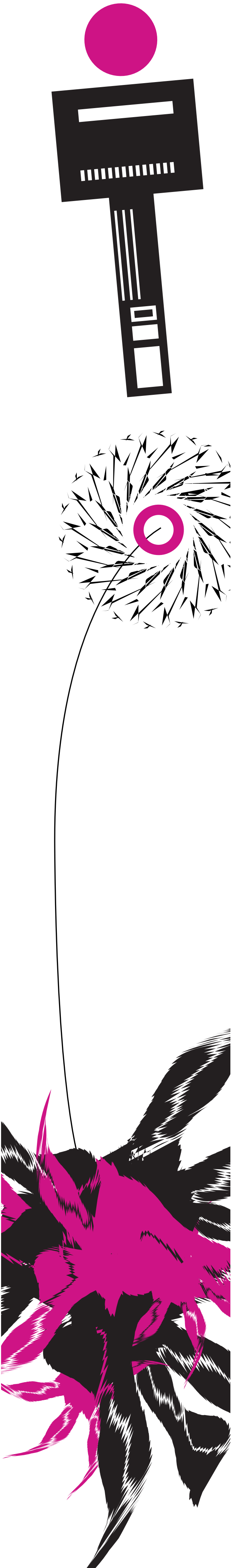


RAM
ROBOTICS
AND
MECHATRONICS

INTERACTION CONTROL WITH A 3D PRINTED TORQUE SENSOR IN AN IMPEDANCE CONTROL FRAMEWORK

P.P. (Pasan) Ottachchige

MSC ASSIGNMENT

Committee:
prof. dr. ir. G.J.M. Krijnen
H.R. Jonkers, MSc
ir. D. Kosmas
dr. J. Dasdemir

January, 2024

003RaM2024
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Summary

In recent years, there has been a notable growth in the utilization of Additive Manufacturing (AM), commonly referred to as 3D printing, as a means to produce a diverse range of sensors. This technological advancement has paved the way for the complete fabrication of sensors through the pure use of AM processes or the integration of sensors within printed structures. As a consequence of these innovations, a great number of applications have begun to materialize, primarily within the realms of soft robotics and biomedical engineering. However, despite the promising prospects associated with embedded 3D printed sensing, a critical challenge that remains relatively unaddressed is the precise integration of these sensors in applications primarily focused on control.

This assignment builds upon the foundational research conducted by the Nature Inspired Fabrication and Transduction program (NIFTy) within the Robotics and Mechatronics (RaM) Research group. The emphasis is placed specifically on the subject of control based on the response of 3D printed sensors. The primary objective of this study is to explore interaction control techniques that enhance the real-world usage of 3D printed torque sensors in control-focused applications. The thesis discusses various interaction control strategies and proceeds to simulate their implementation. Finally, a tangible demonstration setup is constructed to apply the most fitting control strategy in a real-world scenario using the developed torque sensor to explore the potential of interaction control with 3D printed sensors.

In the preliminary phase, modifications were made to the initial sensor design to address the stability concerns in the electrical connection between the sensor and wires. Afterwards, the revised sensor was produced using Fused Filament Fabrication (FFF) technology. This involved the use of two materials: Polylactic Acid (PLA) for the sensor's body and a carbon-doped conductive PLA (ProtoPasta) to form the strain gauges. Subsequently, to gain insights into the characteristics and behavior of the sensor, the sensor was subjected to a sequence of load-unload cycles using a linear actuator within a controlled experimental setup. These cycles induced stress and compression on the sensor's traces, providing a comprehensive understanding of how it responds to varying mechanical conditions during these cycles. Following the data collection, the results were processed including a differential measurement to analyze the sensor's response. This process allowed us to reveal and understand how the sensor behaves under varying conditions, shedding light on its characteristics like non-linearity, hysteresis, drift and repeatability.

After careful consideration of the unique attributes of the proposed 1 degree of freedom (DOF) pendulum system, torque-based impedance control was chosen for implementation in the demonstration setup. The experimentation involved exploring both dynamic and static interaction controls within this setup. The results and insights derived from these experiments are detailed in the final phase of this thesis, providing a comprehensive understanding of how the chosen interaction control method performs with the use of 3D printed torque sensor.

Acknowledgements

I extend my heartfelt appreciation to Professor Gijs Krijnen for providing me with the invaluable opportunity to be part of the NIFTy group and for introducing me to the fascinating realm of 3D printing. Throughout this work, your active supervision and insightful feedback have been instrumental, greatly supporting the completion of this work. Without your guidance, this task would have been more challenging.

I express my sincere gratitude to my daily supervisors, Dimitris Kosmos and Heime Jonkers, for their patience, continuous support, and valuable insights during the execution of experiments and 3D printing processes.

Special thanks to Dr. Janset Dasdemir for agreeing to be a part of the examination committee, and contributing expertise and insight to the evaluation process.

Finally, I am thankful for all the support I received from my friends and family. My gratitude extends to my parents for the sacrifices made in supporting my education and preparing me for the future.

Pasan Ottachchige
Enshede, January 2024

Contents

List of Abbreviations

AM Additive Manufacturing

MM multi material

NIFTy Nature Inspired Fabrication and Transduction program

RaM Robotics and Mechatronics

FFF Fused Filament Fabrication

PLA Polylactic Acid

cPLA conductive PLA

RCC Remote Center Compliance

DOF degree of freedom

LTI linear time-invariant

ODEs ordinary differential equations

GUI Graphical User Interface

EMCL Embedded Motion Control Library

DAQ Data Acquisition

IIR Infinite Impulse Response

PCB printed circuit board

ADC analog-to-digital converter

EOM equation of motion

CoM center of mass

IC integrated circuit

NTI non-linear time invariant

LTI linear time invariant

1 Introduction

1.1 Context

Additive Manufacturing technology has been around since the 1970s, but it's recently gaining popularity, thanks to recent innovations. This has led to the development of advanced manufacturing processes and the introduction of techniques like multi material (MM) Additive Manufacturing, allowing for the utilization of conductive filament-based 3D printed sensors. By blending conductive and dielectric materials within a multi-material 3D printed structure, it becomes possible to measure diverse parameters such as force, pressure, and deformation under various loads. This approach enables a comprehensive understanding of the structure's response to different conditions. Specifically, employing a 3D printed strain gauge proves valuable in this context, as it enables strain measurement. This, in turn, allows for the continuous monitoring of the system under both static and dynamic conditions. This evolution of 3D printing opens up new possibilities for customization and optimization in sensing elements.

3D printed sensors, despite their promise, often exhibit nonlinear characteristics such as creep, drift, and hysteresis. In this study, we delve deeper into these properties to validate the sensor's performance, with a focus on its applicability in interaction control oriented applications. As a result, our literature review primarily centers on interaction control strategies, examining impedance control due to its relevance to the proposed system. Finally, a demonstration setup will be constructed to investigate the application of interaction control strategies using a 3D printed torque sensor in control-focused scenarios.

1.2 Problem Description

This thesis extends the work initiated by R. Heeg in the development of a 3D printed torque sensor using FFF technology [1]. Previous research into the development of this 3D printed single degree of freedom torque sensor based on the piezo-resistive principle showed a closely linear response, within a certain range, particularly when the sensor experienced low torque inputs with a low frequency. However, earlier sensor iterations demonstrated challenging to interpret behaviour, due to their response nonlinearities that are mainly attributed to the inherent viscoelastic properties of the employed materials.

To explore the application of 3D printed torque sensors in real-world interaction control scenarios, a challenge, that needs to be addressed in this research is the impact of nonlinearities observed in the sensor response on control. These nonlinearities pose obstacles to achieving precise and reliable torque measurements, crucial for effective interaction control systems. Furthermore, there is a necessity to improve the mechanical design to address shortcomings identified in earlier iterations. The main focus is on making the sensor's response more reliable. This involves tweaking the mechanical design to create a strong link between the torque sensor and its wiring. By doing so, the goal is to get more accurate and stable torque measurements essential for effective interaction control.

1.3 Goals and Objectives

The primary goal of this research is to explore the potential of 3D printed sensors for interaction control, while simultaneously addressing the challenge of its inherent nonlinear behavior in achieving reliable and precise torque measurements for successful integration into real-world systems. By exploring the possibilities of using 3D printed torque sensors in real-world interaction control scenarios, this research seeks to contribute to the advancement of 3D printed sensor technology.

The project will unfold in two distinct phases. During the initial phase, our primary objectives revolve around refining the mechanical design, fabricating the new sensor iteration, and delving into the characterization of its response across multiple scenarios for understanding the sensor's behavior and identifying any nonlinearities. This phase aims to address the sensor's performance and to lay the groundwork for the subsequent exploration of advanced control strategies and their practical implementation.

The second phase will pivot towards exploring interaction control strategies, with a specific focus on Impedance Control. This phase aims to devise and implement strategies that augment the practical use of the torque sensor in control-oriented applications. To achieve these objectives, following a simulation, a demonstration setup will be established, providing a tangible illustration of the sensor's capabilities in real-world scenarios.

The following main research question will be addressed in this thesis.

- **How can an interaction control strategy be developed for achieving stable, safe operation with a 3D printed torque sensor?**

To answer this research question, it can be broken down into sub-objectives as follows:

1. **Literature Review:**
    - What are the existing interaction control strategies applicable to robotic systems in achieving stable and safe operation?
2. **Method Exploration:**
    - Are there specific methods that have been successfully applied with 3D printed torque sensors?
    - What is the most suitable interaction control method to use with a 3D printed torque sensor?
3. **Simulation:**
    - What performance metrics should be considered during simulations to evaluate stability and safety?
4. **Demonstration Setup:**
    - How can the chosen interaction control method be practically implemented in a real setup using a 3D printed torque sensor?

1.4 Approach

The project starts with a detailed exploration of the literature on interaction control strategies in robotics, commonly referred to as compliance control. The spotlight is particularly directed towards the second phase, where greater emphasis is placed upon interaction control involving 3D printed sensors.

The initial phase targets the improvement of sensor response, emphasizing modifications to the design to ensure robust electrical connections. The fabrication process involves 3D printing the newly developed torque sensor using PLA and conductive PLA (cPLA). Subsequently, the sensor undergoes a characterization process, focusing on addressing its nonlinear behavior through a measurement setup.

The second phase extends from the insights garnered during the literature review on interaction control strategies. This involves a systematic exploration, simulation, and analysis of selected interaction control strategies before transitioning to the design of a demonstration setup. The ultimate objective is to apply the torque sensor in a practical context, specifically in controlling a system considering human interaction. The overarching goal is to craft a control architecture adept at managing system position considering interactions through feedback from these imperfect sensors while mitigating nonlinear behaviors.

Concluding the thesis, the findings are synthesized to draw insights into the feasibility and efficacy of employing 3D printed sensors in real-world control applications. This exploration aims to contribute insights into the evolving landscape of 3D printed sensor integration in control-focused applications in practical scenarios.

1.5 Report Structure

The project unfolds systematically with each chapter commencing with an introduction and concluding with a synthesis of findings. Chapter 1 serves as the project's initiation, outlining its objectives. In Chapter 2, an extensive survey of existing literature is conducted to identify suitable interaction control methods. Within this chapter, the most suitable interaction control method is identified based on the proposed demonstration setup. Although the application of interaction control strategies with the torque sensor in control-focused scenarios is a future work of this project, the method's selection is established here. Chapter 3 delves into the base design of the torque sensor, detailing improvements made to address previous limitations. It also introduces the application that will illustrate the feasibility of interaction control with the torque sensor in real-world scenarios. This will be succeeded by an overview of the fabrication process. Moving on to Chapter 4, a thorough description of the experiments and the subsequent processing of gathered data for sensor characterization is provided. This includes an overview of the measurement setup and an analysis of the resulting data. Chapter 5, the core of the thesis navigates through the system dynamics, control system implementation, and design choices for the selected interaction control method. Simulations of the control system are also discussed. Chapter 6 is dedicated to the application for realizing interaction control, offering insights into the sensor-integrated system setup and presenting results for conclusive observations. Lastly, Chapter 7 sums up the project's entirety, drawing conclusions from results and suggestions for future work are discussed upon.

2 Literature Review

2.1 Introduction to 3D Printing

In recent years, AM technologies have contributed to notable advancements in sensor design and production [2]. Notably, sensors produced through 3D printing techniques are gaining prominence, expanding their applications across various domains. The distinct advantages of 3D printed sensors, in contrast to conventional fabrication methods, are their cost-effectiveness, fast prototyping possibilities, and the ability to create complex geometries with customized functionalities [3, 4]. The precise control over the design parameters of 3D printed sensors enables the creation of tailored sensing elements optimized for specific applications [5]. This allows for the creation of various shapes and functions within a single part, which is not possible with traditional manufacturing. This flexibility also supports the design of compact systems, reducing the size of electronic components and devices. The ability to incorporate fine features, circuitry, and embedded elements contributes to more efficient use of space, resulting in smaller products that require minimal space [5]. This capability not only makes things more efficient and cost-effective but also paves the way for creating and fine-tuning advanced sensing components, bringing a whole new era to sensor technology.

Although 3D printed sensors offer several advantages, challenges exist, including longer manufacturing times and higher economic costs compared to traditional mass production methods. Recent developments in high-speed 3D printing and multi-nozzle systems show promise in addressing these challenges and improving productivity and cost-effectiveness. Additionally, there is a need for 3D printable materials with high performance and compatibility with other structural materials for sensor fabrication, a challenge that advanced materials and techniques like multi-material printing aim to overcome [6].

Schouten et al [7] have conducted a comprehensive review on the 3D printing of sensors, covering various printing technologies, materials, and sensing methods. Additionally, their work presents a list of techniques aimed at addressing challenges related to switching between different materials. The review also highlights efforts in overcoming issues associated with 3D printed sensors, including characterizing the anisotropic conduction of printed conductors, establishing connections to readout electronics, reducing sensor drift, and modeling to subsequently compensate for nonlinear behavior. The findings demonstrate the widespread application of 3D printing. The study also revealed that among 3D printed sensors, piezoresistive sensors offer a simple and cost-effective means of integration, while capacitive sensors provide a path to creating more linear sensors, with added complexity [7].

As for the literature review for this research, there doesn't seem to be evidence of interaction control strategies specifically using 3D printed sensors. While this is within the scope of the current literature search, it's worth noting that existing systems employing traditional force or torque sensors often follow similar control strategies. Hence, in this review, the emphasis has been on exploring interaction control strategies associated with these conventional sensors.

In the context of this thesis, which centres around interaction control (more details on interaction control later), it is crucial to understand the motion of manipulators. Manipulators are one of the most used mechanical devices in various industries, including machinery manufacturing, aerospace, and medicine, where their pivotal role lies in the automation of production processes, contributing significantly to efficiency and precision. The motion of manipulators can be broadly classified into two categories: unconstrained and constrained motion [8]. Most typical and common industrial tasks like painting, welding, and palletizing strictly control both the surroundings and the tasks performed by robots. In such scenarios, manipulators operate without direct contact with the external environment, commonly known as unconstrained or free motion (Figure 2.1). The predominant focus in these situations revolves around position control, given that contact with the environment is unnecessary [8]. Notably, current research in this realm is well-established.

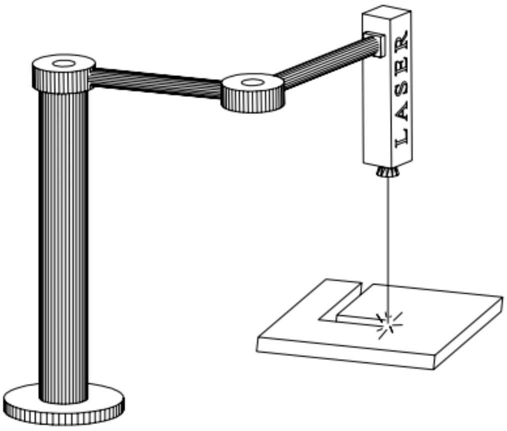


Figure 2.1: A freely moving robot [9].

On the other hand, as technology keeps advancing and robots become more prevalent in various aspects of our lives, it's becoming increasingly important for robots to interact seamlessly with humans and their surroundings. In this scenario, the environment continuously exerts dynamic or kinematic constraints on the manipulator's motion. For complex motion-constrained tasks or unstructured environments where environment knowledge is uncertain, it's crucial for the robot to be able to sense and respond to the forces at play [10, 11]. The conventional paradigm of robotics, which relies on pre-programmed sequences and fixed trajectories, shows limitations when confronted with situations requiring real-time adaptability and responsiveness to uncertain and unpredictable conditions.

Interaction control, addresses the dynamic and complex nature of human-robot and robot-environment interactions [10, 12]. It involves designing and implementing control strategies that allow a robot or autonomous system to dynamically adapt to interact with its surroundings, making decisions based on real-time feedback and sensory information. It encompasses a diverse range of techniques and methodologies aimed at enabling robots to perform tasks collaboratively, safely, and efficiently near humans or within changing environments. Interaction control, generally known as compliance control, addresses these challenges by providing robots with the capability to sense and interpret their environment, respond to external stimuli, and interact in a more natural and human-like manner even when those surroundings are unknown. More specifically, compliance control focuses on ensuring that a robot adjusts to interaction forces smoothly, prioritizing adaptability over resistance to the constraints presented by its environmental contact. Currently in industry, the use of force sensors or torque sensors mounted on robot manipulators to control interactions between the robot and the environment is increasing. An overview of force control of manipulators can be found in [13, 14]. Whitney in [14] refers to explicit control with techniques that prioritize a desired force input over position or velocity inputs. De Schutter et al, [13] have contributed significantly to research efforts focusing on the practical design of interaction control.

One prominent area of application for force/torque sensors with interaction control is in the field of robotics and automation [15]. This integration empowers robots to not only detect but also respond to changes in their surroundings, thereby improving operational safety and efficiency. Another promising area of interest is the development of smart wearables and biomedical devices, where through seamless integration of force sensors, wearables can monitor a spectrum of physiological parameters. Combining these wearables with interaction control, devices dynamically respond to user needs, optimizing both user experience and health outcomes [16].

In summary, the combination of 3D printing technology and interaction control capabilities unlocks exciting possibilities for the design and application of sensors across various industries. As researchers and engineers continue to explore the potential of 3D printed sensors, we can expect further advancements in smart systems, robotics, and wearables, leading to a more interconnected and intelligent world.

2.2 Interaction Control

Numerous control methods and systems have been proposed and developed in order to address the issues associated with regulating compliant motion. This chapter delves into the latest advancements in compliant motion control, tracing the evolution from early theories to current state-of-the-art approaches. While an exhaustive analysis of all compliant motion control methods in the extensive literature would be time-consuming, this chapter focuses on key research and representative approaches. By doing so, it aims to provide a concise exploration of the field's most important developments, acknowledging the substantial growth in recent research endeavors.

Control techniques based on the nature of compliance can be categorized into two distinct groups: passive compliance control and active compliance control [17]. In passive compliance control, mechanical devices are employed to limit interaction forces by either storing or absorbing energy, with examples including springs and damping mechanisms within flexible structures. While this strategy controls the force transmission, a notable drawback is the robot's limited responsiveness to external forces. Moreover, achieving precise position control becomes challenging due to the inherent deformations of elastic components [18]. This can be resolved by active compliance control, which integrates force feedback into the control system, employing sophisticated control algorithms to compensate for both positional and force discrepancies. This approach allows the manipulator not only to actively respond to external forces but also to maintain accurate position control effectively [19].

Both passive and active compliance approaches share the goal of controlling interaction forces arising from contact with the environment. In contrast to the simpler designs of passive methods, active approaches have greater complexity but exhibit versatility across a broader spectrum of applications, adapting swiftly to diverse conditions. While active compliance control methods have garnered recognition in research settings, their integration into industry practices remains limited [18]. Industries tend to address the same interaction challenges by configuring work environments or incorporating mechanical components to achieve passive compliance [17]. The rationale behind this preference is the more robust control offered by passive techniques. However, this reliability comes at the cost of adaptability, as passive solutions may not easily accommodate changes in the environment [18]. This trade-off is illustrated in Table 2.1, which provides a comparative overview of passive and active compliance.

Passive Compliance	Active Compliance
Hardware based	Software based
Application specific	General use
Difficult computation/regulation	Easy computation/regulation
Static compliance	Dynamic Compliance
Complex mechanical structure	Simple mechanical structure

Table 2.1: Active Compliance vs Passive Compliance.

This work will focus on active compliance control methods as we set up a pendulum-based experimental setup for position regulation using 3D printed sensor feedback. Figure 2.2 provides a comprehensive visual guide to traditional compliance control techniques for clarification.

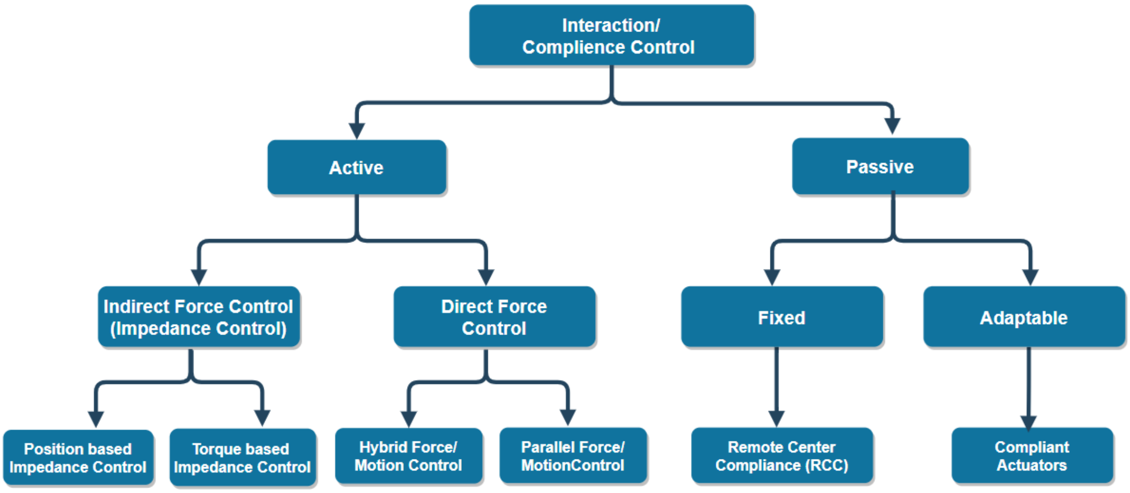

Interaction/
Complience Control
Active
Passive
Indirect Force Control (Impedance Control)
Direct Force Control
Fixed
Adaptable
Position based Impedance Control
Torque based Impedance Control
Hybrid Force/ Motion Control
Parallel Force/ MotionControl
Remote Center Compliance (RCC)
Compliant Actuators

Figure 2.2: General classification of Interaction/Compliance control methods [19].

2.2.1 Passive Compliance Control

In passive interaction control, the trajectory of the robot's end-effector is modified as a result of the interaction forces, due to the inherent compliance of the robot. This compliance can arise from various sources, such as the structural flexibility of the links, joints, and end-effector or inherent compliance in the position servo [18].

The use of compliant actuators proves advantageous in achieving low mechanical output impedance, offering two primary benefits compared to their non-compliant counterparts. The first being that the impedance at frequencies beyond the control bandwidth are dictated by the inherent compliance of the actuator itself [17]. For non-compliant actuators impedance outside of the control bandwidth is influenced by the reflected motor inertia, when gears are involved, the reflected inertia is the product of motor inertia and the square of the gear ratio which is particularly significant in industrial robots [17]. Their second benefit is their force-tracking bandwidth and ability to handle perturbations through their physically compliant elements. It's crucial to acknowledge however, that despite these advantages, compliant actuators do compromise on the precision of positioning compared to their stiffer counterparts [17]. Currently series elastic actuators and pneumatic actuators stand out as the most common compliant actuators in use [19].

When designing robots that adeptly respond to and safely accommodate interaction forces, compliant actuators like series elastic actuators and pneumatic actuators have proven effective [20]. However, this solution has limitations, particularly in industrial settings where robots predominantly use electrical motors without any mechanical compliance. Industrial robots in such environments are intentionally engineered to be rigid for enhanced positioning accuracy, as stiffness directly correlates with precision. In scenarios where the inherent compliance of actuators is not feasible, an alternative approach involves employing an external device attached to the robot's end-effector, such as a Remote Center Compliance (RCC) [21]. This mechanical device addresses inaccuracies in the pre-programmed trajectories' position and orientation. Its unique mechanical design allows it to self-adjust during interaction tasks, making it a cost-effective and straightforward solution for tasks like industrial peg-in-a-hole operations without additional sensors or complex control systems. This solution does however have the drawback of requiring specific design considerations for each task and a constrained effectiveness to managing minor deviations around the desired trajectories.

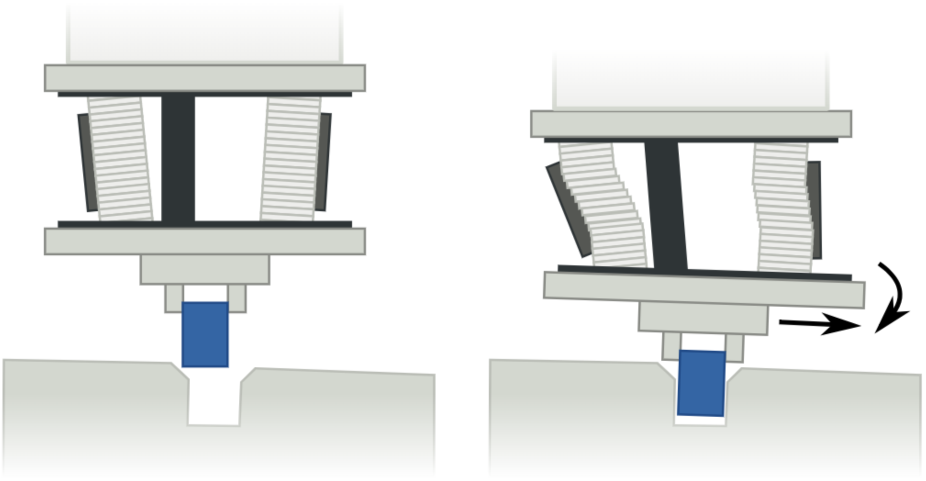


Figure 2.3: Remote Center Compliance in operation [22].

2.2.2 Active Compliance Control

When using active control methods, the generation of precise motion commands for the robot's controller is achieved by assessing the robot's desired behavior. This involves measuring both contact forces, torques and motion, and incorporating this information into real-time feedback for the robot. While active control techniques allow for more flexible design of the controller, this is a trade-off with slower reaction time [18]. The performance of these methods is also influenced by the operating frequency of the controller. Despite their capabilities, these methods also may not always guarantee safe touch interactions, as they heavily rely on sensory feedback signals, which typically become available only after the robot has made contact.

In the past few decades, a surge in research on active methods for precise force control has been observed, with a wealth of available publications. A preliminary summary of the state of the art from the 1980s is provided by [14], while [13] delves into the status of developments throughout the 1990s. Examining the contemporary state of essential sub-components for active compliant systems, recent research is explored in [23]. Furthermore, for a comprehensive understanding of robot force control, the latest edition of the "Handbook of Robotics" includes a dedicated chapter [18].

The essence of active compliance control is to combine motion and force errors, using controllers to optimize inputs for the robot's joint actuators. The control schemes in this field can be broadly classified into two categories: direct and indirect force control. These categories distinguish themselves by the method in which force errors and position errors are integrated.

Direct force control involves a strategy where the manipulator is explicitly commanded to apply a specific force or force profile at the end-effector, closing a force feedback loop. This approach aims purely to control the interaction forces between the manipulator and the environment, making it particularly valuable for tasks requiring precise force application and interaction with delicate or sensitive surfaces. Conversely, indirect force control, commonly known as impedance control [19], operates by establishing a connection between the deviation of the end-effector motion from the intended trajectory and the contact force generated during interaction with the environment. This is achieved through a mechanical impedance framework characterized by adjustable parameters [18]. In simpler terms, when the robot's movement doesn't align perfectly with the intended path due to external factors like contact with surfaces, the impedance control system comes into play. It dynamically adjusts the mechanical properties to modulate the contact force, allowing for a more flexible and adaptive interaction with the environment. Indirect force control finds its niche in scenarios demanding compliant behavior, such as tasks involving human-robot contact, force-based guiding, or secure physical collaboration between humans and robots.

When evaluating interaction control approaches, their performance can be classified into two main categories: static and dynamic. Dynamic model-based control approaches, such as Impedance and Admittance control systems, focus on the system's transient or dynamic response. However, these approaches pose a challenge as they demand a comprehensive dynamic model of the robot, making their design and implementation more complex [17]. Additionally, obtaining force measurements is essential for constructing a linear interaction model. Static model-based control techniques aim to ensure optimal system performance in a steady state [24]. These methods only require knowledge of the manipulator kinematics and gravity generated torques. An in-depth exploration of various dynamic and static model-based interaction control techniques, along with experimental evaluations can be found in [24]. It is shown that in this evaluation, the effectiveness of approaches using dynamic model-based compensation generally outperformed those relying on static model-based compensation.

