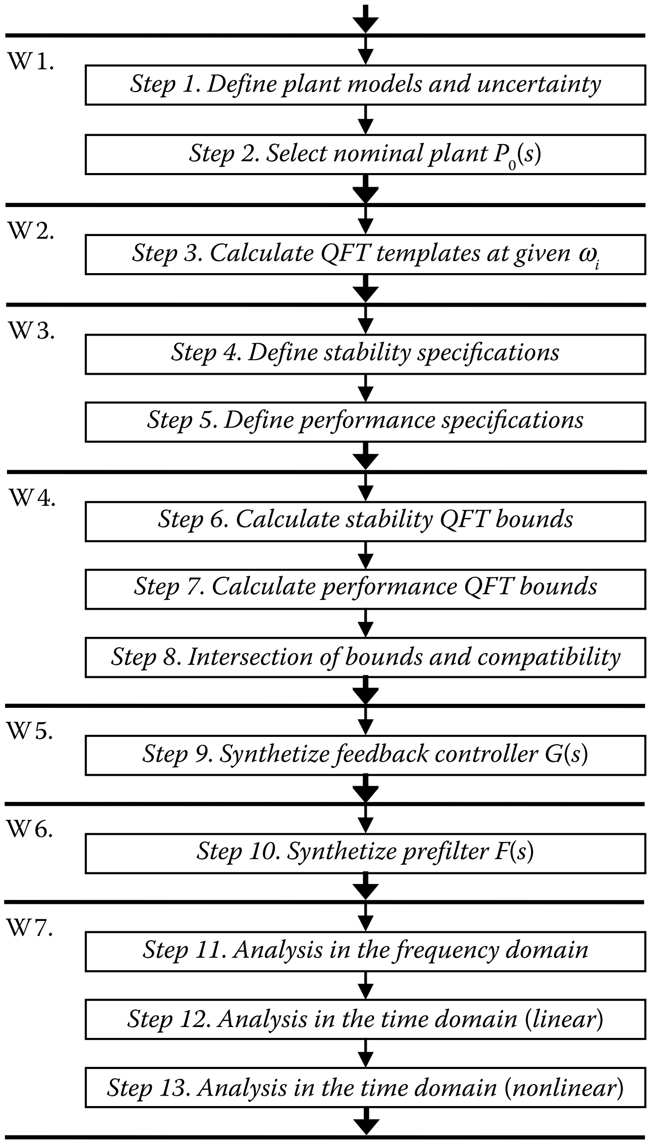

Figure 4.20: Steps of QFT design [47]

including dead zone and backlash effects, as well as physical displacement variation due to wear and potential fault. Motor constants are selected on an interval since they are linear approximations for equations which have actually nonlinear behavior. The uncertainty intervals for the hydraulic actuator system are similar to the values in the literature [42].

Logarithmic intervals are used for assigning values between minimum and maximum values of uncertain parameters. A total of 270 plant transfer function combinations

Table 4.5: Nominal values and parameter ranges

Parameter	Nominal value	Minimum value	Maximum value
C_{eq}	$3.8 \times 10^{-12}\,\text{m}^3/\text{Pa}$	$10^{-1}C_{\text{eq}}$	$10^{1}C_{\text{eq}}$
D_p	$6.684 \times 10^{-7}\,\text{m}^3/\text{rad}$	$0.8D_p$	D_p
β	1.379×10^{7} Pa	0.7β	1.4β
k_T	0.73 Nm/A	$0.8k_T$	$1.2k_T$
k_E	0.53 Nm/(rad/s)	$0.8k_E$	$1.2k_E$

can be obtained. Frequency response of uncertain plant models derived for the torque mode plant is illustrated in Figure 4.21, where plant family of $P_i(s)$ is plotted. The matlab code for the plant generation is demonstrated in Appendix D.

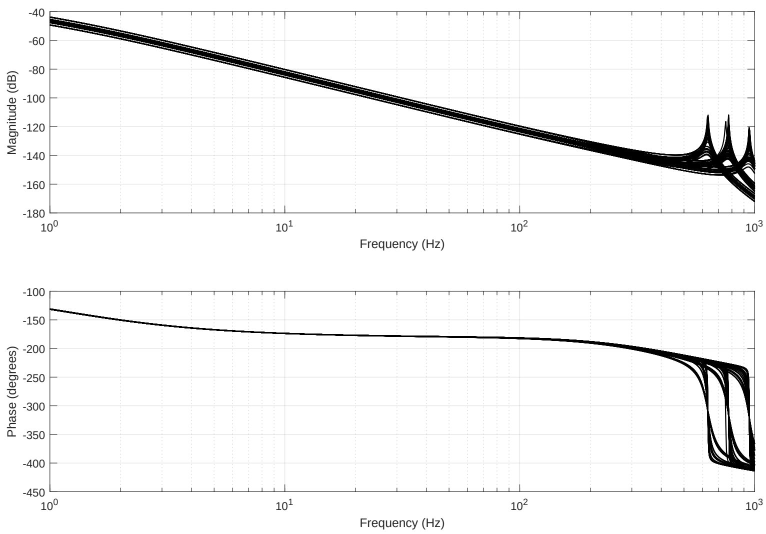

Figure 4.21: Open loop response of uncertain torque controlled EHA plant

For the step 3, discrete QFT templates at specific frequencies will be calculated for the uncertain plant families. Discrete frequency arrays are selected for critical frequencies for the specifications, as well as high and low frequencies for the evaluation of stability and overall performance of the controller. Hence, performance specifications of the velocity loop and position loop can be determined in order to achieve desired actuator performance.

4.3.2 Performance specifications

For the position controller $G(s)$, specifications for stability, disturbance rejection, and reference tracking are defined. For the velocity controller $G_v(s)$, specifications for stability and disturbance rejection are defined. A frequency array of $\{0.01, 0.1, 20, 200, 300, 500\}$ rad/s is selected for the velocity controller in all specifications. The frequency bandwidth of flight control actuators is generally up to 15 Hz [13]. Hence, for the position controller, the frequency interval for tracking, stiffness, and sensitivity is designed based on the intervals defined in the literature [42].

First of all, close-loop robust stability should be satisfied with an instability region defined by the M-locus in the Nichols chart. Hence, the stability specification for the controllers can be shown as:

$$|T_1(s)| = \left|\frac{P_\omega(s)G(s)}{P_\omega(s)G(s)+1}\right| \leq W_s = 1.305, \tag{4.51}$$

$$|T_1^v(s)| = \left|\frac{P_i(s)G_v(s)}{P_i(s)G_v(s)+1}\right| \leq W_s^v = 1.16. \tag{4.52}$$

Frequencies of $\{0.001, 0.01, 0.1, 1, 10, 20, 50, 100, 500\}$ rad/s are selected for stability criteria for the position controller. $T_1(s)$ is transfer function used to assess stability for position controller loop, while $T_1^v(s)$ is defined for the velocity controller. Here, W_s and W_s^v are the constant magnitudes in the Nichols chart of the closed-loop transfer functions, enclosing the corresponding gain and phase margins. Traditionally, Gain Margin (GM) and Phase Margin (PM) are commonly used to measure the stability of a closed loop system. A different but similar method measuring stability is M-locus circles, representing locus of the constant magnitude of the closed loop transfer function. Since the circle encapsulates instability point (0 dB, -180 phase), the circle is related with stability margins. In Figure 4.22, resulting stability circles for the requested closed loop gains are illustrated. Notice that gain and phase margins are shown with dashed lines.

For the stability constant defined, the phase margin and gain margin are defined as follows:

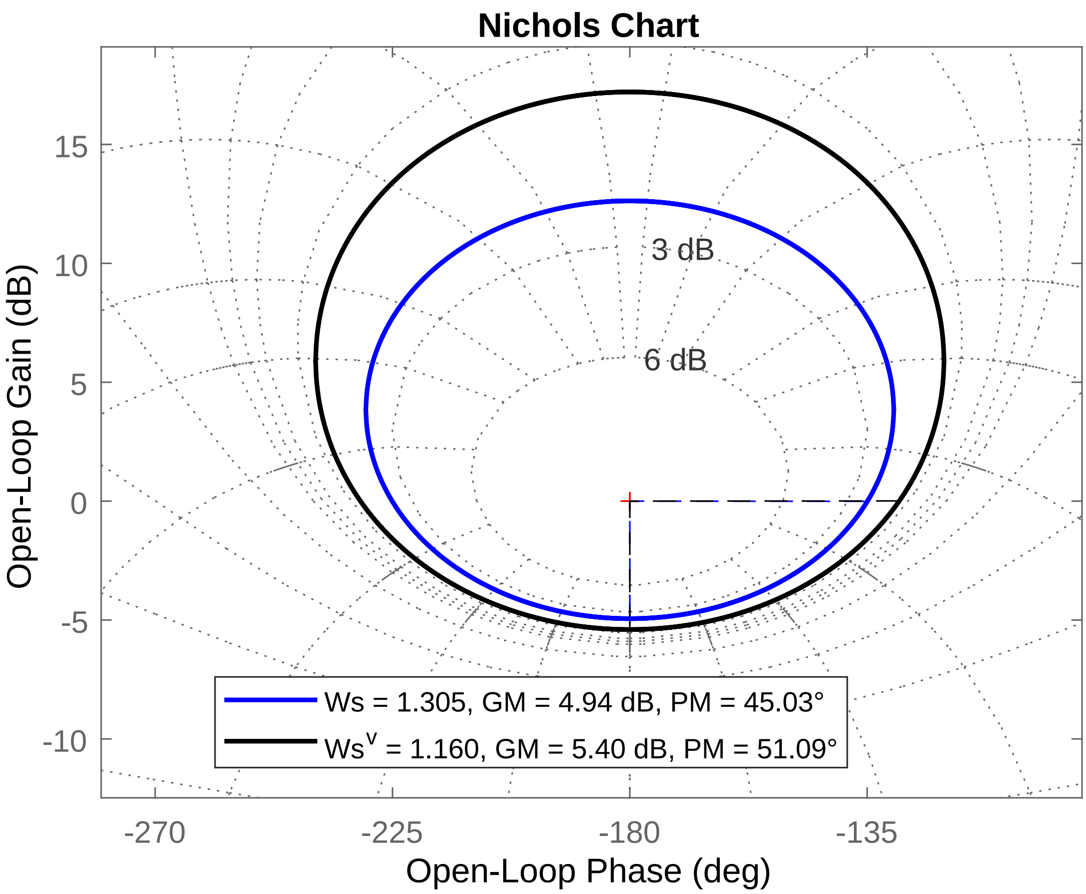

Figure 4.22: Stability margins defined for position and velocity controllers

$$\text{PM} = 180^\circ - 2\left(\frac{180}{\pi}\right)\arccos\left(\frac{0.5}{W_s}\right), \quad \text{in degrees} \tag{4.53}$$

$$\text{GM} = 20\log_{10}\left(1+\frac{1}{W_s}\right), \quad \text{in dB} \tag{4.54}$$

The plotting and calculating of QFT bounds are very similar to each other. The matlab code for the plotting of stability circles is shared on Appendix D. Psedo-code of the matlab function is demonsrated in Table 4.6. A trick here is to solve inequalities of 4.51 and 4.52 in polar forms of the transfer function. The plant transfer function $P(j\omega_i)$ can be written as:

$$P(j\omega_i) = pe^{j\theta} = p\angle\theta \tag{4.55}$$

where:

- p is the magnitude of the plant transfer function at the frequency ω_i.

- θ is the phase angle of the plant at the frequency ω_i.

Similarly, g and ϕ are defined for the magnitude and angle of the controller. Hence, the general form of the equations in 4.51 and 4.52 are derived in equation 4.56.

$$p^2\left(1-\frac{1}{W_s^2}\right)g^2+2p\cos(\phi+\theta)g+1\geq 0 \tag{4.56}$$

Table 4.6: Algorithm to plot Nichols circles

Algorithm 1 Plot Nichols Circles

Require: Desired closed-loop gain W_s

Express plant and controller in polar form:

$$P(j\omega)=p\angle\theta,\quad G(j\omega)=g\angle\phi$$

Define a phase array $\phi\in[-360°,0°]$ for iteration.

for each ϕ in the phase array **do**

    Solve the quadratic equation for two roots g_1 and g_2:

$$p^2\left(1-\frac{1}{W_s^2}\right)g^2+2p\cos(\phi+\theta)g+1=0$$

    Store results for g_1 and g_2 in arrays.

end for

Plot the Nichols circle using:

$$\text{Magnitude}=20\cdot\log_{10}(g_1),\quad 20\cdot\log_{10}(g_2)$$

Calculate Gain Margin (GM):

$$\text{GM}=20\cdot\log_{10}\left(1+\frac{1}{W_s}\right),\quad\text{in dB}$$

Calculate Phase Margin (PM):

$$\text{PM}=180°-2\left(\frac{180}{\pi}\right)\arccos\left(\frac{0.5}{W_s}\right),\quad\text{in degrees}$$

The second objective is to attenuate close-loop disturbances. For rejecting disturbances at plant output, the following sensitivity constraints are used:

$$|T_2(s)| = \left| \frac{1}{P_\omega(s)G(s)+1} \right| \leq S(s), \quad S(s) = \frac{\frac{s}{30}}{\frac{s}{30}+1} \tag{4.57}$$

$$|T_2^v(s)| = \left| \frac{1}{P_i(s)G_v(s)+1} \right| \leq S^v(s), \quad S^v(s) = \frac{\frac{s}{600}}{\frac{s}{600}} \tag{4.58}$$

For the velocity controller, a sensitivity constraint is selected such that cascade loop rejects disturbances up to 100 Hz, whereas for the position controller, sensitivity constraint is selected to respect reference tracking constraints. Frequencies of $\{0.1, 1, 10, 20, 50, 100\}$ rad/s are selected for sensitivity criteria for the position controller. In Figure 4.23, frequency response of the requirement in equations are plotted. Notice that a similar Nichols Chart as in Figure 1 can also be plotted, which also be generated during QFT bounds step. In this step, as in stability margin example, polar forms of the equations 4.57 and 4.58 will be derived. A more comprehensive explanation of the topic is explained in [47].

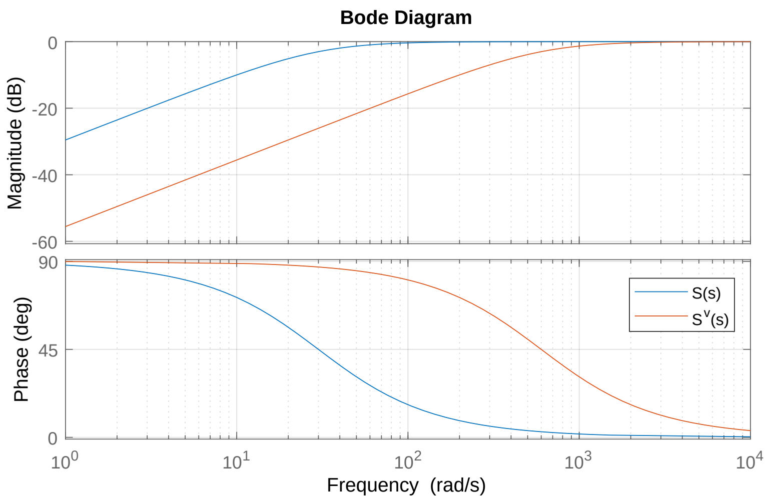

Figure 4.23: Sensitivity or disturbances at plant output specification frequency plot

Additionally, a a requirement for the the actuator's dynamic stiffness may be integrated such that stiffness should be above 49 kN/mm for $\omega \leq 7$ Hz. The stiffness magnitude is selected based on [42] and frequency intervals are selected parallel with the operating frequency range as well as frequency ranges in literature [68]. Closed-loop compliance restriction can be given as:

$$|T_3(s)| = \left| \frac{P_d(s)}{P_\omega(s)G(s)+1} \right| \leq M(s) \tag{4.59}$$

Notice P^d is the transfer function between external force and actuator displacement, which was derived from equation 3.41. $M(s)^{-1}$ is defined as the lower bound on actuator stiffness. For the performance specification, $M(s)$ is chosen in a similar structure as in [42] by trial and error as:

$$M(s) = 2 \times 10^{-6} \frac{(s+600)^2}{(s+6000)^2}. \tag{4.60}$$

Frequencies of $\{1, 10, 30, 50\}$ rad/s are selected for dynamic stiffness criteria in position controller. The frequency response of the constraint is illustrated in Figure 4.24.

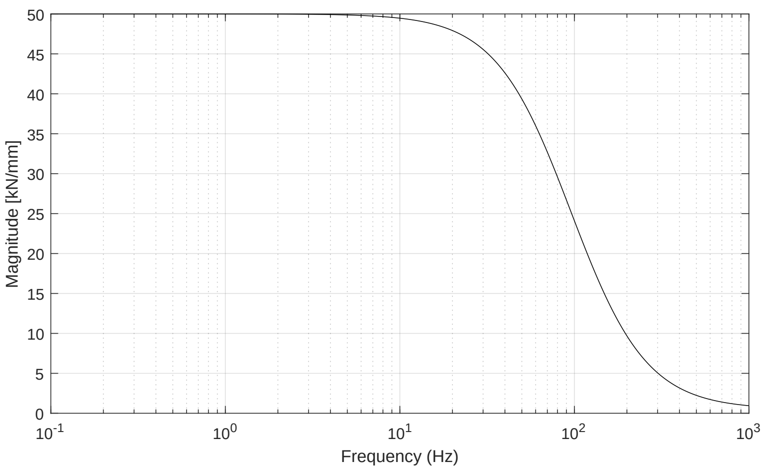

Figure 4.24: Frequency response of the stiffness lower bound $M(s)^{-1}$

Reference tracking specification is determined as:

$$T_{\text{low}}(s) \leq |T_4(s)| = \left| F(s)\frac{P_\omega(s)G(s)}{P_\omega(s)G(s)+1} \right| \leq T_{\text{up}}(s) \tag{4.61}$$

Where upper and lower bounds are defined as:

$$T_{\text{up}}(s) = \frac{\frac{s}{a_{\text{up}}} + 1}{\frac{s}{\omega_n^2} + \frac{2\zeta}{\omega_n}s + 1}, \quad a_{\text{up}} = 30\,\text{rad/s}, \quad \zeta = 0.8, \quad \omega_n = \frac{1.25a_{\text{up}}}{\zeta} \tag{4.62}$$

$$T_{\text{low}}(s) = \frac{1}{\left(1 + \frac{s}{85}\right)^2} \tag{4.63}$$

Frequencies of $0.01, 0.1, 1, 10, 30, 50$ rad/s are selected for reference tracking criterion in position loop. The lower and upper bounds in equations 4.63 and 4.62 determine performance requirement of frequency response under an interval. Visual representation of such interval is highlighted in Figure 4.25.

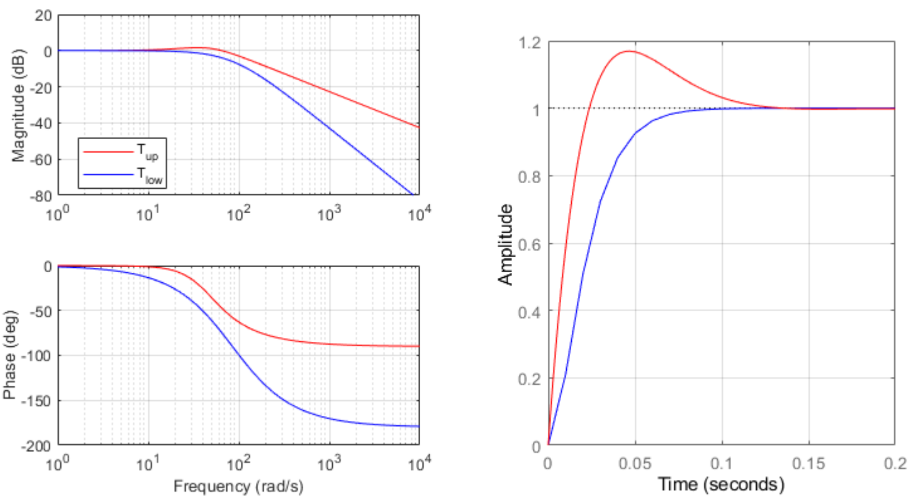

Figure 4.25: Frequency intervals defined for the position controller

Inequalities 4.51,4.52,4.57,4.58, 4.59, 4.62, and 4.63 are constraints on the open-loop transfer function $L(s) = G(s)P(s)$, where nominal transfer functions for the plants are utilized. These constraints refer to dynamic stiffness, tracking performance, output disturbance rejection, and robust stability margin. The array of frequencies for each constraint is merged, and bounds for QFT design are computed with respect to critical frequencies selected. Constraint bounds are merged according to selected frequencies and limiting bounds. Constraints can be solved for $G_v(s)$ and $G(s)$, such that for each uncertain plant, the worst-case bound should be satisfied. Hence, next step, QFT bound generation can be followed before controller design process. Since P_ω depends on velocity controller values, velocity controller will be designed initially.

4.3.3 Velocity controller synthesis

For the design of velocity controller, there are two constraints, 4.52 and 4.58, defined on the uncertain plants. For the stability margin, polar conversion was already explained. Sensitivity polar form for the equation 4.58 is:

$$p^2g^2 + 2p\cos(\phi+\theta)g + \left(1 - \frac{1}{\delta_2^2}\right) \geq 0 \tag{4.64}$$

δ_2 is the magnitude of the specification $T_2^v(s)$ at a particular frequency. Same is valid for the position controller polar form. From polar forms of stability and sensitivity, quadratic insulates will be solved into Nichols chart for the each frequency of interest, taking into account model uncertainty. Then, controller synthesis will take place in order to loop shape. Determining controller structure that is industrially applicable to driver settings, a Particle Swarm Optimization (PSO) algorithm will be conducted to automatic loop shape.

4.3.3.1 QFT bounds of velocity control

QFT bounds are determined according to performance specifications. Once the bounds are visualized on the Nichols chart, the controller design process focuses solely on the nominal plant $P_0(s)$. This is a significant advantage of the QFT methodology. Rather than addressing an infinite number of possible plants, the design step is simplified to consider only the nominal plant $P_0(s)$, as the effects of model uncertainty are already accounted for within the QFT bounds. Matlab code for QFT bound creation in velocity controller is shared in Appendix D. The underlying algorithm for the bounds is demonstrated in 4.7.

Hence, there are two QFT bounds generation processes for stability and sensitivity. The respective bounds for stability and sensitivity are illustrated in Figure 4.26 and Figure 4.27.

Figure 4.26 represents the stability bounds, which ensure that the open-loop transfer

Table 4.7: Algorithm to compute the QFT bounds

Algorithm 2 Compute QFT Bounds

1: Discretize the frequency domain ω into a finite set $\Omega_k = \{\omega_i, i = 1, \ldots, n\}_k$.

2: Define the uncertain plant models $\{P(j\omega)\}$ and map their boundaries for each $\omega_i \in \Omega_k$ on the Nichols chart.

3: Represent the n templates $\{P(j\omega_i)\}$, where $P(j\omega_i) = \{P_r(j\omega_i) = p\angle\theta, r = 0, \ldots, m-1\}$.

4: Choose the nominal plant $P_0(j\omega) = p_0\angle\theta_0$.

5: Define the compensator $G(j\omega) = g\angle\phi$ and discretize $\phi \in \Phi = [-360° : 5° : 0°]$.

6: **for** each frequency $\omega_i \in \Omega_k$ **do**

7:     **for** each compensator phase $\phi \in \Phi$ **do**

8:         **for** each plant $P_r(j\omega_i), r = 0, \ldots, m-1$ **do**

9:             Compute the maximum $g_{\max} = g_{\max}(P_r)$ and the minimum $g_{\min} = g_{\min}(P_r)$ that solve the quadratic inequality for roots g_1 and g_2.

10:         **end for**

11:         Choose the most restrictive $g_{\max}(P)$ and $g_{\min}(P)$ among all plants.

12:     **end for**

13: **end for**

14: Compute $g_{\max}\angle\phi_1$ and $g_{\min}\angle\phi_2$ over $\phi \in \Phi$ for each frequency ω.

15: Represent the open-loop transmission as $L_0(j\omega) = l_0\angle\psi_0$, with $l_{0max} = p_0 g_{\max}\angle\phi$ and $l_{0min} = p_0 g_{\min}\angle\phi$.

16: Note that $\psi_0 = \phi + \theta_0$ and $\phi = [-360° : 5° : 0°]$. Hence bounds are represented as $\{B_k(j\omega), \forall\omega_i \in \Omega_k\}$.

function's gain and phase remain within specified regions to maintain system stability. Dashed lines in the Nichols chart indicate areas that the gain and phase values must avoid. These bounds are critical for ensuring that the controller prevents instability, particularly for systems with higher-order dynamics or significant uncertainties.

Figure 4.27 illustrates the sensitivity bounds, which restrict the open-loop gain and

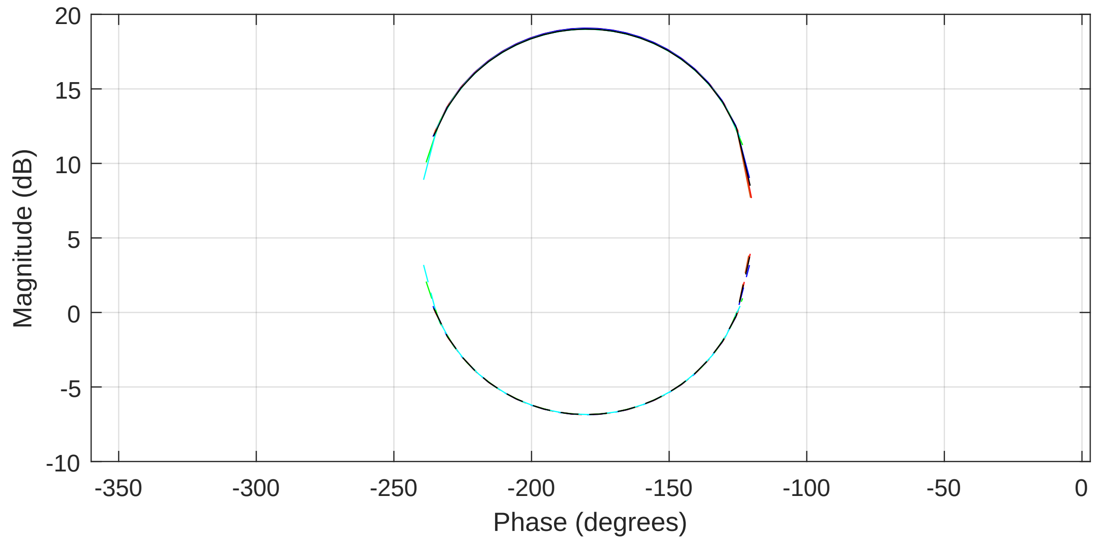

Figure 4.26: Stability bounds for the velocity controller

phase to ensure acceptable disturbance rejection and tracking performance. The solid lines in the Nichols chart represent these sensitivity bounds, emphasizing the regions where the controller must operate to achieve the desired performance criteria. Sensitivity bounds ensure that the system can reject disturbances and maintain robustness against model uncertainties.

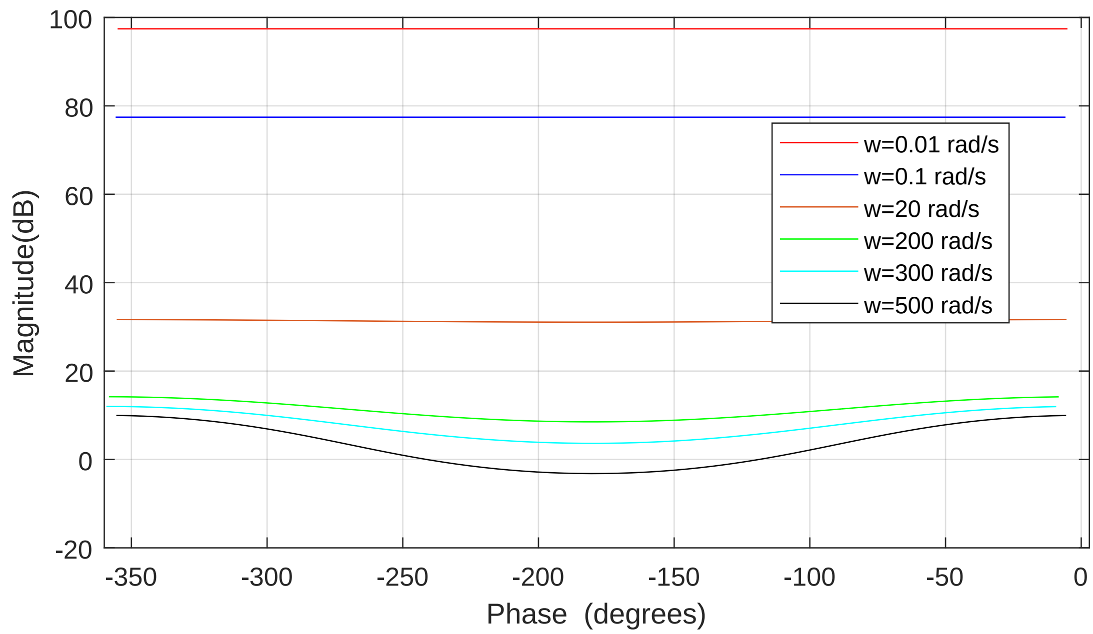

Figure 4.27: Sensitivity bounds for the velocity controller

The resulting QFT bounds that combine both stability and sensitivity constraints are presented in Figure 4.28. These bounds highlight the areas where the open-loop transfer function's gain and phase values must lie to satisfy both stability and sensitivity requirements simultaneously. The bounds are created by merging the most restrictive regions from Figures 4.26 and 4.27, ensuring that the controller satisfies all necessary design criteria.

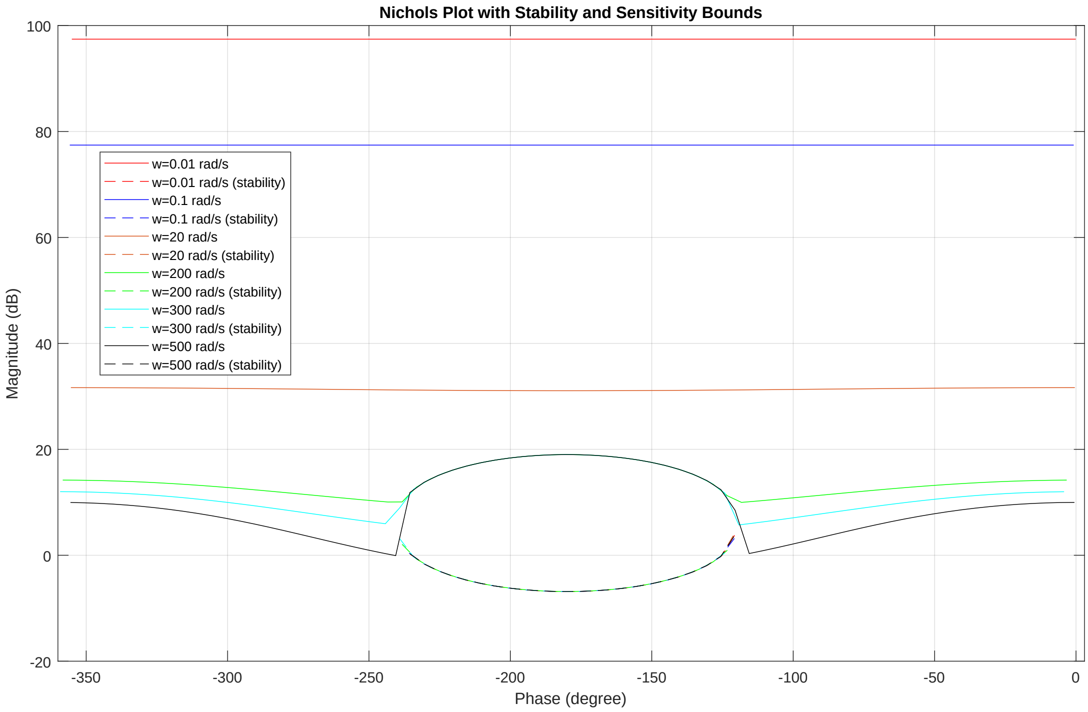

Figure 4.28: QFT bounds for the velocity controller

It is important to note that discrete frequencies are selected during the performance specification step to ensure computational efficiency and clarity in design. The bounds for specific restrictions are merged based on the higher gain values. For the stability case, the dashed lines represent areas where the open-loop transfer function gain and phase values should lie outside the circle. This ensures that the system operates above the solid lines and below the dashed lines to satisfy both stability and robustness criteria effectively.

4.3.3.2 PSO on controller design

PSO is a highly efficient method due to its fast convergence, hence becoming a popular optimization tool [69]. PSO aims to discover a global minimum with respect to a cost function for an optimization problem. A particle represents a point with parameters of interest which are dimensions, with a cost value along given coordinates. Hence in a multidimensional space, a swarm of particles search for the lowest cost with the given information of local and global best positions so far.

Firstly, the motor speed controller $G_v(s)$ is designed automatically, then with the new forming plant, the actuator position controller $G(s)$ is designed. Controller $G_v(s)$ is chosen as a PI controller with a low-pass filter which is applicable to industrial drivers:

$$G_v(s) = \frac{1}{k_T}\left(K_1 + \frac{K_2}{s}\right)\frac{1}{\frac{s}{K_3}+1} \tag{4.65}$$

A 3-dimensional PSO is conducted to determine poles and zeros of the proposed controller structure. Setup related to PSO is given in Table 4.8, that is relevant for velocity and position controller optimization.

Table 4.8: Optimization algorithm constants

Parameter	$G_v(s)$	$G(s)$
Number of particles	49	49
Number of iterations	50	100
Inertia weight (W)	0.9	0.9
C_1 and C_2	2	2

The position update equation can be illustrated as:

$$X_{i,k+1} = X_{i,k} + V_{i,k} \tag{4.66}$$

Where X, V are the position and velocity of the ith particle in the kth iteration. The velocity update equation can be shown as:

$$V_{i,k+1} = WV_{i,k} + C_1 r_1 \left(X_{g,k}^{\text{best}} - X_{i,k} \right) + C_2 r_2 \left(X_{l,k}^{\text{best}} - X_{i,k} \right) \tag{4.67}$$

where r_1 and r_2 are random values between 0 and 1, $X_{g,k}^{\text{best}}$ is the best global position for the particle, and $X_{l,k}^{\text{best}}$ is the best position in the swarm for the current iteration. In Figure 4.29, a diagram for the algorithm implementation is given. The underlying cost function for velocity controller and PSO algorithm is shared in Appendix D.

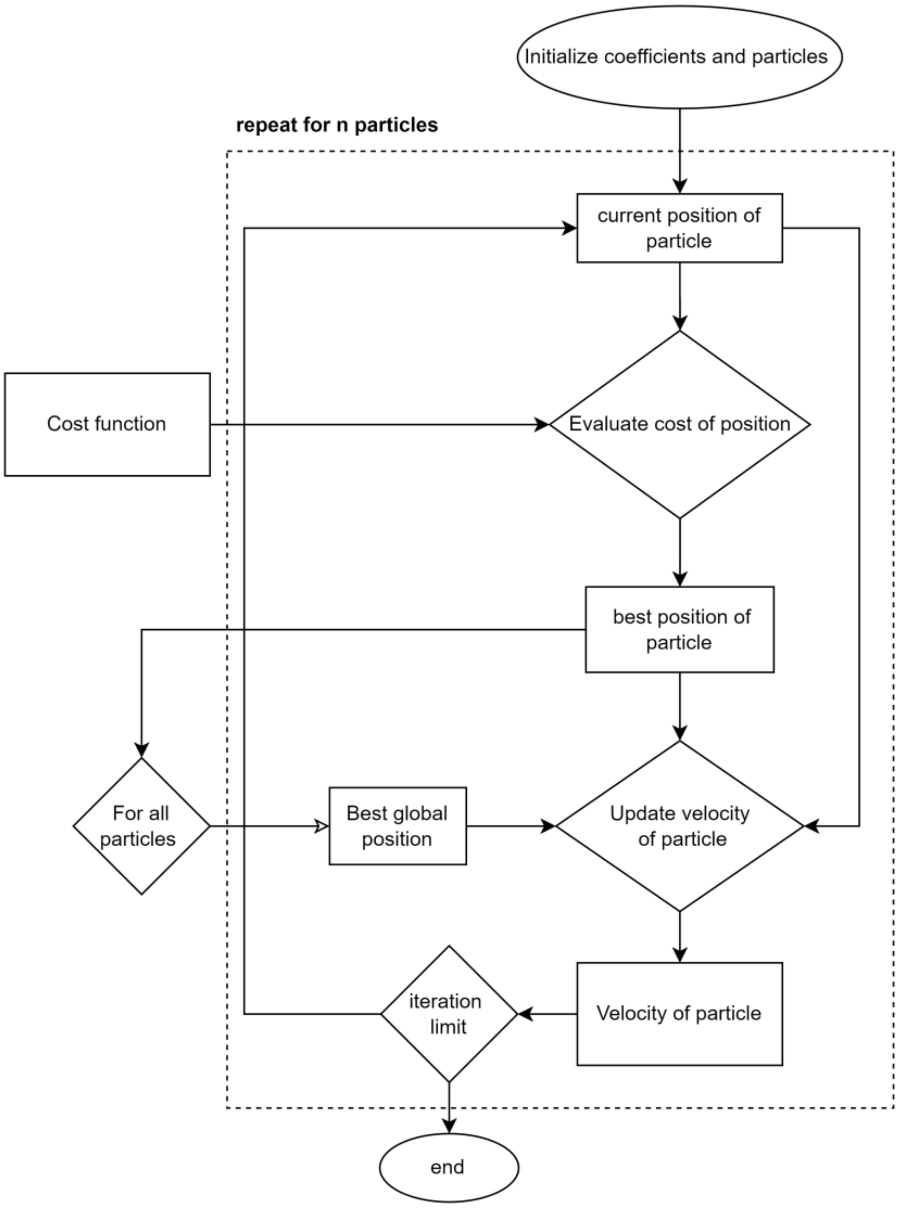

Figure 4.29: Algorithm schematic for PSO algorithm

A cost function is generated to determine the best particle with the corresponding controller coefficients in the form 4.65. Respecting performance specifications and uncertain plant conditions, boundaries for maximum and minimum gains for the $L_v(s) = G_v(s)P_{v0}(s)$ are generated. For the frequencies selected previously, if the gain and phase of L_v at the particular frequency do not satisfy the boundary condition, the cost is updated. Hence, a max function is used for the cost:

$$J_v = \sum_{i=1}^{h} \max\left(0, g_{\text{max}}(j\omega_i) - |L_v(j\omega_i)|\right) \tag{4.68}$$

Where $g_{\text{max}}(j\omega_i)$ refers to the gain boundaries at a particular phase of the $L_v(j\omega_i)$, and h is 6. To prevent high gains in high frequencies, when the gain and phase of L_v at the particular frequency satisfy the boundary condition, the cost at the particular frequency is updated according to an additional cost function:

$$J_v^{\text{high}} = \sum_{i=4}^{h} \max\left(0, |L_v(j\omega_i)| - g_{\text{max}}(j\omega_i) - T_g\right), \quad T_g = 2 \tag{4.69}$$

Where T_g is the gain tolerance constant to allow exceeding boundaries. Stability criteria are implemented with a high-cost weight if stability margins are violated, such that J_v is irrelevant if J_s in Equation (4.70) is not 0:

$$J_s = \max\left(\max\left(\left|\frac{L_v(j\omega)}{L_v(j\omega)+1}\right|, \omega \in \mathbb{R}^+\right) - W_s, 0\right) \tag{4.70}$$

Notice that for the optimization convergence criteria, it is not required; however, convergence can be observed by inspecting the position change of particles. In Figure 4.30, minimum cost explored through iterations is plotted.

In Figure 4.31, moving averages for the controller coefficients are demonstrated.