2.3 Indirect Force Control

Indirect force control schemes, fundamentally operate without the need for measuring contact forces and moments, consequentially the resultant impedance or admittance generally exhibits nonlinearity and coupling [18]. However by incorporating a force-tracking loop alongside the impedance target, the interaction force-tracking issues can be improved [25]. The presence of a force/torque sensor opens the possibility of integrating force measurements into the control scheme, enabling the attainment of a linear and decoupled behavior [18].

Although impedance control schemes are commonly labeled as indirect force methods, impedance control and admittance control share the common goal of actively modifying the robot's mechanical impedance from opposite perspectives [17]. In simpler terms, admittance control assesses force and generates motion, while impedance control evaluates motion and generates force. The fundamental concept behind indirect force control is to establish a dynamic relationship between the interaction force or torque and the position or velocity trajectory of the robot' end-effector.

The precision of the compliance method is dependent on various factors, given the diverse approaches to the control problem. Impedance control's accuracy relies on the precision of position sensors and the bandwidth and precision of force-controlled actuators. Admittance control's effectiveness relies on the force sensors used and the bandwidth and precision of position-controlled actuators. Noting these nuances is crucial, as they directly impact the overall performance and adaptability of the robotic system in achieving the desired mechanical impedance modifications [17].

2.4 Impedance Control

Impedance control establishes a mathematical relationship between interaction forces and reference trajectories, playing a vital role in regulating and stabilizing robot motion [25]. It's primary objective is to manage the forces arising from the coupling between the robot and its dynamic environment by manipulating the mechanical impedance of the robot. Mechanical impedance is defined as the ratio of output force to input velocity or motion whereas mechanical admittance, the inverse of mechanical impedance for linear systems, represents the ratio of input velocity or motion to output force. According to this concept, the robot behaves as an impedance, while the contact environment acts as an admittance. Consider the adaptability humans have when encountering unfamiliar environments; they can adjust their muscular impedance (stiffness) accordingly. Similarly, robots should exhibit flexibility, being soft in rigid surroundings and vice versa. Rigid robots, lacking this adaptability, may encounter challenges in unstructured environments even if they are optimized for precise free motion. Adjusting the stiffness of robots becomes crucial in such scenarios [25], therefore Hogan [10] proposed active impedance regulation based on the bio-mechanics of human motion in both free and confined spaces.

In theory, the control input defining the target impedance relation can take various functional forms. However, it is often implemented as a linear second-order differential equation describing a simple six-dimensional decoupled mass-spring-damper mechanical system. This choice is rooted in the well-known and understood dynamics of second-order systems, impedance control being a technique wherein the robot emulates the behavior of a mass-spring-dashpot system with customizable parameters. Since position control forms the foundation of impedance control, closing the feedback loop requires both position commands and position measurements which are made possible through feedback control and sensors [19].

To illustrate the concept of impedance control, we will investigate a straightforward second-order system (Figure 2.4):

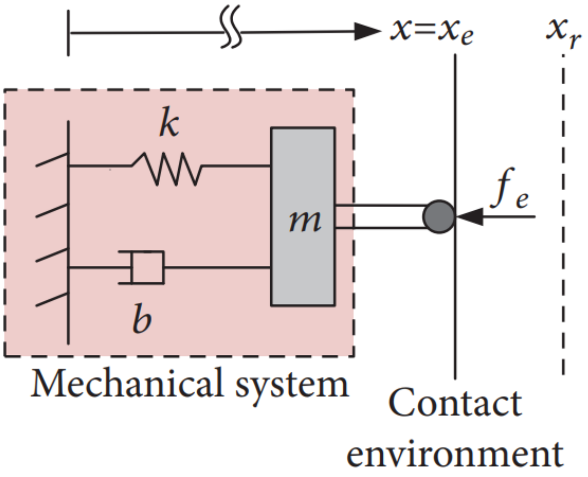


Figure 2.4: The dynamics of the system when in contact with the external environment [25].

$$m\ddot{x} + b\dot{x} + kx = u + f_{\mathrm{e}} \tag{2.1}$$

In this representation, x corresponds to the position of the system mass m, b is the damping coefficient, k denotes the system stiffness, u represents the input control, the variable x_{r} represents the reference equilibrium trajectory, and f_{e} accounts for the external force influencing the system whether it be the interaction contact force or any other external force.

The implementation of impedance control will strive to establish a dynamic correlation between the interaction force and position error. This is achieved by adopting a virtual mass-spring-damper model aligned with the desired trajectory. Consequently, the target impedance function can be represented as Equation 2.2:

$$m_{\mathrm{d}}(\ddot{x} - \ddot{x}_{\mathrm{r}}) + b_{\mathrm{d}}(\dot{x} - \dot{x}_{\mathrm{r}}) + k_{\mathrm{d}}(x - x_{\mathrm{r}}) = f_{\mathrm{e}} \tag{2.2}$$

In the mass spring damper system, $m_\mathrm{d}, b_\mathrm{d}, k_\mathrm{d}$ are positive definite matrices governing the desired target impedance, where k_d is the stiffness matrix, b_d is the damping matrix, and m_d is the inertia matrix (d stands for desired). Adapting the structure of the target impedance dynamics or modifying the behavior of the target impedance coefficients leads to different impedance control strategies. With reference to Figure 2.5, substituting this Equation 2.2 into the closed-loop control system, Equation 2.1 yields the following:

$$(b - m \cdot m_\mathrm{d}^{-1} b_\mathrm{d})\dot{x} + (k - m \cdot m_\mathrm{d}^{-1} k_\mathrm{d})x - (1 - m \cdot m_\mathrm{d}^{-1}) f_\mathrm{d} + m \cdot m_\mathrm{d}^{-1}(b_\mathrm{d}\dot{x}_\mathrm{d} + k_\mathrm{d} x_\mathrm{r}) + m\ddot{x}_\mathrm{r} = u \quad (2.3)$$

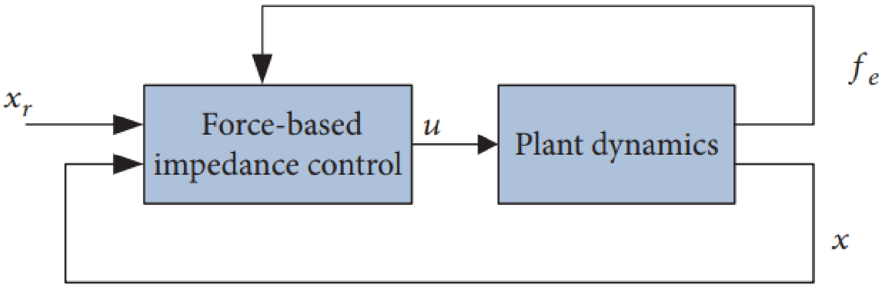


Figure 2.5: General diagram illustrating force/torque-based impedance control. (It is noteworthy that in force/torque-based impedance control, the inclusion of an inner feedback loop is optional and in this example explanation we haven't used an inner force/torque loop) [25].

The feedback controller outlined in Equation 2.3 necessitates the availability of measurements for both the interaction force and the state variables of the end-effector. This requirement can be fulfilled through the implementation of feedback control, which entails the use of both position sensors and force sensors, which relate to our study of using 3D printed sensors for interaction control.

Choosing the target impedance matrices is crucial for achieving varied objectives in a given manipulation task [26]. In scenarios where the environment demands compliance and precise positioning is crucial, opting for high stiffness is a logical choice. Conversely, in directions where maintaining minimal contact forces is a priority, opting for lower stiffness proves more effective. When the task involves energy dissipation, the incorporation of high damping values b_d is required. The mass matrix m_d can be strategically used to induce damped transient behavior, for a controlled system response during contacts [19]. This adaptive approach allows for a strategic adjustment of impedance parameters, addressing the specific requirements of different directions within the manipulation task.

The choice of a control strategy in interaction control is intricately tied to the specific application scenario and its inherent limitations. Unlike a one-size-fits-all solution, there is no universal "off-the-shelf" control strategy. Numerous control systems have been proposed to regulate the relationship between robot motion and interaction forces. In one of the earliest methods introduced by Whitney in 1977 [27], also known as damping or accommodation control, force feedback is closed around the velocity control loop (see Figure 2.6). A constant damping coefficient K_F transforms the contact force into a command for modifying velocity. In a simplified illustration of discrete-time force control, Whitney established the criteria for ensuring system stability during contact [19].

$$0 < T \cdot K_\mathrm{f} \cdot K_\mathrm{e} < 1 \quad (2.4)$$

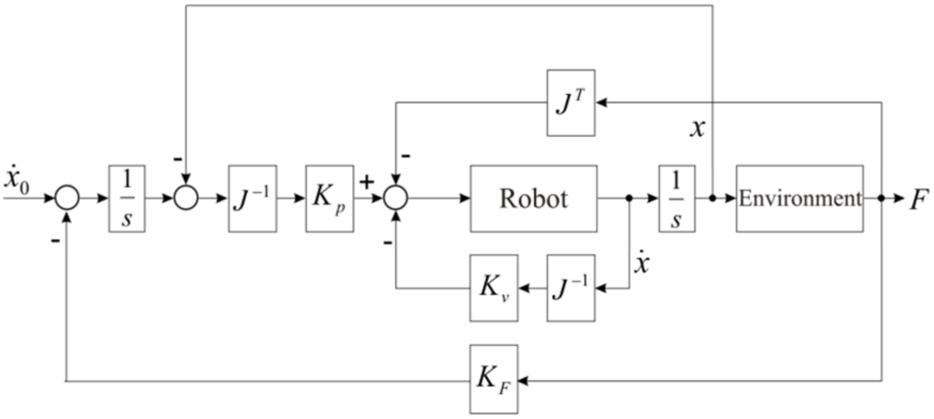


Figure 2.6: Damping control [19].

where T represents the sampling period, K_f is the force control gain acting as a damping coefficient, and K_e denotes the stiffness of the environment. This condition suggests that when K_e is high, the product TK_f should be kept small. To prevent excessive contact forces, achieving a very high sampling rate (small T) becomes necessary. Alternatively, when dealing with the contact of a highly rigid object, Whitney suggested incorporating passive compliance. This addition aims to reduce the effective environmental stiffness K_e,

The majority of impedance control research focuses on electrical manipulators, where the actuator torque is proportional to the input current [28]. For these scenarios, simple force control of the manipulator is possible and internal force loop implementation is simple. Most studies [29], [30], [31] however focused on the position-based application of impedance control. Position-based implementation can prevent the need to redesign the controller because most industrial manipulators already apply a position-based servo controller.

The need for velocity and acceleration feedback into the outer loop significantly increases the complexity of force-based implementation [26]. While it's possible to numerically differentiate the position and velocity signals from the joints' position sensors to obtain these values, this process comes with a drawback. The amplification of quantization noise during differentiation makes the resulting signal challenging to use for control. Alternatives exist however, such as the use of the reference acceleration signal derived from the chosen impedance model to drive the necessary joint forces and the use of a State Variable Filter as a method to obtain the remaining states via integration instead [32].

Salisbury recommends adjusting the end-effector position in response to the contact force, as illustrated in Figure 2.7 [33]. This concept is rooted in the definition of generalized stiffness, expressed as $F = K\delta x$, where δx represents a generalized displacement from the intended nominal end-effector position, and K is a stiffness matrix in six dimensions. In this approach, the transpose of the Jacobian matrix J^T is used to calculate a nominal force by assessing the discrepancy between the desired and actual end positions. This nominal force is then converted into joint torque. Subsequently, the torque error at each joint is determined using this force, and the applied torque is adjusted to ensure the preservation of the desired force at the robot hand [19]. The specific requirements for the elements of the stiffness matrix and their design tailored for particular tasks are discussed in detail in [14].

Although implementing both strategies in a real system is relatively straightforward, achieving high dynamical performance poses a significant challenge resulting from the highly complex dynamics of a typical robot manipulator. The approaches discussed use a simple linear

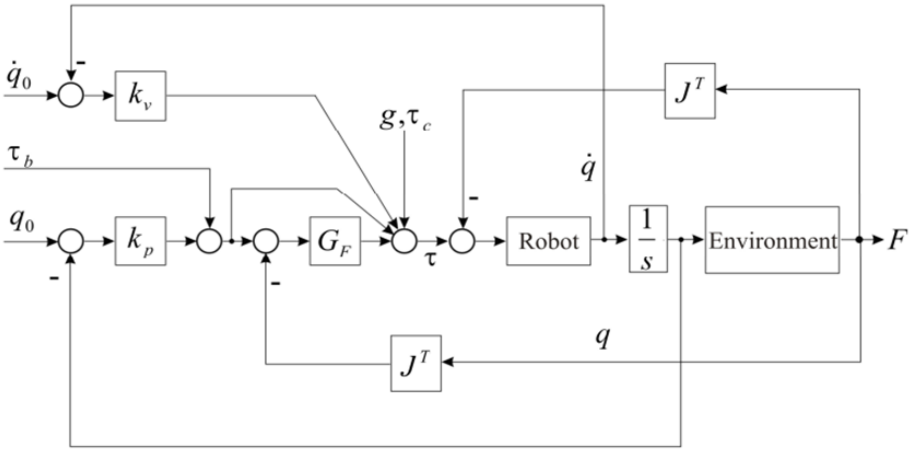


Figure 2.7: Stiffness control [19].

time invariant (LTI) mass-spring-damper target system as a virtual model for the manipulator. While this simplification eases the implementation process, it falls short in accounting for the complex dynamics of the manipulator. One reason is real-world robot manipulators often exhibit nonlinear dynamics, especially in the presence of friction, joint backlash, and other nonidealities. Linear models can fail to capture these complexities.

To eliminate the impacts of nonlinearities on the effectiveness of the impedance control law, these approaches require a control law that will compensate for the appropriate system nonlinearities [19]. A method known as computed torque control has been introduced as a solution to address the inherent nonlinearities within manipulator structures [34]. This approach is a nonlinear control law, which leads to dynamics that are both linearized and decoupled [35]. The efficacy of this method relies significantly on the precision of the manipulator's dynamic model [28]. In 1985, Hogan [10] proposed the widely adopted impedance control idea, grounded in the bio-mechanics of human motion in both free and constrained spaces. Hogan's contribution involves defining a unified theoretical framework for comprehending the mechanical interactions between the robot and its environment. The core concept is to establish a dynamic relationship between the end-effector's reference trajectory and the contact force and torque along each axis. Hogan's impedance control introduces two distinct approaches: namely torque or force-based impedance control and position-based impedance control.

2.4.1 Force/Torque based Impedance Control (Impedance Control)

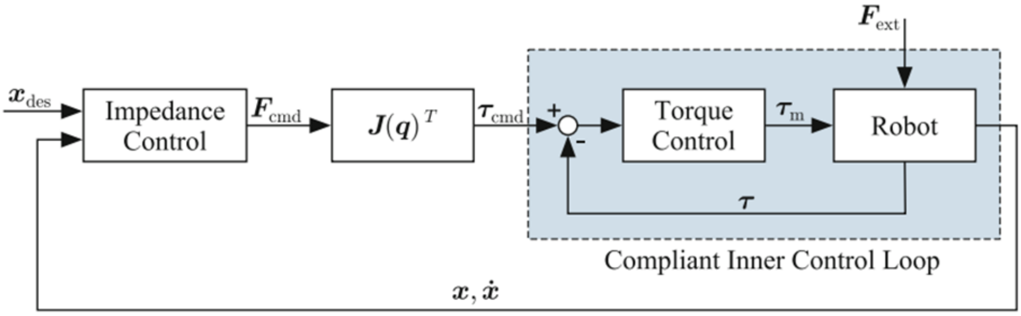


Figure 2.8: Torque/Force-based impedance control scheme [36].

To achieve a linear target impedance behavior, many impedance control algorithms rely on the computed torque approach, aiming to eliminate nonlinearity in robot dynamics. The Force-based impedance control, often simply called "impedance control" in literature, typically has the controller respond to deviations in motion [18]. This method employs two control loops: an optional inner force loop and an outer position loop, also known as the target impedance filter [25]. The controller has the ability to adjust the stiffness of a soft force source. Figure 2.8 provides a general overview of the force/torque-based impedance control approach.

It is however worth noting that this widely used technique involves the computation of a complete dynamic model for the robot's constrained motion, making its implementation quite complex. An important drawback of this method is its sensitivity to uncertainties and variations in model parameters. In industrial robotics, the efforts put into implementing this approach may not proportionally translate into performance improvements that can be achieved [19]. Hogan has also proposed various methods, with and without force feedback, to regulate the end-point impedance of a general nonlinear manipulator [37].

2.4.2 Position-based Impedance Control (Admittance Control)

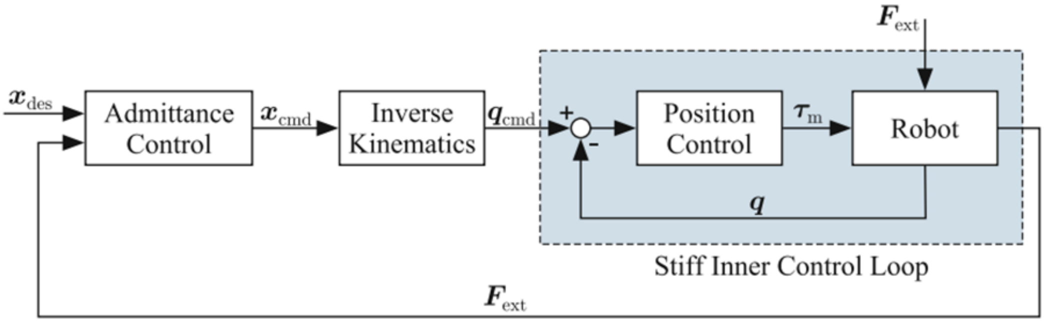


Figure 2.9: Position-based impedance control scheme [36].

By closing a force-sensing loop around the position controller, a straightforward position-mode impedance control can be implemented in commercial robotic systems. This approach, known as position-based impedance control, stands out for its reliability and practicality in industrial robot control systems, as it doesn't require altering the traditional positional controller. In position-based impedance control, the controller, under the influence of contact pressures, introduces a deviation from the intended motion to soften a rigid position source [18,37]. Figure 2.9 provides a general overview of position-based impedance control, featuring two essential control loops: an inner loop for managing compliant position references and an outer loop to establish the desired target impedance dynamics, delivering the instructed compliant references [25]. It's noteworthy that a similar strategy can be applied to velocity-based impedance control by replacing the velocity reference with the desired and commanded position references.

Due to inherent limitations in the precision of position control systems and sensor resolutions, the position-based impedance strategy faces challenges in providing extremely soft impedance [19]. This approach is most effective in scenarios where maintaining stiff and reliable joint control is crucial for achieving high positional accuracy along specific Cartesian directions. It may not be the best choice when the goal is to achieve low impedance, characterized by reduced stiffness and damping while minimizing contact forces [19]. For applications demanding slow motion and encountering low gravitational forces, the force/torque-based technique proves more suitable, given a comprehensive understanding of dynamic models is necessary [19]. Unlike position-based impedance control, force-based control is specifically designed for robotic systems where there is a strong correlation between joint torques and end-effector forces, as seen in direct drive manipulators [19]. This distinction highlights the adaptability of these control strategies based on the specific requirements of different robotic applications.

Summarizing and selecting the most suitable interaction control method for showcasing the potential of 3D printed torque sensors in control-focused applications, a comparative analysis was conducted between torque-based impedance control and position-based impedance control. The goal is to determine the best-suited method that enhances the real-world use of the chosen application setup.

Torque-based Impedance Control	**Position-based Impedance Control**
Ideal for tasks where the robot needs to interact with the environment or humans with a focus on force sensing and compliance [18].	Suited for applications where achieving and maintaining accurate positioning is crucial [18].
Controls interaction forces and adapts compliance.	Controls and maintains accurate positions or trajectories.
More adaptable to uncertain or dynamic environments.	More stable and predictable for tasks with defined trajectories.
Better suited for achieving low impedance (small stiffness and damping) characteristics [19].	Faces challenges in achieving very soft impedance due to limitations in the accuracy of the position control system and sensor resolution [19].
This technique entails the computation of a comprehensive dynamic model for the robot's constrained motion, rendering its implementation rather intricate and complex [18].	Proves to be highly reliable and well-suited for implementation in industrial robot control systems, notably due to its seamless integration without the need for any modification of conventional positional controllers [18].
From a computational perspective, particularly suitable for applications where the influence of manipulator gravity is minimal, and the need for slow motion is prevalent [19].	
Primarily designed for application in robotic systems where there is a relatively good causality between joint torques and end-effector forces, as observed in direct-drive manipulators [19].	
Many impedance control algorithms use this approach to eliminate nonlinearity in robot dynamics [28].	

Table 2.2: Torque-based impedance control vs Position-based impedance control.

2.5 Robot Dynamics

Robot dynamics form the backbone of precise and effective robotic control, offering insights into how these mechanical systems respond to external forces. As we need to explore the dynamics of our robotic system in this thesis, it's crucial to establish a connection between the theoretical principles of robot dynamics and our chosen 1 DOF pendulum system which can also be considered as a simple 1 DOF manipulator. The term "robot" is currently used to refer to autonomous machines. These machines can be broadly categorized as follows:

- Mobile Robots
  - Ground Robots
  - Aerial Robots
  - Aquatic Robots
- Robot Manipulators (Fixed Base Robots)

It is noteworthy that this classification is a general overview, and the robot manipulators can be also integrated into mobile platforms. Examples of this integration can be observed in various rovers and even drones. Mobile robots and manipulators are both integral components of the modern robotics landscape. However, this chapter of this thesis is exclusively dedicated to robot manipulators and their control.

The term "industrial robot manipulator" has seen various interpretations, leading to differences among authors. The International Federation of Robotics, following the ISO/TR 8373 standard, defines a manipulating industrial robot as an automatically controlled, reprogrammable, multipurpose manipulator [9]. This manipulator can either be fixed in place or mobile, designed for industrial automation applications [9]. For the purposes of this thesis, a more specific definition is used: a robot manipulator, or simply a manipulator, is described as a mechanical articulated arm composed of interconnected links. These links are joined through hinges or joints, facilitating relative movement between consecutive links. The joints enable various types of movement, including prismatic (linear), revolute (rotational), or a combination of both.

Despite its apparent simplicity with only one link, the chosen 1 DOF pendulum system aligns with the definition. The motor-driven joint, equipped with a torque sensor, serves as a rotational joint analogous to those found in traditional robot manipulators. This intentional simplification allows for a focused investigation into the core principles of interaction control. By using the proposed 1 DOF manipulator as a model, we aim to gain comprehensive insights into the dynamics and control strategies essential for the integration of 3D printed torque sensors in interaction control applications.

In robotics, the degree of freedom (DOF) refers to the number of independent parameters or ways a robotic system can move. A one degree of freedom (1 DOF) manipulator has only one independent way of motion. For example, In the context of robotics, a rotational DOF manipulator could be, a simple rotational joint or actuator that allows a connected link or arm to rotate around a specific axis. This type of manipulator is often used for tasks where rotation around a single axis is sufficient, such as in some pick-and-place operations or simple assembly tasks.

2.5.1 Joint Space

The variables q_1, q_2, and q_3 represent the joint positions of the robot manipulator in figure 2.10. These positions, defined within an appropriate reference frame, signify the displacements of the robot's joints, which can be either linear or angular. Analytically, in the context of an n-degree-of-freedom (n DOF) robot manipulator, the joint positions and joint velocities are gathered in the vector $\boldsymbol{q}$ and vector $\dot{\boldsymbol{q}}$ respectively.

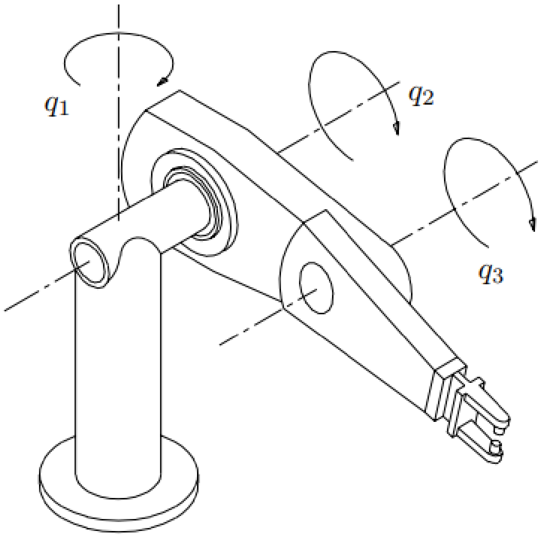


Figure 2.10: A robot manipulator with three revolute joints [9].

$$\boldsymbol{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{bmatrix} \tag{2.5}$$

Each joint in the robot is associated with an actuator. The primary role of actuators is to generate forces or torques, facilitating the movement of the links and, consequently, the overall motion of the entire robot. In analytical terms, these torques and forces are collectively represented by the vector τ

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \vdots \\ \tau_n \end{bmatrix} \tag{2.6}$$

2.5.2 Coordinate Frames

In the field of robotics, accurately determining the positions of objects, sensors, robot joints, and the end-effector in space is crucial [9]. This task is commonly achieved through the use of coordinate frames. To articulate the positions of frames in space, measurements are taken with respect to some other frame [38], often with the origin designated as the base frame. Additionally, each frame is characterized by a combination of its position and orientation in space, commonly referred to as its pose.

2.5.3 Kinematics

In robotics, kinematics is all about establishing a connection between how the joints of a robot are positioned and the resulting pose of its end-effector. The end-effector is a crucial element for the effective use of a manipulator in various applications. Consequently, kinematics plays a vital role in developing robotic control systems and applications. The fundamental approach involves strategically placing coordinate frames along the robot's joints to derive the desired relationship. The translation and rotation connections between these frames are typically expressed using homogeneous transformation matrices.

Determining the position of the end-effector involves solving a set of nonlinear equations, which are then used to translate the configuration space in equation 2.5 into the Cartesian pose. This process, known as forward kinematics, can be defined mathematically as follows:

$$\boldsymbol{x} = f(\boldsymbol{q}) \tag{2.7}$$

Where $\boldsymbol{q}$ is a vector of a set of joint angles, x is the resulting pose of the end-effector.

Once the pose of the end-effector is identified, there is often the need to identify the robot configuration. This is achieved by solving the inverse of equation 2.7, that describes the inverse kinematics.

$$\boldsymbol{q} = f^{-1}\boldsymbol{x} \tag{2.8}$$

2.5.4 Dynamics

In the use of robot manipulators, there exists an array of outputs, usually denoted as $\boldsymbol{y}$, the behavior of which one might aim to control. In the scenario of robots freely navigating in their workspace without external interactions, the output $\boldsymbol{y}$ that requires control may consist of joint positions $\boldsymbol{q}$ and joint velocities $\dot{\boldsymbol{q}}$. However, for robots that physically interact with their environment, the output $\boldsymbol{y}$ may have the torques and forces $\boldsymbol{\tau}$ exerted by the end-effector on its surroundings [9].

Therefore, the resulting output $\boldsymbol{y}$ of a robot system, engaged in a particular class of tasks, typically follows the general form :

$$\boldsymbol{y} = \boldsymbol{q}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\tau}) = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} \tag{2.9}$$

On the other hand, the input variables, those that can be adjusted to influence the output, are the torques and forces $\boldsymbol{\tau}$ applied by the actuators on the robot's joints [9].

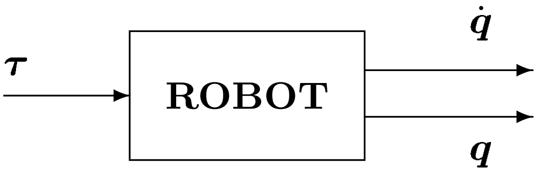


Figure 2.11: Input output representation of a manipulator [9].

At this point, to implement control strategies, one establishes the mathematical model that connects the input variables to the output variables. Typically, this mathematical representation of the system is formulated through ordinary differential equations (ODEs). In essence, the model establishes a connection between the generalized forces or torques (input) applied to the robot and the motion of the robot, depicting how the configuration $\boldsymbol{q}$ changes over time. Various approaches exist, leading to different formulations, yet all converging to equivalent equations [39].

The equations that represent the motion of robots are frequently obtained through Lagrangian mechanics, a formalism grounded in energy principles for deriving dynamic equations [9]. In the case of an n DOF system with joint values $\boldsymbol{q}$ and joint velocities $\dot{\boldsymbol{q}}$, the Lagrangian is defined in Equation 2.10 as the difference between the kinetic and potential energies of the system [9].

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = T(\boldsymbol{q}, \dot{\boldsymbol{q}}) - V(\boldsymbol{q}) \tag{2.10}$$

Here, $T(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $V(\boldsymbol{q})$ denote the kinetic and potential energies, respectively.

By applying Newton's second law within the framework of the Lagrangian, a set of equations known as the Euler-Lagrange equations is obtained.

$$\frac{d}{dx}\left(\frac{\partial L(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}\right) - \left(\frac{\partial L}{\partial \boldsymbol{q}}\right) = \boldsymbol{u} \tag{2.11}$$

Equation 2.10 and Equation 2.11 are key for deriving the dynamic equations governing the motion of a system. The dynamic equations are obtained by evaluating the Euler-Lagrange equations, which involve the generalized joint forces τ, external loads τ_{ext}, and non-conservative generalized forces, including factors like friction. These forces are collectively represented by the vector $\boldsymbol{u}$. One of the key advantages of this method lies in its simplicity for analytically determining both the kinetic and potential energy of the system, contributing to an enhanced understanding of the system's motion and dynamics [36].