In the figures, one iteration contains an array of n particles, hence moving averages are taken to observe convergence. The obtained controller is:

$$G_v(s) = \frac{1}{k_T}\left(0.45 + \frac{56}{s}\right)\frac{1}{\frac{s}{2555}+1} \tag{4.71}$$

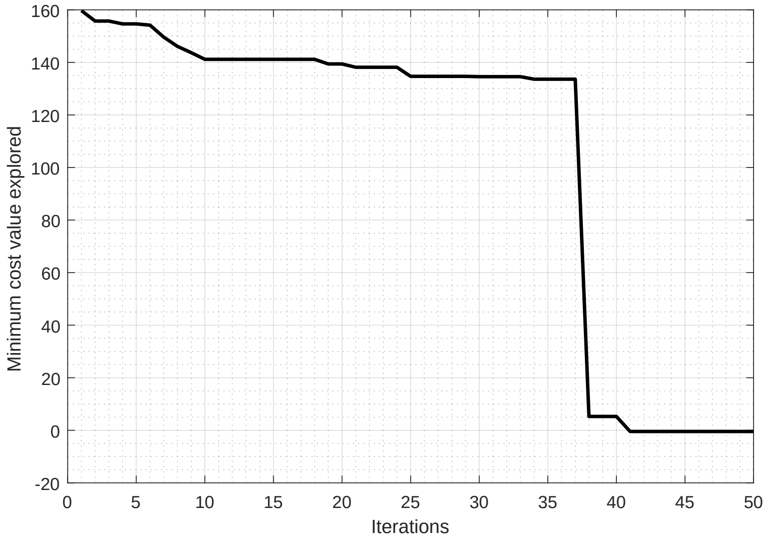

Figure 4.30: Minimum global cost value derived through iterations

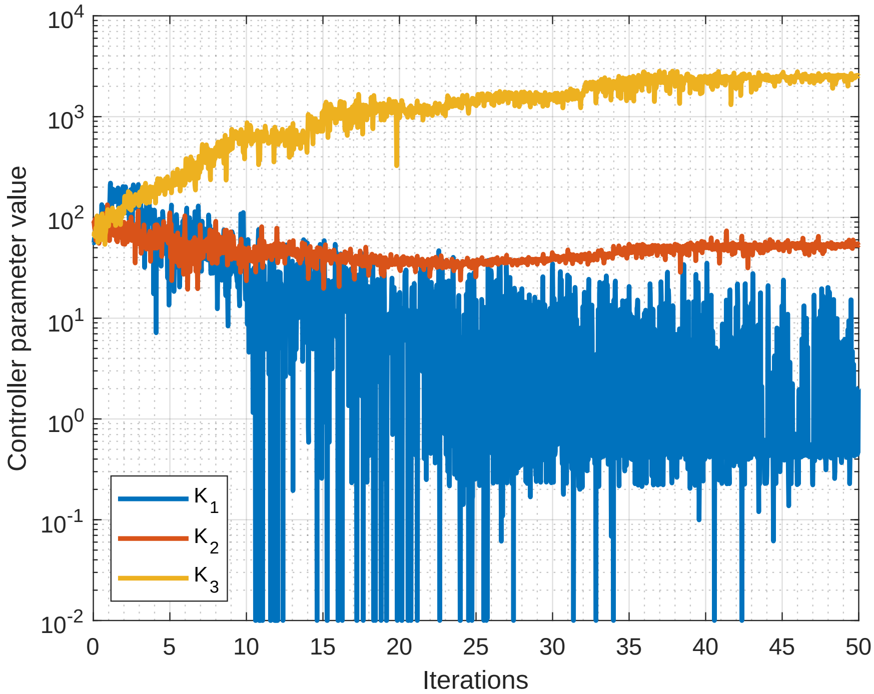

Figure 4.31: Controller coefficients through iterations

In Figure 4.32, the implemented controller with satisfied performance specifications is given. Notice that for each frequency of interest shown by circles on the transfer function $L_v(s)$, they are above the sensitivity bounds defined for the uncertain plants. The impact of the cost function in Equation (4.69) is displayed where high-frequency gains satisfy constraints, however, without exceeding them within a margin. For stability, notice that the open-loop plant does not interfere with the stability margin circle defined by W_s, which corresponds to a 63° phase margin approximately.

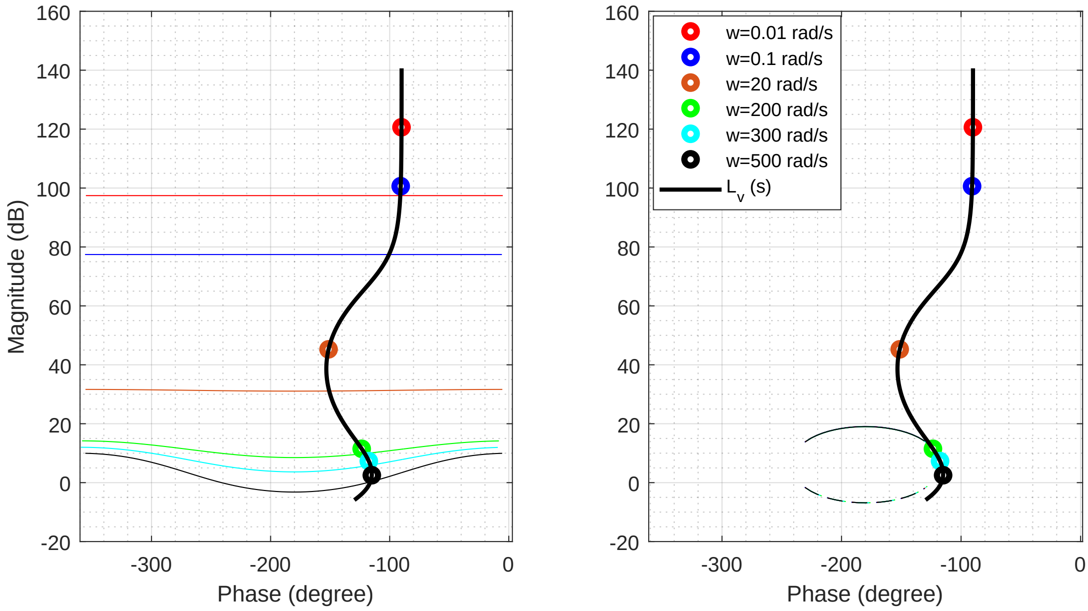

Figure 4.32: Resulting open loop response with designed controller on design constraints

A 3D representation of the convergence of controller coefficients, as illustrated in Figure 4.31, is demonstrated in Figure 4.33. The particle positions are plotted in the parameter space (K_1, K_2, K_3) for each iteration. The color gradient, transitioning from blue (iteration 1) to red (iteration 50), visually represents the progression of the optimization process. Early iterations show a wide spread of particles, indicative of exploration, whereas later iterations demonstrate a focused clustering of particles as they converge toward optimal solutions. This plot provides insight into the behavior of the particle swarm and the dynamic adjustment of controller coefficients over the optimization process.

In Figure 4.34, the resulting cost values corresponding to the particle locations in parameter space (K_1, K_2, K_3) are represented using a logarithmic color scale. This visualization highlights regions of the parameter space associated with higher or lower cost values.

A summary of this section is generation of performance specifications boundaries regarding parametric uncertainties and then merging them. PSO framework is utilized

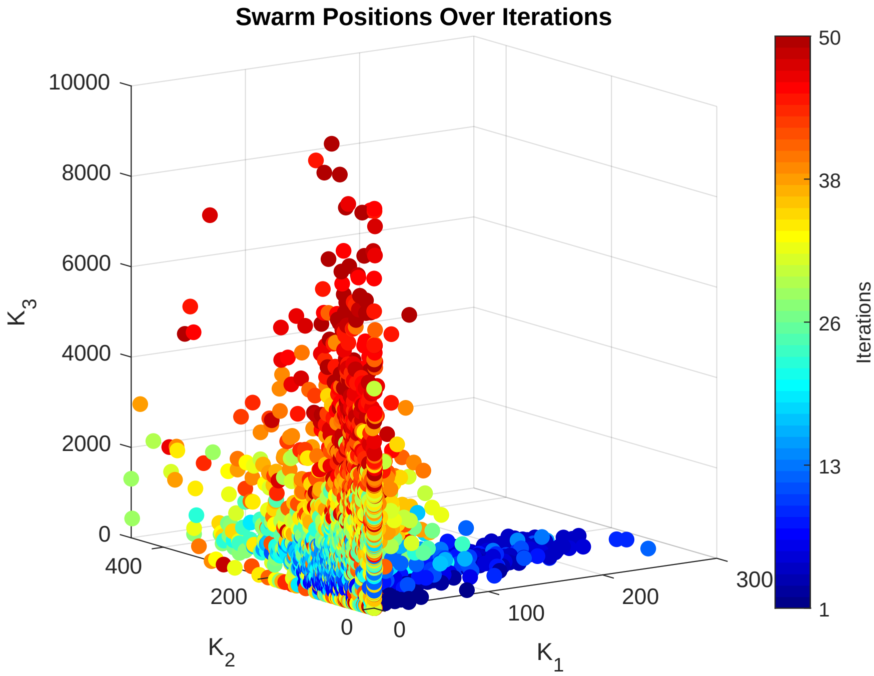

Figure 4.33: Particle locations with respect to iterations

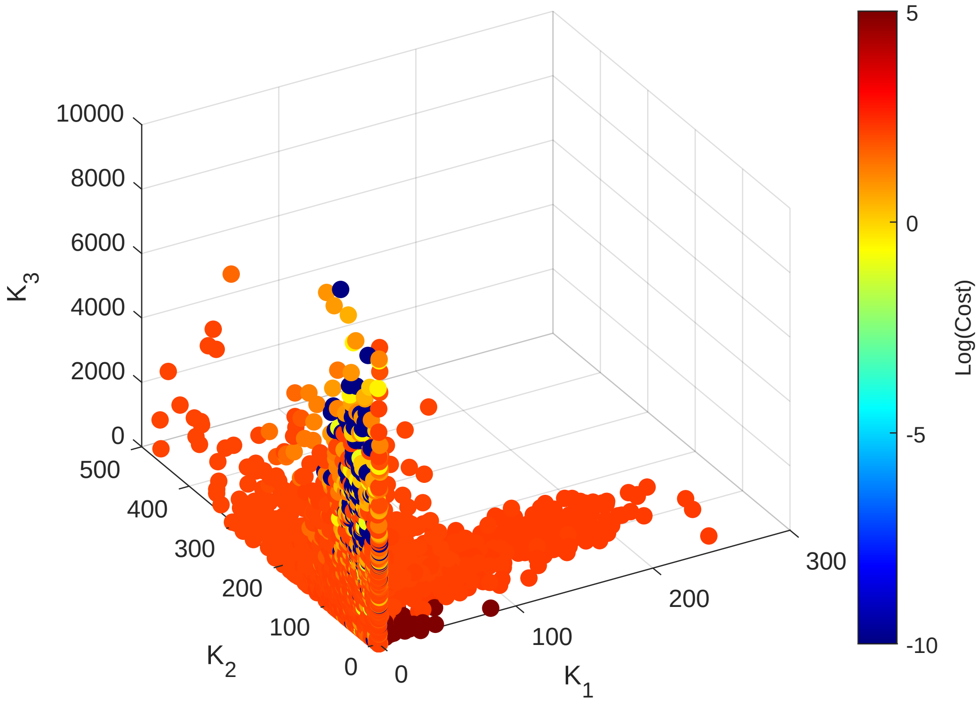

Figure 4.34: Resulting cost values for the particle locations in space

to synthesize velocity controller, where benefits of this optimization process will be more clear during the design of a more complex position controller.

Since, controller coefficients are defined in 4.71 ,from state space model in equation 3.41, transfer function between reference motor speed and actual motor speed can be as in equation 4.72.

$$\frac{\omega(s)}{\omega_{ref}(s)} = \frac{3.9019 \times 10^6 (s + 1250)(s + 125)}{(s + 530.6)(s + 158.3)(s^2 + 4913s + 7.257 \times 10^6)} \tag{4.72}$$

Resulting frequency response of the transfer function is displayed on Figure 4.35. Notice that bandwidth result is desired, an expected from the analogy that there is an order of magnitude between bandwidths of torque, velocity and position controllers.

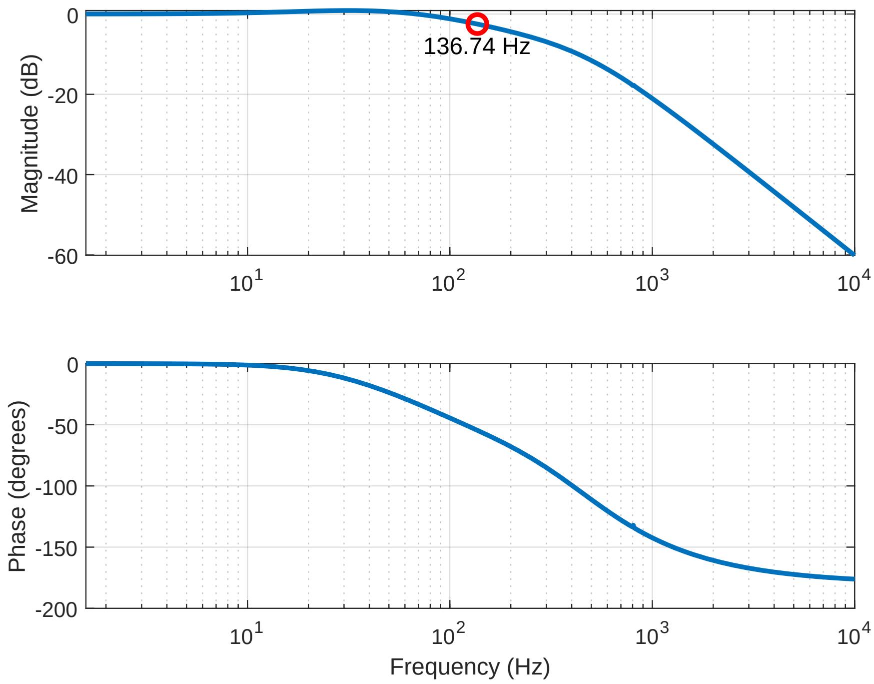

Figure 4.35: Frequency response of the velocity controller

4.3.4 Position controller design

Since velocity loop controller coefficients are determined in equation 4.71, neglecting low-pass term of the velocity controller $P(s)$ can be defined:

$$P_\omega(s) = \frac{2.5998 \times 10^{10}(s+1250)(s+125)}{s(s+530.6)(s+158.3)(s^2+4913s+7.257\times 10^6)(s^2+38.77s+2.564\times 10^7)} \tag{4.73}$$

Refer to Table 4.5 for the parametric uncertainties of the plants, hence plant families of uncertain $P_\omega(s)$ functions are plotted in Figure 4.36.

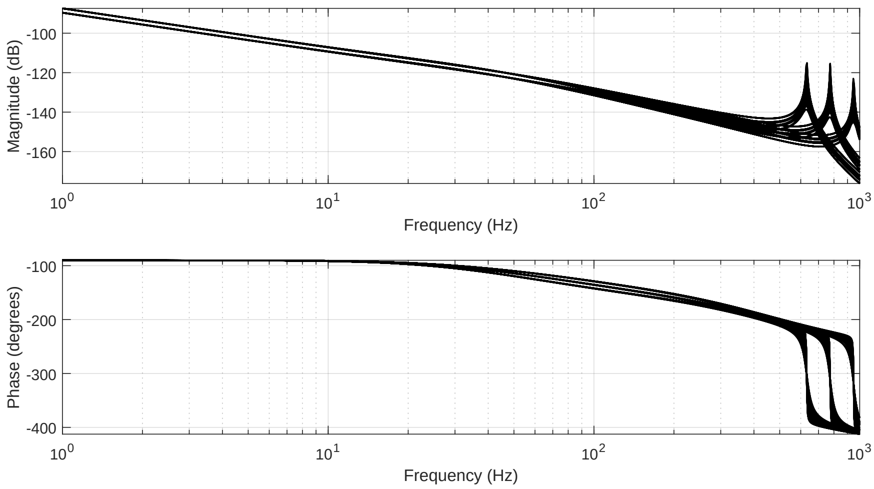

Figure 4.36: Frequency response of open loop uncertain plants $P_\omega(s)$

4.3.4.1 QFT bounds of position control

In previous section, in equations 4.51, 4.57, 4.59 4.61; stability, sensitivity, stiffness and tracking requirements of the actuator were defined. For the velocity controller, the polar code transformation along with QFT bounds generation for the uncertain plants were demonstrated. In a similar process, QFT bounds for the specific requirements are illustrated in Figures 4.37, 4.38, 4.39, 4.40 respectively.

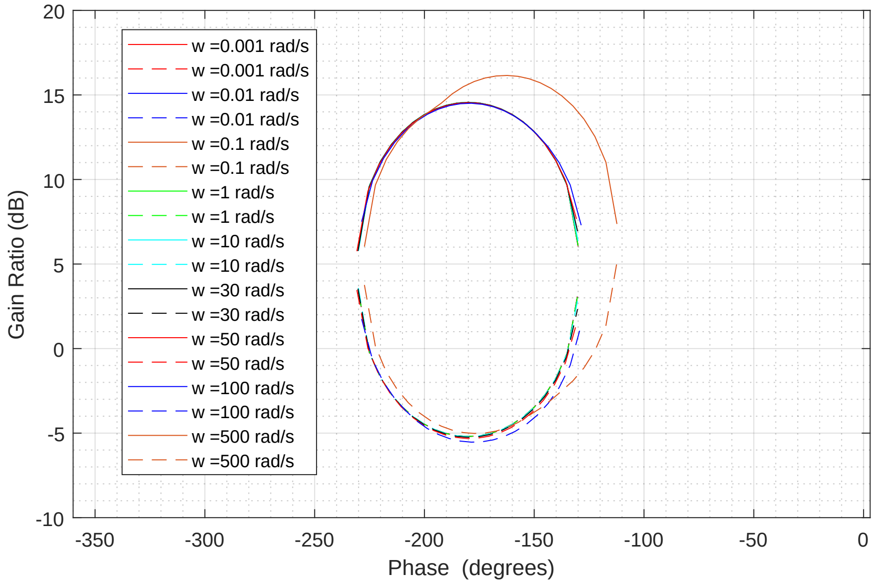

Figure 4.37: Stability restriction for $P_\omega(s)$

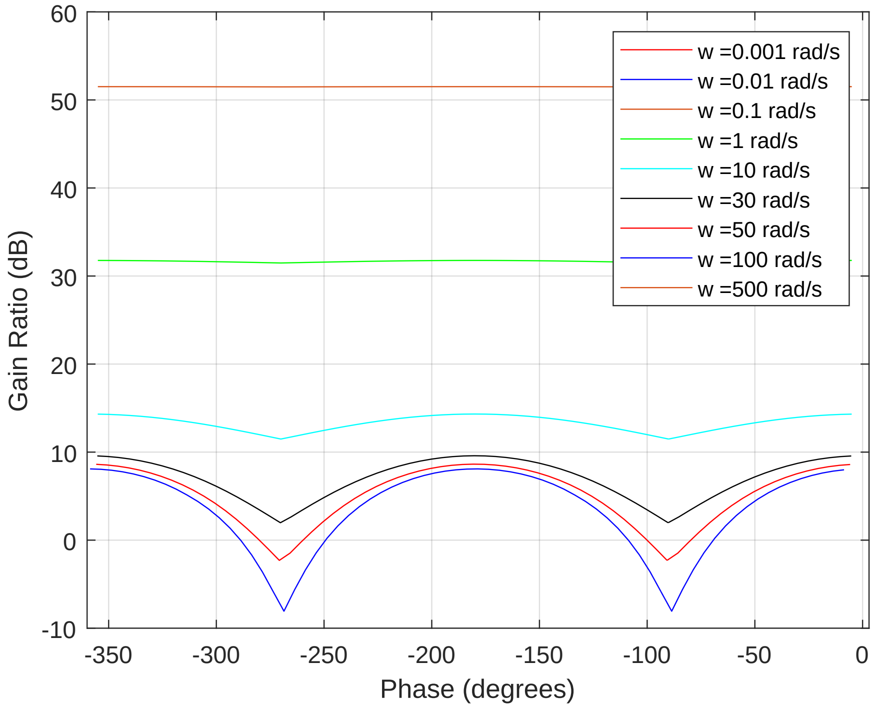

Figure 4.38: Sensitivity restriction for $P_\omega(s)$

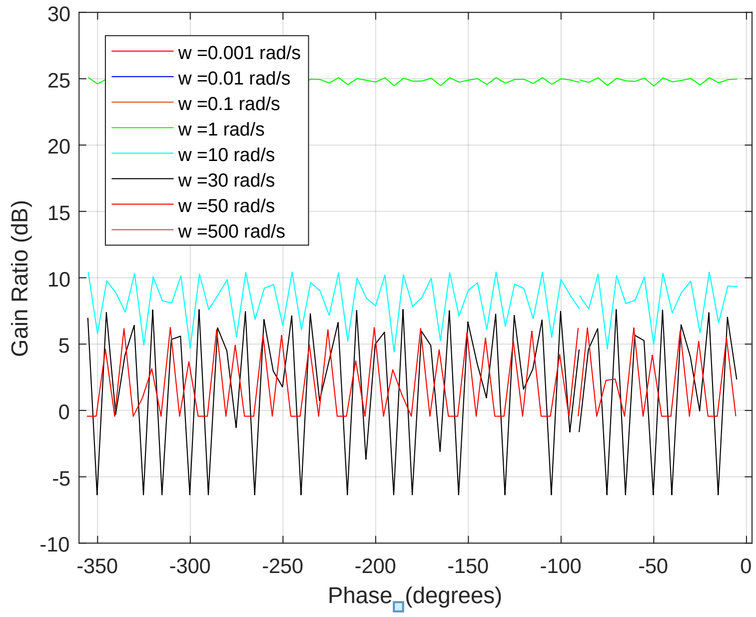

Figure 4.39: Stiffness restriction for $P_\omega(s)$

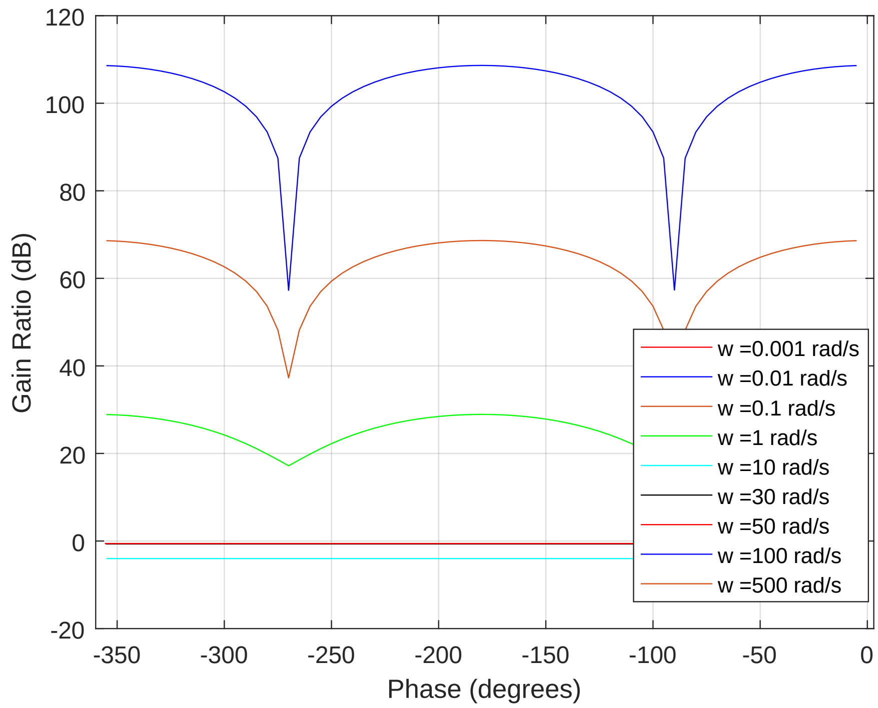

Figure 4.40: Tracking restriction for $P_\omega(s)$

4.3.4.2 Position controller synthesis

One advantage of QFT is defining specifications and designing controller according to robust system characteristics. In performance specifications, a controller effort reduction specifications into QFT boundaries could also be integrated. However problem of such boundary is that it may conflict with other boundaries, such at other boundaries requiring higher gain at a particular phase and controller effort reduction requiring lower gain. Hence, to solve this conflict, designer should cautiously determine specifications and respective boundaries formed by uncertain plant. This approach requires a lot of effort, even then as an end result control effort reduction would be designed according with respect to significance of other specifications. A simpler an a more elegant solution of this using optimization tools, as in previous velocity controller synthesis example a PSO was used. In the position controller design, performance specifications are considered as hard constraints, however it is not desired to satisfy these constraints by large margins due to high controller effort. Hence a cost function is designed to consider performance specifications as hard constraints with high cost values, together with a cost function on controller effort reduction as a soft constraint with low cost value.

The position loop can be designed with defined specifications in QFT bounds. The controller $G(s)$ is chosen in the form:

$$G(s) = \frac{K}{s} \prod_{i=1}^{3} \frac{\left(1 + \frac{s}{K_{2i-1}}\right)}{\left(1 + \frac{s}{K_{2i+1}}\right)} \tag{4.74}$$

The value of K is chosen initially with the objective to reduce control effort at high frequencies. A default integrator is present in order to decrease steady-state errors. In the literature, it is common to add a cost function for high-frequency gain, which is related to the gain of the controller [48, 70]. This problem is solved by preselecting the controller gain, where a cost for control effort is added at high frequencies. A six-dimensional PSO is conducted to determine poles and zeros of the proposed controller structure, where the setup was previously formed in Table 4.8. The cost function is designed as:

$$J = m_1 \sum_{i=1}^{k} \max(0, g_{\text{max}}(\omega_i) - |L(\omega_i)|) - m_2 \sum_{j=k}^{l} \max(0, g_{\text{min}}(\omega_j) - |L(\omega_j)|) \quad (4.75)$$

Where $g_{\text{max}}(\omega_i)$ refers to gain boundaries at a particular phase of $L(\omega_i)$. To reduce control effort, at higher frequencies, the cost is reduced for the lower gains. Weight m_1 is significantly greater than m_2, hence control effort is rather a soft constraint for the frequency limits $\{0.001, 0.01, 0.1, 1, 10, 30, 50, 100, 500\}$; $k = 7$ and $l = 9$.

Additionally, a cost function to ensure stability margins as in the velocity controller is used in equation 4.70. In Figure 4.41, controller coefficients during optimization process is illustrated. The dots refer to each particle coordinate during an iteration, while darker lines correspond to mean of these particles for specific coefficient. Notice that as iteration continue, deviation of particles from mean decrease. The solution is achieved around the 50th iteration, and iteration is continued to observe convergence, where the optimal controller parameter results with lowest cost are selected. In Appendix D, relevant cost function for the position controller design is shared.

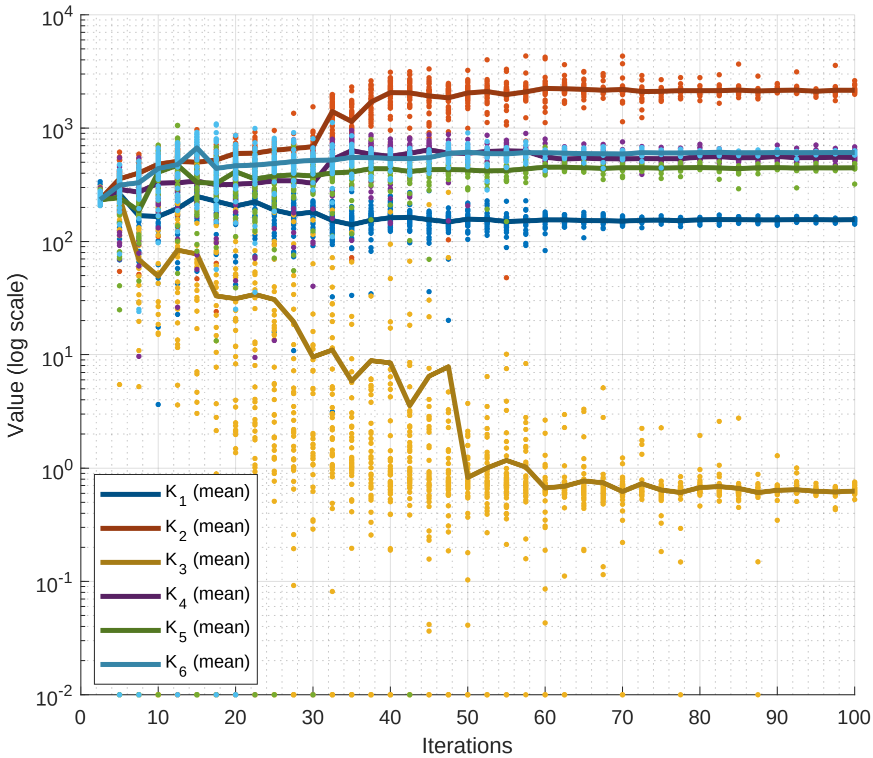

Figure 4.41: Position controller coefficients through PSO process

As a result, the following position controller is obtained with a pre-selected gain:

$$G(s) = 2.668 \times 10^9 \frac{(s+447)(s+153)(s+0.61)}{s(s+2155)(s+613)(s+522)} \tag{4.76}$$

Scaling factor for the position controller $G(s)$ is 165200. Notice that a transformer ratio was introduced between pump speed and actuator speed due to flowrate relation of pump displacement and effective piston area. Considering the transformer ratio between systems, the gain of the controller is 44.2 s^{-1}. Resulting EHA's open-loop response satisfies constraints specified as in Figure 4.42. QFT bounds are merged forms of Figures 4.37,4.38,4.39,4.40.

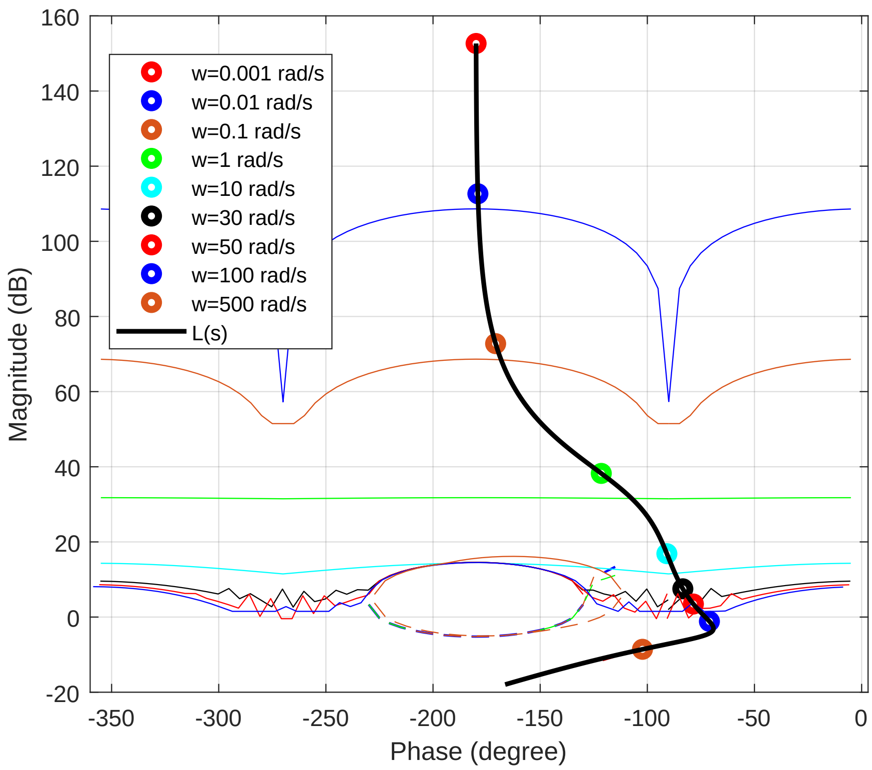

Figure 4.42: Open loop response of EHA with QFT bounds

Notice that at high frequencies, 100 rad/s and 500 rad/s are taken as low-effort high frequencies in the cost function. Hence, they impose low gain behavior, and the controller effort is taken as a soft constraint along with specified constraints. The Kalman filter introduced also decreases control effort due to reduced noise at higher frequencies. The next step in QFT design is synthesis is prefilter design. No optimization method is utilized prefilter-design, since it has a straightforward systematic tuning method. The prefilter is designed with systematic trial and error to meet reference tracking requirements:

$$F(s) = \frac{\left(\frac{s}{38}+1\right)}{\left(\frac{s}{77}+1\right)\left(\frac{s}{65}+1\right)} \tag{4.77}$$

Resultant closed-loop response can be observed in Figure 4.43. A bandwidth frequency larger then 9 Hz is achieved by the pre-filter design, where uncertain plants are within desired bounds.

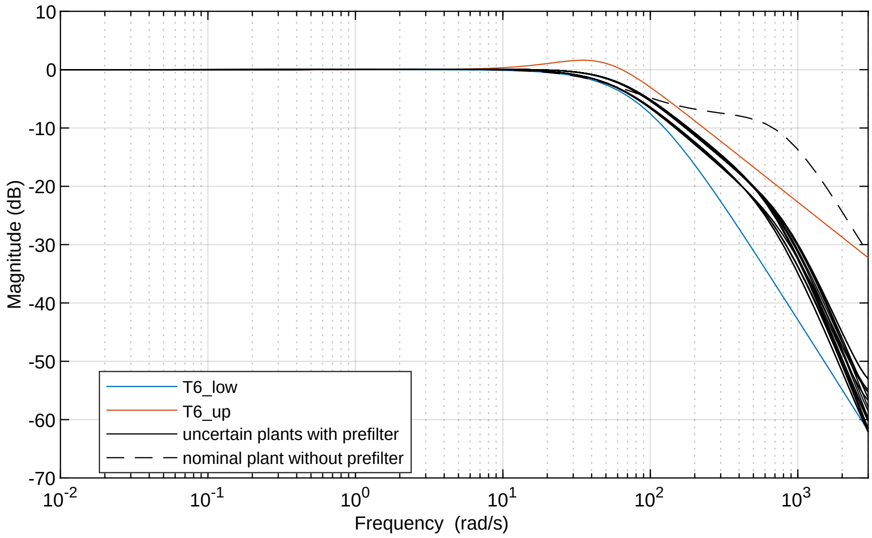

Figure 4.43: Frequency response of uncertain plants with designed pre-filter

In nonlinear simulations, due to friction and delay, the profile is not exactly same. However, since aim of the prefilter and tracking requirement is to essentially provide a high bandwidth, that goal is achivied in simulation analysis. In Figure 4.44, stiffness frequency response of uncertain EHA plants under position controller is illustrated. EHa stiffness holds the lower boundary condition for the desired frequency range.

Evaluation of sensitivity will be observed in experimental tests, especially considering faulty position sensor with high amplitude noise. Hence controller output, motor speed reference shall be investigated.

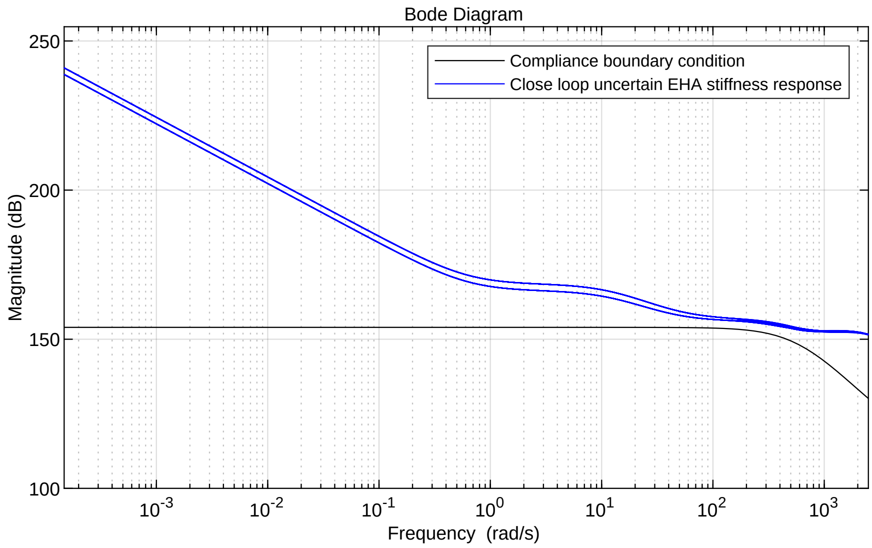

Figure 4.44: Lower bound of stiffness and closed loop uncertain EHA responses

4.4 Evaluation of the proposed EHA design and controller

In this section, the evaluation of the controller design is conducted. For this purpose, the performance of the Kalman filter, designed to estimate piston displacement, and the QFT-designed controller will be discussed.

In Figure 4.45, the position estimation of the Kalman filter for small step references is illustrated. This test is conducted to observe the system's sensitivity to reference commands and to determine if the Kalman filter exhibits any steady-state error in the actual piston measurement. The results indicate that the piston position is precisely controlled for commands larger than $10\ \mu$m. As observed in the figure, Kalman filter

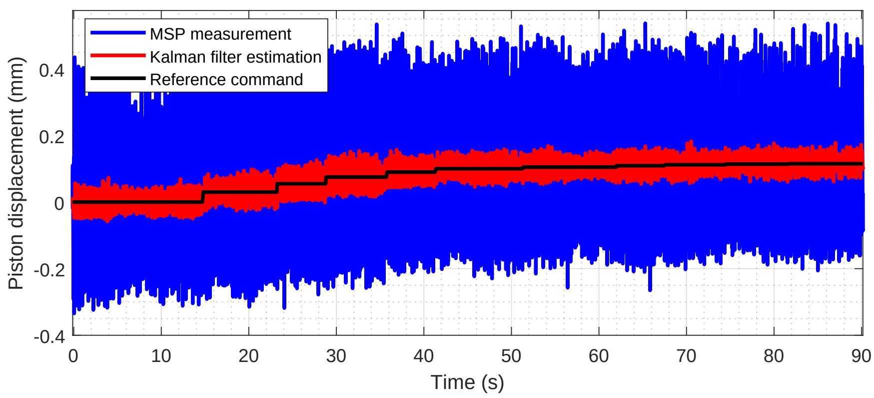

Figure 4.45: Evaluation of position Kalman filter on small step commands

does not completely suppress measurement noise along with low frequency behavior of noise, however reducing it significantly. In 4.46,the output of the controller along with the reference command with filter estimation is illustrated. Notice that there is a steady oscillation in controller output while the system command is stationary. This undesired behavior occurs due to noise of the position sensor, dead-zone of the hydraulic pump and high gain of the controller. Hence there is a peak $50\ \mu$m displacement on the actuator while it is stationary, and unnecessary energy consumption due to motor oscillation.

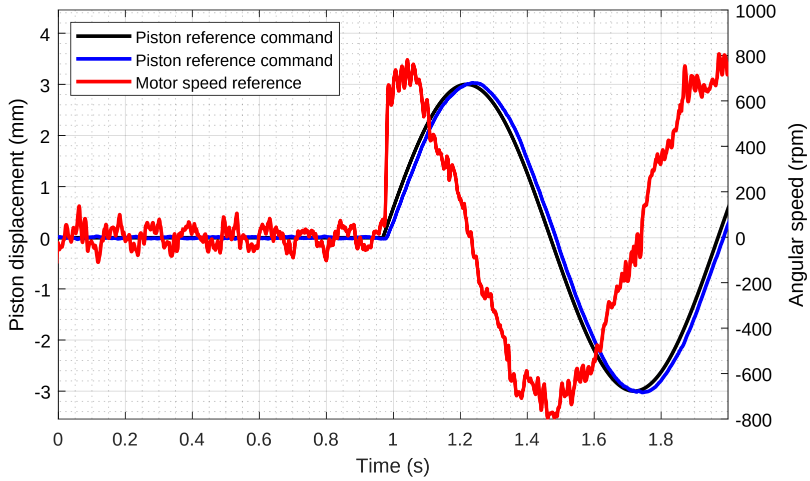

Figure 4.46: Controller output with respect to reference signal

From Table 3.1, the design specifications of EHA are tested. The load test result is illustrated in Figure 4.47, where motor torque steadily holds EHA at a constant position. In Figure 4.48, the maximum speed of the piston is observed and its relation with the motor speed. In Figure 4.49, the linearity of EHA is investigated, the maximum difference between the fit line and measurement value being 0.112%. A low-pass filter with a cut-off frequency of 10 Hz is applied at the measurement signal, in order to eliminate measurement noise. Test speed is selected in order to eliminate measurement noise more effectively. In Figure 4.50, the switching of a redundant motor is illustrated during a sinusoidal position reference. The system continues its operation successfully without major changes on the piston position profile, hence validating the redundant property of the system.