2.5.5 General Form of Dynamics

An illustration of an RP (Prismatic-Revolute) manipulator is presented in Appendix A to enrich the understanding of a manipulator system's motion and dynamics. By employing Lagrange and Euler-Lagrange equations, we obtained the force u_2 and torque u_1 exerted on the prismatic and revolute joints of the manipulator, respectively.

$$u_1 = \left(I_1 + I_2 + m_1 {r_1}^2 + m_2 {q_2}^2\right) \ddot{q}_1 + 2 m_2 q_2 \dot{q}_1 \dot{q}_2 + a_g cos(q_1)(m_1 r_1 + m_2 q_2) \tag{2.12}$$

$$u_2 = m_2 \ddot{q}_2 - m_2 q_2 {\dot{q}_1}^2 + a_g cos(q_1) m_2 \tag{2.13}$$

In practical applications, it is often more convenient to express the dynamic model of a robot with more than two links in matrix form. Examining equations 2.12 and 2.13 becomes insightful, as they exhibit much of the interesting structure common to many second-order mechanical systems [40]. On the right-hand side of each equation, there exists a term dependent on the second derivatives of the configuration variables, a term quadratic in the first derivatives of the configuration variables, and a term dependent solely on the configuration variables. These terms can be organised to present the dynamics in the following standardised form [40]:

$$u = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) \tag{2.14}$$

Where $M(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is a symmetric, positive definite mass or inertia matrix with n degrees of freedom, and $C \in \mathbb{R}$ is the Coriolis/Centrifugal matrix, $g(\boldsymbol{q}) \in \mathbb{R}$ is a vector of gravitational forces.

The dynamics outlined in equation 2.14 are particular to mechanical systems where the actuators directly influence the generalised coordinates. Typically in robot arms, each joint is equipped with an actuator. Additionally, in mechanical systems, dissipative forces like dry Coulomb friction or viscous damping are common. These forces are considered external and are incorporated into the equations of motion after deriving them using Lagrange's equations.

While various models for friction and damping exist, these forces typically depend on the generalized velocity $\dot{\boldsymbol{q}}$ and possibly the generalized coordinates $\boldsymbol{q}$. A broader expression capturing the dynamics of second-order mechanical systems is needed to encompass a more general scope [40], leading to a general form written as:

$$u = M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) + F(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{2.15}$$

2.5.6 Dynamics of a Rigid Body

When dealing with robots, it's important to move beyond the concept of point masses, as robots are not point masses. To extend these concepts to the dynamics of a translating and rotating rigid body, a formal definition is necessary. Consider a rigid body that occupies a volume $V \subset \mathbb{R}^3$, where r is a vector from the origin to a point in V, and $\rho(r)$ represents the mass density of the rigid body as a function of location r. The mass of the rigid body is then calculated as the volume integral of the mass density, and the center of mass is determined as the weighted average of the mass density [40].

$$m = \int_v \rho(r)\, dV \tag{2.16}$$

$$r_{CoM} = \frac{1}{m}\int_v r\rho(r)\, dV \tag{2.17}$$

When a rigid body rotates around a fixed axis, its orientation can be described by a single configuration variable, denoted as q. For the analysis, a stationary x, y, z inertial frame can be established, where the z axis aligns with the rotation axis. The kinetic energy of a body rotating in a plane is then computed by integrating the differential kinetic energy at each point $r = (x, y, z)^T$ over the entire body [40]:

$$K = \int_v \frac{1}{2}\rho(r)v^2\, dV \tag{2.18}$$

In the equation above, q represents the angle of the body, $\dot{q}$ denotes the angular velocity, and $v(r) = \dot{q}\sqrt{x^2 + y^2}$ expresses the linear velocity at the point r. Therefore, the kinetic energy can be expressed in the form of equation:

$$K = \frac{1}{2}\dot{q}^2 \int_v \rho(r)(x^2 + y^2)\, dV \tag{2.19}$$

The I_{zz} is the inertia of the body about the z-axis:

$$I_{zz} = \int_v \rho(r)(x^2 + y^2)\, dV \tag{2.20}$$

For which the kinetic energy can be written as:

$$K = \frac{1}{2}\dot{q}^T I_{zz}\dot{q} \tag{2.21}$$

And if the rigid body has uniform density, equation 2.20 can be written as:

$$I_{zz} = m\int_v (x^2 + y^2)\, dV \tag{2.22}$$

If the z axis was chosen to pass through the center of mass of the body and another parallel z' axis at a distance d away, the scalar inertia's are connected by the parallel-axis theorem for planar rotation [40]:

$$I_{z'z'} = I_{zz} + md^2 \tag{2.23}$$

2.6 Conclusion

This chapter comprehensively reviewed the fundamental classical methods for robot interaction control, covering both passive and active approaches. In focusing on active control systems, the chapter delves into various control schemes, with a particular emphasis on impedance control. The discussion highlights the advantages, challenges, and suitable application areas for these schemes. Importantly, the chapter underscores that there is no one-size-fits-all 'universal' interaction controller presently found in the state of the art. The choice among the presented alternatives depends on a careful analysis of factors such as the available information about the environment, computational complexity, and the feasibility of modifying the internal position controller of the robot. Moreover, this chapter provides an in-depth exploration of the theoretical foundation essential for realizing the dynamic equations of motion in the context of a basic mobile manipulator robot. This understanding serves as a crucial prerequisite for effectively controlling the 1 DOF pendulum system.

In conclusion, the selection of the torque-based impedance control scheme for the proof of concept in this project stems from a careful consideration of the unique characteristics of the setup (as discussed in Chapter 6). The 1 DOF pendulum system, actuated by a Mechaduino [41] controlled actuator, and equipped with the 3D printed torque sensor directly connected to the motor shaft, presents a scenario where force/torque sensing and compliance are of utmost importance. The system's lightweight nature and pendulum like behavior, introduce nonlinear dynamics that torque-based impedance control can effectively address. This method is well-suited for tasks requiring interaction with the environment or humans, offering adaptability to uncertain or dynamic conditions. In comparison to position-based impedance control, torque-based control excels in achieving low impedance characteristics, making it an ideal choice for this application. The decision is further supported by the computational advantages in scenarios where manipulator gravity is minimal, aligning with the prevalent slow-motion requirements of the system. Overall, torque-based impedance control proves to be an effective strategy for realizing the potential of 3D printed torque sensors in interaction control, providing a robust and adaptable solution for the proposed demonstration setup.

3 Sensor Development

3.1 Introduction

This chapter serves to provide an overview of sensor design and fabrication. This chapter initiates with the introduction of the torque sensor designed by R. Heeg, which serves as the base model for this project [1]. The changes made to optimize the results of the existing design are then outlined, followed by the fabrication procedure of the sensor. The sensor will then be tested experimentally and the characterization of the non-linearities of the 3D printed sensor will be discussed. Finally, it introduces the sensor design fabricated for the demonstration setup.

3.2 Previous Design

The CAD model displayed in Figure 3.1a represents the sensor designed by R. Heeg, serving as the basis of our redesign [1]. This design draws inspiration from the work of Lou et al. [42], who detailed a one-dimensional torque sensor used in robot collision detection. The sensor comprises an outer and an inner ring subjected to applied torque. These rings are interconnected by four equally spaced spokes or beams, as depicted in the Figure 3.1. Application of torque induces rotation along the central axis of the sensor, perpendicular to its surface, leading to strain in the beams and subsequent deformation.

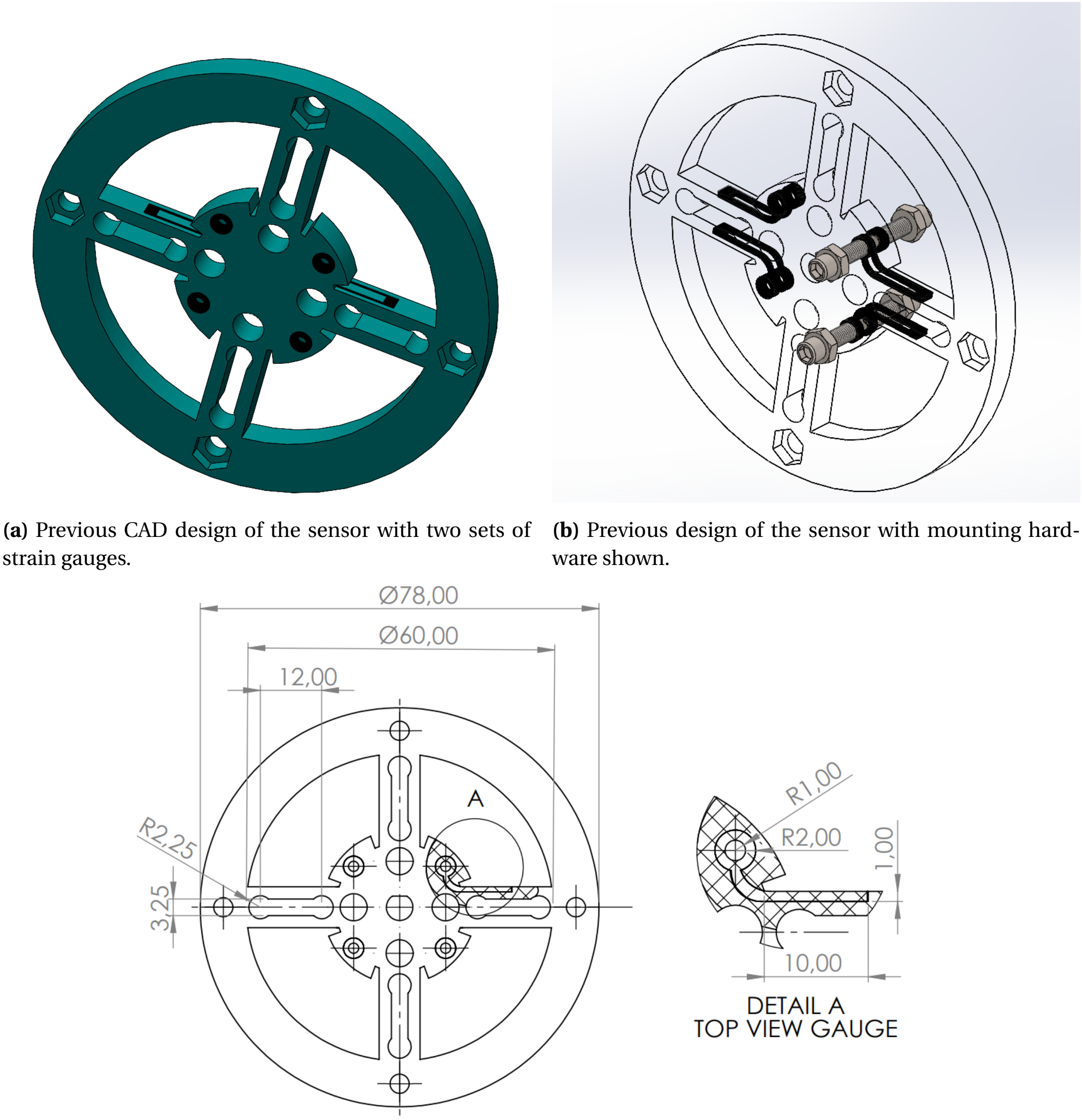
As depicted in Figure 3.1a, two sets of strain gauges are positioned on each side of the model from the centre. Each set of gauges consists of two gauges located on opposite sides of a spoke. Achieving a systematic arrangement involves placing the strain gauges in a mirrored fashion across the neutral axis of each beam, ensuring that one gauge experiences tension (resulting in positive strain), while the other undergoes compression (resulting in negative strain) when a torque is applied. This symmetrical configuration is essential for a differential measurement. Several studies have shown that using differential measurements improve the linearity of 3D printed sensors [43,44]. Additionally, such measurements help mitigate the impact of temperature changes on the sensors. Screws are strategically placed in designated holes (depicted in black) on both sides of the sensor. These holes, constructed from the same conductive material as the gauges, provide a method for establishing the necessary electrical connections between the strain gauges and the wiring system, facilitating the readout of measurements. For a more detailed view of the strain gauge connections, refer to Figure 3.1b.

3.3 New Sensor Design

The objective for the redesign of the current sensor is to retain the current design while addressing determined shortcomings. As outlined in the introduction chapter, the redesign aims to improve the dynamic range, accuracy, and repeatability of the sensor response compared to the previous design. These improvements are important for the ultimate objective, which is to employ the sensor in an interaction control scheme, given its response is a critical input to the controller. Therefore, ensuring a stable sensor response is important for the effective integration of the sensor into the control system.

3.3.1 Shortcomings of Previous Design

The previously designed sensor configuration involves establishing the connection between wires and the conductive surface using screws, nuts, and washers to secure the wires in place, as illustrated in Figure 3.1b [1]. This approach allows for a modular connection facilitated by a bolt that fastens the wires. However, a drawback of this method is the limited depth to which the screws penetrate into the designated holes connecting to the strain gauges. This slight



(a) Previous CAD design of the sensor with two sets of strain gauges.

(b) Previous design of the sensor with mounting hardware shown.

(c) CAD drawing of the previous sensor with main dimensions in mm.

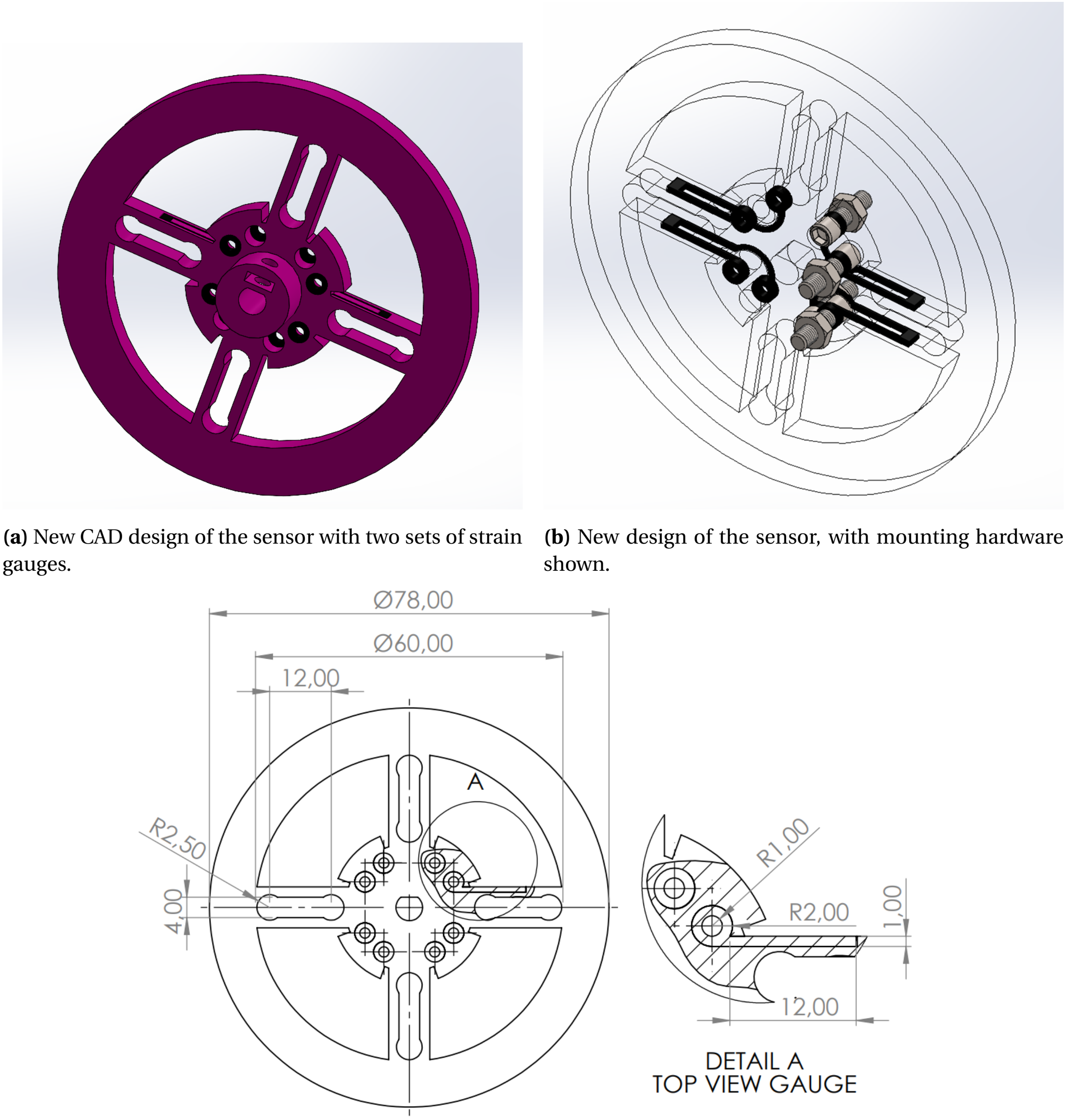
Figure 3.1: Previous design of the sensor.

penetration was found to result in screws potentially loosening and dislodging due to vibrations and movements and also increases the risk of contamination in the measurement signal as the electrical contact resistance incurs variations due to mutual movements between the screws and the sensing structure. A more robust connection design is required, which will be addressed in the redesign of the sensor.

3.3.2 Improvements

The primary objective for the new design was to ensure stable connections between the sensor, wires, and printed contact traces. Consequently, adjustments were made to the design to allow screws to pass through designated holes. However, it was crucial to maintain an adequate gap between the screws to prevent shorting.

Secondly, given that the experiments conducted with this sensor involved smaller loads, the thickness of the beams was also reduced to enhance sensitivity to these smaller loads. This can be seen clearly with the dimensions of the previous sensor design and new sensor design as



(a) New CAD design of the sensor with two sets of strain gauges.

(b) New design of the sensor, with mounting hardware shown.

(c) CAD drawing of the new sensor with main dimensions in mm.

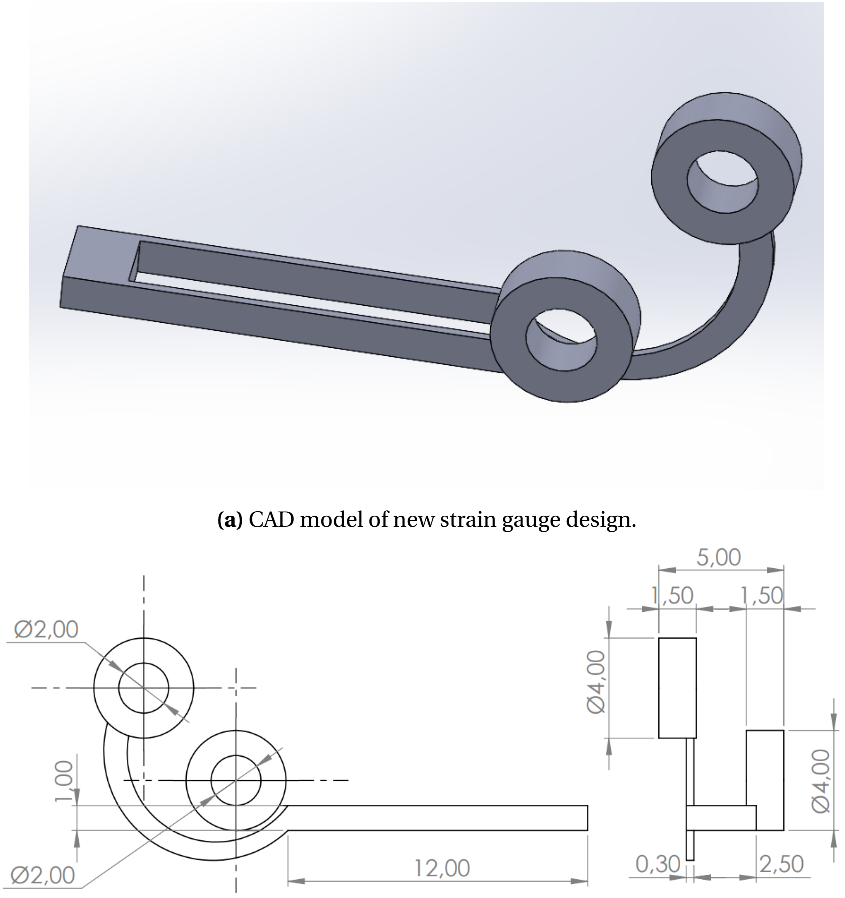
Figure 3.2: New design of the sensor.

illustrated in Figure 3.1c and Figure 3.2c The sensitivity of a strain gauge is influenced by the amount of strain the material experiences in response to an applied load, thinning the material leads to more deformation for the same load, resulting in a higher strain.

The sensitivity (S) of a strain gauge sensor is expressed as the ratio of the relative change in electrical resistance ($\Delta R/R$) of the strain gauge to the applied strain (ε):

$$S = \frac{\Delta R/R}{\varepsilon} \tag{3.1}$$

Strain is related to the change in length per unit length, and for a beam under bending, the strain is influenced by the beam's geometry. For a simple cantilever beam, the bending strain ($\varepsilon_{\text{bend}}$) is proportional to the applied moment (M) and inversely proportional to the thickness qubed (h^3). In these cases, the moment of inertia (I) of the beam's cross-section, which appears in the denominator of the formula for bending strain, is proportional to h^3:



(a) CAD model of new strain gauge design.

(b) CAD drawing of the modified strain gauge. Dimensions in [mm].

Figure 3.3: New design of the strain gauge.

$$\varepsilon_{\text{bend}} \propto \frac{M}{E \cdot h^3} \tag{3.2}$$

Therefore, if we decrease the thickness of the beam while keeping other factors constant, the bending strain will increase, resulting in a higher sensitivity of the strain gauge attached to the beam. The redesigned sensor is shown in Figure 3.2a. The modifications to ensure a robust connection can be seen in Figure 3.2b.

Emphasizing the importance of symmetry of individual strain gauges, the length of the strain gauges along the beam was also kept consistent. The design, as illustrated in Figure 3.3, reflects the adjustments made to these elements.

3.4 Sensor Integrated System

To evaluate the performance of the developed torque sensor, it was integrated into a mechanical system. A simple starting point of a 1 DOF single pendulum system with an integrated torque sensor was selected. This initial and straightforward setup fulfills the project's objectives of evaluating the sensor's performance in interaction control. By fixing the sensor directly to the rotating shaft of the motor, the accuracy was improved and outside influences were reduced, as this arrangement isolates the torque from the motor and minimizes parasitic forces.

The mechanical system was specifically designed in a way such that it can be actuated by a stepper motor. This motor was controlled by a Mechaduino board, which allowed for torque-controlled excitations, aligning with the initial project concept. The modified sensor design, illustrated in Figure 3.4, features one end that can be directly connected to the motor shaft, while the other end remains free. Some additional holes for attaching bolted weights were placed towards the open end to enhance the pendulum-like dynamics, potentially increasing the overall torque required at the rotation point. The anticipated interaction involves physically interrupting the system's movement at the far end, like in real-world scenarios where external forces, such as a hand or an obstacle, affect the free motion of the system.

This choice, emphasizing a balance between simplicity and effectiveness in the mechanical design, brings out advantages that enhance the capabilities of the entire setup. To name a few:

1. **Minimal Mechanical Interference:** By fixing the pendulum directly to the rotating shaft, there are fewer mechanical components and linkages, reducing the chances of mechanical issues.

2. **Improved Dynamic Response:** The direct connection enhances the dynamic response of the system. It reduces delays and offsets that could arise from additional mechanical elements such as gears (backlash), enabling quicker and more accurate motion control.

3. **Simplified Control Algorithm:** A direct connection simplifies the control algorithm, particularly in the modelling phase. With fewer intermediate components, the control system can focus on the essential dynamics of the pendulum.

4. **Reduced Mechanical Compliance:** Direct attachment minimizes mechanical compliance, ensuring that the pendulum responds more precisely to control inputs. This is particularly advantageous when implementing control strategies.

5. **Increased Precision in Torque Application:** The direct connection allows for more accurate torque application from the stepper motor to the pendulum. This precision is crucial when studying and implementing interaction control methods, as it provides a reliable and consistent input.

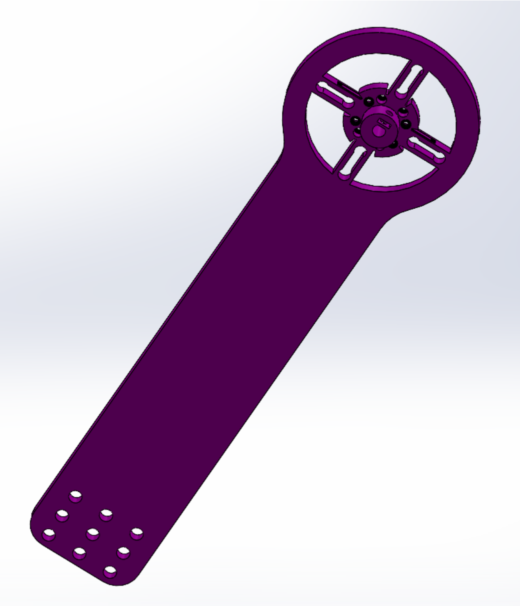


Figure 3.4: CAD model of 1 DOF pendulum like system with an embedded torque sensor.

3.5 Fabrication

The primary equipment used for fabrication includes:

1. Printing Materials :
    - Polylactic Acid :

      PLA is environmentally friendly 3D printing material [45, 46]. It has a printing temperature of around 210 °C which is lower compared to many other filaments, which makes it compatible with a wide range of 3D printers [47].
    - ProtoPasta conductive PLA [48] :

      cPLA is a type of 3D printing filament designed to have a finite electrical conductance. It is a variation of traditional PLA filament where conductance is promoted by incorporating conductive materials, such as carbon black, into the PLA filament.
2. Diabase H5 3D printer :

   This multi-material 3D printer, is part of the Diabase H-series. It enables simultaneous printing with up to 5 filaments [7]. The filament switching mechanism operates based on a turret configuration, which by design places idle nozzles at a safe distance from the printing platform. For the fabrication of the sensor, two of the 5 nozzles were used.

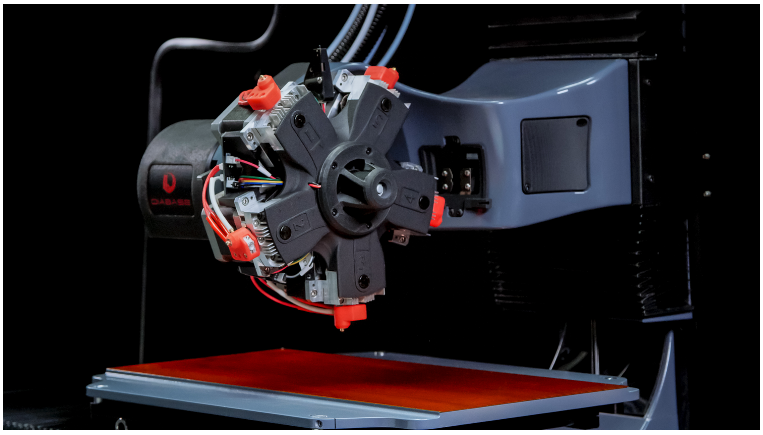


Figure 3.5: Diabase H-series 3D printer [49].

3. PrusaSlicer software :

   PrusaSlicer is a 3D printing slicer developed and maintained by Prusa Research, the company behind the Prusa line of 3D printers [50]. A slicer is a crucial component in the 3D printing workflow that converts 3D volume models into machine instructions (G-code) that the 3D printer can understand and execute.

The design featuring the integrated sensor was successfully printed using the parameters outlined in the Table 3.1, as illustrated in Figure 3.6.

The printing process began with default settings, producing decent prints. However, some warping issues were noticed with initial prints, and an increase of the first layer thickness to 300 µm, resulted in better prints without warping. While the altered layer thickness may have contributed to the improvement, it became apparent that bed levelling also might have played a role. Because some parts of the print were a bit off-centre, imperfect bed levelling may potentially have caused warping. This issue was not apparent with the circular sensor printed at the center of the bed but was noticeable with the sensor integrated pendulum model. Additionally, variations in layer thickness were assessed to evaluate their impact on print quality, resulting in visibly improved prints with lower layer thicknesses but at the expense of increased printing time. As a compromise between print quality and time efficiency, a 200 µm layer thickness was

Parameter	Value
Layer Settings	
Layer thickness	200 µm
First layer thickness	300 µm
Infill	40%
Temperature	
PLA	210 °C
ProtoPasta	225 °C
Bed temperature	60 °C
Printing speed	
PLA	60 mm s^{-1}
Protopasta	15 mm s^{-1}

Table 3.1: Printing parameters.