System type requirements are satisfied successfully. A frequency test is conducted for

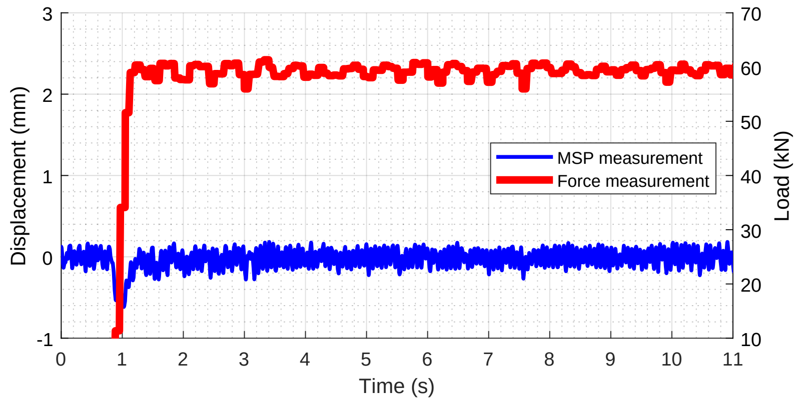

Figure 4.47: Steady position of EHA on load

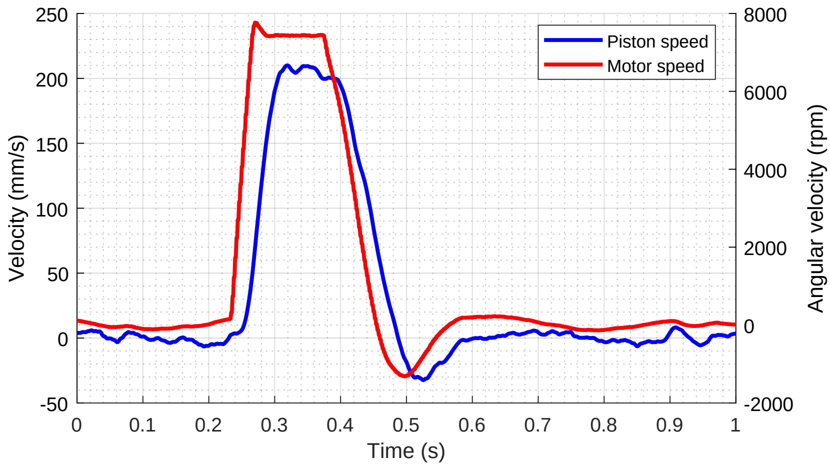

Figure 4.48: Maximum piston speed of EHA

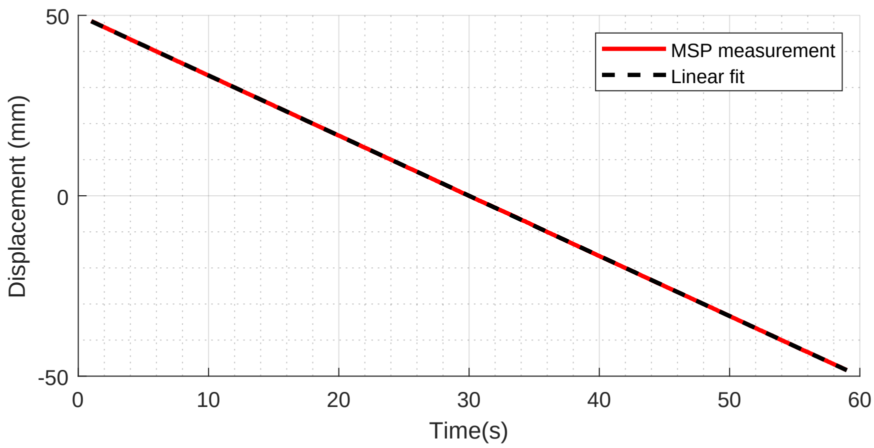

Figure 4.49: Linearity test of EHA

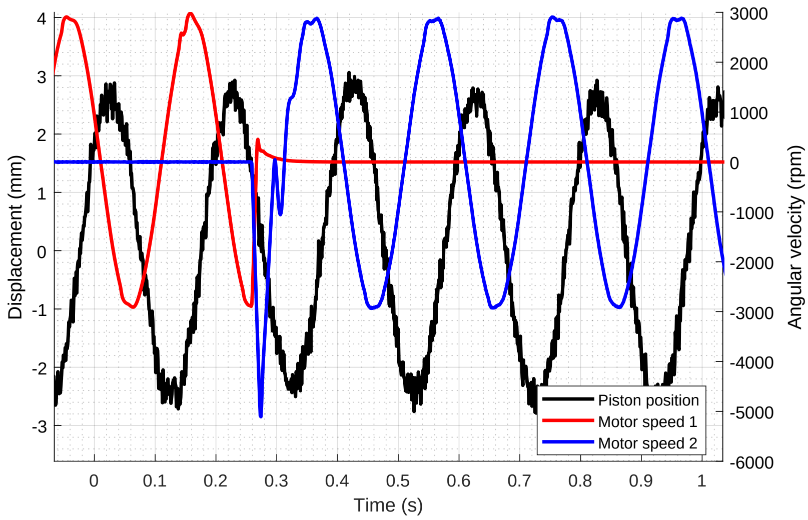

Figure 4.50: EHA redundant motor switch during operation

the actuator position on discrete frequency intervals. FFT method is applied on steady state signals to observe the frequency response of the actuator on specific frequencies. In Figure 4.51, the frequency response of test results is illustrated and compared with nonlinear simulation results for the same test signal. The resulting EHA performs on a high bandwidth, up to 11 Hz, satisfying design constraints. There is a high gain on lower frequencies due to the frequency behavior of Kalman filter that is explained in the previous section. While it is not on QFT design intervals, gain at low frequencies up to 1 dB is allowed on reference specification document [13]. The strict requirement is the bandwidth of the actuator.

In frequency response graphs, notice the high slope phase drop and hence gain increase. This behavior occurs due to the saturation of motor speed. The nonlinear simulation precisely matches the test results. In order to evaluate the performance of stiffness criteria, a nonlinear simulation is conducted with external force with an amplitude of 5 kN. In Figure 4.52, the frequency response of EHA stiffness under the proposed controller structure is demonstrated.

In conclusion, a Kalman filter for the position measurement is designed to estimate piston displacement even if sensor noise has faulty characteristics. A robust QFT design is applied for both velocity and position controllers. An optimization algorithm is operated in order to synthesize position and velocity controllers, where trial and er-

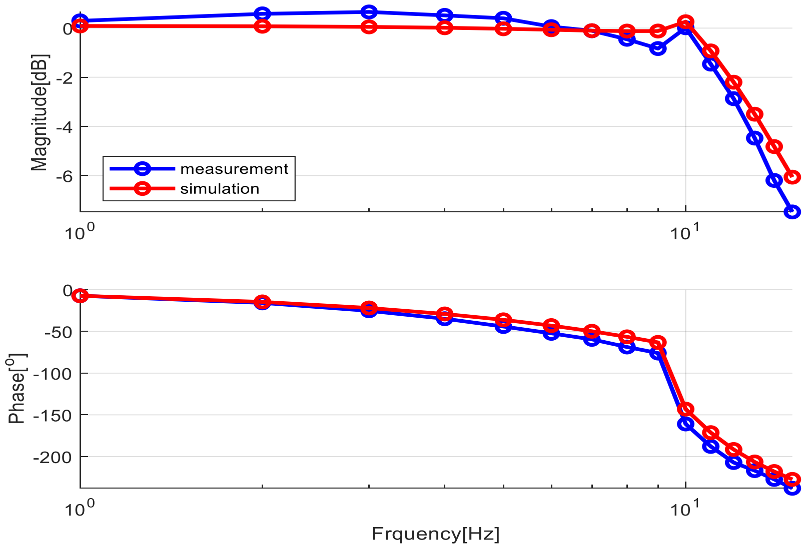

Figure 4.51: Frequency response of EHA with proposed controller structure

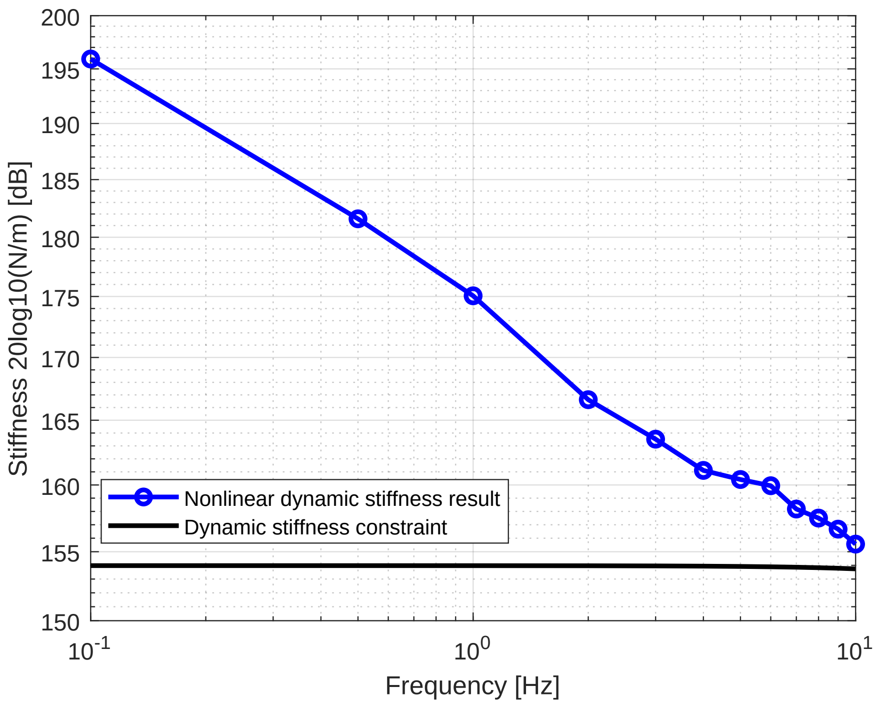

Figure 4.52: Nonlinear simulation result on EHA stiffness

ror process is eliminated and a low effort position controller is designed satisfying the performance specifications. The performance results of the developed EHA system exhibit success of the proposed controller structure.

CHAPTER 5

SUMMARY AND DISCUSSION

This thesis proposes a robust methodology for designing controllers for flight control actuators. The resulting controller design process is automated, eliminating the need for traditional trial-and-error methods typically employed in loop shaping during QFT. This approach significantly reduces the time required by designers and introduces a novel perspective on controller design that has not been explored in the literature on EHAs. Integrating control effort reduction as a secondary objective aligns well with the QFT boundaries, as these boundaries are strictly defined for the parametric uncertainties, thereby enabling the formulation of an optimization problem. Additionally, a fault-tolerant position estimator is developed from the hydraulic model of an EHA and proposes a novel robust solution.

Linear and nonlinear simulations are developed throughout the design process of EHA, both for component selection and deciding on the redundant properties of EHA. After EHA manifold is manufactured and sub-components are assembled, simulations are reconstructed especially on nonlinear properties of the subsystems. Such approach allowed designing the controllers in the simulation environment, as well as evoking a framework for the quantification of parametric uncertainties of the system. Kalman Filters, which aim to reduce noise of the position sensor, also designed and tested through simulation environment before integrating into the EHA system. From the literature, a Kalman Filter suitable for fault-detection algorithms is also developed through simulation environment.

There are numerous directions for future work. For instance, experimental evaluation of EHA uncertainty could be conducted under varying environmental conditions. Research on friction compensation for the controller could also be pursued to achieve a more precise control structure. The position controller design lacks saturation compensation, such as anti-windup strategies, which are critical for such systems. Integrating a saturation strategy into the proposed controller would be a valuable enhancement.

The optimization method employed in this study is relatively simple, with limited consideration of detailed coefficient determination. While the optimization successfully satisfies QFT bounds, a more comprehensive approach could suggest new tools and metrics for the design process. Additionally, the pre-determined controller structures used in this study are not necessarily optimal. Future research could focus on developing an inclusive optimization strategy for controller design, incorporating multi-objective functions to address various design goals.

Moreover, studying the performance of optimization methods for different types of specifications could help create a framework capable of synthesizing controllers for diverse conditions. In a continuously evolving technological environment, automated design methodologies are inherently valuable. To completely automate the design process, system modeling, along with an understanding of how parametric changes due to environmental conditions affect system properties, is essential for developing robust solutions.

REFERENCES

[1] McGraw-Hill, *McGraw-Hill Dictionary of Engineering*. New York: McGraw-Hill, 2nd ed., 2003.

[2] A. Vacca and G. Franzoni, *Hydraulic Fluid Power: Fundamentals, Applications, and Circuit Design*. Hoboken, NJ: Wiley, 2021.

[3] H. E. Merritt, *Hydraulic Control Systems*. New York, NY: John Wiley & Sons, 1967.

[4] AOPA, "How it works: Ailerons," *AOPA Flight Training Magazine*, September 2019.

[5] R. Pratt, *Flight Control Systems: Practical Issues in Design and Implementation*. London: Institution of Electrical Engineers, 2000.

[6] A. S. Technology, "Large aircraft hydraulic systems," 2024. Accessed: 2024-11-30.

[7] D. Li, S. Dong, J. Wang, and Y. Li, "Thermal dynamics and thermal management strategy for a civil aircraft hydraulic system," *Thermal Science*, vol. 24, no. 4, pp. 2311–2318, 2020.

[8] J. Li, W. Wang, and X. Zhang, "Hydraulic actuation systems: A review," *International Journal of Hydromechatronics*, vol. 3, no. 4, pp. 345–362, 2020.

[9] J.-C. Derrien, "Electromechanical actuator (ema) advanced technologies for flight controls," in *28th International Congress of the Aeronautical Sciences (ICAS)*, (Brisbane, Australia), Sagem Défense Sécurité, 2012.

[10] J.-J. Charrier and A. Kulshreshtha, "Electric actuation for flight and engine control; evolution and current trend," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, (Reno, Nevada), American Institute of Aeronautics and Astronautics, 2007.

[11] G. Online, “Entwicklung der flugzeug-steuerungssysteme,” *Gazette Online*, 2021. Accessed: 2024-12-01.

[12] ASME, “A review of high-speed electro-hydrostatic actuators,” *ASME Digital Collection*, vol. 141, no. 5, p. 050801, 2006.

[13] R. Navarro, “Performance of an electro-hydrostatic actuator on the f-18 systems research aircraft,” tech. rep., Dryden Flight Research Center, 1997.

[14] D. van den Bossche, “The a380 flight control electrohydrostatic actuators, achievements and lessons learnt,” in *25th International Congress of the Aeronautical Sciences*, (Hamburg, Germany), 2006.

[15] D. Robbins, J. D. Bobalik, D. Stena, N. Martin, K. Plag, K. Rail, and K. Wall, “F-35 subsystems design, development & verification,” in *AIAA Aviation Forum*, AIAA, 2018.

[16] C. Wiegand, B. A. Bullick, J. A. Catt, J. W. Hamstra, G. P. Walker, and S. Wurth, “F-35 air vehicle technology overview,” in *AIAA Aviation Forum*, AIAA, 2018.

[17] G. K. Fourlas and G. C. Karras, “A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles,” *Machines*, vol. 9, no. 9, p. 197, 2021. Published: 13 September 2021.

[18] M. Żugaj, “Reconfiguration of fixed-wing uav control system in autonomous flight,” *The Institute of Aeronautics and Applied Mechanics, Warsaw University of Technology*, 2021. Contact author for further details on publication source.

[19] H. Shen and N. Hovakimyan, “Adaptive flight control design for actuator fault compensation,” in *IEEE Conference on Decision and Control*, pp. 3340–3345, IEEE, 2014.

[20] G. Langford, J. Schmiedeler, and C. Cotting, “Adaptive flight control for next-generation aircraft,” tech. rep., NASA Langley Research Center, 2008.

[21] T. M. Jahns and R. C. V. Nocker, “High-performance eha controls using an interior permanent magnet motor,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 3, pp. 534–542, 1990.

[22] J. Anderson, “Variable displacement electro-hydrostatic actuator,” in *Proceedings of the IEEE 1991 National Aerospace and Electronics Conference (NAECON)*, vol. 2, (Dayton, OH), pp. 529–534, IEEE, 1991.

[23] S. Frischemeier, “Electrohydrostatic actuators for aircraft primary flight control - types, modelling and evaluation,” in *Proceedings of the 5th Scandinavian International Conference on Fluid Power (SICFP ’97)*, (Linköping, Sweden), May 1997.

[24] S. Habibi and A. Goldenberg, “A mechatronics approach for the design of a new high performance electrohydraulic actuator,” in *SAE Technical Paper Series*, no. 1999-01-2853, SAE International, 1999.

[25] J. Kim and Y. Hong, “Improvement of backdrivability of a force controlled eha by introducing bypass flow control,” *International Journal of Precision Engineering and Manufacturing*, vol. 21, pp. 819–830, 2020.

[26] W. S. C. L. Kang Rongjie, Jiao Zongxia, “Design and simulation of electro-hydrostatic actuator with a built-in power regulator,” *Chinese Journal of Aeronautics*, vol. 22, no. 6, pp. 700–706, 2009. Received 16 December 2008; accepted 15 May 2009.

[27] T. Guo, X. Han, T. Minav, and Y. Fu, “A preliminary design method of high-power electro-hydrostatic actuators considering design robustness,” *Actuators*, vol. 11, no. 11, p. 308, 2022.

[28] T. M. Chi Zhang, Xu Han and Y. Fu, “Multi-objective optimization design of a 30 kw electro-hydrostatic actuator,” in *Proceedings of the First International Electronic Conference on Actuator Technology: Materials, Devices, and Applications*, (Beihang University, Beijing, China and Tampere University, Tampere, Finland), 2020.

[29] L. Xue, S. Wu, Y. Xu, and D. Ma, “A simulation-based multi-objective optimization design method for pump-driven electro-hydrostatic actuators,” *Processes*, vol. 7, no. 5, p. 274, 2019.

[30] S. Habibi and A. Goldenberg, “Design of a new high performance electrohydraulic actuator,” in *Proceedings of the 1999 IEEE/ASME International Con-*

ference on Advanced Intelligent Mechatronics, (Atlanta, USA), IEEE/ASME, September 19–23 1999. Member, IEEE, and Fellow, IEEE.

[31] M. A. E. Sayed and S. Habibi, "Inner-loop control for electro-hydraulic (eha) actuation systems," in *Proceedings of the ASME 2009 Dynamic Systems and Control Conference (DSCC2009)*, (Hollywood, California, USA), pp. DSCC2009–2586, McMaster University, Department of Mechanical Engineering, ASME, October 12–14 2009.

[32] D. A.-A. Xiang Hu and S. Habibi, "A new sliding mode controller for electro-hydraulic actuator (eha) applications," *Mechanical Engineering*, 2014.

[33] Y. Chinniah, R. Burton, and S. Habibi, "Failure monitoring in a high performance hydrostatic actuation system using the extended kalman filter," *Mechatronics*, vol. 16, no. 10, pp. 643–653, 2006.

[34] H.-A. T. Van Du Phan and K. K. Ahn, "Backstepping control of an electro-hydraulic actuator using kalman filter," in *Proceedings of the 2023 26th International Conference on Mechatronics Technology (ICMT)*, (Busan, Korea, Republic of), pp. 1–6, IEEE, October 18–21 2023.

[35] H. Chen, X. Miao, W. Mao, S. Zhao, G. Yang, and Y. Bo, "Fault diagnosis of eha with few-shot data augmentation technique," *Smart Materials and Structures*, vol. 32, p. 044005, mar 2023.

[36] S. H. Cho and R. Burton, "Position control of high performance hydrostatic actuation system using a simple adaptive control (sac) method," *Mechatronics*, vol. 21, no. 1, pp. 109–115, 2011. Received 13 February 2010; Accepted 10 September 2010; Available online 13 October 2010.

[37] S.-R. Lee and Y.-S. Hong, "A dual eha system for the improvement of position control performance via active load compensation," *International Journal of Precision Engineering and Manufacturing*, vol. 18, no. 7, pp. 937–944, 2017. Accessed: 2024-12-04.

[38] Y. S. Yang Lin and R. Burton, "Modeling and robust discrete-time sliding-mode control design for a fluid power electrohydraulic actuator (eha) system," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 1–10, February 2013.

[39] J. W. Hui Zhang, Xiaotao Liu and H. R. Karimi, “Robust h_∞ sliding mode control with pole placement for a fluid power electrohydraulic actuator (eha) system,” *International Journal of Advanced Manufacturing Technology*, vol. 74, pp. 195–206, 2014. Received: 25 January 2014; Accepted: 29 April 2014.

[40] Z. Q. Shi, Z. Tang, and C. Pei, “Sliding mode control for electrohydrostatic actuator,” *Computer Measurement & Control*, 2014.

[41] R. Yang, Y. Fu, L. Zhang, H. Qi, X. Han, and J. Fu, “A novel sliding mode control framework for electrohydrostatic position actuation system,” *Mathematical Problems in Engineering*, vol. 2018, no. 1, p. 7159891, 2018.

[42] D. Thompson, J. Pruyn, and A. Shukla, “Feedback design for robust tracking and robust stiffness in flight control actuators using a modified qft technique,” in *Proceedings of the 1999 American Control Conference*, (San Diego, CA), IEEE, 1999.

[43] K. H. Kang, “Electro-hydrostatic actuator controller design using quantitative feedback theory,” Master’s thesis, Air Force Institute of Technology, Air University, 1994. Approved for public release; distribution unlimited.

[44] D. N. C. N. Kyoung Kwan Ahn and M. Jin, “Adaptive backstepping control of an electrohydraulic actuator,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, pp. 987–995, June 2014.

[45] G. Ren, M. Esfandiari, J. Song, and N. Sepehri, “Position control of an electrohydrostatic actuator with tolerance to internal leakage,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2224–2232, 2016.

[46] I. Horowitz, *Synthesis of Feedback Systems*. New York: Academic Press, 1963.

[47] M. Garcia-Sanz, *Robust Control Engineering: Practical QFT Solutions*. Boca Raton, FL: CRC Press, 1st ed., 2021.

[48] L. Meng and D. Xue, “Automatic loop shaping in fractional-order qft controllers using particle swarm optimization,” in *IEEE International Conference on Control and Automation*, (Christchurch, New Zealand), 2009.

[49] C. K. B. Satpati and S. Datta, “Robust pid controller design using pso enabled automated qft approach for first order lag system with minimal dead time,” *Systems Science & Control Engineering*, pp. 477–485, 2014.

[50] S. B. H. I. Ali, S. B. B. M. Noor and M. H. Marhaban, “Quantitative feedback theory control design using particle swarm optimization method,” *Transactions of the Institute of Measurement and Control*, vol. 0, no. 0, pp. 1–14, 2011.

[51] S. J. Chapman, *Electric Machinery Fundamentals*. McGraw-Hill Education, 5th ed., 2011.

[52] C. Zhang, C. Zhu, B. Meng, and S. Li, “Challenges and solutions for high-speed aviation piston pumps: A review,” *Aerospace*, vol. 8, no. 12, p. 392, 2021.

[53] P. H. Corporation, “Compact electro-hydrostatic actuator (eha),” 2021. Accessed: 2024-11-30.