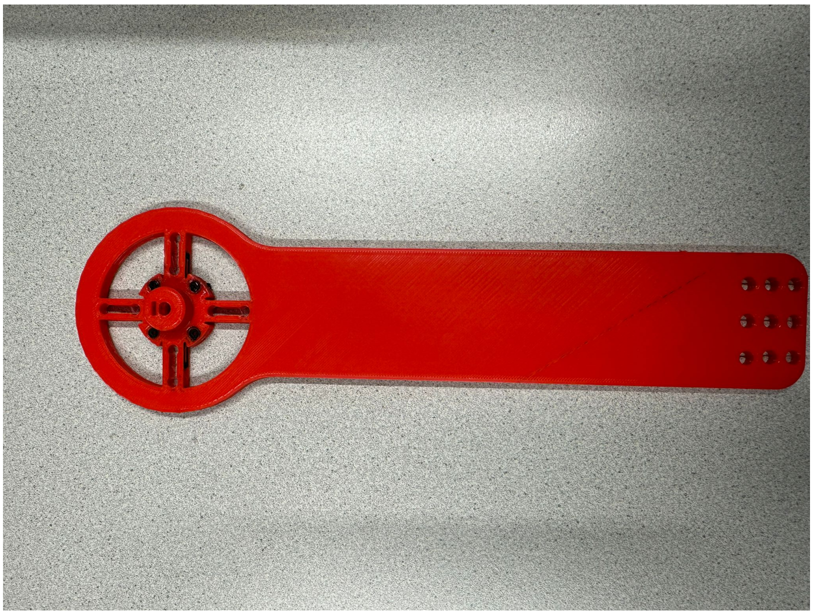


Figure 3.6: 3D print of the sensor integrated 1 DOF pendulum system.

selected. Printing the sensor integrated pendulum faced challenges with nozzle offsets and bed levelling, causing gauge misalignment. We resolved this by recalibrating the nozzles for proper alignment.

According to Abeykoon et al. [51], increasing infill density contributes to the improvement of PLA material's tensile strength. Additionally, according to Naranag and Chhabra [52], the highest tensile strength is achieved when utilizing a rectilinear pattern with 100 % infill density. Given the changes to the sensor design, such as reduced beam thickness can cause the beams to become weaker. Therefore, the infill was increased and a rectilinear infill pattern was chosen. Figure 3.6 provides a visual representation of the successfully fabricated design.

3.6 Conclusion

In summary, the initial design was modified to ensure a stable connection between the strain gauges and readout wires during sensor excitation. Additionally the strain gauges were redesigned, emphasizing the need to maintain symmetry along the beam for a reliable differential response. The refined design was then fabricated through FFF, resulting in the successful fabrication of the new sensor design.

A 1 DOF pendulum-like system with an integrated sensor was chosen as the experimental platform for exploring interaction control methods. This integrated design, containing the printed sensor, was successfully printed as shown in the Figure 3.6. Various printing parameters were also explored to optimize both the functionality and quality of the print.

Improving both the strength and quality of the prints can be accomplished by tweaking the printing parameters. In this study, these specific printing parameters were carefully selected to ensure the accurate translation of the 3D model into a physical object by starting with the default settings recommended in the literature and tweaking printing parameters by the visual inspection of the print.

4 Sensor Characterisation and Validation

4.1 Introduction

This chapter describes the characterization of the printed torque sensor. The measurement setup used for the characterization is explained, including details about all equipment in the configuration. The subsequent sections delve into the analysis and post-processing of the acquired data. Additionally, measurement protocols and the readout techniques are thoroughly explored. During characterization, aspects such as accuracy, non-linearity, hysteresis and repeatability are assessed.

4.2 Experimental setup

With the fabricated sensor, the next step is to establish an experimental setup that allows us to assess the sensor's behavior and performance. This work was performed in the NIFTy laboratory, with specialized equipment to measure voltage changes of the strain gauges of the sensor under varying excitation. The core components of this setup are:

1. 3D printed torque sensor.
2. DEWE-43A Data Acquisition System (DAQ) [53].
3. Power Supply Unit 1 V to 30 V.
4. SMAC linear actuator LCA25-050-15F [54].
5. LCMFD-50 N load-cell.
6. A PC with the required software.
   - SMAC controller software.
   - MATLAB.
   - DewesoftX.

4.2.1 Fabricated Sensor

The fabrication process was already explained in Chapter 3. Figure 4.1 below illustrates the successfully 3D printed torque sensor as used in the characterization setup.

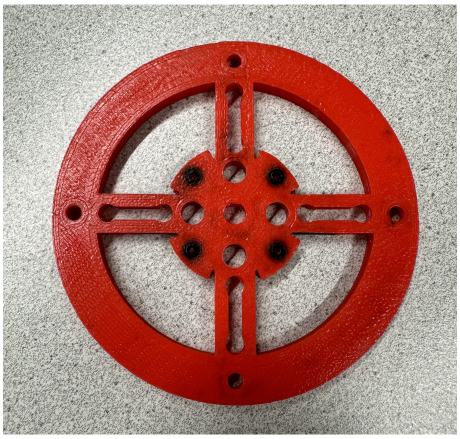


Figure 4.1: Fabricated sensor for characterisation setup.

4.2.2 DEWE-43A Data Acquisition System (DAQ)

The DEWE-43A, serves as a data acquisition system for capturing and processing the relevant signals. To read the signals of the strain gauges and the load cell, three of eight analog input ports of the DEWE were used. These inputs were time-synchronized within the DEWE, delivering a maximum resolution of 24 bit and a peak sampling rate of 200 kS. Furthermore, It is worth noting that the DEWE offers the flexibility to use various filters, and for this experiment, an anti-aliasing filter was used. Additionally, a digital counter port is used to reconstruct the actuator position from the extracted encoder pulses.

Figure 4.2: DEWE-43-A DAQ [55].

4.2.3 SMAC linear actuator

The linear actuator used in this experimental setup is the SMAC LCA25-050-15-F [54], a solenoid-based actuator featuring a 50 mm stroke and a peak force of 15 N, operating at 24 V. This multipurpose linear actuator supports both position and force control. The actuator is controlled by the LCC-10 controller [56]. The controller uses an Embedded Motion Control Library (EMCL) to oversee all SMAC functions [57]. The measurement PC communicates via a USB serial interface, offering multiple operational choices. A Graphical User Interface (GUI) is available for ease of use, or alternatively, users can manually establish communication channels and provide specific commands [57]. For experimental purposes, a combined approach is adopted, using the provided interface for actuator initialization and subsequently custom scripts for either position or force control implementations run through MATLAB.

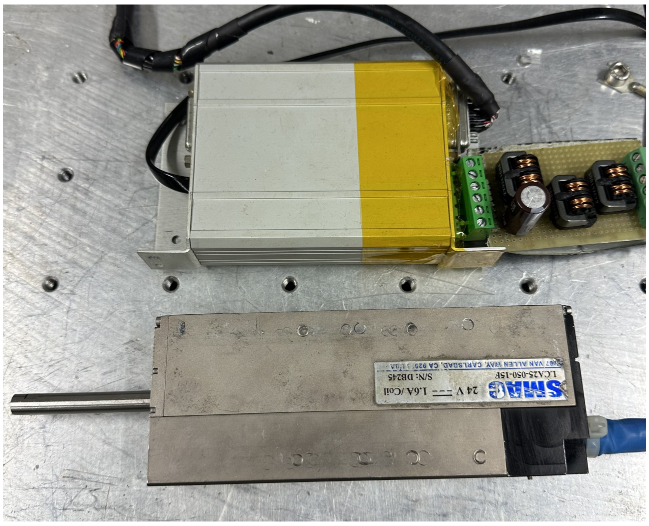


Figure 4.3: SMAC actuator and LC-10 controller.

4.2.4 Software

During the sensor characterization process, three software tools provided with the lab measurement PC are used. As previously mentioned, the LCC-10's software is used for the initialization of the SMAC actuator via its GUI. However, the commanding scripts that control the actuator motion, are executed through the provided MATLAB scripts [58]. The Dewesoft DAQ system is initialized and controlled by its dedicated firmware. The entire process of data acquisition is seamlessly integrated into the MATLAB environment relying on MATLAB serial communication commands provided by Dewesoft. Beyond acquisition, the subsequent post-processing of the obtained data is also conducted within the MATLAB environment.

4.3 Measurement Setup

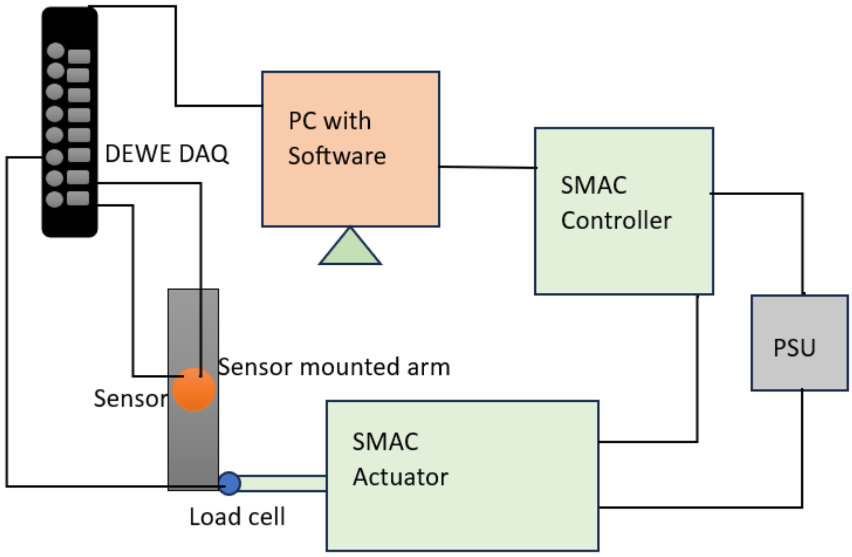


Figure 4.4: Overview of the setup.

The assembly of the experimental setup is based on an aluminum plate featuring 6 mm threaded bores arranged in a symmetric grid of 5 by 5 cm. This design allows for easy and fast assembly of various geometric structures. The SMAC and the sensor are securely fixed to the plate using a combination of 3D printed and metal supporting structures. In this characterisation experimental setup, the objective is to apply a mechanical load on the sensor, intending to induce deflection in the beams, consequently causing resistance changes in the strain gauges. As shown in Figure 4.1, the sensor comprises of both an outer ring and an inner ring, subject to a torque applied around the central axis. These rings are linked by four equally spaced beams. When a torque is exerted, the sensor undergoes rotation along its central axis, perpendicular to the sensor's surface. This torque induces strain in the beams, resulting in their deformation. Strain gauges are placed on the beams where the maximum strain occurs during deformation. As the beams deform, the strain gauges experience changes in geometry, leading to corresponding change in resistance.

In order to secure the sensor in a way that a constrained load can be applied, a 3D printed bracket is bolted to the aluminium plate (refer to Figure 4.6). The bracket secures the sensor horizontally at its centre, enabling rotation around the central axis. Furthermore, an extension arm, as shown in Figure 4.5, is connected to the sensor and the SMAC at other end. Given that the SMAC generates a linear force at a distance from the rotation point, a torque is produced that deforms the strain gauges resulting in a sensor response.

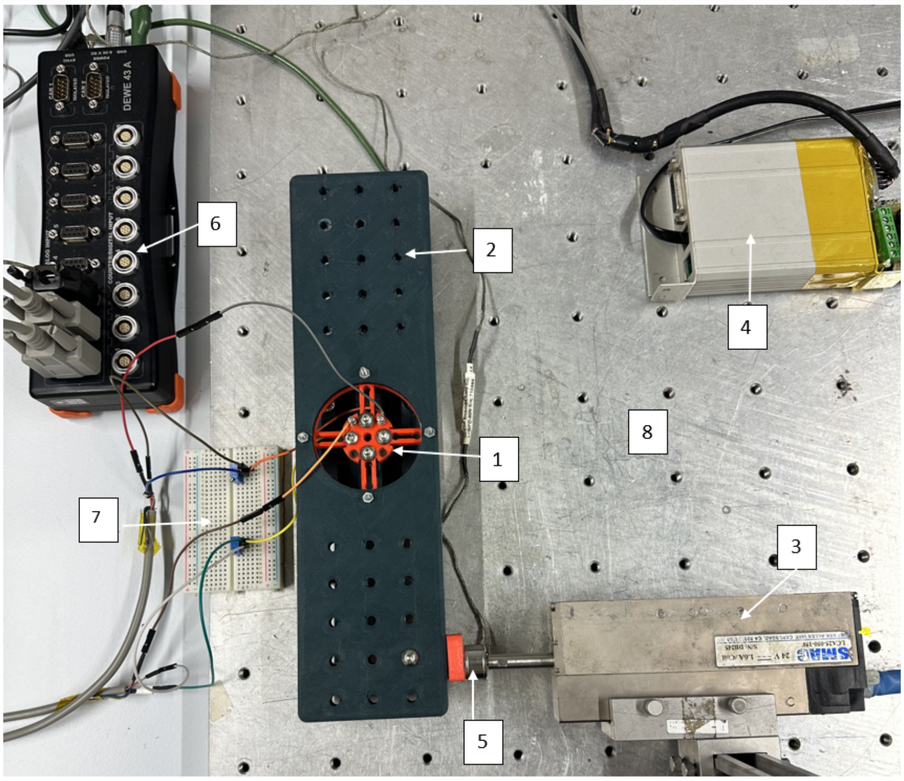


Figure 4.5: Measurement setup (1-Sensor, 2-Sensor mounted extentions arm, 3-SMAC actuator, 4-SMAC controller, 5-Load cell, 6-DEWE, 7-Half bridge setup, 8-Mounting plate).

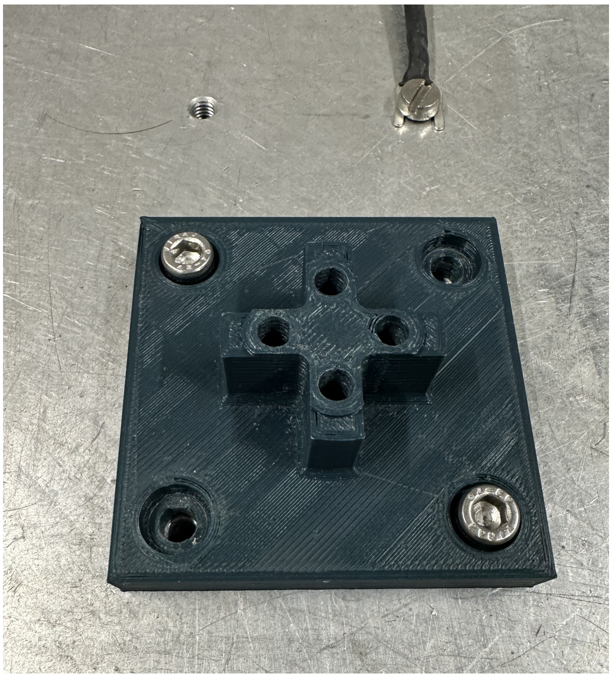


Figure 4.6: 3D printed bracket that allows to fix the sensor.

For the measurement of resistance changes, two strain gauges on one beam are directly linked to the Dewesoft Data Acquisition (DAQ) System via separate RS232 cables. To achieve this connection, the cable setup is modified to use the strain gauges in a half-bridge configuration, following the guidelines in the Dewesoft manual [53]. Figure 4.7 illustrates the Wheatstone bridge circuit configuration for the chosen half-bridge system. The use of a Wheatstone bridge is preferred over direct resistance measurement due to its sensitivity, temperature compensation capability, and linear response. Two half bridge systems were configured, one for each strain gauge. Since the sensors were connected individually, two potentiometers, one for each bridge, are incorporated, and both set to a balanced state before initiating experiments. This balancing involves adjusting the potentiometer associated with each bridge, aiming to achieve a state where the bridge is electrically balanced. The need for a balanced state in a Wheatstone bridge configuration is tied to the principle of operation (will be elaborated later in this chapter) of the bridge and its impact on measurement accuracy.

As shown in Figure 4.7, the right side features the potentiometer (R_2, top right) and one of the gauges (R_1, bottom right). On the left side, R_3 and R_4 represent the internal resistors integrated into the DEWESoft hardware. These internal resistors are essential components needed to complete the half-bridge circuit. In this connection excitation lines (`Exc+` and `Exc−`) connect the excitation voltage source to the bridge circuit. In strain gauge measurements, 10V is applied to the bridge. Input lines (`In+` and `In−`), connect the outputs of the bridge circuit to the designated input connections on the DEWESoft hardware. The voltage difference across the bridge is captured through the sense lines. The two sense lines (`Sns+` and `Sns−`) are connected to the excitation lines and these voltages will be supplied to the fixed resistors integrated into the DEWESoft hardware, which is visibly positioned on the left side of the Figure 4.7

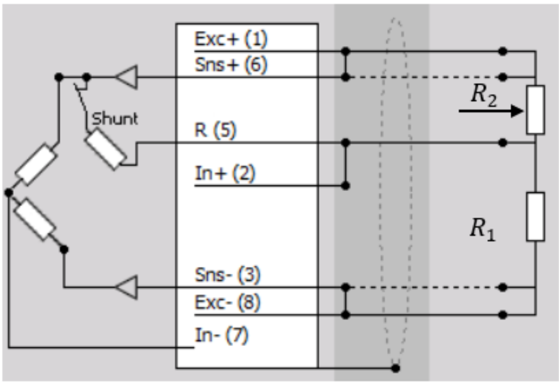


Figure 4.7: Half bridge system [59].

Additionally, the load cell is connected to the SMAC to measure the actuation force, see Figure 4.5. All three readout cable assemblies, two for the sensors and one for the load cell are connected to the analogue inputs of the DEWE DAQ for data acquisition during measurements.

4.4 Measurements

To capture the sensor response, strain gauges are placed in a half-bridge configuration, incorporating 20 kΩ potentiometers. The Dewesoft system conducts measurements at a high sample rate of 20 kHz, employing a 10 V input voltage for the bridges. Furthermore, filtering is applied within DEWEsoft to reduce the aliasing of the acquired data. This is achieved through the application of a low-pass Infinite Infinite Impulse Response (IIR) filter.

The linear actuator is initialized in force control mode. Then MATLAB scripts are used to command a sinusoidal motion, generating torque at the center of the ring due to the specific geometric configuration of the experiment. Each experiment yields three distinct data sets. During the force excitation from the linear actuator, the voltages resulting from the ΔR/R of the two strain gauges, and the load cell measurement of the actuation force are recorded.

The chosen maximum force for exertion was selected to be 10 N. Taking an arm extension of l =0.125 m, the maximum torque applied is calculated to be $T_{\text{max}} = 10l = 1.25\,\text{N m}$. Multiple experiments with a range of amplitudes and frequencies were carried out. Each measurement combination was repeated two times for repeatability with a duration of 60 s per measurement.

Calculation of relative resistance changes $\Delta R/R$ from the half-bridge voltage measurements of the strain gauges is described below using a Wheatstone bridge configuration.

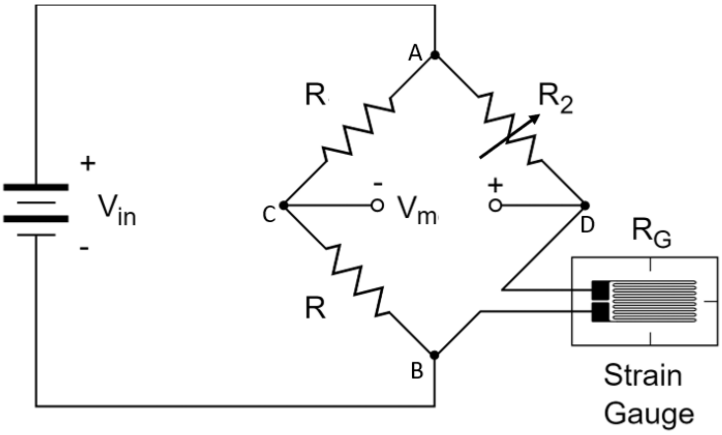


Figure 4.8: Wheatstone bridge circuit.

The relative resistance change of one strain gauge can be generally calculated by,

$$\frac{\Delta R}{R} = \frac{R_{\text{cal}} - R_{\text{G}}}{R_{\text{G}}} \tag{4.1}$$

In this calculation, the initial resistance of the strain gauge at the balanced state is R_{G} and the resistance of the strain gauge when beams are deforming is R_{cal} are required.

At sthe balancef state $V_{\text{C}} = V_{\text{D}}$

R is the resistance of the internal resistors which we can read from Dewesoft. R_2 is the potentiometer resistance at the balanced state and R_{G} is the gauge resistance at the balanced state which we are looking to find.

$$V_{\text{C}} = \frac{R}{R+R} \cdot V_{\text{in}} \tag{4.2}$$

$$V_{\text{D}} = \frac{R_{\text{G}}}{R_2 + R_{\text{G}}} \cdot V_{\text{in}} \tag{4.3}$$

When bridge in balance state

$$\frac{R}{R+R} \cdot V_{\text{in}} = \frac{R_{\text{G}}}{R_2 + R_{\text{G}}} \cdot V_{\text{in}} \tag{4.4}$$

By changing the potentiometer setting R_2 we can bring the bridge in to the balanced state and measure that resistance of potentiometer using a multi-meter. Therefore the initial resistance of the strain gauge can be calculated using Equation 4.4.

$$R_{\text{G}} = R_2 \tag{4.5}$$

Now we need to calculate the resistance of the gauge when it deforms with the beam, this can be done by using Equation 4.2 and Equation 4.3

$$V_{\mathrm{m}} = V_{\mathrm{C}} - V_{\mathrm{D}} \tag{4.6}$$

$$V_{\mathrm{m}} = \frac{R}{R+R} \cdot V_{\mathrm{in}} - \frac{R_{\mathrm{cal}}}{R_2 + R_{\mathrm{cal}}} \cdot V_{\mathrm{in}} \tag{4.7}$$

$$V_{\mathrm{m}} = \left(\frac{1}{2} - \frac{R_{\mathrm{cal}}}{R_2 + R_{\mathrm{cal}}}\right) \cdot V_{\mathrm{in}} \tag{4.8}$$

From Equation 4.8 we can derive an equation for the resistance of the strain gauge R_{cal} in terms of the Input voltage V_{in}, measured voltage V_{m} and potentiometer resistance R_2

$$R_{\mathrm{cal}} = \frac{R_2\,(V_{\mathrm{in}} - 2V_{\mathrm{m}})}{2V_{\mathrm{m}} + V_{\mathrm{in}}} \tag{4.9}$$

Substituting Equation 4.9 and Equation 4.5 to Equation 4.1 we can find the relative resistance change of the strain gauge.

4.5 Results

The primary objective of this characterization experiment is to analyse various aspects of the 3D printed torque sensor's performance, including accuracy, non-linearity, hysteresis, creep, and drift. The initial step involves an examination of the relationship between the measured voltage and the applied torque on the sensor, as well as hysteresis by establishing a connection between the relative resistance change and torque. It is essential to note that the characterisation within this study is aimed at providing a phenomenological perspective on the behaviour of the 3D printed sensor, rather than offering an analytical description of the sensor's physical behavior, the study focuses on understanding its characteristics through empirical observations.

The findings presented in this chapter are the result of the analysis of two distinct data sets. The first data set involves a sine wave measurement with a maximum torque of 1.25 N m and a maximum frequency of 1 Hz, while the second data set explores the sensor's response to a minimum torque of 0.5 N m at a minimum frequency of 0.2 Hz. The selection of these data sets not only aims to capture the differences between the results but also serves the purpose of evaluating the sensor's behaviour across a wide range of operational conditions. For a detailed breakdown of each conducted experiment and an overview of the results, please refer to Appendix B.

The measured voltage (V_{m}) for each experiment is visually represented over time, clearly demonstrating the expected behaviour. This is evident from the signal patterns illustrated in Figure 4.9 and Figure 4.10.

In the process of preparing acquired data for analysis, a digital filter was applied to eliminate noise from the data. This was done during post-processing, using a zero-phase filter that operates in both the forward and backward directions [60]. Specifically, a 2$^{\mathrm{nd}}$ order Butterworth low-pass filter was chosen for this purpose. To determine the suitable cut-off frequency, an analysis of the frequency spectrum of each signal was conducted, as shown in Figure 4.11. The data set acquired with 1.25 N m torque input with a frequency of 1 Hz is used to illustrate the concept of cut-off frequency for the filter. Looking at the Figure 4.11, it seems that a minimum cut-off frequency of around 5 Hz is required due to the presence of fundamental frequency peaks up to 5 Hz. It is necessary to make sure that the higher harmonics are captured in the

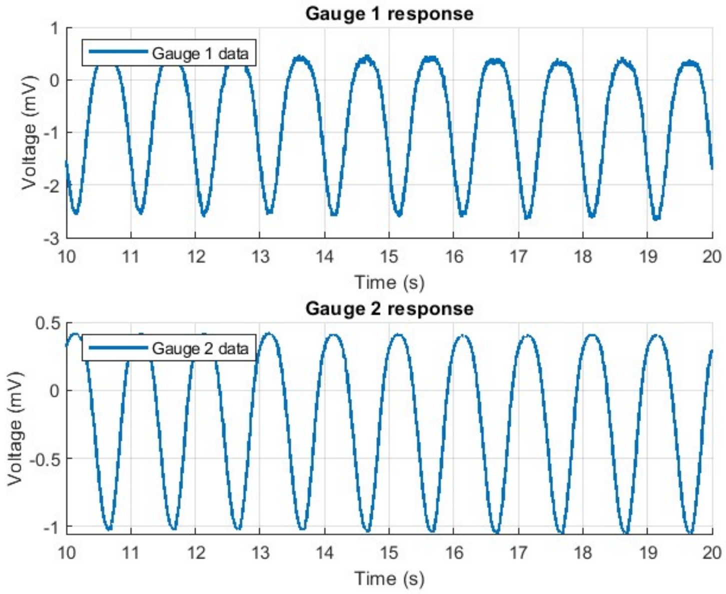


Figure 4.9: Individual response of strain gauges for a 1.25 N m torque at a 1 Hz sinusoidal excitation.

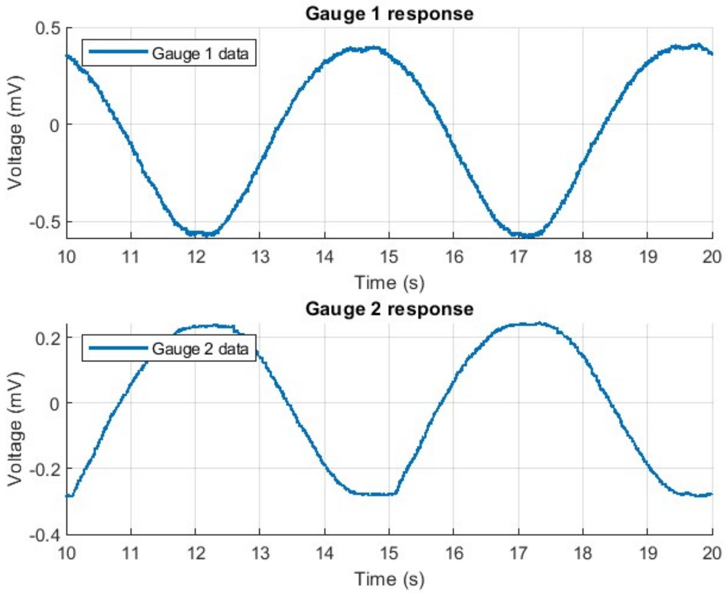


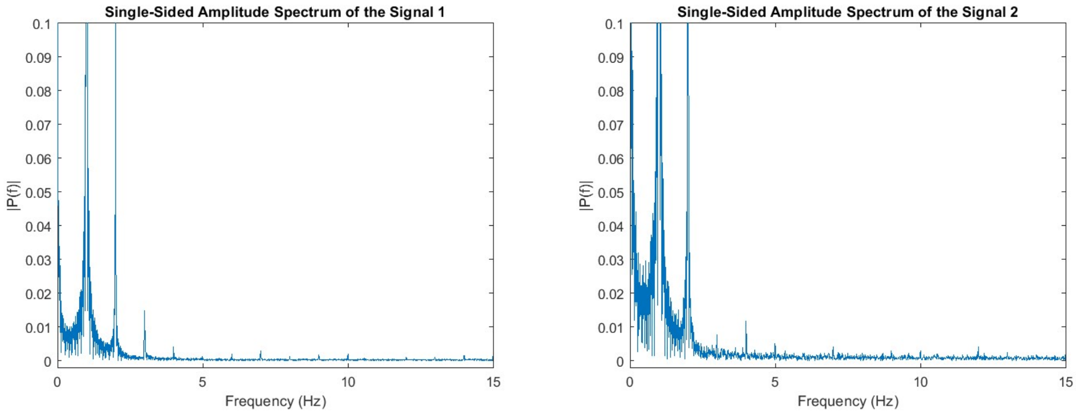
Figure 4.10: Individual response of strain gauges for a 0.5 N m torque at a 0.2 Hz sinusoidal excitation.

signal, up to a point where the noise in the signal hides the magnitude of the highest harmonic. Therefore, to keep the impact of noise under control a cut-off frequency of 10 Hz was chosen.

With the chosen cutoff frequency, a 2$^{\text{nd}}$ order Butterworth filter was designed and applied as represented by the transfer function in Equation 4.10.

$$H(z) = \frac{1.542\times10^{-9}z^2 + 3.084\times10^{-9}z + 1.542\times10^{-9}}{z^2 - 2z + 0.9999} \tag{4.10}$$

The filter was applied to all four data sets, and the filtered voltage responses of each half-bridge signal are depicted alongside the raw data in Figure 4.12 and Figure 4.13.



(a) Frequency spectrum of strain gauge 1.

(b) Frequency spectrum of strain gauge 2.

Figure 4.11: Frequency spectrum of strain gauges for a 1 Hz sinusoidal torque input.

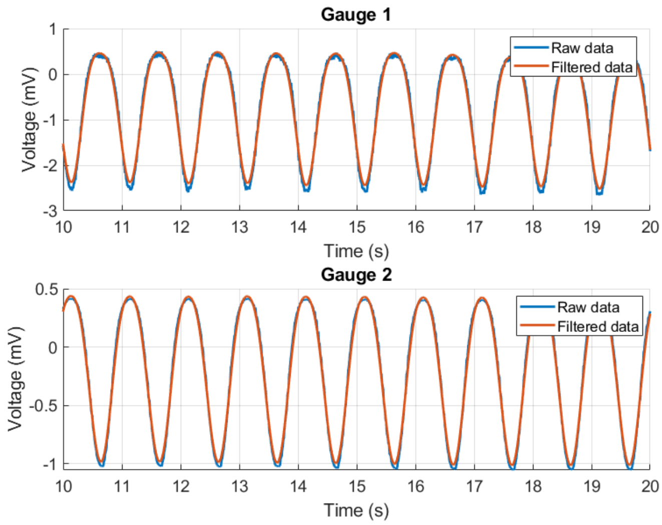


Figure 4.12: Filtered and raw responses of strain gauges for a 10 N force at 1 Hz sinusoidal excitation.

Following the acquisition and post processing of input and output data, the next step involves establishing the relationship between input torque and output voltage over time. Figure 4.14 illustrates the force exerted by the linear actuator on the sensor, measured by the load cell, across two distinct experiments.

The plotted input-output relationship over time as shown in Figure 4.15, shows the sensor's hysteresis and also highlights the sensor drift. Notably, the observed drift is higher with higher forces.

Additionally, individual gauge observations indicate opposing responses, a phenomenon that aligns with expectation of opposing strains. This is reflected in Figure 4.16.

The impact of differential measurement on drift becomes more evident when visualizing a restricted number of cycles, each separated by a substantial time interval, as illustrated in Figure 4.17 and Figure 4.18. It can be seen that the differential measurement partially compensates for sensor drift compared to each gauge's individual response. This behaviour also has been seen in other studies such as [43].

Furthermore, examining the individual responses of the gauges alongside the differential response reveals an observable improvement in response linearity with the differential meas-

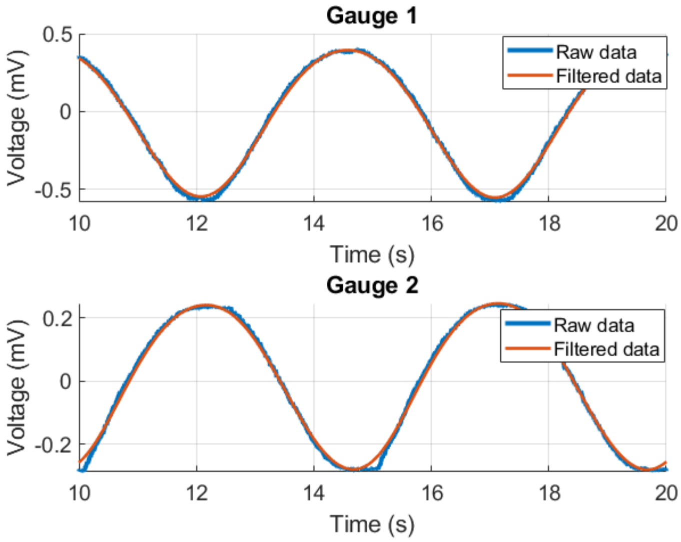
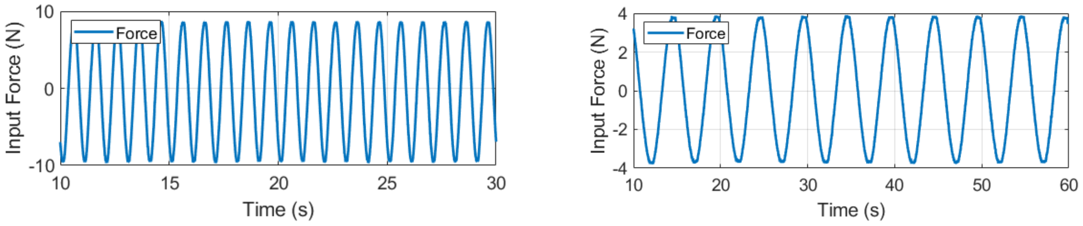


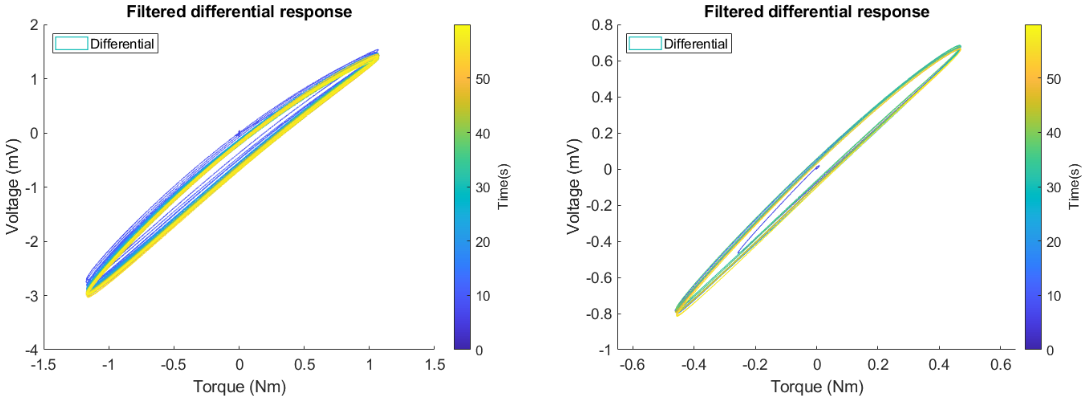
Figure 4.13: Filtered and raw responses of strain gauges for a 4 N force at 0.2 Hz sinusoidal excitation.



(a) 10 N output force of SMAC at 1 Hz.

(b) 4 N output force of SMAC at 0.2 Hz.

Figure 4.14: Output force of sinusoidal excitation from SMAC actuator.



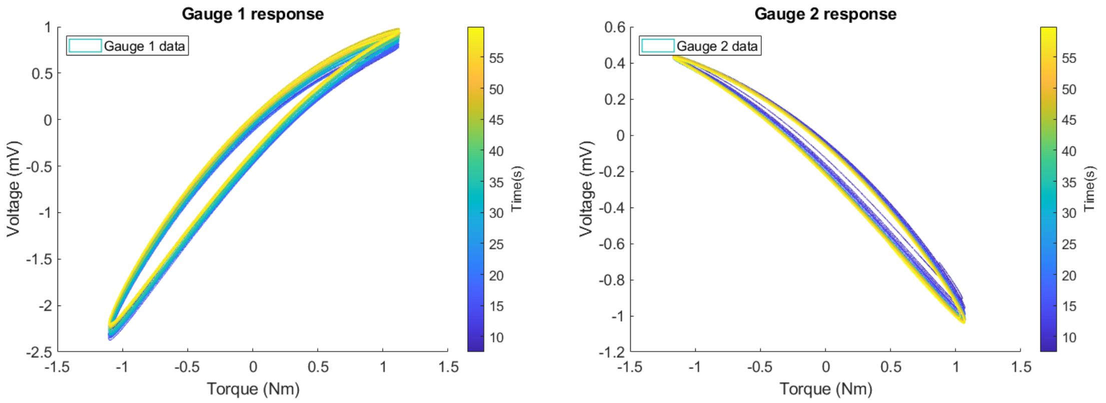
(a) Response for 10 N force at 1 Hz sinusoidal excitation.

(b) Response for 4 N force at 0.2 Hz sinusoidal excitation.

Figure 4.15: Filtered differential response of datasets.

urement than individual responses. Similar behavior has been observed in other studies, as documented in [44] and [1]. Consequently, opting for the differential response derived from the sensor's collective response, rather than considering individual responses, is preferable for effective control system implementation.

Furthermore, a first-order polynomial is applied to the filtered data. This linear fit serves as a representation of the sensor's characterized response, illustrating the voltage difference corresponding to the applied torque. The purpose behind this step is to use this as a starting point



(a) Gauge 1 response

(b) Gauge 2 response

Figure 4.16: Strain gauge responses to sinusoidal excitation force of 10 N at 1 Hz.

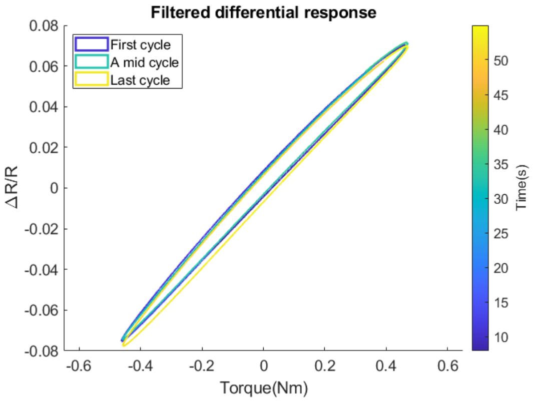
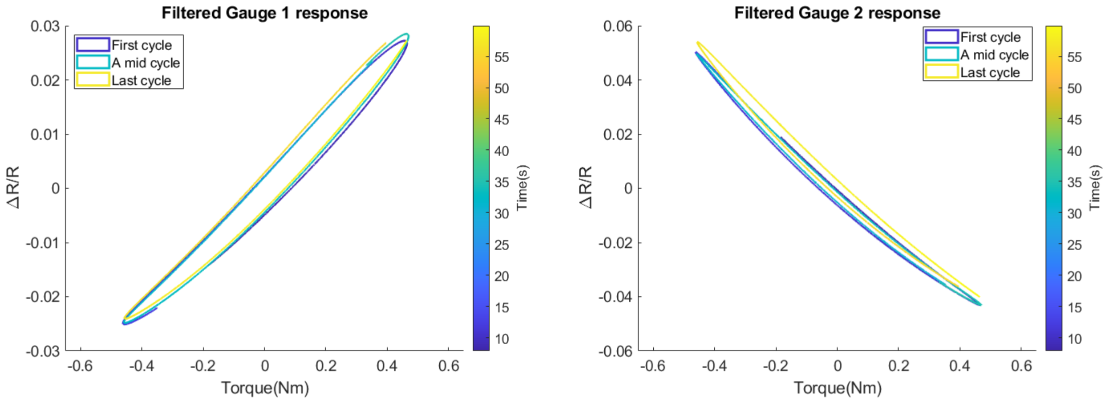


Figure 4.17: Filtered differential response of 3 cycles.



(a) Filtered gauge 1 response with 3 cycles.

(b) Filtered gauge 2 response with 3 cycles.

Figure 4.18: Filtered response of both gauges for 3 cycles from data set 2.

for the control system implementation mapping the voltage readouts of the sensor to the input torques.

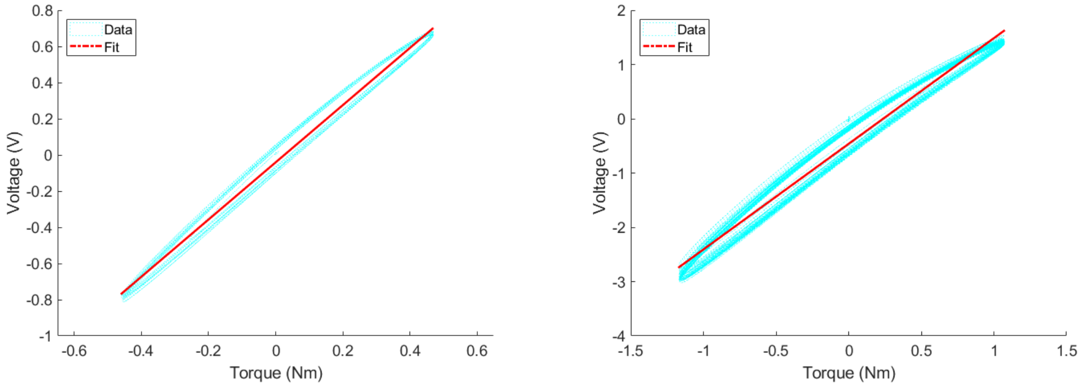
It is worth mentioning that the estimated linear fit varies with changes in both input frequency and amplitude. This observation can be seen with the linear fit representations of both datasets.

The fit for data set 1 with 10 N force input at 1 Hz is,

$$y = 1.9470x - 0.461 \tag{4.11}$$

The fit for data set 2 with 4 N force input at 0.2 Hz is,

$$y = 1.7185x - 0.07408 \tag{4.12}$$



(a) For 4 N force at 0.2 Hz.

(b) For 10 N force at 1 Hz.

Figure 4.19: Linear fit with the filtered differential voltage against the input torque for both data sets.

However, when subjected to the same torque at different frequencies, the linear fits closely follow consistent behaviour. This observation leads to the conclusion that variations in input torque have a greater impact on the approximation. These results can be further seen in the Appendix A.

4.6 Conclusion

In conclusion, this chapter presented a comprehensive characterisation of the 3D printed torque sensor. An experimental setup, featuring the sensor in combination with a linear actuator and a load cell, was used to evaluate the sensor's behavior under varying conditions. The applied methodology involved sinusoidal torque inputs with different frequencies and amplitudes, capturing a broad spectrum of operational scenarios.

The obtained measurement results have been presented, and their subsequent post-processing is also detailed in this chapter. These results revealed key sensor characteristics, including hysteresis, non-linearity, and drift. As an initial step towards representing the sensor behaviour for control purposes, a linear fit was used.

The differential measurement approach, considering collective responses from strain gauges, demonstrated improved linearity and partial drift compensation. Furthermore, the responses indicate a closely linear behaviour when the sensor is subjected to low frequency, low torque inputs. However, as the input frequency and torque are increased, the sensor's response becomes less linear, mainly due to the influence of hysteresis.

5 Impedance Control

5.1 Introduction

In robotics applications like manipulation, grasping, assembly, and locomotion, physical contacts play a fundamental role. Effectively managing these contacts requires the ability to regulate interaction forces or, more broadly, to control the compliance of the robot.

The primary focus of this chapter is to delve deeper into the selected interaction control scheme. This will be achieved by breaking down the main goal into several sub-goals. The initial objective being to understand the dynamics of the pendulum system. Following this, the aim is to simulate the dynamics of the plant for validation using MATLAB and Simulink. Next the interaction control scheme will be developed, along with further analyses and simulations of the control system in Simulink. This will also include a MATLAB simulation of the interaction control for a 1 DOF pendulum system.

In the experimental setup, a control system will be implemented for a 1 DOF pendulum system. Therefore, this chapter will introduce the pendulum system and its characteristics, with the following mathematical derivations being tailored exclusively for this particular type of robot. Working with a 1 DOF pendulum simplifies the control and analysis process making it an ideal starting point for understanding and developing control strategies before expanding to more complex systems with multiple degrees of freedom. Furthermore, it allows us to investigate the impact of the characteristics of the incorporated 3D printed sensor on the system dynamics and control performance for a simple 1 DOF movement.

Before delving into the subject, it's important to clarify the term "compliance". In mechanical terms, compliance typically describes a material's inherent property, not that of a structure. However, for the sake of historical consistency and clarity, this thesis continues to use the term "compliance" in the context of robotics, although the accurate term should be "flexibility." [61]. In this thesis, "compliance" refers to the relation between motion and forces at a contact point. This includes both the stiffness and damping of the structure, excluding its inertia.

5.2 Equation of Motion of Proposed System

To derive the equation of motion (EOM) for the proposed 1 DOF pendulum system and subsequently develop the control system based on the theory explained in Chapter 2, it is essential to initially define the inertia matrix M, Coriolis effects C, generalized forces or friction F and gravitational forces g in the system dynamics as explained in Equation 2.15. However, understanding the system kinematics is necessary before delving into the dynamics. Kinematics provide insight into how the system moves. First the pose (position and orientation) is determined of the center of mass (CoM) of the system, expressed in terms of the generalized coordinates q. In this case, the system is fixed on the z axis plane, with rotation occurring around the z axis. The schematic overview of the pendulum system is shown in Figure 5.1. The pose comprises of the 2D (x, y) positions and the orientation angle θ, which is represented in a 3-dimensional vector [62],

$$P = \begin{bmatrix} x_\text{c} \\ y_\text{c} \\ \theta \end{bmatrix} \tag{5.1}$$

Similar to the pose, the linear and angular velocity can be expressed as the twist V,

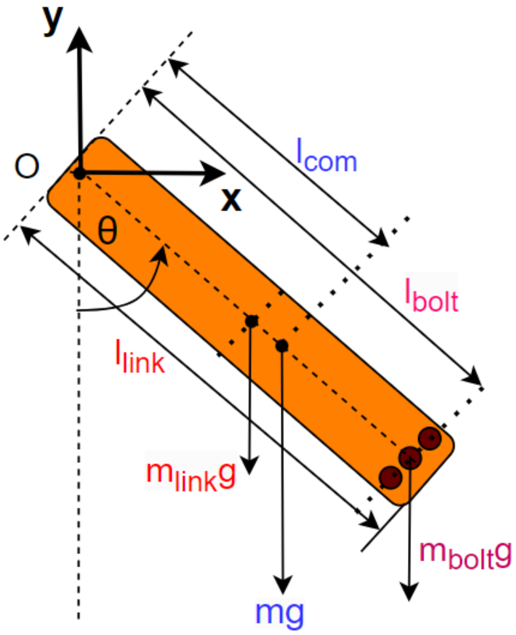


Figure 5.1: The schematic overview of the pendulum system.

$$V = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} \tag{5.2}$$

With the generalised coordinates, the twist of the link can be expressed as:

$$V = \dot{P} = \frac{\partial P}{\partial q}\left(\frac{dq}{dt}\right) = J(q)\dot{q} \tag{5.3}$$

where,

$$J(q) = \frac{\partial P}{\partial q} \tag{5.4}$$

is the analytical Jacobian of the link's centre of mass. For a 1 DOF pendulum system, q is the angle θ that describes the state of the system.

To get the pose of the CoM of the system , first we need to find the CoM of our pendulum system. In modelling the pendulum system, the centre of mass for the rigid link was determined by assuming uniform density for the link. Bolted weights that could be added to the end of the pendulum were treated as point masses attached at specific locations. The overall CoM was calculated by summing the contributions from the original link and bolted weights.

The formula for calculating the CoM distance (l_{com}) is derived based on the principle of weighted averages. This principle recognizes that the overall CoM position is influenced by both the mass distribution within the rigid link and the discrete point masses represented by the bolted weights. Considering the masses of the link and bolts and their respective distances from the rotation point. The CoM of the system can be calculated as follows,

$$l_{\text{com}} = \frac{m_{\text{link}} \cdot \frac{l_{\text{link}}}{2} + m_{\text{bolt}} \cdot l_{\text{bolt}}}{m_{\text{link}} + m_{\text{bolt}}} \tag{5.5}$$

Where,

m_{link} - Mass of the rigid link.

l_{link} - Length of the link.

m_{bolt} - Total weight of the bolted weights, equivalent to the sum of all the point masses.

l_{bolt} - Distance from the rotation point to the point masses.

l_{com} - Centre of total mass of the system as a whole.

Making use of Equation 5.5 the l_{com} can be calculated and the pose of the CoM can be written with respect to the local coordinate system attached to the rotation point of the pendulum as depicted in the Figure 5.2 as follows,

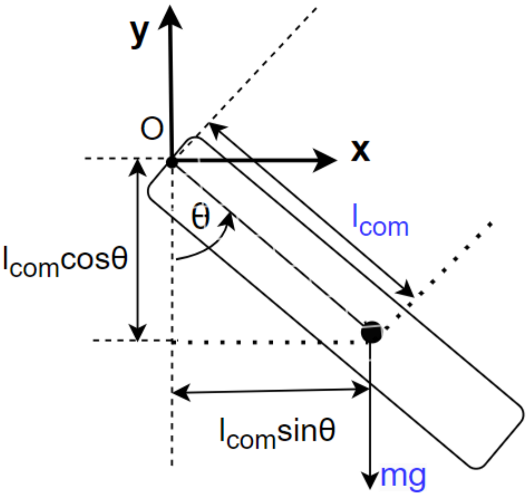


Figure 5.2: Pose of the pendulum system with respect to the rotation point.

$$P_{\text{com}} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} l_{\text{com}} \sin(\theta) \\ -l_{\text{com}} \cos(\theta) \\ \theta \end{bmatrix} \tag{5.6}$$

The CAD model of the pendulum system, including bolted weights, ensures that CoM is precisely located along the neutral axis, as illustrated in Figure 5.3. The mass properties, along with the coordinates of the CoM relative to the rotation point, have been computed from the CAD model. For detailed information, please refer to Appendix C.

5.2.1 Inertia Matrix

After determining the center of mass, the subsequent step involves deriving the inertia matrix M. The inertia matrix can be obtained through an analysis of the system's kinetic energy. Recalling that the kinetic energy K of a rigid body with mass m, linear velocity v, the moment of inertia I and angular velocity $\dot{\theta}$ is expressed as,

$$K = \frac{1}{2} m v^2 + \frac{1}{2} I \dot{\theta}^2 \tag{5.7}$$

The kinetic energy of a mechanical system can also be reformulated using generalized velocities $\dot{q}$ and the inertia matrix $M(q)$ [40, p. 356], [62]

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \tag{5.8}$$

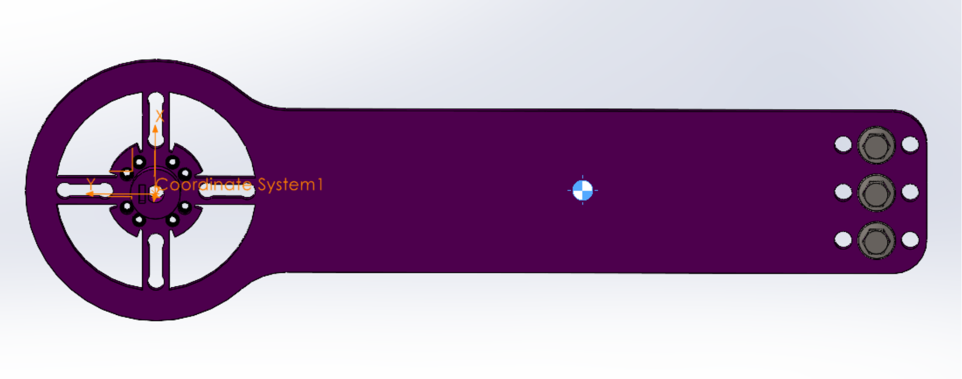


Figure 5.3: Center of mass of the CAD model of the pendulum with bolted weights.

From equation 5.8 it can be observed that kinetic energy depends on the inertia matrix. Therefore we can derive the inertia matrix from the kinetic energy of the system. By representing the kinetic energy of our system using the expression provided in Equation 5.7, which describes the kinetic energy of a rigid body. We arrive at the following matrix form:

$$K = \frac{1}{2} V^T G V \tag{5.9}$$

Where,

$$V = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}$$

twist of the link and

$$G = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}$$

the inertia tensor. To find an expression for the inertia matrix we can express Equation 5.9 in terms of generalized velocity. Substituting Equation 5.3 into Equation 5.9 we can derive K as follows,

$$K = \frac{1}{2} \dot{q}^T J(q)^T \cdot G \cdot J(q) \dot{q} \tag{5.10}$$

Now using the above equation and Equation 5.8 the intertia matrix M can be derived as follows,

$$M(q) = J(q)^T \cdot G \cdot J(q) \tag{5.11}$$

To compute the inertia matrix $M(q)$ as per Equation 5.11 we need to find J_{com} first. In this analysis, the vector q represents the joint variable, and for the specific case of a single degree of freedom system, it is denoted as θ.

$$J_{\text{com}} = \frac{\partial P_{\text{com}}}{\partial q} = \begin{bmatrix} \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial \theta} \\ \frac{\partial \theta}{\partial \theta} \end{bmatrix} = \begin{bmatrix} l_{\text{com}} \cos(\theta) \\ l_{\text{com}} \sin(\theta) \\ 1 \end{bmatrix} \tag{5.12}$$

Combining this result with Equation 5.11, the inertia matrix can be written as,

$$M(q) = J_{\text{com}}(q)^T G J_{\text{com}}(q) = \begin{bmatrix} l_{\text{com}}\cos(\theta) \\ l_{\text{com}}\sin(\theta) \\ 1 \end{bmatrix}^T \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} l_{\text{com}}\cos(\theta) \\ l_{\text{com}}\sin(\theta) \\ 1 \end{bmatrix} \tag{5.13}$$

$$M(q) = m l_{\text{com}}^2 + I \tag{5.14}$$

Estimating the moment of inertia (I) involves a simplification approach where the pendulum, now equipped with bolted weights, is modelled as a solid rectangular plate. This simplification is used for ease of estimation, recognizing that the obtained value may not be entirely precise. The resulting equation for the moment of inertia around its centre of mass under this simplified representation can be written as,

$$I = \frac{1}{12} m (l_{\text{link}}^2 + w^2) \tag{5.15}$$

where m is the total mass of the system. l_{link} is the length of the body, assumed to be the length of the rectangular plate. w is the width of the body, which is the width of the rigid link.

This equation holds when system rotates around the center of mass, or in other words, when the z axis is located through the center of the mass of the system. However for our system, the rotation axis is located away from the CoM by l_{com} . By applying the parallel axis theorem the new inertia can be written as,

$$I' = I + m l_{\text{com}}^2 \tag{5.16}$$

Allowing us to rewrite the inertia matrix $M(q)$ as below by combining Equation 5.14 and Equation 5.16,

$$M(q) = 2 m l_{\text{com}}^2 + I \tag{5.17}$$

Following from Equation 5.17 the Inertia matrix for our 1 DOF pendulum system is a constant scalar.

5.2.2 Coriolis/Centrifugal matrix

The Coriolis term is linear in $\dot{q}$ and represents the forces that arise from the velocities of the system [40]. While numerous methods exist for expressing the coriolis matrix $C(q,\dot{q})$ in terms of Christoffel symbols, a frequently used approach is [63], [40, p. 354],

$$C_{ij}(q,\dot{q}) = \sum_{k}^{n} \Gamma_{ijk}(q)\dot{q}_k \tag{5.18}$$

where Γ_{ijk} are the Christoffel symbols, calculated from the mass matrix and n is the number of joints in the system. In order to determine the Coriolis matrix, the initial step involves obtaining the array of Christoffel symbols,

$$\Gamma_{ijk}(q) = \frac{1}{2}\left[\frac{\partial M_{ij}}{\partial q_k} + \frac{\partial M_{ik}}{\partial q_j} - \frac{\partial M_{jk}}{\partial q_i}\right] \tag{5.19}$$

where M_{ij} is ij-th element of the mass matrix $M(q)$. The symbol $\frac{\partial M_{ij}}{\partial q_k}$ denotes a partial derivative of an element of inertia matrix with respect to the joint coordinate q_k.

It follows from the equation that, to compute the Christoffel symbols and subsequently the Coriolis matrix $C(q,\dot{q})$, we need to know the mass matrix $M(q)$ and its partial derivatives. With the Inertia Matrix $M(q)$ computed as a constant scalar, it's noteworthy that the Coriolis forces in this configuration reduce to zero $C(q,\dot{q}) = 0$. This simplification arises as the derivative of the Inertia Matrix, responsible for Coriolis forces, is equal to zero for the 1 DOF pendulum.

5.2.3 Gravity vector

To derive the gravity vector $g(q)$, we look into the system's potential energy. The potential energy U associated with a mass m at a height y is expressed as $U = mgy$, where g signifies the gravitational acceleration. In a robotic system which has multiple links the potential energy is given by,

$$U = g\sum m_i \cdot y_i \tag{5.20}$$

Now, the gravitational force acting on each mass element of the system can be derived from the potential energy. This involves taking the partial derivatives of the potential energy with respect to the joint variable associated with the link (joint angle) q.

$$g(q) = \left[g_1(q), g_2(q), g_3(q)....g_n(q)\right] \tag{5.21}$$

n represents the number of links. Each $g_i(q)$ represents the gravitational force acting on the $i-th$ link. The resulting torque generated by gravity on the joint is then computed as follows:

$$\tau_{\mathrm{g}} = m \cdot g \cdot l_{\mathrm{com}} \sin(\theta) \tag{5.22}$$

5.2.4 Generalized forces or friction

The term $F(q.\dot{q})$ typically represents additional generalized forces or friction in the joint. We assumed that the system is considered to be free of any additional generalized forces other than friction. In this study, friction is characterized by viscous damping, a type of damping that is proportional to the velocity. Frictional torques can be modelled as linearly dependent on the rotational velocity,

$$F = B\dot{q} \tag{5.23}$$

B is damping coefficient.

5.2.5 Equation of motion

The next stage involves the computation of the derived inertia matrix, Coriolis matrix, and gravity vector with the system parameters and variables. Subsequently, these derived matrices and vectors will be used to formulate the equations of motion for the 1 DOF pendulum system, as explained in the aforementioned theoretical framework. The calculated/measured system parameters are detailed in the Table 5.1.