[54] Ömer Faruk Turhan, “Multi-objective optimization of electro-hydrostatic actuators with octagon architecture,” master’s thesis, Middle East Technical University, Graduate School of Natural and Applied Sciences, August 2024.

[55] M. Stewart, *5 - Rotary pumps*, vol. IV, pp. 415–440. Elsevier: Gulf Professional Publishing, 2019. Pump and Compressor Systems: Mechanical Design and Specification.

[56] R. Kang, J. C. Mare, and Z. Jiao, “Nonlinear modeling and control design of electro-hydrostatic actuator,” in *Proceedings of the 7th JFPS International Symposium on Fluid Power, Toyama, Japan*, (Toyama, Japan), pp. P2–15, JFPS, September 15-18 2008. School of Automation Science and Electrical Engineering, Beihang University, Beijing, China, and Institut National des Sciences Appliquées de Toulouse, France.

[57] B. Armstrong and C. C. de Wit, “Friction modeling and compensation,” in *The Control Handbook* (W. S. Levine, ed.), pp. 1369–1382, CRC Press, 1995.

[58] M. Inc., “Dual redundant electrohydrostatic actuator.”

[59] Qihaitao, Hanxu, Tengyating, and Liuzilong, “Double-redundancy electro-hydrostatic actuator (eha),” 2015. Accessed: 2024-12-04.

[60] M. Inc., "Aircraft electrohydrostatic actuators," *Moog*, 2024.

[61] C. Özbaş, A. C. Afatsun, and H. Çalışkan, "Improvement of dynamic stiffness in electro-hydrostatic actuators," in *Proceedings of the 24th National Automatic Control Meeting (TOK 2023)*, pp. 127–135, ITU, 2023.

[62] B. Armstrong-Hélouvry, P. Dupont, and C. C. de Wit, "A survey of models, analysis tools and compensation methods for the control of machines with friction," *Automatica*, vol. 30, no. 7, pp. 1083–1138, 1994.

[63] A. Becker, *Kalman Filter from the Ground Up*. Alex Becker, first ed., 2023.

[64] Y. Chinniah, R. Burton, S. Habibi, and E. Sampson, "Identification of the nonlinear friction characteristics in a hydraulic actuator using the extended kalman filter," *Transactions of the CSME / de la SCGM*, vol. 32, no. 2, p. 121, 2008. Received February 2005, Accepted July 2008.

[65] Y. A. Chinniah, *Fault Detection in the Electrohydraulic Actuator Using Extended Kalman Filter*. Ph.d. thesis, University of Saskatchewan, Saskatoon, Canada, 2004.

[66] R. G. Brown and P. Y. Hwang, *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises*. Hoboken, NJ: John Wiley & Sons, 4th ed., 2012.

[67] I. Horowitz, "Application of quantitative feedback theory (qft) to flight control problems," in *Proceedings of the 29th Conference on Decision and Control*, (Honolulu, Hawaii), University of California, Davis, IEEE, 1990.

[68] H. M. S. Ballesteros, R. d. N. Calvo, and A. A. Filho, "Dynamic stiffness enhancement of a flight control actuator using control techniques," in *IEEE International Conference on Mechatronics (ICM)*, (Churchill, VIC, Australia), IEEE, 2017.

[69] Z. Xinchao, "A perturbed particle swarm algorithm for numerical optimization," *Applied Soft Computing*, vol. 10, no. 1, pp. 119–124, 2010.

[70] P. D. P. Baldini, G. Calandrini and H. Bambill, "Pso algorithm-based robust design of pid controller for variable time-delay systems: Aqm application," *Journal of Computer Science and Technology*, vol. 15, no. 2, pp. 100–106, 2015.

APPENDICES

A Hydraulic schematic of EHA for mechanical design

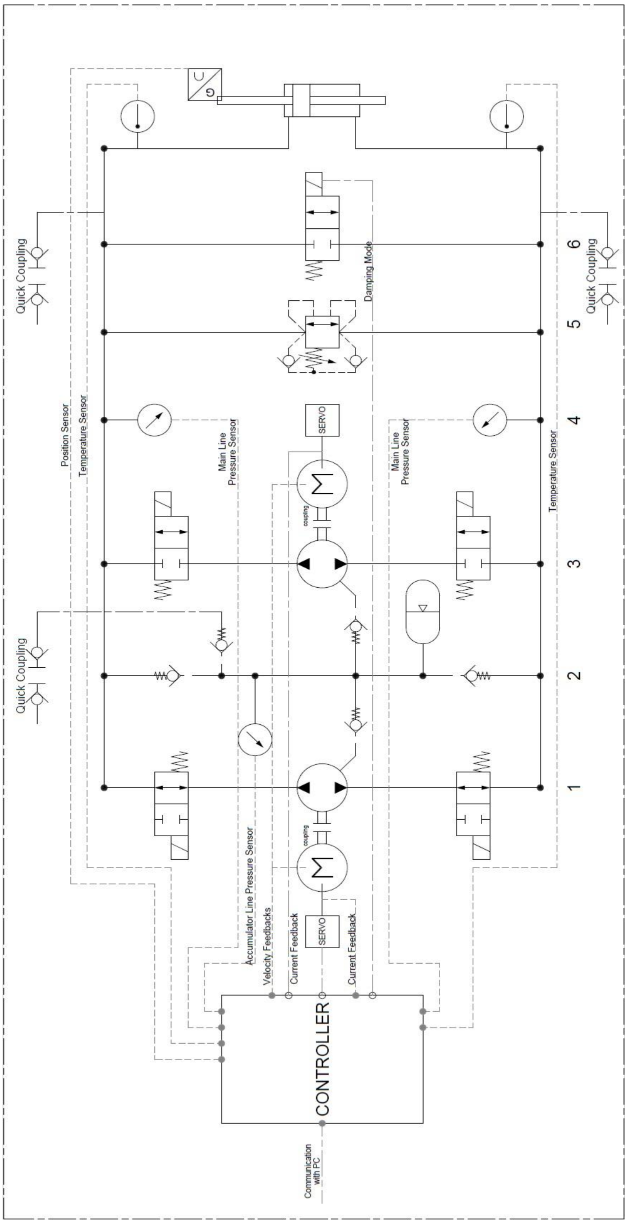

Figure A.1: Frequency response of EHA with proposed controller structure

B Frequency response algorithms

B.1 Sine sweep test function in TWINCAT

```
FUNCTION_BLOCK Fr_Test
VAR_INPUT
    max_freq : LREAL;
    period_count : LREAL;
    starting_freq : LREAL;
    freq_increment : LREAL;
    freq_interval : LREAL;
END_VAR
VAR_OUTPUT
    bBode_out : BOOL;
    sin_out : LREAL;
END_VAR
VAR
    bSinTime : BOOL := TRUE;
    relTime : LREAL := 0.0;
    firstTime : ULINT := 0;
    bDelay : BOOL := FALSE;
    bodeStartTime : LREAL := 0.0;
    bode_relax_time : LREAL := 0.0;
    bodeCounter : LREAL := 0.0;
END_VAR

METHOD bode : BOOL
VAR_INPUT
    bBode : BOOL;
    w : LREAL;
    amp : LREAL;
END_VAR
VAR
    relax_constant : LREAL := 5.0;
END_VAR

// Initialize Outputs
```

```
bBode_out := bBode;
sin_out := 0.0;

// Bode Logic
IF bBode THEN
    bode_relax_time := ULINT_TO_LREAL(F_GetSystemTime()) / 10000000 - bodeStartTime;
    w := bodeCounter;

    IF bode_relax_time > freq_interval THEN
        sin_wave(bPistonSin := TRUE, w := w, amp := amp, sin_out => sin_out);
    END_IF

    IF relTime > period_count / w THEN
        sin_wave(bPistonSin := FALSE, w := w, amp := amp, sin_out => sin_out);
        relTime := 0;
        IF (bodeCounter + freq_increment) > max_freq THEN
            bodeCounter := starting_freq;
        ELSE
            bodeCounter := bodeCounter + freq_increment;
        END_IF
        bodeStartTime := ULINT_TO_LREAL(F_GetSystemTime()) / 10000000;
    END_IF
END_IF

IF bodeCounter > max_freq THEN
    bodeCounter := starting_freq;
    bBode_out := FALSE;
END_IF
END_METHOD

METHOD sin_wave : BOOL
VAR_INPUT
    bPistonSin : BOOL;
```

```
    w : LREAL;
    amp : LREAL;
END_VAR
VAR_OUTPUT
    sin_out : LREAL;
END_VAR

IF bPistonSin THEN
    IF bSinTime THEN
        firstTime := F_GetSystemTime();
    END_IF
    relTime := ULINT_TO_LREAL(F_GetSystemTime() - firstTime) / 10000000;
    sin_out := SIN(2 * 3.14159265359 * w * relTime) * amp;
    bDelay := TRUE;
    bSinTime := FALSE;
END_IF

IF NOT bPistonSin AND bDelay THEN
    sin_out := 0;
    relTime := 0;
    bDelay := FALSE;
    bSinTime := TRUE;
END_IF
END_METHOD
```

B.2 Python code for frequency response analysis

```
  -*- coding: utf-8 -*-
"""
Created on Fri Feb 16 16:30:22 2024

@author: gcozb
"""

import matplotlib.pyplot as plt
```

```
import numpy as np
import warnings


def fftPlot(sig, dt=None, plot=True):
      Here it's assumes analytic signal (real signal...) - so only half of the axis is required

    if dt is None:
        dt = 1
        t = np.arange(0, sig.shape[-1])
        xLabel = 'samples'
    else:
        t = np.arange(0, sig.shape[-1]) * dt
        xLabel = 'freq [Hz]'

    if sig.shape[0] % 2 != 0:
        warnings.warn("signal preferred to be even in size, autoFixing it...")
        t = t[0:-1]
        sig = sig[0:-1]

    sigFFT = np.fft.fft(sig) / t.shape[0]    Divided by size t for coherent magnitude

    freq = np.fft.fftfreq(t.shape[0], d=dt)

      Plot analytic signal - right half of frequence axis needed only...
    firstNegInd = np.argmax(freq < 0)
    freqAxisPos = freq[0:firstNegInd]
    sigFFTPos = 2 * sigFFT[0:firstNegInd]    *2 because of magnitude of analytic signal

    if plot:
        plt.figure()
        plt.plot(freqAxisPos, np.angle(sigFFTPos))
```

```
        plt.xlabel(xLabel)
        plt.ylabel('mag')
        plt.title('Analytic FFT plot')
        plt.show()

    return sigFFTPos, freqAxisPos


if __name__ == "__main__":
    dt = 1 / 1000

      Build a signal within Nyquist - the result will be the positive FFT with actual magnitude
    f0 = 200    [Hz]
    t = np.arange(0, 1 + dt, dt)
    sig = (
        1 * np.sin(2 * np.pi * f0 * t)
        + 10 * np.sin(2 * np.pi * f0 / 2 * t)
        + 3 * np.sin(2 * np.pi * f0 / 4 * t)
        + 10 * np.sin(2 * np.pi * (f0 * 2 + 0.5) * t)    <--- not sampled on grid so the peak will not be actual height
    )
      Result in frequencies
    fftPlot(sig, dt=dt)
      Result in samples (if the frequencies axis is unknown)
    fftPlot(sig)

def dB_phaseGen (sig1, sig2, freqAxis,debug):
    maxval = np.max(np.abs(sig1))
    maxval2 = np.max(np.abs(sig2))
    ind = np.where(np.abs(sig1)==maxval)
    freq = freqAxis[ind]
    dB= 20*np.log10(np.abs(maxval2)/np.abs(maxval))
    phase1 = np.arctan2(np.imag(sig1[ind]),np.real(sig1[ind]))*180/np.pi
    phase2 = np.arctan2(np.imag(sig2[ind]),np.real(sig2[ind]))*180/np.pi
```

```
    phase = -(phase1-phase2)
    if debug:
        print("result : " + str(np.abs(maxval2)))
        print("ref : " + str(np.abs(maxval)))
    return dB,phase[0]


def dB_phaseResult(res,ref, cur_freq, sample_time, freq_repeat,
    ↪ plot=True, debug=False):

    period_sample = int(2/cur_freq/sample_time)
    large_sample = int(1/cur_freq*freq_repeat/sample_time)
    sigFFTRef, freqAxisPos1 =
    ↪ fftPlot(ref[period_sample:large_sample], dt=sample_time,
    ↪ plot=False)
    sigFFTRes, freqAxisPos2 =
    ↪ fftPlot(res[period_sample:large_sample], dt=sample_time,
    ↪ plot=False)
    if plot:
        plt.figure(1)
        plt.plot(freqAxisPos1,np.abs(sigFFTRes))
        plt.figure(2)
        plt.plot(freqAxisPos1,np.abs(sigFFTRef))

    dB,phase =dB_phaseGen(sigFFTRef, sigFFTRes, freqAxisPos1,debug)
    return dB,phase
```

C TWINCAT programs and Matlab function codes of Kalman Filtering functions

C.1 Chinniah Friction Model

```
function Ff = chinniah_friction_model(xdot)
% codegen

m = 20; % kg
linear_factor = 1;
a1 = 2.1 * 10^4 * linear_factor; % Ns/m^2
a2 = -1.45 * 10^3 * linear_factor; % Ns/m
a3 = 46; % N
quadratic_F = a1 * sign(xdot) * xdot^2 + a2 * xdot + sign(xdot) * a3;
Ff = quadratic_F / m;

end
```

C.2 Kalman Equations with 3 States

```
function [rbegin, K_out, X_out, P_out] = Kalman_equations_3states(dP, Z_measure, X_est, P_est, begin)
coder.extrinsic('exist')

tc_relation = 1;
X_out = [0 0 0]';
P_out = diag([1e9, 1e9, 1e9]);

Zk = Z_measure;
rbegin = begin;
% codegen
M = 20; % kg
A = 5.05 * 1e-4; % m^2 area
Ts = 1e-4; % s, sampling time

```

```
X0 = [0 0 0]';
Rk = diag([1e-9, 1e-3]) * tc_relation;
P0 = diag([1e9, 1e9, 1e9]) * tc_relation;
Qk = diag([1e-9, 1e-12, 1e-4]) * tc_relation;
Hk = [1 0 0; 0 1 0];
Gk = [0 1 0]';
K_out = P0 * Hk' * inv(Hk * P0 * Hk' + Rk);

if rbegin == 1 % If we already have estimations for X and P
    x2 = X_est(2);
    x3 = X_est(3);
    phi22 = 1 - 0.6 * x3 * Ts / M;
    phi23 = -0.4 * x2 * Ts / M;

    state_trans = [1 Ts 0; 0 1 -Ts * sign(x2) / M; 0 0 1];
    K_k = P_est * Hk' / (Hk * P_est * Hk' + Rk);
    X_k = X_est + K_k * (Zk - Hk * X_est);
    P_k = (eye(3) - K_k * Hk) * P_est;
    X_k1 = state_trans * X_k + Gk * dP * A * Ts / M;
    P_k1 = state_trans * P_k * state_trans' + Qk;
else
    % Using initial estimate for the first iteration
    x2 = X0(2);
    x3 = X0(3);
    phi22 = 1 - 0.5 * x3 * Ts / M;
    phi23 = -0.5 * x2 * Ts / M;
    state_trans = [1 Ts 0; 0 1 -Ts * sign(x2) / M; 0 0 1];

    K_k = P0 * Hk' * inv(Hk * P0 * Hk' + Rk);
    X_k = X0 + K_k * (Zk - Hk * X0);
    P_k = (eye(3) - K_k * Hk) * P0;
    X_k1 = state_trans * X_k + Gk * dP * A * Ts / M;
    P_k1 = state_trans * P_k * state_trans' + Qk;
end

X_out = X_k1;
P_out = P_k1;
```

```
K_out = K_k;
end
```

C.3 Kalman Equations with 4 States

```
function [K_out, X_out, P_out] = Kalman_denklemleri_4durum(Z_measure, a3, dP, X_est, P_est, begin)
coder.extrinsic('exist')

X_out = [0 0 0 0]';
P_out = diag([1e9, 1e9, 1e10, 1e9]);
beginout = begin;

Zk = Z_measure;

Ts_constant = 1;

% codegen
M = 20; % kg
A = 5.05 * 1e-4; % m^2 area
Ts = 1e-4; % s, sampling time
X0 = [0 0 0 0]';
Rk = diag([1e-12, 1e-3]) * Ts_constant;
P0 = diag([1e9, 1e9, 1e9, 1e9]) * Ts_constant;
Qk = diag([1e-12, 1e-7, 1e-7, 1e-4]) * Ts_constant;

Hk = [1 0 0 0; 0 1 0 0];
Gk = [0 1 0 0]';
K_out = P0 * Hk' * inv(Hk * P0 * Hk' + Rk);

if begin == 1
    x2 = X_est(2);
    x3 = X_est(3);
    x4 = X_est(4);

```

```
    state_share = 0.0;
    phi22 = 1 - state_share * sign(x2) * x2 * x3 * Ts / M - state_share * x4 * Ts / M;
    phi23 = -(1 - state_share) * sign(x2) * x2^2 * Ts / M;
    phi24 = -(1 - state_share) * x2 * Ts / M;

    a3_f = -sign(x2) * a3 * Ts / M;
    state_trans = [1 Ts 0 0; 0 phi22 phi23 phi24; 0 0 1 0; 0 0 0 1];

    K_k = P_est * Hk' / (Hk * P_est * Hk' + Rk);
    X_k = X_est + K_k * (Zk - Hk * X_est);
    P_k = (eye(4) - K_k * Hk) * P_est;

    X_k1 = state_trans * X_k + Gk * (dP * A * Ts / M + a3_f);
    P_k1 = state_trans * P_k * state_trans' + Qk;
else
    x2 = X0(2);
    x3 = X0(3);
    x4 = X0(4);
    phi22 = 1;
    phi23 = -sign(x2) * x2^2 * Ts / M;
    phi24 = -x2 * Ts / M;
    a3_f = -sign(x2) * a3 * Ts / M;
    state_trans = [1 Ts 0 0; 0 phi22 phi23 phi24; 0 0 1 0; 0 0 0 1];

    K_k = P0 * Hk' * inv(Hk * P0 * Hk' + Rk);
    X_k = X0 + K_k * (Zk - Hk * X0);
    P_k = (eye(4) - K_k * Hk) * P0;

    X_k1 = state_trans * X_k + Gk * (dP * A * Ts / M + a3_f);
    P_k1 = state_trans * P_k * state_trans' + Qk;
end

X_out = X_k1;
P_out = P_k1;
```

```
end
```

C.4 PVA Kalman filter matlab function in Simulink

```
function [K_out,X_out,P_out,beginout]= Kalman_PVA(Z_measure, X_est,P_est, begin,time,K_prev)
coder.extrinsic('exist')
coder.extrinsic('ss')
coder.extrinsic('c2d')
coder.extrinsic('diag')
coder.extrinsic('inv')
Ts=1e-3;
X_out=[0 0 0 ]';
P_out=zeros(3,3);
Rk=[0];
P0=zeros(3,3);
Qk=zeros(3,3);
K_out=zeros(3,1);
P_out= diag([1e5,1e8,1e11]);
beginout=begin;
if mod(time,Ts)==0
state_trans =  [  1    Ts   Ts^2/2;
             0      1 Ts;
              0   0    1];
Zk=Z_measure;
Ts_constant=1;
X0=[0 0 0]';
W=1e2 %white noise jerk
Rk=[(2.5e-8)]*Ts_constant;
P0=diag([1e2,1e1,1e2])*Ts_constant;
Qk=[W/20*Ts^5 W/8*Ts^4 W/6*Ts^3; W/8*Ts^4 W/3*Ts^3 W/2*Ts^2; W/6*Ts^3 W/2*Ts^2 W*Ts];
Hk=[1 0 0];
K_out=P0*Hk'/(Hk*P0*Hk'+Rk);
```

```
if begin==1 % if we have already estimations for X and P

%5 kalman equations
K_k=P_est*Hk'/(Hk*P_est*Hk'+Rk);
X_k=X_est+K_k*(Zk-Hk*X_est);
%X_k=X_est;
P_k=(eye(3)-K_k*Hk)*P_est;
X_k1=state_trans*X_k;
P_k1=state_trans*P_k*state_trans'+Qk;
K_out=K_k;

else % If we are using initial estimate, just for once.
%Zk, Hk?
%5 kalman equations
K_k=P0*Hk'/(Hk*P0*Hk'+Rk);
X_k=X0+K_k*(Zk-Hk*X0);
P_k=(eye(3)-K_k*Hk)*P0;
X_k1=state_trans*X_k;
P_k1=state_trans*P_k*state_trans'+Qk;
end
X_out=X_k1;
P_out=P_k1;
else
X_out=X_est;
P_out=P_est;
K_out=K_prev;

end
```