Taking the dynamic equation of motion for a robotic system derived in Chapter 2,

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + F(q,\dot{q}) \tag{5.24}$$

Variable	Variable name	Varibale value	Units
m_{link}	mass of the rigid link	0.0550	kg
m_{bolt}	total mass of the 3 bolts	0.0180	kg
l_{link}	length of the link	0.230	m
l_{bolt}	distance to the bolts from the rotation point	0.215	m
w	width of the rigid link	0.048	m

Table 5.1: Parameters of the pendulum system

and proceeding with this equation to calculate the torque acting on the robot joint. We arrive at the following:

$$\tau = (ml_{\text{com}}^2 + I')\ddot{q} + m \cdot g \cdot l_{\text{com}} \sin(q) + B\dot{q} \tag{5.25}$$

This formulation also serves as the inverse dynamic representation of the pendulum system. The Equation of Motion (EOM) for the system can be obtained by solving this expression to determine the system's acceleration as follows;

$$\ddot{q} = \frac{\tau - m \cdot g \cdot l_{\text{com}} \sin(q) - B\dot{q}}{ml_{\text{com}}^2 + I'} \tag{5.26}$$

5.3 Impedance Control

In Chapter 2, various interaction control methods were discussed, emphasizing the importance of effectively handling interaction forces in the design of control strategies for robotic systems. Traditional control theories, assume accurate representation of system dynamics up to a certain frequency [64], but may fall short especially when low-frequency behaviour altering interaction forces are introduced. In these scenarios, conventional control methods prove insufficient to design control strategies that allow the system to interact with its surroundings in a stable, responsive, and controlled manner and this is where interaction control methods become important. Recognizing this limitation, impedance control was identified as a suitable approach for effectively managing interaction forces within the proposed pendulum robot system. Impedance control adjusts the robot's mechanical impedance, defining the relationship between robot motion and resulting forces/torques [36]. By viewing the environment as an admittance that maps forces to velocities [10], implementing an impedance behavior on the manipulator becomes a suitable choice for defining interaction behavior during contact.

Starting from the general equation of motion for a multi DOF mechanical system as expressed in Equation 5.24, variable τ represents all input torques and is occasionally divided into control torques and external torques.

$$\tau = \tau_{\text{e}} + \tau_{\text{u}} \tag{5.27}$$

In the domain of control systems, the objective is to select control torques τ_{u} that guide the robot's behavior toward a desired state. The perspective of 'impedance control' offers the flexibility to develop these control architectures either in joint space or Cartesian space (Section 2.5.1). In our specific scenario, where we use a torque sensor to detect joint torques, the focus will be on implementing impedance control in joint space. The use of a 3D printed torque sensor allows us to directly measure joint torques, providing a direct insight into the robot's interaction with the environment. Joint space impedance control proves advantageous in this context as it enables us to directly regulate the robot's joint-level dynamics.

5.3.1 Joint Space Impedance Control

The external force exerted on the end of the manipulator under pure position control will not affect the movement of the end-effector, because in this case the manipulator can be considered completely rigid. If active compliance control is to be achieved, even if the robot arm shows a certain degree of flexibility, the interaction between it and the environment needs to be considered. At this time from Equation 5.27 the joint driving torque can be written as,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(q,\dot{q}) + g(q) + \tau_{\mathrm{e}} = \tau_{\mathrm{u}} \tag{5.28}$$

τ_{e} represents the external environmental moment acting on the robot joints. τ_{u} is the driving torque of the robot joint.

The fundamental impedance controller in joint space is derived by shaping the effective potential energy of the robot. The torque generated by the interaction between the manipulator and the environment can be written as,

$$\tau_{\mathrm{e}} = D(\dot{\theta}_{\mathrm{d}} - \dot{\theta}) + K(\theta_{\mathrm{d}} - \theta) \tag{5.29}$$

This is also called the target impedance behaviour. Here, θ and θ_{d} represent the target and actual angular position respectively, $\dot{\theta}_{\mathrm{d}}$ and $\dot{\theta}$ represent the target and actual angular velocity respectively. This control law introduces a virtual spring and damper to the robot's joints. Matrices K and D represent the stiffness and damping respectively, during the interaction with the environment. The objective of this impedance controller is to impart impedance-like behavior to the entire system. Adjusting the stiffness and damping parameters of this control law allows us to tailor the system's behavior according to our desired specifications, especially when engaging with external forces.

5.3.2 Torque-based Impedance Control

Considering the thesis's objective to explore interaction control possibilities using a 3D printed torque sensor, the decision was made to implement the controller using torque-based impedance control in Chapter 2.

The control architecture developed uses a cascade control architecture, which consists of an impedance controller closed around an inner torque loop at the joint level. This is shown in Figure 5.4, where by integrating a nested torque controller, the use of a computed torque control law becomes possible. This control law effectively addresses system non-linearities under the assumption of an ideal torque sensor which, however, is not the case in this work. The presence of the inner joint space torque controller allows for treating the robot joints as torque sources, which significantly simplifies the implementation of advanced control approaches that utilize joint torques as their output.

The outer loop controls the robot's compliance by providing feedback on the joint positions, and generating torques necessary to exhibit desired stiffness and damping characteristics. θ and θ_{d} represent the actual and desired joint positions respectively. The impedance controller's output serves as the reference torque T_{r} for the inner torque loop. These reference torques are then directed to the inner torque loop, which uses feedback from the actual torque at the robot joints, to track and the reference torques. Joint torque T and position θ measurements are obtained through torque and position sensors respectively. The torque controller's output U commands the actuator. It's also important to highlight that this architecture does not currently incorporate feed-forward torque for compensating robot dynamics, as will be discussed in more detail in a later section.

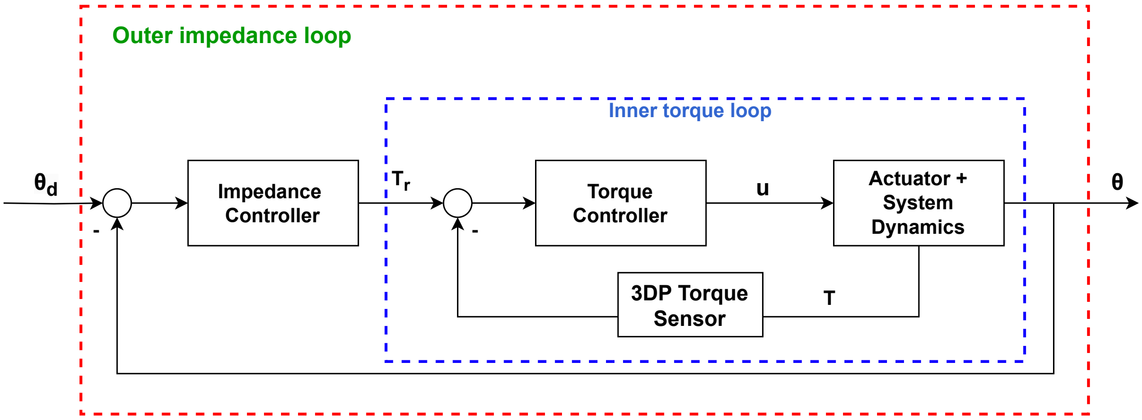


Figure 5.4: Block diagram of impedance control scheme.

Focussing on each loop separately. In typical robotic systems, the motor system is connected to a rigid link, constituting an inertial load. Inertial loads, as per [10], are admittances where the application of force is feasible, but not of velocity. Consequently, for a robot, characterized as a rigid body system, the input is fundamentally identified as a torque (or generalized force). Employing a joint-space torque controller simplifies the application of such inputs.

5.3.3 Torque Control Loop Design

As mentioned earlier, robots are commonly represented as rigid body systems, with torques serving as inputs to their dynamics. Achieving optimal control performance necessitates the precise measurement and control of the inputs exerted on the system. It is crucial to ensure the highest possible accuracy in managing these inputs for effective system control.

The PID error feedback controller stands out as the most widely used controller due to its simplicity and robustness [65]. However, when implementing torque control, the derivative term can introduce stability issues, especially when numerical differentiation is sensitive to the high noise levels inherent in strain gauge-based torque sensors. In such instances, the derivative term may not be practical, making a PI controller more suitable [61].

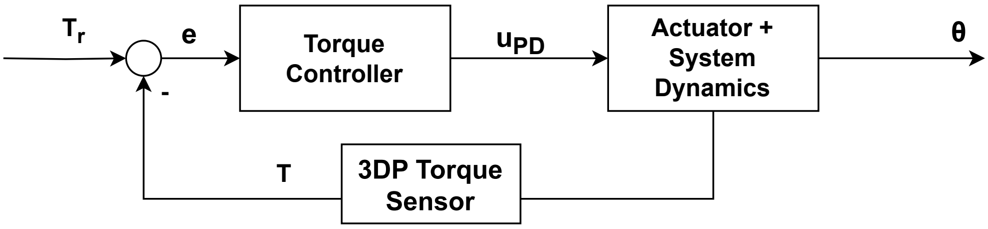


Figure 5.5: Block diagram of the PI inner torque control loop.

5.3.4 Outer Impedance Loop Design

One of the simplest ways to implement an impedance controller is likely by using a PD joint-space position loop. The integral term is often deemed unnecessary since achieving zero steady-state position error is typically not a critical requirement in a compliant system. Additionally, the physical counterparts of the PD control terms are directly linked to the intuitive understanding of compliance [61].

- Proportional Term P: This term can be linked to a spring, generating a control action proportional to the position error.

- Derivative Term D: This term can be linked to a damper, producing a control action proportional to the velocity error.

In the cascaded system, the output of the PD controller, denoted as T_r is necessarily a torque measured in Newton-meters (Nm). The proportional gain is expressed in units of Nm/rad, which aligns with the characteristics of a rotational spring. Similarly, the derivative gain corresponds to units of a rotational damper Nms/rad [61]. The advantage of using these units for the controller gains lies in the straightforward physical interpretation they provide.

Without an inner torque loop, the output of the feedback controller T_r directly serves as an analog input to the actuator, such as voltage or current. If T_r is considered as a voltage, the proportional gain K_p would have units of V/rad, and the derivative gain K_d would have units of Vs/rad. While these units lack a clear physical interpretation, the actuator would still exhibit behaviour like complex spring-damper, reacting proportionally to position and velocity errors. However, tuning such gains is less intuitive compared to scenarios where an inner torque loop is present.

Given the control architecture depicted in Figure 5.4 the outer PD control law can be formulated as:

$$\tau_\mathrm{fb} = K_\mathrm{p}(\theta_\mathrm{d} - \theta) + K_\mathrm{d}(\dot{\theta_\mathrm{d}} - \dot{\theta}) \tag{5.30}$$

where, τ_fb : output of the impedance controller. K_p : proportional gain. K_d : derivative gain. θ_d : desired joint position. θ : actual joint position. $\dot{\theta_\mathrm{d}}$: desired joint velocity. $\dot{\theta}$: actual joint velocity.

5.4 Implementation

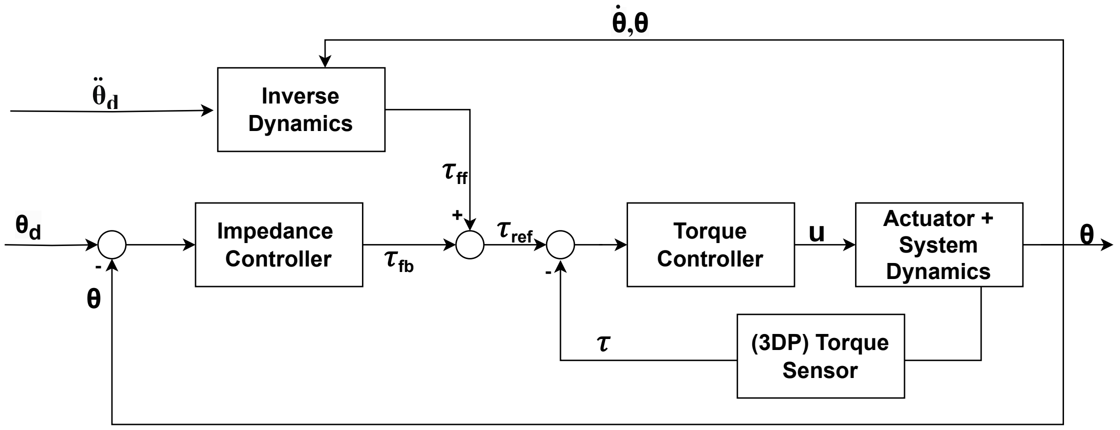


Figure 5.6: Block diagram of the joint-space impedance controller implemented for the pendulum system.

The implemented controller incorporates both the inner torque controller and the outer impedance loop, as previously discussed. In this configuration, the outer feedback loop, with a feedforward term derived from the rigid body inverse dynamics, generates the torque reference τ_ref for the inner torque loop.

Inverse dynamics proves to be valuable in improving robot performance. When a predetermined trajectory is available, the desired accelerations $\ddot{\theta}_\mathrm{d}$ can be computed and introduced into the inverse dynamics process to derive the necessary feedforward torques τ_ff.

Given our proven calculation demonstrating that Coriolis forces are zero, we can express the feedforward torque obtained from the inverse dynamics of the system as follows:

$$\tau_{\mathrm{ff}} = M(q)\ddot{q}_{\mathrm{d}} + B(\dot{q}) + g(q) \tag{5.31}$$

where, τ_{ff} : feedforward torque vector. q : joint angles vector. $M(q)$: joint space inertia matrix. $\ddot{q}_{\mathrm{d}}$: desired acceleration vector. B : damping coefficient. $\dot{q}$ is the joint velocity vector. g : gravity vector.

As indicated in the work by Buchli et al., [66] one notable advantage of using inverse dynamics is its ability to support compliant and robust locomotion. This advantage arises from the ability to decrease position gains without compromising tracking performance. This works because inverse dynamics control uses the robot's body model to predict the torques needed for certain movements. If the model is accurate, these predicted torques should help the robot follow the desired path with minimal position errors. The better the model, the fewer errors in position. A perfect model would generate torques that perfectly follow the planned trajectory, even without a feedback position controller [61]. Additionally, it's worth highlighting that in Equation 5.31, the control signal from inverse dynamics isn't influenced by position errors. To illustrate, if the robot encounters an unforeseen obstacle, the inverse dynamics won't change the estimated torques to maintain the desired position.

These attributes of inverse dynamics control become particularly advantageous with the inclusion of an inner torque loop. Incorporating the predicted torque τ_{ff} as a feedforward term, as illustrated in Figure 5.6 and adding to the feedback torque τ_{fb} results the reference torque τ_{ref} necessary for the inner torque loop. It's worth emphasizing that in Equation 5.31, τ_{ff} is a vector, whereas in this context, represents one of the components of this vector.

$$\tau_{\mathrm{ref}} = \tau_{\mathrm{ff}} + \tau_{\mathrm{fb}} = \tau_{\mathrm{ff}} + K_{\mathrm{p}}(\theta_{\mathrm{d}} - \theta) + K_{\mathrm{d}}(\dot{\theta}_{\mathrm{r}} - \dot{\theta}) \tag{5.32}$$

Hence, in theory, with a precise torque controller and an accurate rigid body model, the robot should follow the predicted motion, resulting in minimal position errors. Consequently, the impact of the feedback control action τ_{fb}, which depends on the position error would also be small during position tracking. These expectations will be further analysed in the simulation section.

5.5 Simulation

The simulation involved both MATLAB and Simulink platforms. The interaction control of a 1 DOF pendulum was simulated in MATLAB providing insightful visualizations to illustrate its functionality. Subsequently, for a more in-depth analysis of the control system and system dynamics, the system was implemented in Simulink. Initially, the simulation aimed to access and validate the derived plant dynamics. It's worth noting that, for simplicity, concepts like the controller and transfer functions were introduced in continuous time.

5.5.1 Plant dynamics simulation

The previously obtained equation of motion for the pendulum,

$$\ddot{q} = \frac{\tau - m \cdot g \cdot l_{\mathrm{com}} \sin(\theta) - B\dot{q}}{m l_{\mathrm{com}}^2 + I'} \tag{5.33}$$

Can be simulated using the measured and calculated system variables outlined in Table 5.2.

This EOM is then used to simulate the pendulum system in MATLAB and Simulink. To access and verify the system dynamics in both simulations, the pendulum is released from an initial

Variable	Variable name	Varibale value	Units
m	total mass of the system	0.0730	kg
B	Damping coefficient of the system	0.01	$\mathrm{N\,m\,s\,rad^{-1}}$
l_{com}	distance to the CoM from rotation point	0.128	m
I'	Inertia of the system around rotation point	0.0015	$\mathrm{kg\,m^{-3}}$

Table 5.2: Parameters of the system for the EOM

angle of $\frac{\pi}{2}$ rad with zero input torque. The ensuing response is visually simulated using MATLAB, and the angle is plotted against time for detailed analysis. The implementation code for this MATLAB simulation is provided in the Appendix C.

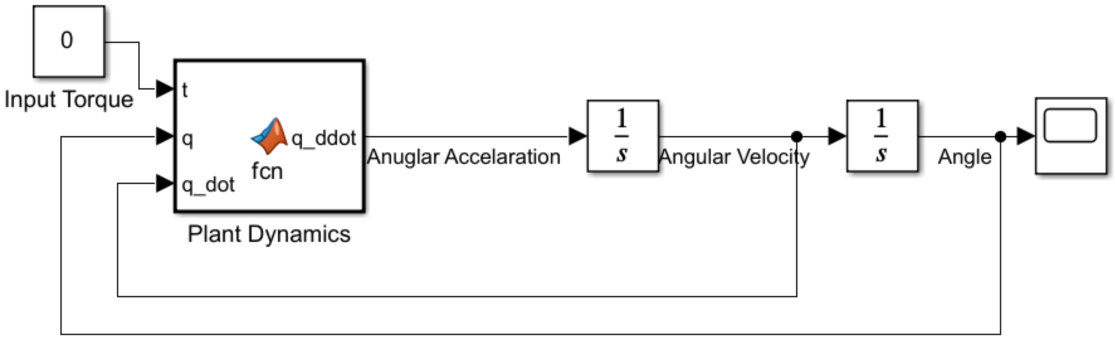


Figure 5.7: Simulink model to verify pendulum dynamics.

The Simulink block diagram for the zero input torque scenario is depicted in Figure 5.7. Figure 5.8 illustrates the system's behavior as it begins with an initial angle of $\frac{\pi}{4}$ rad , falling under the influence of gravity and oscillating for several seconds until reaching stability. This observation signifies that the implementation of plant dynamics yields the anticipated behavior, showcasing the expected characteristics of the pendulum system.

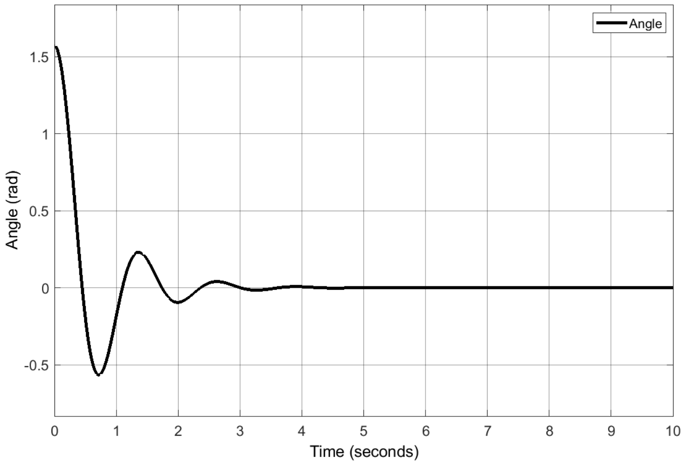


Figure 5.8: Zero torque input response with initial angle $\frac{\pi}{2}$ rad.

5.5.2 Impedance control simulation

Following the derivation and validation of the pendulum system dynamics, the subsequent phase involves simulating the control algorithm using MATLAB and Simulink. This simulation unfolds in two distinct phases: initially, without any interaction, followed by a simulation incorporating interaction dynamics. The goal of the simulation is to is to regulate the interaction forces and motions between the robotic system and its environment, ensuring stable and predictable behaviour during interactions.

Initially, an ideal impedance control simulation of a 1 DOF pendulum system was conducted in MATLAB using the same control architecture, showcasing how the system responds under the various interaction forces. Using the block diagram presented in Figure 5.6, the control architecture was implemented in Simulink for further analysis.

Figure 5.9 depicts the Simulink implementation of the interaction control architecture for the pendulum system.

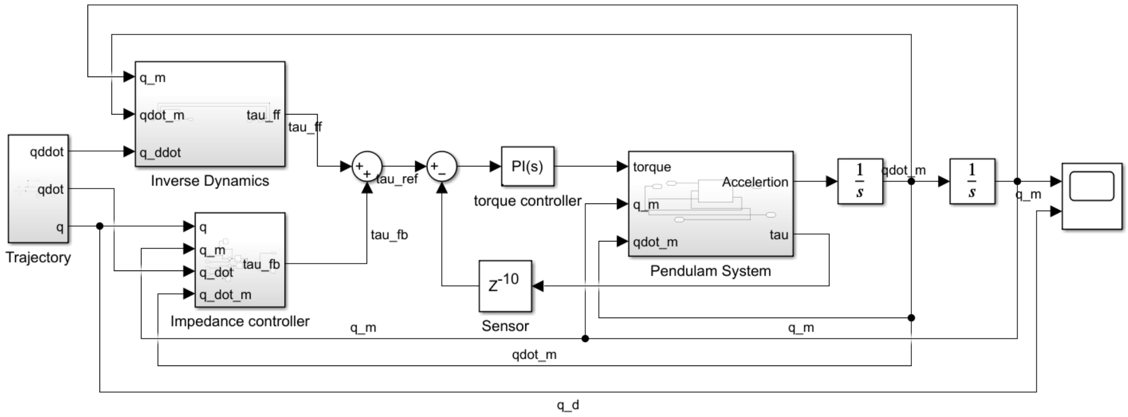


Figure 5.9: Simulink model of the control system.

The control system is tuned using a step response. In the process of tuning a cascade control system, such as a torque-based impedance control, the most common way is to initially fine-tune the inner-loop gains to maximize the bandwidth, followed by outer-loop gain tuning [67]. This iterative tuning process was executed through the trial and error method. Generally, expanding the bandwidth of the inner loop proves advantageous, facilitating a broader range of frequencies for real impedance to follow the desired trajectory. However, this comes with a trade-off, as it reduces the stability range for overall system impedance parameters [68]. In other words, as the inner loop becomes more responsive to rapid changes and high frequency components, the system becomes more responsive to disturbances and uncertainties that may exist in the environment or within the system itself. This increased sensitivity to high frequency dynamics can lead to a compromise in the overall stability of the impedance control system.

A strategy called velocity compensation is proposed to overcome this limitation and maximize the torque bandwidth [69]. This strategy shows that by keeping the torque controller gains constant, the bandwidth can be increased only by using velocity compensation.

In the implementation of the control system, a systematic tuning approach was followed to optimize the performance of the cascade control architecture. The inner torque control loop, responsible for generating the desired torque response, was initially tuned. Following this, an iterative process was employed to determine the outer impedance loop gains, resulting in proportional (P) and derivative (D) values of $K_\mathrm{p} = 45$ and $K_\mathrm{d} = 1.1$, respectively. The tuning process involved iterative adjustments to both the torque and impedance control loops, with stability analysis. With these controller gains, the system exhibits good transient behaviour, achieving a settling time of approximately $t_\mathrm{s} = 0.2\,\mathrm{sec}$, characterized by no overshoot and no undershoot. Furthermore, the steady-state error is reduced to zero at $t = 0.7\,\mathrm{sec}$, see Figure 5.10.

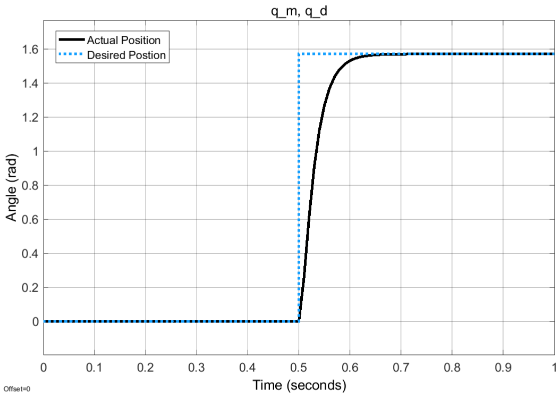
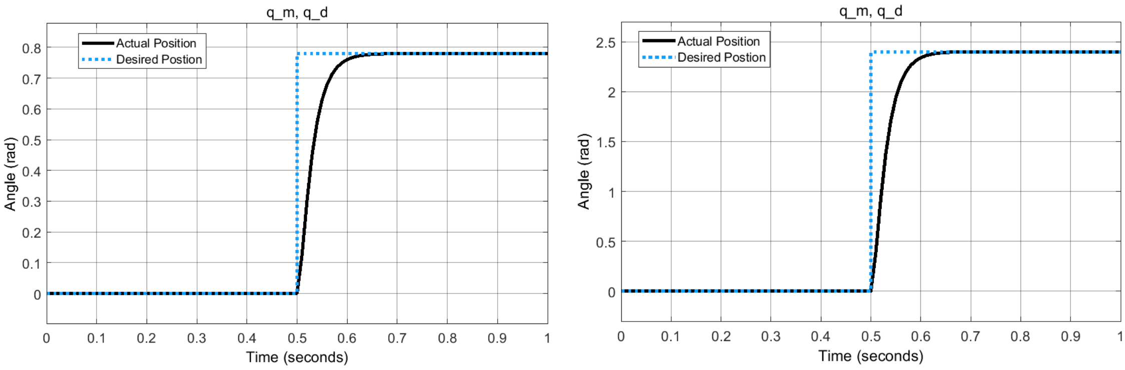


Figure 5.10: Step response for $\theta = 90°$ set point.

The performance of the system, characterized by the specified gains, was further verified using various set points, as illustrated in Figure 5.11.



(a) Step response for $\theta = \frac{\pi}{4}$ rad set point.

(b) Step response for $\theta = \frac{3\pi}{4}$ rad set point.

Figure 5.11: Step responses for$\theta = \frac{\pi}{4}$ rad and $\theta = \frac{3\pi}{4}$ rad set points.

To evaluate the performance of the torque controller and inverse dynamics model, a sinusoidal input with a frequency of 2 Hz and an amplitude of $\frac{\pi}{4}$ was used as the desired trajectory. As previously mentioned, an accurate model should generate torques capable of tracking the specified trajectory with minimal position errors. A more accurate model correlates with smaller deviations from the desired positions.

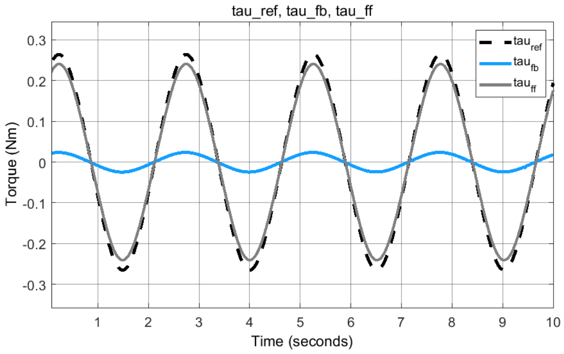


Figure 5.12: Simulation result with torque tracking.

Figure 5.12 demonstrates the close alignment between the feedforward torque, generated by the inverse dynamic model, and the reference torque. Additionally, it is evident that the feedback torque, derived from position error, remains low as anticipated. This observation highlights the accuracy of both the rigid body model and the torque controller.

In Figure 5.13, the desired and actual angles of the pendulum system are presented for the same sinusoidal input. This illustrates that the actual angle of the pendulum system closely follows the predefined trajectory, This shows that the developed control system is achieving accurate tracking or trajectory following under ideal circumstances (i.e. no noise, linear sensor).

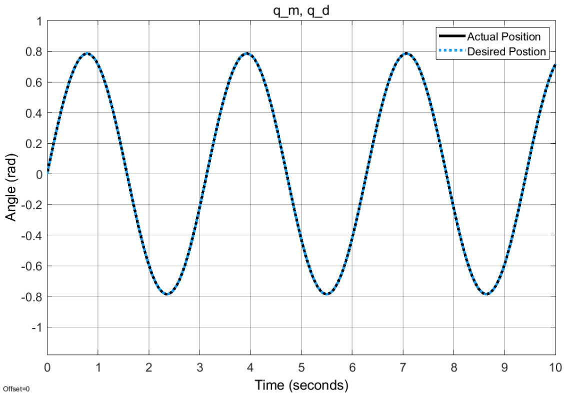
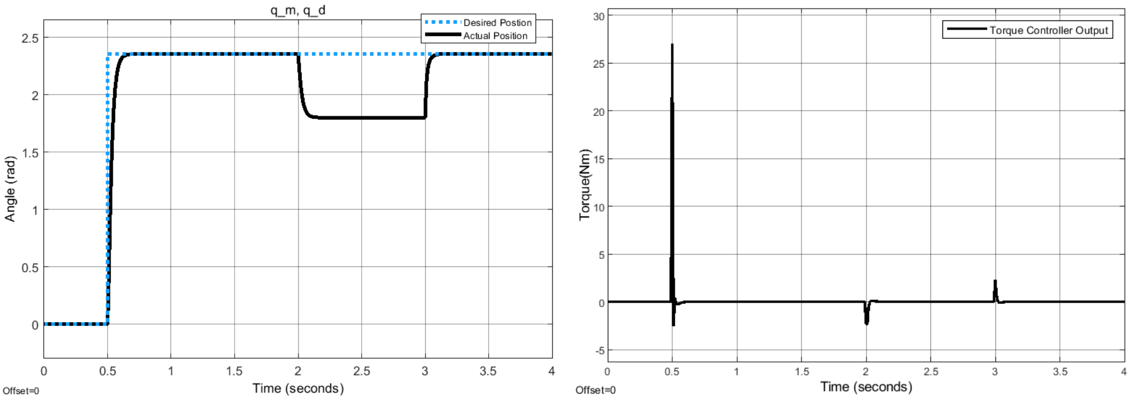


Figure 5.13: Simulation results with actual position and desired position.