C.5 PVA Kalman filter code in TWINCAT

```
FUNCTION_BLOCK PUBLIC KalmanFilter_PVA
VAR_INPUT
rk_variance1 : LREAL;
W: LREAL;
A_cells : ARRAY[0..8] OF LREAL;
```

```
B_cells : ARRAY[0..5] OF LREAL;

END_VAR

VAR_OUTPUT

END_VAR
VAR
    //Rk_Data : ARRAY[1..3,1..3] OF LREAL := [2.5E-10,  0, 0,0, 160,   0,0, 0,     1E5]; //This array will act as the memory for the intial column vector, prepopulated
    //Qk_Data : ARRAY[1..3,1..3] OF LREAL := [1E-12,  0, 0,0, 1E-7,   0,0, 0,     1E-11];
    //Pk_Data : ARRAY[1..3,1..3] OF LREAL :=[1E5,  1E-9, 1E-9,1E-9,    1E7,   1E-9,1E-9, 1E-9,     1E9];
    Ts: LREAL :=1E-3;

    Pk: Array2DStaticMatrix; //This instance is the matrix that the code will interact with
    Pk_Data : ARRAY[1..3,1..3] OF LREAL :=[0,  0, 0,0,    0, 0,0, 0,     0]; //This array will act as the memory for the matrix
    Rk: Array2DStaticMatrix; //This instance is the initial column vector

    //rK first element increase increases affect of Kalman, reduces affect of measurement 5.5E-9,
    Rk_Data : ARRAY[1..1,1..1] OF LREAL := [5.5E-7];
    Qk : Array2DStaticMatrix; //This instance is the resulting column vector                //1e-11
    Qk_Data : ARRAY[1..3,1..3] OF LREAL := [1E-11,  0, 0,0, 1E-9,   0,0, 0,     1E-7]; //This array will act as the memory for the resulting column vector


    Kk: Array2DStaticMatrix; //This instance is the matrix that the code will interact with
```

```
    Kk_Data : ARRAY[1..1,1..3] OF LREAL :=[0,  0, 0];

    D : Array2DStaticMatrix;
    D_Data : ARRAY[1..2,1..2] OF LREAL := [0,0,0,0];
    H: Array2DStaticMatrix; //This instance is the matrix that the code will interact with
    H_Data : ARRAY[1..3,1..1] OF LREAL := [1,  0, 0];

    Xk : Array2DStaticMatrix;
    Xk_Data : ARRAY[1..3,1..1] OF LREAL :=[0,0,0];
    //state matrix
    A : Array2DStaticMatrix; //This instance is the matrix that the code will interact with




    A_T : Array2DStaticMatrix;
    A_T_Data : ARRAY[1..3,1..3] OF LREAL;

    B : Array2DStaticMatrix; //This instance is the initial column vector

    //From matlab
A_Data : ARRAY[1..3,1..3] OF LREAL :=[1,1E-3,EXPT(1E-3,2),0,1,1E-3,0,0,1];
B_Data : ARRAY[1..3,1..2] OF LREAL :=[0,0,0,0,0,0];
END_VAR

// initialize
Pk(Data:=Pk_Data);
Rk(Data:=Rk_Data);
Qk(Data:=Qk_Data);
Kk(Data:=Kk_Data);
Xk(Data:=Xk_Data);
Rk.SetRC(0,0,rk_variance1);

```

```
H(Data:=H_Data);

//state matrix
A(Data:=A_Data);
B(Data:=B_Data);

//set state matrices

Qk.SetRC(0,0,W/20*EXPT(Ts,5));
Qk.SetRC(0,1,W/8*EXPT(Ts,4));
Qk.SetRC(0,2,W/6*EXPT(Ts,3));
Qk.SetRC(1,0,W/8*EXPT(Ts,4));
Qk.SetRC(1,1,W/3*EXPT(Ts,3));
Qk.SetRC(1,2,W/2*EXPT(Ts,2));
Qk.SetRC(2,0,W/6*EXPT(Ts,3));
Qk.SetRC(2,1,W/2*EXPT(Ts,2));
Qk.SetRC(2,2,W/1*EXPT(Ts,1));

//*)
A_T(Data:=A_T_Data);
Matrix_Transpose(A,A_T);
METHOD PUBLIC fb_statespace : LREAL

VAR_INPUT
Xin : ARRAY[0..2] OF LREAL; // states
w : LREAL; // speed
fL : LREAL; // friction force
END_VAR
VAR

    //A : Array2DStaticMatrix; //This instance is the matrix that the code will interact with
    //A_Data : ARRAY[1..3,1..3] OF LREAL :=[1,  -0.0001937,  -1.221E-10,0,     -0.5778,   2.502E-07,0,  -2.522E+06,     -0.581]; //This array will act as the memory for the matrix
    //B : Array2DStaticMatrix; //This instance is the initial column vector
```

```
    //B_Data : ARRAY[1..3,1..2] OF LREAL := [4.78E-08,  -3.096E-07, -9.687E-05,  -0.0004113,615.3 ,     -647.3]; //This array will act as the memory for the intial column vector, prepopulated
    C : Array2DStaticMatrix; //This instance is the resulting column vector
    C_Data : ARRAY[1..2,1..3] OF LREAL := [1, 0,0, 0,1,0]; //This array will act as the memory for the resulting column vector
    D : Array2DStaticMatrix;
    D_Data : ARRAY[1..2,1..2] OF LREAL := [0,0,0,0];
    X_new : Array2DStaticMatrix;
    X_new_Data : ARRAY[1..3,1..1] OF LREAL :=[0,0,0];

    X : Array2DStaticMatrix;
    X_Data : ARRAY [1..3,1..1] OF LREAL := [Xin[0],Xin[1],Xin[2]];

    Ax : Array2DStaticMatrix;
    Ax_Data : ARRAY[1..3,1..1] OF LREAL;

    Bu : Array2DStaticMatrix;
    Bu_Data : ARRAY[1..3,1..1] OF LREAL;

    U : Array2DStaticMatrix;
    U_Data : ARRAY[1..2,1..1] OF LREAL :=[0,0];

    // inits

    read_u1: LREAL;
    read_u2: LREAL;
    read_bu1: LREAL;
    read_bu2: LREAL;
    read_bu3: LREAL;
    read_x1: LREAL;
    successif: BOOL;
END_VAR

VAR_OUTPUT
```

```
X_k1: ARRAY[0..2] OF LREAL;
END_VAR




U(Data :=U_Data);
U.SetRC(1,0,w);
U.SetRC(0,0,fL);
X_new(Data:=X_new_Data);
read_u1 := u.GetRC(0,0);
read_u2 := u.GetRC(1,0);
//A(Data:=A_Data);

Ax(Data:=Ax_Data);
Bu(Data:=Bu_Data);
X(Data:=X_Data);

Matrix_Product(A,X,Ax);
successif :=Matrix_Product(B,U,Bu);

read_bu1 := Bu.GetRC(0,0);
read_bu2 := Bu.GetRC(1,0);
read_bu3 := Bu.GetRC(2,0);
Matrix_ElementSum(Ax,Bu,X_new);
//X_new := Ax;
read_x1 := X_new.GetRC(0,0);

X_k1[0]:=X_new.GetRC(0,0); //return
X_k1[1]:=X_new.GetRC(1,0); //return
X_k1[2]:=X_new.GetRC(2,0); //return
METHOD numeric_div : LREAL
VAR_INPUT
num : LREAL;
denum :LREAL;
END_VAR
```

```
VAR_OUTPUT
    result :LREAL;
END_VAR
IF denum=0 THEN
result:=0;
ELSE
result:=num/denum;
END_IF

//result:=num/denum;
METHOD PUBLIC update5eqs

VAR_INPUT
Yin : LREAL; // measurements, X

END_VAR



VAR
//matrix elements

p11 :LREAL; p12:LREAL; p13:LREAL; p21:LREAL; p22:LREAL; p23:LREAL; p31:LREAL; p32:LREAL; p33:LREAL;
r11 :LREAL;
q11 :LREAL; q12 :LREAL; q13 :LREAL; q21:LREAL; q22:LREAL;q23:LREAL;  q31:LREAL; q32:LREAL;q33:LREAL;

k11: LREAL;  k21:LREAL;  k31:LREAL;
xi1 : LREAL; xi2: LREAL; xi3: LREAL;
z1: LREAL; z2:LREAL; z3:LREAL;


Xk_eq4: ARRAY[0..2] OF LREAL;
Xout: ARRAY[0..2] OF LREAL;
pout11 :LREAL; pout12:LREAL; pout13:LREAL; pout21:LREAL; pout22:LREAL; pout23:LREAL; pout31:LREAL; pout32:LREAL;
```

```
↪ pout33:LREAL;

Pk_inter: Array2DStaticMatrix; //This instance is the matrix that the code will interact with
Pk_inter_Data : ARRAY[1..3,1..3] OF LREAL :=[1E2,  0, 0,0,    1E2,   0,0, 0,     1E2]; //This ar

Pk_AT: Array2DStaticMatrix; //This instance is the matrix that the code will interact with
Pk_AT_Data : ARRAY[1..3,1..3] OF LREAL ; //This ar
A_Pk_AT: Array2DStaticMatrix; //This instance is the matrix that the code will interact with
A_Pk_AT_Data : ARRAY[1..3,1..3] OF LREAL ; //This ar
END_VAR

VAR_OUTPUT
X_display: ARRAY[0..2] OF LREAL;
K_out:LREAL;
END_VAR
//init elements
p11 := Pk.GetRC(0,0);
p12 := Pk.GetRC(0,1);
p13 := Pk.GetRC(0,2);
p21 := Pk.GetRC(1,0);
p22 := Pk.GetRC(1,1);
p23 := Pk.GetRC(1,2);
p31 := Pk.GetRC(2,0);
p32 := Pk.GetRC(2,1);
p33 := Pk.GetRC(2,2);


r11 := Rk.GetRC(0,0);

q11 := Qk.GetRC(0,0);
q22 := Qk.GetRC(1,1);
q33 := Qk.GetRC(2,2);

```

```
xi1 :=Xk.GetRC(0,0);
xi2 :=Xk.GetRC(1,0);
xi3 :=Xk.GetRC(2,0);

//measurements
z1:= Yin;


//init P_k matrix
Pk_inter(Data:=Pk_inter_Data);
Pk_AT(Data:=PK_AT_Data);
A_Pk_AT(Data:=A_Pk_AT_Data);
//update equations
//Kk
numeric_div(p11,p11 + r11,result=>k11);
numeric_div(p21,p11 + r11,result=>k21);
numeric_div(p31,p11 + r11,result=>k31);
//X_k
Xk_eq4[0]:=xi1 - k11*(xi1 - (z1));
Xk_eq4[1]:=xi2 - k21*(xi1 - (z1));
Xk_eq4[2]:=xi3 - k31*(xi1 - (z1));

//P_k
p11:=-p11*(k11 - 1);
p12 := -p12*(k11 - 1);
p13 :=  -p13*(k11 - 1);
p21 := p21 - k21*p11;
p22 := p22 - k21*p12;
p23 :=  p23 - k21*p13;
p31 := p31 - k31*p11;
p32 :=  p32 - k31*p12;
p33 :=   p33 - k31*p13;
//state space est
fb_statespace(Xin:=Xk_eq4,w:=0,fL:=0, X_k1=> Xout);
X_display := XOut;
//P_k1
Pk_inter.SetRC(0,0,p11);
```

```
Pk_inter.SetRC(0,1,p12);
Pk_inter.SetRC(0,2,p13);
Pk_inter.SetRC(1,0,p21);
Pk_inter.SetRC(1,1,p22);
Pk_inter.SetRC(1,2,p23);
Pk_inter.SetRC(2,0,p31);
Pk_inter.SetRC(2,1,p32);
Pk_inter.SetRC(2,2,p33);

Matrix_Product(Pk_inter,A_T,Pk_AT);
Matrix_Product(A,Pk_AT,A_Pk_AT);
Matrix_ElementSum(A_Pk_AT,Qk,Pk);

Xk.SetRC(0,0,Xout[0]);
Xk.SetRC(1,0,Xout[1]);
Xk.SetRC(2,0,Xout[2]);
K_out:=k11;
(*
Pk.SetRC(0,0,pout11);
Pk.SetRC(0,1,pout12);
Pk.SetRC(0,2,pout13);
Pk.SetRC(1,0,pout21);
Pk.SetRC(1,1,pout22);
Pk.SetRC(1,2,pout23);
Pk.SetRC(2,0,pout31);
Pk.SetRC(2,1,pout32);
Pk.SetRC(2,2,pout33);
*)
```

D Matlab code used for QFT design

D.1 Uncertain torque plant

```

beta = 1.355e9;         %[Pa] Bulk modulu
ro      = 860;              %[kg/m^3]    Hidroligin k tlesel yogunlugu
mu      = .0155;                 %[kg/(m*s)]         hidrolik vizkozite
k_T = .73;              %[Nm/A] Motorun tork katsayi
K_T=k_T;
k_E = 0.53;          %[Nm/(rad/s)] Motorun zit EMK kuvveti

T_0 = 11.4;               %[Nm] Stall torque
T_l = 2.7;               %[Nm]  Y  k l   hiz testindeki tork
w_l = 733;              %[rad/s]  Y  k l   hiz testindeki donme hizi
R =0.45; % V*k_T/T_0;          %[ohm] Armatur direnci
L = 2.1e-3;              %[H] Enduktans
J_m = 5.93*1e-4;            %[kg*m^2] Rotor ataleti
J_eq = J_m;             %[kg*m^2] Motor-pompa komplesinin toplam ataleti (Pompanin ataleti cok daha dusuk oldugu icin yoksayilmistir)
b_m = 0.426*1e-2;           %[Nm/(rad/s)] Rotorun sonumleme katsayisi
D_p = 6.6845e-07;     %[m^3/rad] Deplasman
c_eq = b_m + 1*D_p*mu;  %[Nm/(rad/s)] Motor-pompa komplesinin sonumleme katsayisi
A_p = 2572e-6;           %[m^2] Etkin piston alani
V_d = A_p*120e-3*1.3;      %[m^3] Olu hacim (akis yollarini da kapsamasi icin 1.3 ile carpildi)
m_p = 2;               %[kg] Piston kutlesi
c_c = 850;               %[N/(m/s)] Silindirdeki viskoz  s rt nme
C_p = 1/9.6e11;         %[(m^3/s)/Pa] Pompanin ic kacak katsayisi
C_r = C_p;              %[(m^3/s)/Pa] Pompanin dis kacak katsayisi
C_c = 1/5e12;           %[Pa/(m3/s)] Silindir ic kacak katsayisi
C_1 = C_p + C_c + C_r;
```

```
C_2 = C_p + C_c;
C_3 = C_1 + C_2;
C_eq = C_3;
C_d     = .625;                    %[]                  Discharge coef.
stroke=0.12;
%%tork controller PI
TORK_KP  = 11.3;
TORK_KI = 11.3/(.8*1e-3);
% state space with tork controller states are i, w, x ,xdot, pD,
% integral(iref-iactual)
A = [-R/L-TORK_KP/L, -k_E/L, 0,0, 0, TORK_KI/L ;...
    k_T/J_eq, -c_eq/J_eq, 0, 0,  -D_p/J_eq, 0;...
    0, 0,  0,1, 0, 0;...
    0, 0,  0, -c_c/m_p, A_p/m_p,0 ;...
    0, (2*beta*D_p)/V_d, 0, -(2*A_p*beta)/V_d, -(beta*C_eq)/V_d , 0;...
    -1,         0,         0,         0,        0,     0];
Atorque = [-R/L-TORK_KP/L, -k_E/L, 0,0, 0, TORK_KI/L ;...
    k_T/J_eq, -c_eq/J_eq, 0, 0,  -D_p/J_eq, 0;...
    0, 0,  0,1, 0, 0;...
    0, 0,  0, -c_c/m_p, A_p/m_p,0 ;...
    0, (2*beta*D_p)/V_d, 0, -(2*A_p*beta)/V_d, -(beta*C_eq)/V_d , 0;...
    -1,         0,         0,         0,        0,     0];
gainer=2;
A2 = [-R/L-TORK_KP/L*gainer, -k_E/L, 0,0, 0, TORK_KI/L-gainer ;...
    k_T/J_eq, -c_eq/J_eq, 0, 0,  -D_p/J_eq, 0;...
    0, 0,  0,1, 0, 0;...
    0, 0,  0, -c_c/m_p, A_p/m_p,0 ;...
    0, (2*beta*D_p)/V_d, 0, -(2*A_p*beta)/V_d, -(beta*C_eq)/V_d , 0;...
    -1,         0,         0,         0,        0,     0];
% input is iref
B2= [TORK_KP/L, 0, 0 ,0 ,0, 1]';
B3= [0, 0, 0 ,-1/m_p ,0, 0]';
B_noise = [0, 0, 0 ,0 ,0, 1]';
% output is position
```

```
C = [0 0 1 0 0 0;...
    ];
C3= [0 0 0 1 0 0;...
    ];
A_torque=A;
D = zeros(1,1);
%P


%%
[b,a]= ss2tf(A2,B2*gainer,C,D);
%Plant=ss(A,B,C,D);
Phigh_gainT = tf(b,a);
gainer=1;
[b,a]= ss2tf(A2,B2*gainer,C,D);
PnomT = tf(b,a);
Parray={};
[b,a]= ss2tf(A,B2,C,D);
[bF,aF]= ss2tf(A,B3,C3,D);
F_tf = tf(bF,aF);
[bP,aP]= ss2tf(A,B2,C3,D);
P_tf = tf(bP,aP);
P_F_tf{1} = P_tf;
P_F_tf{2} = F_tf;
Parray{1}=tf(b,a);
Pnom = Parray{1};

% -- Parameters: minimum "m", maximum "M", and grid
C_eqm = 1/5e11; C_eqM = 1/5e13; i1m = 5;
D_p_m= 6.6845e-07*0.8;  D_p_M = 6.6845e-07*1.03; i2m = 2;
betam = 9.555e8; betaM = 2.155e9; i3m = 3;
c_cm = 50; c_cM= 900; i4m = 3;
k_Tm = .5920; k_TM = 0.8640; i5m = 3;

% -- Gridding
C_eqv = logspace(log10(C_eqm),log10(C_eqM),i1m)*20;
D_pv = logspace(log10(D_p_m),log10(D_p_M),i2m);
```

```
betav = logspace(log10(betam),log10(betaM),i3m);
c_cv = logspace(log10(c_cm),log10(c_cM),i4m);
k_Tv = logspace(log10(k_Tm),log10(k_TM),i5m);
% -- Plants
c = 1;
for i1=1:i1m
C_eq = C_eqv(i1);
for i2=1:i2m
D_p = D_pv(i2);
for i3=1:i3m
beta = betav(i3);
for i4=1:i4m
c_c = c_cv(i4);
for i5=1:i5m
k_T = k_Tv(i5);
c = c + 1;


c_eq = b_m + 1*D_p*mu;  %[Nm/(rad/s)] Motor-pompa komplesinin sonumleme katsayisi


A = [-R/L-TORK_KP/L, -k_E/L, 0,0, 0, TORK_KI/L ;...
    k_T/J_eq, -c_eq/J_eq, 0, 0,  -D_p/J_eq, 0;...
    0, 0,  0,1, 0, 0;...
    0, 0,  0, -c_c/m_p, A_p/m_p,0 ;...
    0, (2*beta*D_p)/V_d, 0, -(2*A_p*beta)/V_d, -(beta*C_eq)/V_d , 0;...
    -1,         0,         0,         0,       0,    0];
% input is iref
B2= [TORK_KP/L, 0, 0 ,0 ,0, 1]';
% output is position
C = [0 0 1 0 0 0];
D = zeros(1,1);
%P
%Plant=ss(A,B,C,D);
[b,a]= ss2tf(A,B2,C,D);
```

```
Parray{c}=tf(b,a);
end
end
end
end
end
uncertainbode_v2(Parray,1,1,240);

function [] = uncertainbode_v2(Plants, F, G, it)
    hold on;

    % Bode plot options
    opts = bodeoptions('cstprefs');
    opts.PhaseVisible = 'on';
    opts.FreqUnits = 'Hz';
    opts.Title.String = '';
    opts.XLabel.String = 'Input Frequency';
    opts.YLabel.String = {'Magnitude Ratio', 'Phase Difference'};
    opts.Title.FontSize = 12;

    % Pre-define the frequency range
    freqRange = {1*2*pi, 1000*2*pi};

    Plength = length(Plants);
    count = min(it, Plength);

    % Generate Bode plots for each selected plant
    for i = 1:count
        random = randi(Plength); % Randomly select a plant
        pla = Plants{random};

        if G == 1
            sys = pla; % Open-loop system
        else
            sys = (pla * G * F) / (pla * G + 1); % Closed-loop system
        end
```

```
    % Plot the bode diagram
    [mag, phase, w] = bode(sys, logspace(log10(freqRange{1}),
  ↪ log10(freqRange{2}), 1000));
    mag = squeeze(mag);
    phase = squeeze(phase);

    % Plot Magnitude
    subplot(2, 1, 1);
    semilogx(w / (2 * pi), 20 * log10(mag), 'k', 'LineWidth',
  ↪ 1); % Black lines
    grid on;
    hold on;
    ylabel('Magnitude (dB)');
    xlabel('Frequency (Hz)');

    % Plot Phase
    subplot(2, 1, 2);
    semilogx(w / (2 * pi), phase, 'k', 'LineWidth', 1); %
  ↪ Black lines
    grid on;
    hold on;
    ylabel('Phase (degrees)');
    xlabel('Frequency (Hz)');
  end

  hold off;
end



end
```