5.5.3 Interaction force control

An external impulse was introduced to the system during a steady-state condition to assess the impact of external forces and observe the functionality of the interaction control in static interaction scenarios. The impulse was initiated one second after the system started, and it persisted for one second. Notably, the introduction of the impulse led to a deviation in the pendulum angle, maintaining a displacement from the desired angle until the external disturbance was removed. Subsequently, the pendulum smoothly returned to the desired angle without oscillations, demonstrating an anticipated and controlled response as shown in Figure 5.14. The disturbance signal is also shown in Figure 5.15.



(a) Actual position vs desired position of the system.

(b) Corresponding torque controller output.

Figure 5.14: Static interaction control with $K_{\mathrm{p}} = 45$ and $K_{\mathrm{d}} = 1.1$.

To explore the dynamic interaction control, a sinusoidal input with a frequency of 2 Hz and an amplitude of $\frac{\pi}{4}$ was used as the desired trajectory. A disturbance or an interaction was introduced to the system every 5 seconds with a pulse width of 1 second and with a delay of 2 seconds. It is worth mentioning that, in this dynamic interaction scenario, the disturbance was applied in the opposite direction compared to the static interaction case. This decision

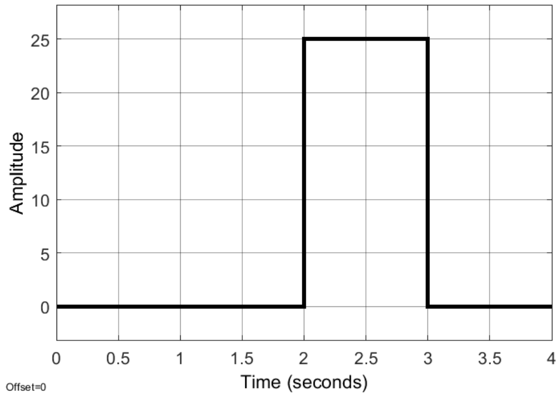
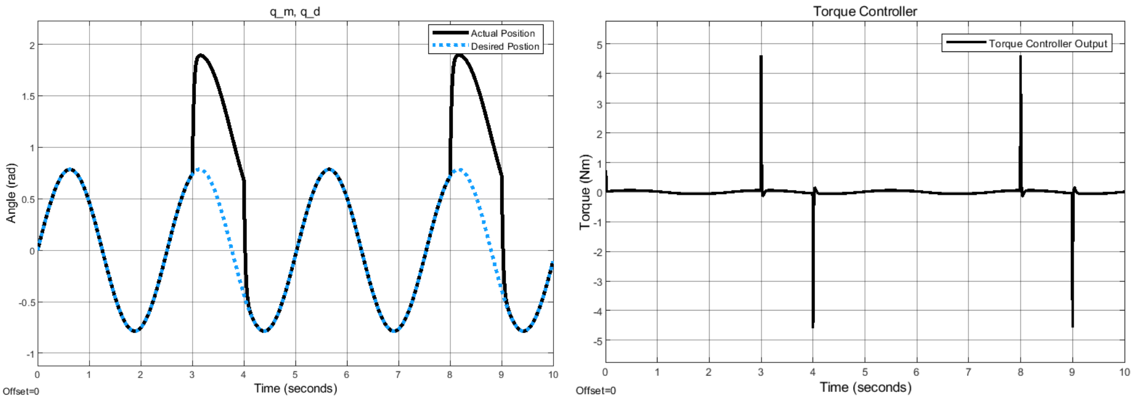


Figure 5.15: Disturbance signal for static interaction control.

was made to assess the system's response under varying conditions, specifically to investigate whether changing the direction of the external force results in a corresponding response.



(a) Actual position vs desired position of the system.

(b) Corresponding torque controller output.

Figure 5.16: Dynamic interaction control with $K_\mathrm{p} = 45$ and $K_\mathrm{d} = 1.1$.

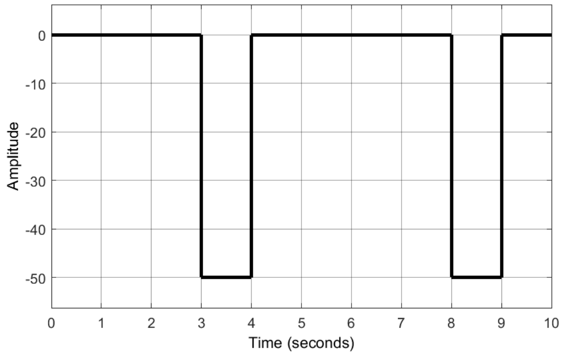


Figure 5.17: Disturbance signal for dynamic interaction control

The resulting trajectory of the pendulum system which is shown in the Figure 5.16 reveals the impact of interactions or disturbances, showcasing a distinct shift in the actual angle compared to the predefined trajectory. It can be seen that the actual angle lags behind the desired angle during the presence of interaction forces. However, upon the removal of these forces, the system closely aligns itself with the desired trajectory. Furthermore, Introducing disturbance in the opposite direction moved the system in the opposite direction compared to the static scenario as expected. The disturbance signal is also shown in Figure 5.17.

5.5.4 Impedance parameter investigation

The stiffness and damping coefficient values in impedance control are important parameters that influence the behaviour and performance of the control system. Stiffness determines how rigidly the system resists deformation or displacement when interacting with an external force or disturbance. Damping controls the rate at which the system dissipates energy and influences the system's ability to handle external disturbances smoothly without excessive oscillations. Achieving the right balance between stiffness and damping is crucial for effective impedance control. The optimal values depend on the specific application, the nature of interactions, and the desired system behavior. Different applications may require different combinations of stiffness and damping based on safety, precision, and performance requirements.

To investigate the impact of stiffness and damping coefficient values on interaction control, an analysis was conducted by varying these coefficients at two levels each. Specifically, for the stiffness coefficient, two levels (-1 and $+1$) were considered, corresponding to the previously tuned value of 45 and a higher value of 70. Similarly, the damping coefficient was varied at two levels, -1 and $+1$, representing the previously tuned damping value of 1.1 and a new value of 10. The selection of these new values resulted from several simulations, ensuring a good step response. For clarity, the parameter variations are summarised in the table.

Coefficient	**Level**	
	-1 (lower)	+1 (higer)
Stiffness coefficient	45	70
Damping coefficient	1.1	10

Table 5.3: Coefficient value levels.

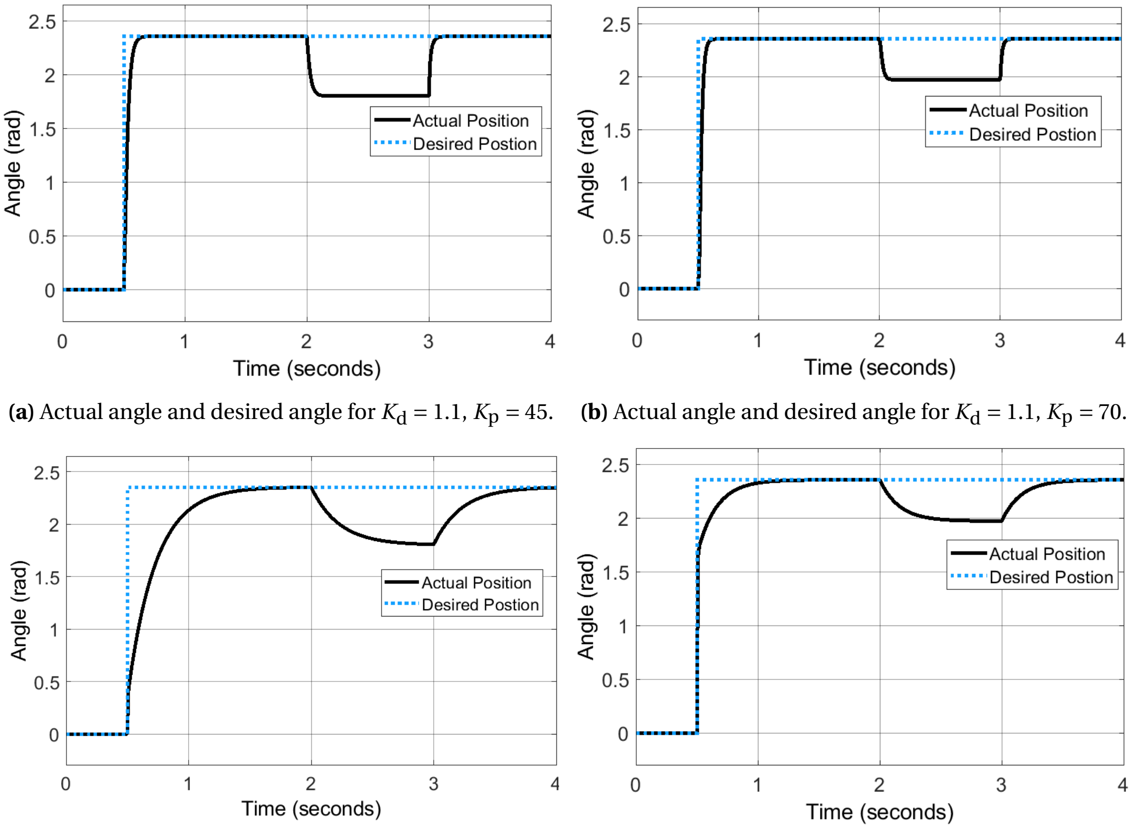
The simulations included both static and dynamic scenarios, with results being analyzed separately. Each scenario comprised four simulations. A detailed simulation plan is summarized in the table for enhanced clarity, with distinct names assigned to each sub-scenario to facilitate understanding and analysis.

	Scenario	**Stiffness coefficient level**	**Damping coefficient level**
	A	-1	-1
Static	B	+1	-1
Interaction	C	-1	+1
	D	+1	+1
	E	-1	-1
Dynamic	F	+1	-1
Interaction	G	-1	+1
	H	+1	+1

Table 5.4: Simulation plan with different coefficient values for static and dynamic interaction.

Static interaction

In the first scenario (Scenario A), both the damping coefficient and stiffness coefficient were maintained at the lower level (K_d = 1.1, K_p = 45, respectively). In the second scenario (Scenario B), the stiffness coefficient was increased to the higher level (K_p = 70) to explore its influence on the overall system dynamics, while maintaining the damping coefficient at the same level as in Scenario A. Moving to the third scenario (Scenario C), the damping coefficient was increased to the upper level (K_d = 10), while keeping the stiffness coefficient at the lower level (K_p = 45). In the fourth scenario (Scenario D), both the damping coefficient and stiffness coefficient were increased to higher values (K_d = 10, K_p = 75, respectively) to investigate their combined impact on the overall system dynamics during the interaction. It is noteworthy that, in all these scenarios, the amplitude of the interaction force remained constant.



(**a**) Actual angle and desired angle for $K_\mathrm{d} = 1.1$, $K_\mathrm{p} = 45$.

(**b**) Actual angle and desired angle for $K_\mathrm{d} = 1.1$, $K_\mathrm{p} = 70$.

(**c**) Actual angle and desired angle for $K_\mathrm{d} = 10$, $K_\mathrm{p} = 45$.

(**d**) Actual angle and desired angle for $K_\mathrm{d} = 10$, $K_\mathrm{p} = 70$.

Figure 5.18: Static interaction control scenarios for different damping and stiffness coefficients.

Analysis of the obtained results reveals that the implemented control architecture effectively handles interaction forces, ensuring both safety and stability in the system's response. Furthermore, the observed system behavior shows a significant influence from the chosen impedance parameters.

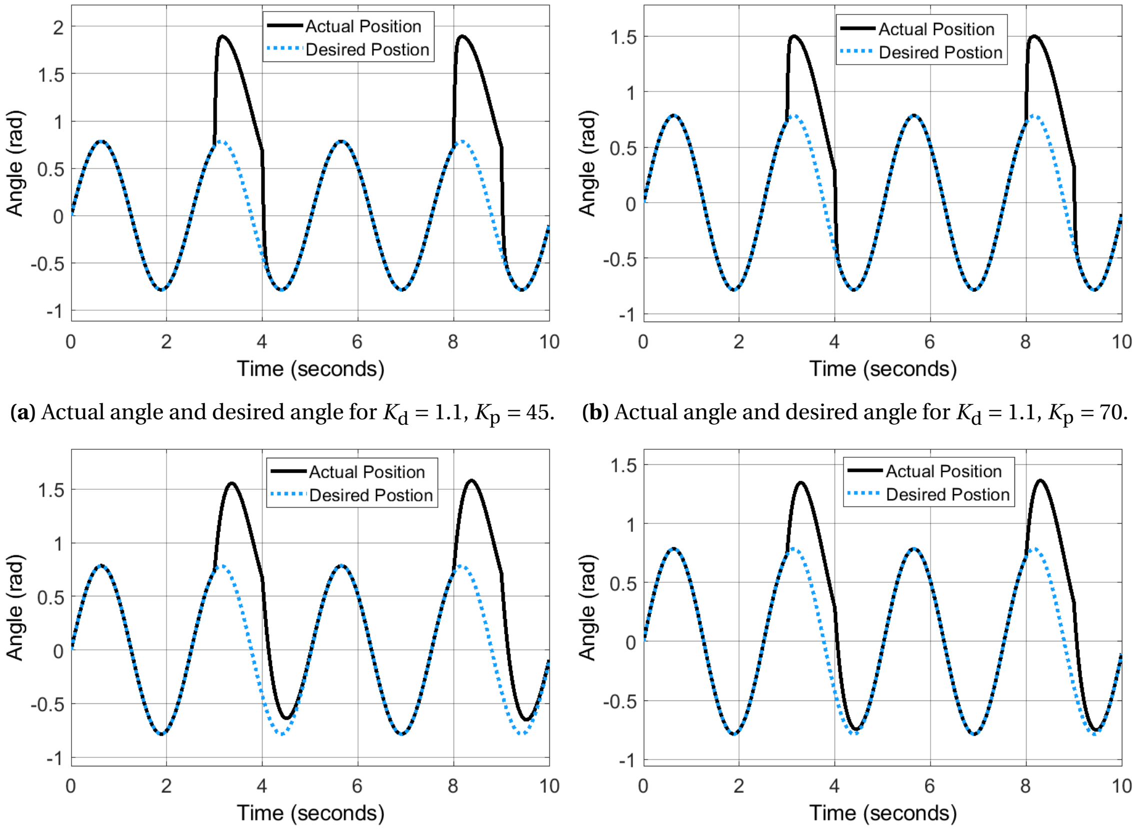
Dynamic interaction control

To investigate the influence of stiffness and damping on dynamic interaction control, a similar procedure was applied to Scenarios E, F, G, and H, mirroring the approach used for the static scenarios. Scenario E involved lower damping and stiffness coefficients, Scenario F featured higher stiffness while maintaining lower damping, Scenario G had higher damping while keeping stiffness lower, and Scenario H included both coefficients at higher levels. Importantly, the interaction force remained consistent across all scenarios.

For interaction simulation, a pulse generator was utilized with a 3-second delay, repeating every 5 seconds, and a pulse length of 1 second. The figure illustrates the system response (actual angle) of the pendulum plotted against the desired angle for each scenario.

As depicted in Figure 5.18 and Figure 5.19 the system demonstrates similar responses in both static and dynamic scenarios when subjected to interactions, specifically when maintaining same damping and stiffness coefficients.

In Scenario A and E, characterized by lower damping and lower stiffness, as shown in Figure 5.18a and Figure 5.19a, can be seen as a more flexible system that is more responsive to



(a) Actual angle and desired angle for $K_d = 1.1$, $K_p = 45$.

(b) Actual angle and desired angle for $K_d = 1.1$, $K_p = 70$.

(c) Actual angle and desired angle for $K_d = 10$, $K_p = 45$.

(d) Actual angle and desired angle for $K_d = 10$, $K_p = 70$.

Figure 5.19: Dynamic interaction control scenarios for different damping and stiffness coefficients.

external force and displays faster reactions to disturbances. The system demonstrates a rapid response when interaction occurs suggesting that the lower stiffness allows for greater movement and the lower damping permits quicker adjustments, potentially resulting in a less controlled trajectory.

Moving to Scenario B and F, where higher stiffness is combined with lower damping, In Figure 5.18b and Figure 5.19b it can be seen as a more rigid, (less flexible) system than scenario A and H. The higher stiffness reduces deformation caused by interaction forces, resulting in a more stable and controlled trajectory, also with the potential for a faster response.

In Scenario C and G, featuring higher damping and lower stiffness, shown in Figure 5.18c and Figure 5.19c, we can see that the system exhibits smoother responses to external disturbances. The higher damping coefficient provides a more gradual adjustment not only when the interaction force has occurred but also when it is removed. This smoother response, however, comes with a slight trade-off in settling time. Higher damping is preferred in scenarios prioritizing safety, as it results in less reactive force on the environment.

Finally, in Scenario D and H, with both higher damping and higher stiffness, as illustrated in Figure 5.18d and Figure 5.19d, we can see the system combines the benefits of increased rigidity and smoother responses. The higher stiffness provides a more rigid response, while the increased damping ensures controlled adjustments, resulting in a more balanced response to interaction forces.

Overall, these simulations offer insights into how variations in damping and stiffness coefficients influence the dynamic behaviour of the torque-based impedance control system during interaction scenarios. Selecting the optimal values for the damping and stiffness coefficients in an interaction control system involves a trade-off between various factors such as system dynamics, safety requirements, and task specifications. For our 1 DOF pendulum, characterized by its less rigid and more flexible nature, the findings suggest that higher stiffness becomes advantageous for resisting large deformations. Furthermore, considering the importance of safe interactions, a higher damping coefficient is also preferred. These decisions hold significant importance as they lay the foundation for the development and tuning of the control system intended for use in our demonstration setup with the 3D printed torque sensor. It is essential to note that the simulated control architecture assumed an ideal torque sensor, a condition not reflective of the real setup. The primary rationale behind using an ideal sensor was to assess the performance and functionality of the control architecture development.

5.6 Stability Analysis

For stability analysis, working in the Laplace domain using transfer functions is a common and convenient approach. However, our system's dynamics are inherently nonlinear, falling into the category of a non-linear time invariant (NTI) system. Such systems, like our pendulum, cannot be fully expressed in the Laplace domain. The Laplace transform is used to convert NTI systems into LTI at specific initial conditions.

In the case of our pendulum, linearization is essential to transform its NTI nature into a more manageable LTI form. This involves approximating the non-linear response of the pendulum, which is a sinusoidal function of its angle θ, through a linearized version. Specifically, we apply the small angle approximation by considering $\sin\theta \approx \theta$. From Figure 5.20 we can see there is a stable equilibrium at a vertically downward position and it repeats after every 2π rad . The linearization process centres around the vertically downward equilibrium position and assumes that the system remains in a small neighbourhood around this equilibrium.

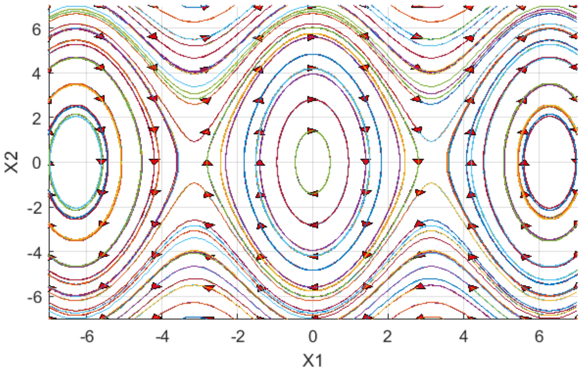


Figure 5.20: Phase portrait of the pendulum system

The linearization is done manually and using the Simulink model linearizer around the equilibrium. Both methods yield the same 2nd order system transfer function in the Laplace domain. The manual calculation is given below.

Let's find the linearized transfer function around the equilibrium point in the Laplace domain using the continuous time transfer function of the plant in the time domain,

$$\tau = (ml_{\text{com}}^2 + I')\ddot{q} + m \cdot g \cdot l_{\text{com}} \sin(q) + B\dot{q} \tag{5.34}$$

Assume $(ml_{\text{com}}^2 + I') = M$ and $m \cdot g \cdot l_{\text{com}} = K$ (K and M are constants).

Then,

$$\tau = M\ddot{q} + K\sin(q) + B\dot{q} \tag{5.35}$$

Let's assume $x_1 = q$ and $x_2 = \dot{q}$ Then, $\dot{x}_1 = \dot{q} = x_2$ and $\dot{x}_2 = \ddot{q}$

Using the state-space matrices,

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} \\ \mathbf{y} &= \mathbf{Cx} + \mathbf{Du} \end{aligned} \tag{5.36}$$

Where:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{\tau}{M} - \frac{Bx_2}{M} - \frac{K\sin(x_1)}{M} \end{bmatrix}, \quad \mathbf{u} = [\tau], \quad \mathbf{y} = [q]$$

The state-space matrices **A**, **B**, **C**, and **D** are given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{K}{M} & -\frac{B}{M} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{D} = [0]$$

Now using the state-space matrices **A**, **B**, **C**, and **D** we can derive the open loop transfer function $T(s)$ of the plant as below,

$$G(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D} \tag{5.37}$$

$$G(s) = \frac{1}{Ms^2 + Bs + K} \tag{5.38}$$

Substituting parameter values defined in Table 5.2 we can find the linearized transfer function $T(s)$ of the system around equilibrium point (0,0) as below,

$$G(s) = \frac{120.2}{s^2 + 1.202s + 9.058} \tag{5.39}$$

The transfer function derived from the Simulink linearization yielded the same formulation.

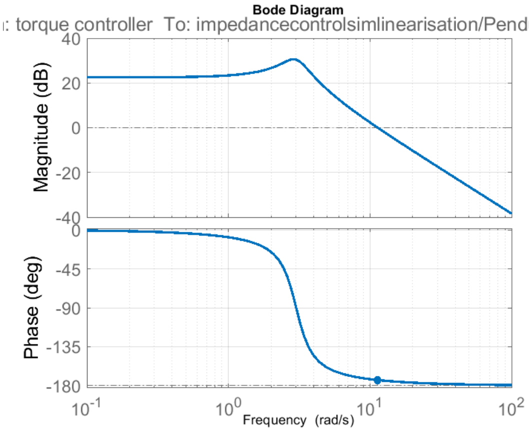


Figure 5.21: Bode plot of linearized system

In exploring the closed-loop behavior, an insightful analysis involves examining the frequency response of the open-loop system. For instance, identifying frequencies where the open-loop system produces a gain of 0 dB and a phase of −180° is important, as these conditions may lead to undamped oscillations in the closed-loop system. A careful inspection of the Bode plot ensures that the phase isn't −180° for frequencies corresponding to a magnitude of 0 dB. This observation assures that the closed-loop system avoids undamped oscillations.

To assess closed-loop stability, another effective approach involves examining the Bode plot under the condition that the open-loop system $G(s)$ is both minimum-phase and stable. A minimum-phase system is a system where all the zeros (roots of the numerator polynomial) are located inside the left-half plane of the complex plane, and all the poles (roots of the denominator polynomial) are also in the left-half plane. A stable system is a system where all the poles are located in the left-half plane.

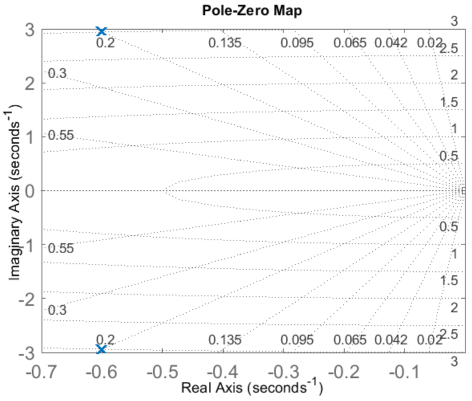


Figure 5.22: Pole-zero map of the open loop system

In our case, as shown in the pole-zero map in Figure 5.22, the presence of complex conjugate poles with real parts equal to −0.6010 confirms the stability of the system. The absence of zeros, coupled with all poles possessing negative real parts, establishes the system as a minimum-phase configuration. Therefore we can determine closed-loop stability from the bode plot, observing that the phase remains below −180° at frequencies corresponding to a magnitude of 0 dB. Therefore we can conclude that the closed-loop system is stable around the equilibrium point. This can be further seen from the step response of the closed-loop system.

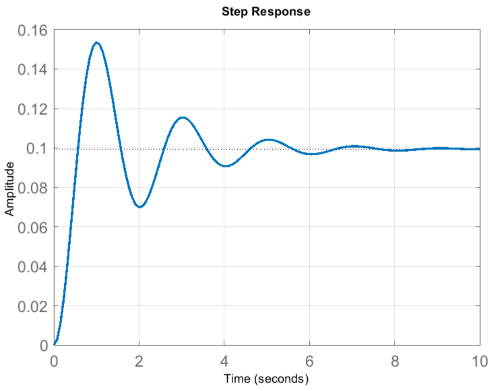


Figure 5.23: Step response of the closed-loop system

5.7 Conclusion

In this chapter, a comprehensive exploration of the implementation and simulation of torque-based impedance control for a 1 DOF pendulum system was presented. Through simulations conducted in MATLAB and Simulink, we verified the effectiveness of the control architecture in regulating the pendulum's behavior during static and dynamic interactions.

The initial phase of the simulation focused on validating the plant dynamics derived from the system's equations of motion. The pendulum system's behavior was simulated in MATLAB, showcasing the expected characteristics when released from an initial angle. The Simulink model further confirmed the accurate representation of the pendulum's dynamics, aligning with theoretical expectations. Subsequently, the torque-based impedance control architecture was implemented and analyzed in both MATLAB and Simulink. The cascade control structure, consisting of an inner torque loop and an outer impedance loop, was detailed. The simulation results demonstrated the effectiveness of the control architecture in achieving accurate trajectory tracking and stable behavior.

Furthermore, an investigation into the influence of stiffness and damping coefficients on the system's performance was conducted. The analysis considered various scenarios with different combinations of these coefficients in both static and dynamic interaction setups. The results highlighted the trade-offs involved in selecting optimal values for stiffness and damping, emphasizing the need for a balanced approach based on safety, task requirements, and system dynamics.

Overall, the developed torque-based impedance control system demonstrated promising capabilities in regulating interaction torques and ensuring stable behavior in the simulated 1 DOF pendulum system. The insights gained from these simulations laid the groundwork for the implementation of a demonstration setup to explore the potential of interaction control with 3d printed torque sensors in an impedance control framework.

6 Demonstration Setup

6.1 Introduction

This chapter is dedicated to establishing the demonstration setup and investigating the practical application of the implemented control algorithm. It will provide a comprehensive overview of the demonstration setup, offering insights into the specific configuration and detailing each piece of equipment used. Experiments have been conducted to analyze the system's behaviour in both static and dynamic interaction control scenarios. Subsequent sections focus on the thorough analysis and post-processing of the gathered data during these experiments. The experimental results are then presented, with a comparative discussion against the simulation results. The objective is to validate sensor performance and assess the potential of the 3D printed torque sensor in interaction control scenarios.

6.2 Setup and components

This phase involves the establishment of the demonstration setup, where the developed control strategy is applied in a real-world context using the 3D printed torque sensor, aiming to explore the potential of interaction control in real-world scenarios. This phase unfolds within the boundaries in the NIFTy laboratory environment. The fundamental components used in the setup are:

1. Sensor integrated pendulum system.
2. Mechaduino [70].
3. Power supply unit 1 V to 30 V.
4. Load cell amplifier - HX711 [71].
5. Wheatstone bridge circuit.
6. A PC with the required software:
   - Arduino.
   - MATLAB.

6.2.1 Sensor integrated pendulum system

The fabrication process of the sensor integrated pendulum system is already explained in Chapter 3. The below illustrates the successfully 3D printed sensor integrated pendulum system used in the demonstration setup.

6.2.2 Mechaduino

Mechaduino, an affordable servo system focused on NEMA17 stepper motors, stands out as an open-source solution that serves to a spectrum of control modes, including position, torque, velocity, and custom configurations [70]. Developed by Tropical Labs, this self-contained motion control platform operates on the Arduino Zero architecture, providing compatibility with Arduino programming. Mechaduino is equipped with the SAMD21G18A microcontroller that facilitates precise control of a stepper motor. It incorporates an encoder AS5047D, which is used for closed-loop control by providing accurate feedback on the motor's position. The integration of the printed circuit board (PCB) directly onto the rear of the stepper motor, as illustrated in the Figure 6.2, enables a connection for direct encoder readouts.