D.2 Plotting of stability margins

```

```

```
% Open a new figure
figure;
dummy_tf = tf([1 0],[1 0]);
nichols(dummy_tf,1); % Display Nichols chart grid

% Call the function for different Ws values and retrieve GM, PM

[GM2, PM2] = plot_nichols_circle(1.305, 'b'); % Red for Ws = 1.305
[GM3, PM3] = plot_nichols_circle(1.16, 'k');  % Black for Ws = 1.16

% Create dummy lines for the legend
hold on;

h2 = plot(nan, nan, 'b', 'LineWidth', 1.5); % Red line
h3 = plot(nan, nan, 'k', 'LineWidth', 1.5); % Black line
hold off;

% Add the legend with colored lines
legend([h2, h3], ...
       {
        sprintf('Ws = 1.305, GM = %.2f dB, PM = %.2 f  ', GM2, PM2), ...
        sprintf('Ws^v = 1.160, GM = %.2f dB, PM = %.2 f  ', GM3, PM3)}, ...
       'Location', 'best');



grid on;

% Function Definition
function [GM, PM] = plot_nichols_circle(Ws, lineColor)
    % Function to plot a symmetric circle on Nichols chart for a given Ws and line color
    % Returns GM and PM values for use in external legends
    % Inputs:
    %   Ws        - Desired closed-loop gain
    %   lineColor - Line color for the plot (e.g., 'b', 'r', 'g')
```

```
% Parameters
p = 1; % Parameter p
pphase = 0; % Offset phase

% Wide phase range for searching delta_phi
phi_range = linspace(-360, 0, 2000); % Fine phase range for accurate solution

% Preallocate storage for solutions
g_array = nan(length(phi_range), 2);

% Loop to calculate gains for each phase
for a = 1:length(phi_range)
    phi_current = phi_range(a) + pphase; % Current phase
    a1 = p^2 * (1 - 1 / Ws^2); % Coefficient a1
    b1 = 2 * p * cosd(phi_current); % Coefficient b1
    c1 = 1; % Coefficient c1

    % Solve the quadratic equation if the discriminant is non-negative
    discriminant = b1^2 - 4 * a1 * c1;
    discriminant = max(discriminant, 0); % Clamp negative discriminant to 0
    if discriminant >= 0
        g1 = (-b1 - sqrt(discriminant)) / (2 * a1); % First root
        g2 = (-b1 + sqrt(discriminant)) / (2 * a1); % Second root

        % Store the results
        g_array(a, 1) = double(g1); % Store first root
        g_array(a, 2) = double(g2); % Store second root
    end
end

% Filtered solutions
```

```
    g1 = g_array(:, 1);
    g2 = g_array(:, 2);

    % Find the phase where the gains are within 0.1 dB and closest to -180
    gain_diff = abs(20*log10(g1) - 20*log10(g2)); % Difference in dB
    valid_indices = find(gain_diff < 0.1); % Indices where gain difference < 0.1 dB
    [~, idx_closest] = min(abs(phi_range(valid_indices) - (-180))); % Closest to -180
    idx_closest = valid_indices(idx_closest); % Map to original indices
    delta_phi = abs(phi_range(idx_closest) + 180); % Calculate delta_phi as distance from -180

    % Define the phase bounds using delta_phi
    phi_center = -180; % Center phase
    phase_lower = phi_center - delta_phi; % Lower bound
    phase_upper = phi_center + delta_phi; % Upper bound
    filtered_phase_range = linspace(phase_lower, phase_upper, 1000); % Symmetric phase range

    % Preallocate storage for filtered solutions
    filtered_g_array = nan(length(filtered_phase_range), 2);

    % Loop over the filtered phase range
    for a = 1:length(filtered_phase_range)
        phi_current = filtered_phase_range(a) + pphase; % Current phase
        a1 = p^2 * (1 - 1 / Ws^2); % Coefficient a1
        b1 = 2 * p * cosd(phi_current); % Coefficient b1
        c1 = 1; % Coefficient c1

        % Solve the quadratic equation if the discriminant is non-negative
        discriminant = b1^2 - 4 * a1 * c1;
```

```
        discriminant = max(discriminant, 0); % Clamp negative discriminant to 0
        if discriminant >= 0
            g1 = (-b1 - sqrt(discriminant)) / (2 * a1); % First root
            g2 = (-b1 + sqrt(discriminant)) / (2 * a1); % Second root

            % Store the results
            filtered_g_array(a, 1) = double(g1); % Store first root
            filtered_g_array(a, 2) = double(g2); % Store second root
        end
    end

    % Filtered solutions
    g1_filtered = filtered_g_array(:, 1);
    g2_filtered = filtered_g_array(:, 2);

    % Calculate Gain Margin (GM)
    [~, idx_closest_to_minus180] = min(abs(filtered_phase_range - (-180))); % Closest to -180 degrees
    GM = 20 * log10(1 / g1_filtered(idx_closest_to_minus180)); % Gain Margin in dB

    % Calculate Phase Margin (PM)
    valid_indices = find(filtered_phase_range > -180); % Indices where phase > -180
    [~, idx_closest_to_0db] = min(abs(20*log10(g1_filtered(valid_indices)) - 0)); % Closest to 0 dB
    idx_closest_to_0db = valid_indices(idx_closest_to_0db); % Map back to original indices
    PM = abs(filtered_phase_range(idx_closest_to_0db) - (-180)); % Phase Margin in degrees

    % Overlay the filtered symmetric circle
```

```
    hold on;
    plot(filtered_phase_range, 20*log10(g1_filtered), 'Color', lineColor, 'LineWidth', 1.5);
    plot(filtered_phase_range, 20*log10(g2_filtered), 'Color', lineColor, 'LineWidth', 1.5);

    % Add thin guide lines for GM and PM
    plot([-180, -180], [20*log10(g1_filtered(idx_closest_to_minus180)), 0], ...
         '--', 'Color', lineColor, 'LineWidth', 0.5); % GM guide line
    plot([filtered_phase_range(idx_closest_to_0db), -180], ...
          [20*log10(g1_filtered(idx_closest_to_0db)), 0], ...
          '--', 'Color', lineColor, 'LineWidth', 0.5); % PM guide line

    hold off;
end
```

D.3 Matlab function, generation of QFT bounds on velocity controller

```

function [g_array,g_array2] = create_g4_v02vel(w_m,P,Ws,ad_rad)
phi_a=-360: 5: 0;
w_a = merge_frequency_points(w_m);
lengthP=length(P);
N=length(phi_a)*length(w_a)*lengthP;


g_array=(zeros(N,2));
g_array2=(zeros(N,2));

for i=1:length(w_a)
[p1,p1angle]= mag_phase(P{1},w_a(i)*j);
[p2,p2angle]= mag_phase(P{lengthP-1},w_a(i)*j);
w_freq=w_a(i);
```

```
    for a=1:length(phi_a)
        disp([i,a]);
        phi=phi_a(a);
        for jcount=1:lengthP
            %disp([i,a,jcount]);

            [p,pphase] = mag_phase(P{jcount},w_a(i)*j);
            %%% 1
            %type1

            a1=p^2*(1-1/(Ws^2));
            b1=2*p*cos((phi_a(a)+pphase)/180*pi);
            c1=1;
            if ismember(w_freq, w_m{1}) && abs(mod(phi_a(a)+pphase,-360)+180)<52
                g_array((i-1)*(73)*lengthP+(a-1)*lengthP+jcount,1)=double(  (  (-b1)-sqrt(b1^2-4*a1*c1)  )  /(2*a1) );
                g_array((i-1)*(73)*lengthP+(a-1)*lengthP+jcount,2)=double(  (  (-b1)+sqrt(b1^2-4*a1*c1)  )  /(2*a1) );
            else
                g_array((i-1)*(73)*lengthP+(a-1)*lengthP+jcount,1)=NaN;
                g_array((i-1)*(73)*lengthP+(a-1)*lengthP+jcount,2)=NaN;
            end
            %%% 2
            %type3

            T3=tf([1/ad_rad 0],[1/ad_rad,1]);
            [t3,t3phase]=mag_phase(T3,w_freq*j);
            % eqn2=p^2*g^2+2*p*cos((phi+pphase)/180*pi)*g+(1-1/t3^2)==0;
            a1=p^2;
            b1=2*p*cos((phi+pphase)/180*pi);
            c1=(1-1/t3^2);
```

```
                if ismember(w_freq, w_m{2})
                g_array2((i-1)*(73)*lengthP+(a-1)*lengthP
    ↪ +jcount,1)=double(  (  (-b1)-sqrt(b1^2-4*a1*c1)  )  /(2*a1)
    ↪ );
                g_array2((i-1)*(73)*lengthP+(a-1)*lengthP
    ↪ +jcount,2)=NaN; %no below line
                else
                g_array2((i-1)*(73)*lengthP+(a-1)*lengthP
    ↪ +jcount,1)=NaN;
                g_array2((i-1)*(73)*lengthP+(a-1)*lengthP
    ↪ +jcount,2)=NaN;
                end
                %disp(double(  (  (-b1)-sqrt(b1^2-4*a1*c1)  )  /(2*a1)
    ↪  )*p);


                %}


            end
        end
end

end
function [w_general] = merge_frequency_points(w_cell)
dim = length(w_cell);
w_general = w_cell {1};
for i=2:dim
w_general = cat(2,w_general,w_cell{i});
end
w_general = unique(w_general);
end
```

D.4 Matlab function to merge intersection of QFT bounds

```
function [wL,wL2,wL3] = g_restrictor4(w_a,Parray,Pnom,Gcell)

phi_a=[-360: 5: 0];
N=length(phi_a)*length(w_a)*900;
wL={};
wL2={};
wL3={};
lengthP=length(Parray);

for i=1:length(w_a)
    w_freq=w_a(i);
    disp(i);
    disp(w_freq);

    G_phi_min=zeros(1,length(phi_a));
    G_phi_max=zeros(1,length(phi_a));
    Gphases=zeros(1,length(phi_a));
    %[p,pphase] = mag_phase(pmotor,w_freq*j);  %replaced p11 with
    ↪ pmotor% REVERT WHEN USING EHA
    % actually not need p,pphase
    maxphi=0;
    minphi=0;

    for jc=1:length(phi_a)
        [p,pphase] = mag_phase(Pnom,w_freq*j);
        maxb=0;
        minb=1000000000000;
        for k=1:lengthP
            gmin=0;
            gmax=0;
            glist1={};
            glist2={};
            for ig=1:length(Gcell)
                g_array = Gcell{ig};
                glist1{ig} =
    ↪ abs((g_array((i-1)*lengthP*73+(jc-1)*lengthP+k,1)));
```

```
                glist1{ig+length(Gcell)} = abs((g_array((i-1)*lengthP*73+(jc-1)*lengthP+k,2)));

            end
            %disp(size(glist1))
            gmax=max(cell2mat(glist1));
            gmin=min(cell2mat(glist1));

            if gmax>=maxb
                maxb=gmax;
            end
            if gmin<=minb
                minb=gmin;
            end
            bound_min=minb;
            bound_max=maxb;

        end

            G_phi_min(jc)=vpa(bound_min*p);
            G_phi_max(jc)=vpa(bound_max*p);
            Gphases(jc)=mod((phi_a(jc)+pphase),-360);
    end
    wL=[wL,G_phi_min];
    wL2=[wL2,G_phi_max];
    wL3=[wL3,Gphases];
end
end
```

D.5 Matlab code for particle swarm optimization of velocity controller

```

clear;
rng(3);
%% EHA transfer function and plant
EHA_cascade_plant_velTf;
```

```
w_m={ [0.01, 0.1 ,20, 200, 300,500],[]
        };
w_a =merge_frequency_points(w_m);
Ws = 1.16;
ad_rad = 600; % rad/s sensitivity

%%
[g_array1,g_array2]=create_g4_v02vel(w_m,Parray,Ws,ad_rad);
Gcell{1} = g_array1;
Gcell{2} = g_array2;
%%
Gcell = importdata("velSPO Gcell_sens.mat");

%% restrict controller gs

[gmin,gmax,EHAphase] = g_restrictor4(w_a,Parray,Pnom,Gcell);
 g_array1=Gcell{1};
 g_array2=Gcell{2};
%% g1
plot_plainNichols(g_array1,w_a,Pnom,length(Parray))
%% restrict controller gs
Gcont = 1; %default controller
plot_NicholsVel(gmin,gmax,EHAphase,w_a,Pnom,Gcont)


%% Initialization
parameter_count = 3;
% Parameters
K_array={};
Farray={};
iterations =50;
W = 0.9;
C1 = 2;
C2 = 2;
n = 49;
% ---- initial swarm position -----

```

```

mu = 1;
sigma = 35;

for nelor=1:n
    for parts=1:parameter_count
        particle(nelor, 1, parts) = abs(random('Normal',mu,sigma));
    end
end
particle(:, 4, 1) = 100000;           % best value so far
particle(:, 2, :) = 6.9e1;              % initial velocity


%% Iterations

for iter = 1 : iterations

for i = 1 : n
for pcounter =1: parameter_count
particle(i, 1, pcounter) = max(1e-2,particle(i, 1, pcounter) + particle(i, 2, pcounter)/1.3); %update y position
end


for counter=1:parameter_count
if particle(i,1,counter)>1e4
    particle(i,1,counter)=1e4;
end
end

K1 = particle(i, 1, 1);
K2 = particle(i, 1, 2);
K3 = particle(i, 1, 3);

K_array=[K_array,[K1 ; K2; K3]];

% e=Z.SettlingMax;
```

```
% alpha=0.5;

G1 =  pid(K1,K2)*tf([1],[1/K3 1])/k_T;
GQFT = G1;



F =  QFTcostvel(Ws,GQFT,Pnom,EHAphase,w_a,gmin,gmax) ;          % fitness evaluation

if F < particle(i, 4, 1)                    % if new cost is better
for counter=1:parameter_count
particle(i, 3, counter) = particle(i, 1, counter);    % update best x,
end
particle(i, 4, 1) = F;                  % and best value
end
end
Farray =[Farray,particle(i, 4, 1)]; %cost of best positions array
[temp, gbest] = min(particle(:, 4, 1));         % global best position
%--- updating velocity vectors
for i = 1 : n
for counter=1:parameter_count
particle(i, 2, counter) = rand*W*particle(i, 2, counter) + C1*rand*(particle(i, 3, counter) - particle(i, 1, counter)) + C2*rand*(particle(gbest, 3, counter) - particle(i, 1, counter));   %x velocity component

end
if particle(i,2,1)>1e4
    particle(i,2,1)=1e4;
end
if particle(i,2,2)>1e4
    particle(i,2,2)=1e4;
end
end
```

```
%% Plotting the swarm
clf
plot(particle(:, 1, 1), particle(:, 1, 2), 'x')
axis([-1e3 1e3 -1e3 1e3]);
pause(0)
end


%% analyze
lelor=pid(0.43,53)*tf([1],[1/3000 1])/k_T; %vel controller in the
    ↪ driver
bode(1/(1+Pnom*GQFT))
hold on
bode(1/(1+Pnom*lelor))
T3=tf([1/ad_rad 0],[1/ad_rad,1]);
bode(T3)
bode(Pnom*GQFT/(1+Pnom*GQFT))
%%

plot_NicholsVel(gmin,gmax,EHAphase,w_a,Pnom,GQFT)
F =  QFTcostvel(Ws,GQFT,Pnom,EHAphase,w_a,gmin,gmax)
%%
%%
K_arrayPI=cell2mat(K_array);
its_array = 0: iterations/length(K_array):iterations;
for i=1:3
semilogy(its_array(2:end), movmean(K_arrayPI(i,:),2)
    ↪ ,"DisplayName", sprintf("K_%d",i), "LineWidth",2)
hold on
end
xlabel("iterations");
xlim([0,iterations])
ylabel("value");
grid on
grid minor
%%
F_arrayPI=cell2mat(Farray);
```

```
plot(F_arrayPI,'k',"DisplayName","cost","LineWidth",2)

xlabel("iterations");
xlim([0,iterations])
%ylim([-10,1000])
ylabel("value");
grid on
grid minor

%%
Gsens = importdata("D:EHA - Control &
    ↪ Simulation QFT PSO velSPO Gcell_sens.mat");
Gstab = importdata("D:EHA - Control &
    ↪ Simulation QFT PSO velSPO Gcell_stab.mat");
Farray = importdata("D:EHA - Control &
    ↪ Simulation QFT PSO velSPO Farray.mat");
K_array = importdata("D:EHA - Control &
    ↪ Simulation QFT PSO velSPO Karray.mat");
GQFT = importdata("D:EHA - Control &
    ↪ Simulation QFT PSO velSPO PIControllerGOOD.mat");
%%
[gmin1,gmax1,EHAphase1] = g_restrictor4(w_a,Parray,Pnom,Gsens);
[gmin2,gmax2,EHAphase2] = g_restrictor4(w_a,Parray,Pnom,Gstab);
%%
subplot(1,2,1)
plot_NicholsVel_plain(gmin1,gmax1,EHAphase1,w_a,Pnom,GQFT)
grid minor
%%
subplot(1,2,2)
plot_NicholsVel2(gmin2,gmax2,EHAphase2,w_a,Pnom,GQFT)
grid minor
```

D.6 Cost function of the velocity controller

```
function[cost] =
    ↪ QFTcostvel(Ws,Gtf,Pnom,EHAphase,wspan,gmincell,gmaxcell)
cost=0;


for i=1:length(wspan)
    gmax = gmaxcell{i};
    gmin = gmincell{i};
    [p,pphase] = mag_phase(Pnom*Gtf,wspan(i)*j);
    if pphase>0
    pphase=-360+pphase;
    end
    c=EHAphase{i};

    eha_phaseindex =  min(find(min(abs(c-pphase)) ==
    ↪ abs(c-pphase)));
    %disp(pphase)
    if(pphase+180<0)
        cost = cost + abs(pphase+180);
    end

    if ~isnan(gmax(eha_phaseindex))
    Gdiff=abs(20*log10(gmax(eha_phaseindex))-20*log10(p));
    if(gmax(eha_phaseindex)>p)
    cost=cost+Gdiff;
    else
        if wspan(i)>=100

            cost = cost+max(0,Gdiff-2);   %control effort reduction
        end

        if wspan(i) <=1
            cost = cost - Gdiff*0.01;
        end
    end
    %disp(cost)
    end
```

```
end
[mag,phase,wout] = bode(Pnom*Gtf/(1+Pnom*Gtf));
Wsmax=20*log10(max(mag));
if Wsmax>Ws
    cost= cost+abs(Wsmax-Ws)*10;
    %disp(Wsmax)
end
%disp(cost)


% if(length(cost)>1)
%     cost=cost(1);
% end
end
```

D.7 Cost function of the position controller

```
function[cost] = QFTcost(Gtf,Pnom,EHAphase,wspan,gmincell,gmaxcell)
cost=0;
gmargin = 10;
pmargin = 107;
for i=1:length(wspan)
    gmax = gmaxcell{i};
    gmin = gmincell{i};
    [p,pphase] = mag_phase(Pnom*Gtf,wspan(i)*j);
    if pphase>0
    pphase=-360+pphase;
    end
    c=EHAphase{i};

    eha_phaseindex =  min(find(min(abs(c-pphase)) ==
    ↪ abs(c-pphase)));
    if 20*log10(p)>0 && wspan(i)<80
        if(gmax(eha_phaseindex)>p)
```

```
            cost=cost+abs(20*log10(gmax(eha_phaseindex))-20*log10(p));
        end
    else
            if(gmin(eha_phaseindex)<p)  &&  wspan(i)>=100
                cost=cost+abs(20*log10(gmin(eha_phaseindex))-20*log10(p));
            else
                cost=cost-0.02*abs(20*log10(gmin(eha_phaseindex))-20*log10(p)); %control effort reduction
            end
    end

end

%margin test
[gainM,gainPhase] = margin(Pnom*Gtf);
if(max(step(Gtf*Pnom/(Gtf*Pnom+1)))>1.05)
    cost = cost+max(step(Gtf*Pnom/(Gtf*Pnom+1)))-1.05;
end
if(gainM<gmargin)
    cost=cost+gmargin-gainM;
end
if(gainPhase<pmargin)
    cost=cost+pmargin-gainPhase;
end

% if(length(cost)>1)
%     cost=cost(1);
% end
end
```