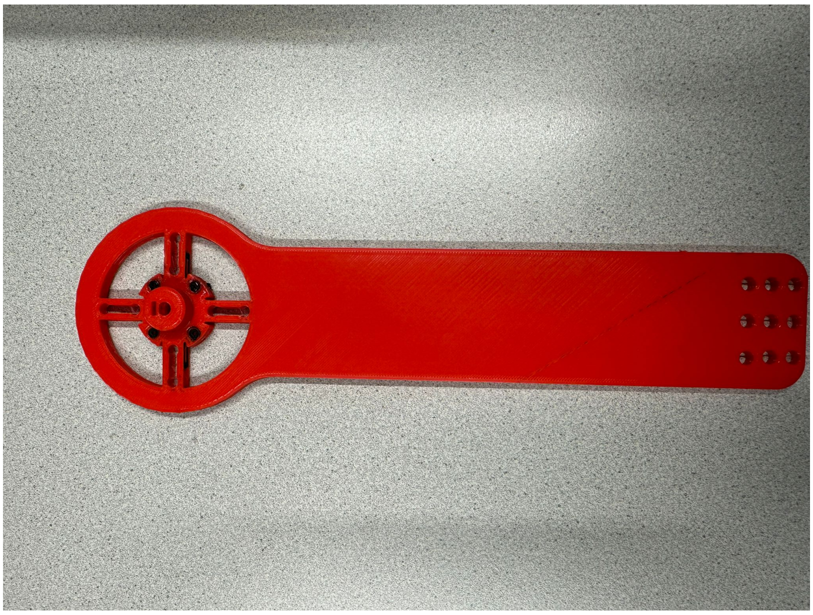


Figure 6.1: The sensor integrated 1 DOF pendulum system.

Within this application, the focus was on using torque control mode, where the user directs the output torque or sets the torque setpoint for the motor. Returning to the dynamics description of the pendulum system, the calculated setpoint torque was derived from the desired position of the pendulum. This torque value is then converted to the input signal u for the motor, where u is expressed as an 8 bit value, ranging from 0 to 255

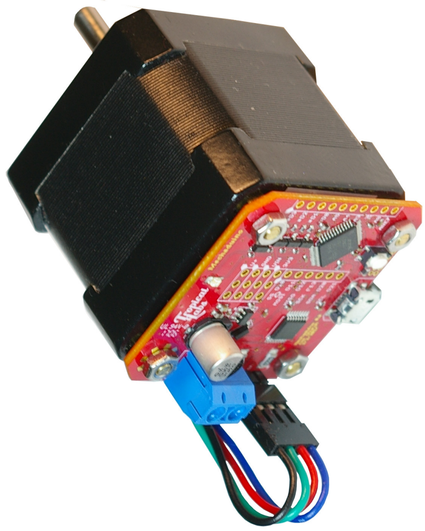


Figure 6.2: Mechaduino 0.2 [72].

6.2.3 Load cell amplifier - HX711

During the sensor characterization, it was observed that the strain gauges exhibited minimal changes in resistance, in response to low torque inputs. This indicates the need of amplification of the sensor response to accurately capture sensor response for small torque variations. Addressing this need, the Load Cell Amplifier emerges as a compact breakout board designed for the HX711 integrated circuit (IC), facilitating the reading and amplification of sensor responses within a Wheatstone bridge configuration. Through integration with the Mechaduino, this amplifier is capable of capturing and amplifying the changes in the resistance of strain gauges of the pendulum system. This is possible because the HX711 serves as an analog-to-digital converter (ADC) and signal amplifier. The guidance provided in Sparkfun's hookup guide ensured the effective connection of the amplifier within the pendulum system [71].

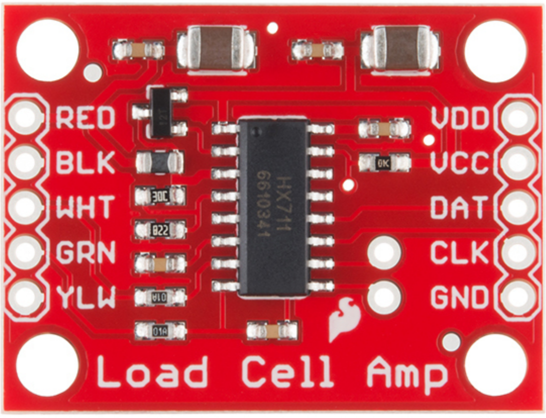


Figure 6.3: Load cell amplifier - HX711

6.2.4 Wheatstone Bridge circuit

To measure resistance changes caused by external forces or torques on strain gauges, the Wheatstone bridge configuration is used. The Wheatstone bridge is commonly used for the precise measurement of small changes in resistance, particularly in applications such as strain gauge sensors. This bridge arrangement becomes particularly useful when dealing with the dynamic changes in the electrical resistance of the strain gauges. The typical Wheatstone bridge, depicted below in Figure 6.4, is comprised of four resistive arms, and an excitation voltage, V_{EX}, is applied across the bridge [73].

V_0, the output voltage of the bridge, can be expressed as:

$$V_0 = \left[\frac{R_3}{R_3 + R_4} - \frac{R_2}{R_1 + R_2} \right] \cdot V_{\text{EX}} \tag{6.1}$$

This equation reveals that when the ratio of R_1 to R_2 equals the ratio of R_4 to R_3, the output voltage V_0 becomes zero, indicating a balanced state for the bridge. In this equilibrium condition, any change in the resistance of any arm of the bridge will lead to a nonzero output voltage.

In considering the Wheatstone bridge configurations for our pendulum system, three common options were explored. Quarter, half, and full bridges, using one, two, and four strain gauges respectively. Initially, the plan was to use the full bridge configuration for our pendulum system which is equipped with four strain gauges, as it offers increased sensitivity compared to the half and quarter bridge configurations.

However, a decision was later made to transition to the half-bridge configuration, incorporating two potentiometers. The primary motivation for transitioning to the half-bridge configura-

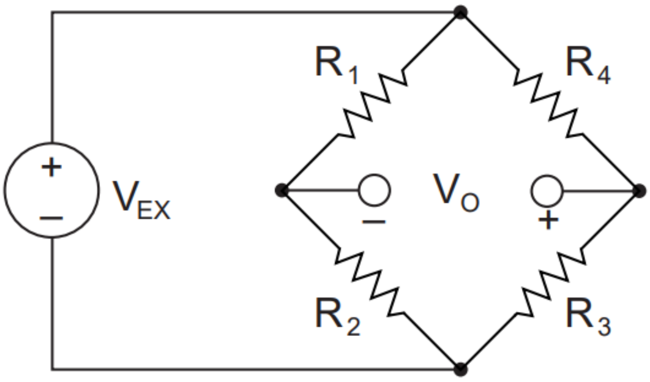
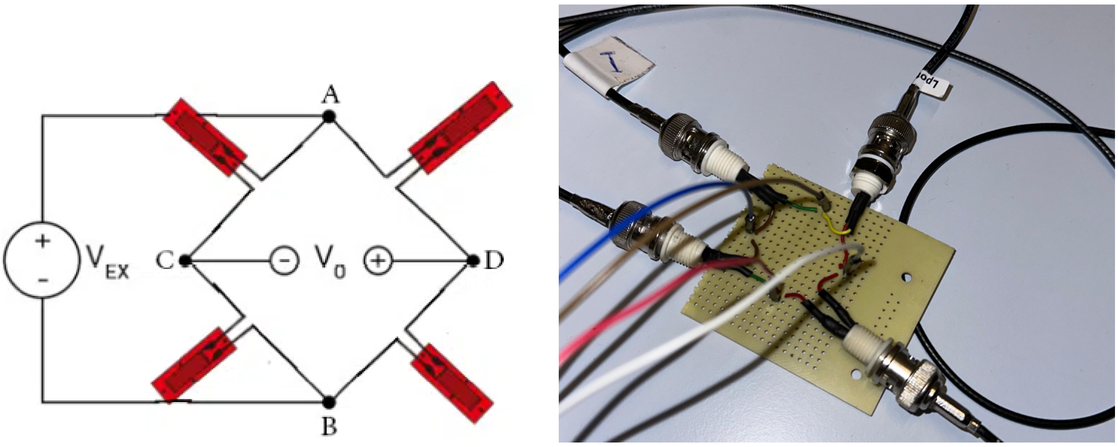


Figure 6.4: Wheatstone Bridge [73]



(a) Wheatstone full bridge configuration.

(b) Implemented Wheatstone full bridge circuit.

Figure 6.5: Wheatstone Full bridge.

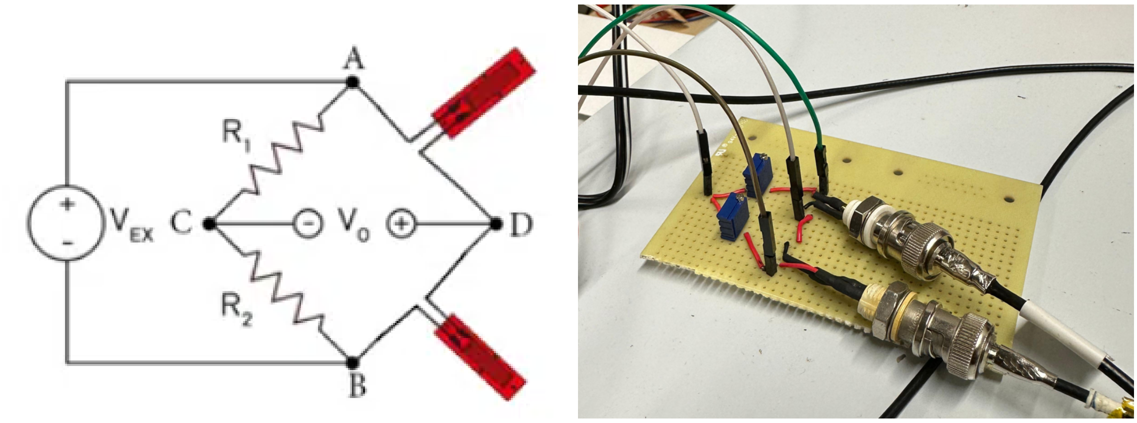
tion stemmed from the requirement in the Wheatstone bridge circuits that assumes an initially balanced bridge, resulting in zero output when no strain is applied. This is important for our control system, ensuring that the bridge voltage matches zero when no torque is applied. However, in practical scenarios, challenges arise due to resistance tolerances leading to some initial offset voltage.

Handling this initial offset voltage can be achieved through two approaches. The first involves the use of a special offset-nulling or balancing circuit to adjust the bridge's resistance, rebalancing it to zero output. The second approach entails measuring the initial unstrained output of the circuit and compensating for it in software. With the full bridge configuration, the former is not feasible as the resistance of the strain bridge cannot be altered. Furthermore, the latter approach appears challenging with the integration of the amplifier.

Therefore, the half-bridge configuration was chosen for its practical advantages, enabling us to balance the bridge using potentiometers. This choice was made for the sake of simplicity over the full bridge configuration.

6.2.5 PC with the required software

The control and monitoring of the demonstration setup are facilitated by a computer equipped with essential software.



(a) Wheatstone half bridge configuration.

(b) Implemented Wheatstone half bridge circuit.

Figure 6.6: Wheatstone half bridge.

Arduino: The Arduino programming environment facilitates the programming of the AVR chip on the Mechaduino, enabling the implementation of the control algorithm. The Mechaduino, equipped with a microcontroller based on the Arduino Zero architecture, interprets and executes control signals. These signals command the servo motor to execute torque commands, using the developed control algorithm. The Arduino uses the HX711 Arduino library to read the Wheatstone bridge sensor outputs. The library ensures the measurement and amplification of the sensor responses, which are important for the control system.

MATLAB: MATLAB is used for data acquisition, analysis, and visualisation, providing a comprehensive understanding of the system's behaviour based on experimental results. Also used to compare real world data with simulation results, facilitating an assessment of the control algorithm's performance.

Figure 6.7 shows a simple diagram of the setup.

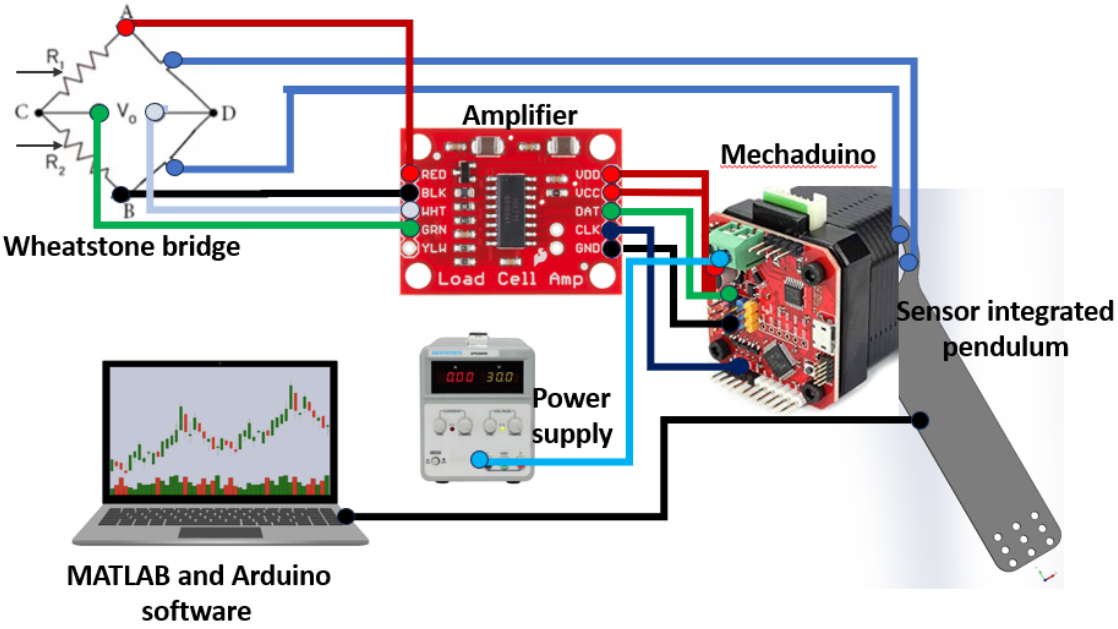


Figure 6.7: Diagram of the demonstration setup.

6.2.6 Visual Representation of the Demonstration Setup

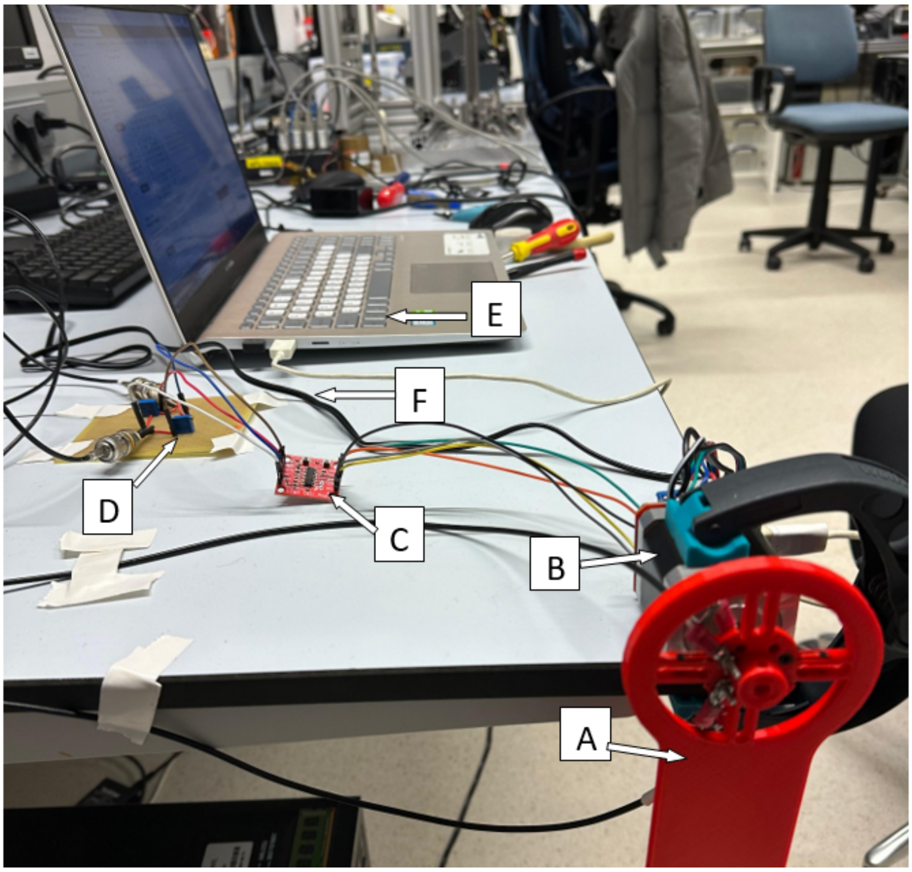


Figure 6.8: Overview of the demonstration setup (A: Sensor integrated pendulum, B: Mechaduino, C: Amplifier, D: Wheatstone bridge, E: PC with software, F: power cable).

The illustrated Figure 6.8 provides a visual representation of the implemented setup. In this configuration, the pendulum is firmly attached to the shaft of the stepper motor, as visible in the figure. Power is supplied to the amplifier through the microcontroller. The sensor readouts, obtained from the Wheatstone bridge configuration, are connected to the amplifier. Subsequently, these readings are recorded in MATLAB, paving the way for in-depth analysis in subsequent stages.

6.3 Experiments

6.3.1 Mechaduino control architecture

The control algorithm implemented in Arduino follows the architectural principles explained in Chapter 5, illustrated in Figure 5.4. This algorithm can be categorized by its two primary functions: impedance control and sensor signal processing.

A first-order low-pass filter typically, represented by the transfer function $F(s)$,

$$F(s) = \frac{\omega_0}{s + \omega_0} \tag{6.2}$$

where,the cutoff frequency, denoted as ω_0, has been selected as 5 Hz, by analysing the frequency spectrum of the sensor responses as shown in Figure 6.9.

While the transfer function derived from this cutoff frequency is suited for continuous-time systems, it is not directly applicable to real-time signal processing on the Mechaduino plat-

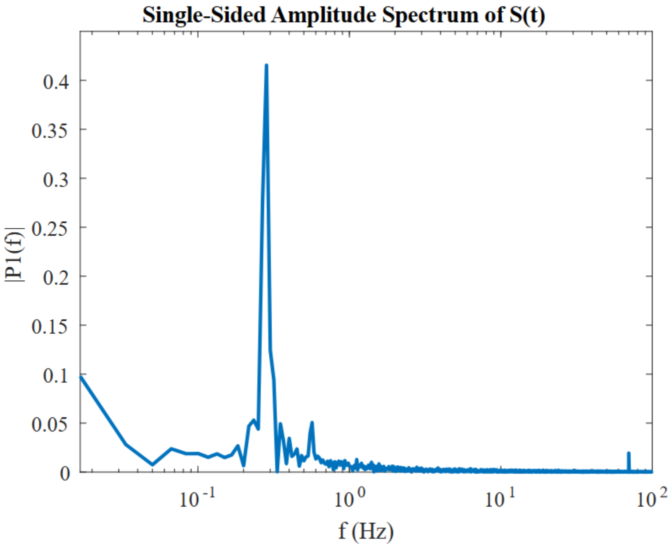


Figure 6.9: Frequency spectrum of the sensor response.

form. To adapt the filter for practical implementation on the Mechaduino, it is necessary to transform the continuous-time transfer function into its discrete-time equivalent. This is essential because digital implementations of filters operate in discrete time. Although real-time filtering occurs in the time domain, the conversion to a constant coefficient difference equation is also necessary for effective implementation on the Mechaduino. This implementation was done in MATLAB and the resultant constant coefficient difference equation obtained was used in Arduino to filter the sensor response.

$$y[n] = 0.9691y[n-1] + 0.01555x[n] + 0.0155x[n-1] \tag{6.3}$$

where,

y : filtered signal. n : discrete time index x : filter input signal.

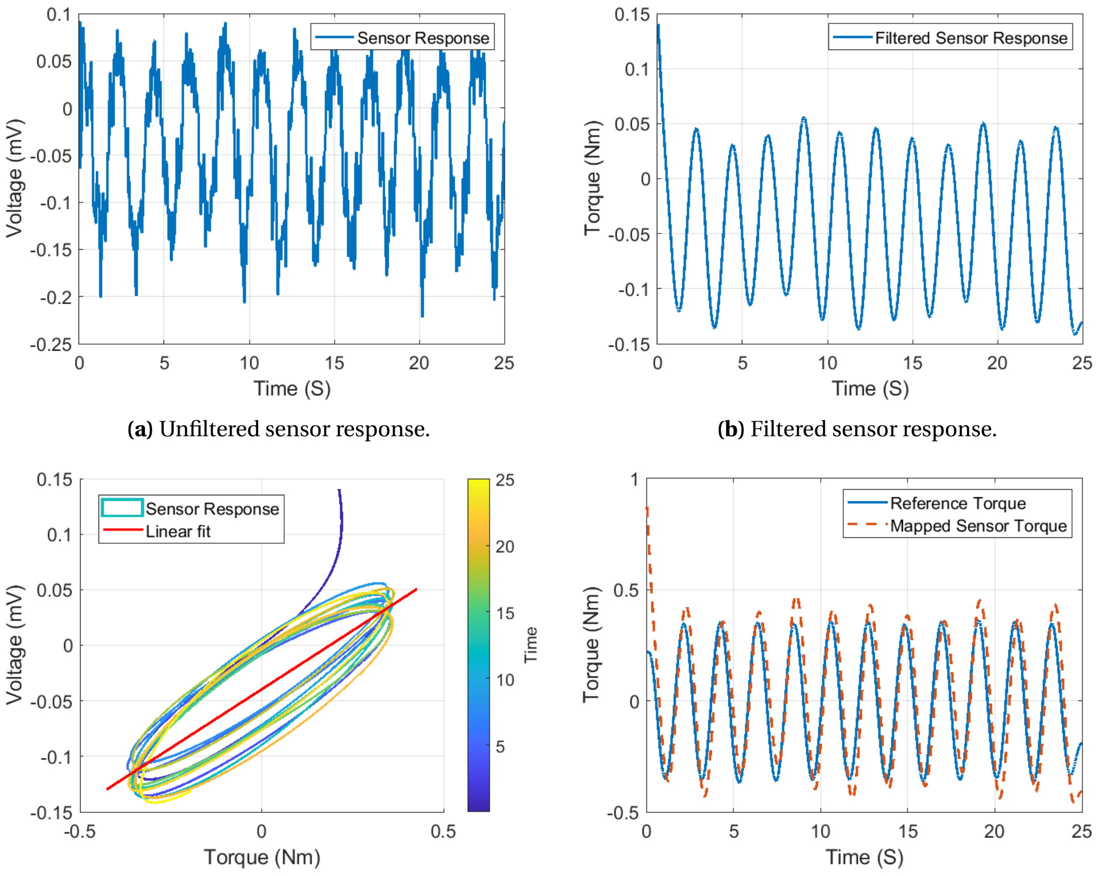
This Mechaduino control algorithm is presented in Appendix D for reference.

6.3.2 Sensor Performance Evaluation

The experiments were conducted for both static and dynamic interaction control scenarios. The initial phase involved assessing the performance of the torque sensor without any external interactions. Subsequently, interaction dynamics were introduced to the system to investigate the potential and effectiveness of the 3D printed torque sensor for interaction control.

In the torque sensor performance evaluation experiment, a sinusoidal input with a frequency of 1 Hz and an amplitude of $\frac{\pi}{4}$ was used as the desired trajectory. The low pass filter obtained in Equation 6.3 was used to remove the high-frequency noise of the sensor.

A series of experiments was conducted to validate the sensor's behavior with the characteristics observed during the characterization phase. To assess the torque sensor's performance further, the sensor response was recorded in MATLAB against the reference torque. The findings presented in this section are the result of the analysis of four experiments with the same sinusoidal input with a frequency of 1 Hz and an amplitude of $\frac{\pi}{4}$ rad . The results shown below are based on only four experiments, conveniently named as Experiment 1 to Experiment 4 for clarity.



(a) Unfiltered sensor response.

(b) Filtered sensor response.

(c) Sensor response against input torque.

(d) Reference torque and mapped sensor torque.

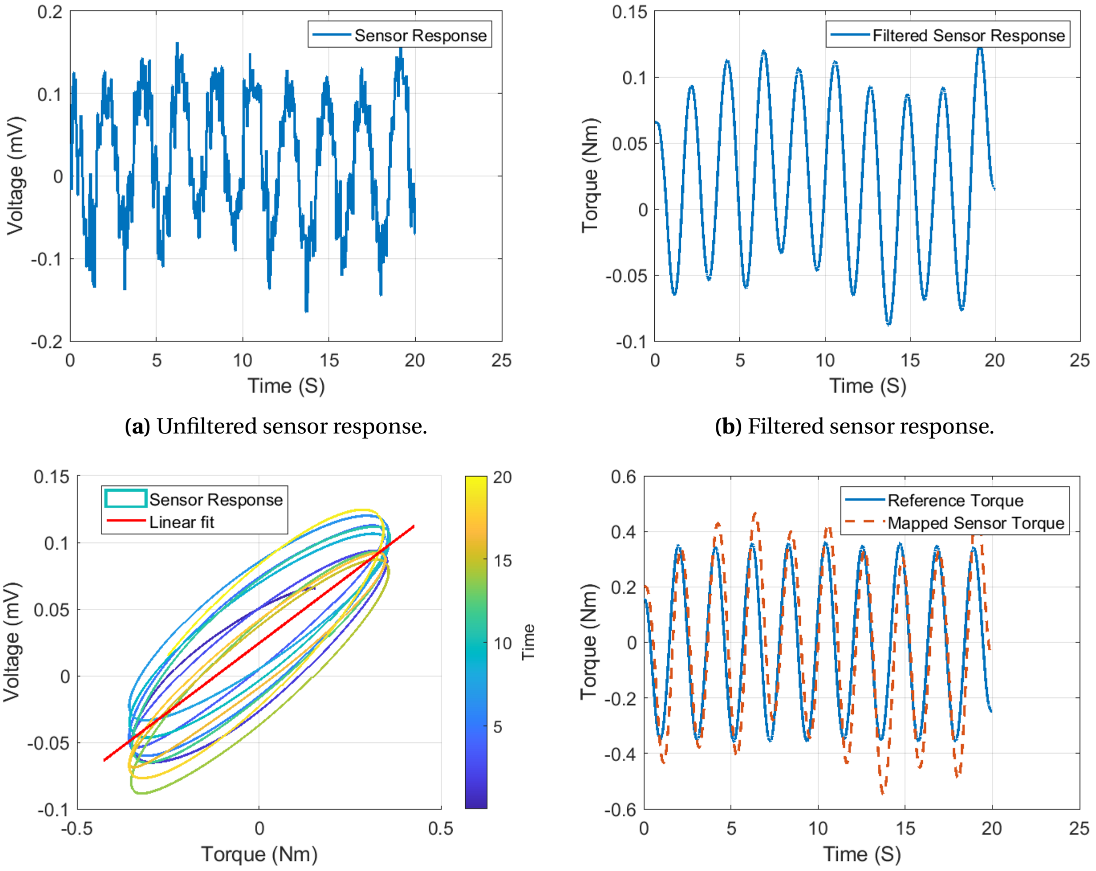
Figure 6.10: Experiment 1 results for sinusoidal input with a frequency of 1 Hz and an amplitude of $\frac{\pi}{4}$.

Experiments 1 and 2 yielded positive outcomes, illustrating the pendulum system sticks to the anticipated behavior within a sinusoidal trajectory featuring an angle close to $\frac{\pi}{4}$ rad , but with some vibrations. The experimental data from these experiments were simultaneously captured in MATLAB and used for further analysis, as demonstrated below.

Experiment 1 validates that the sensor's behavior aligns with the observed characteristics during the characterization phase. In Figure 6.10a, the raw sensor response voltage is depicted. Figure 6.10b showcases the filtered response, and Figure 6.10c illustrates the relationship between this filtered voltage and torque over time. The presence of drift and hysteresis behavior is evident in Figure 6.10c. Furthermore, Figure 6.10d presents the torque that is calibrated using the linear fit derived in the characterisation against the reference torque which is calculated using inverse dynamics torque and impedance torque, demonstrating that the mapped torque closely tracks the reference. These visualizations affirm the expected functionality of our sensor within the control system.

To assess the repeatability of the sensor's performance, additional experiments were conducted with the same sinusoidal input(see Figure 6.11. Experiment 2 exhibits behaviour similar to the previously described Experiment 1, though with a notable increase in drift over time, as evident in Figure 6.11d. The mapped torque gradually diverges from the reference torque as time progresses. This may be due to material fatigue or varying temperatures after performing multiple experiments

The sensor responses obtained from experiments 3 and 4 exhibit distinct behaviours compared to the initial two experiments. In experiment 3, as shown in Figure 6.12a and in experiment 4 as



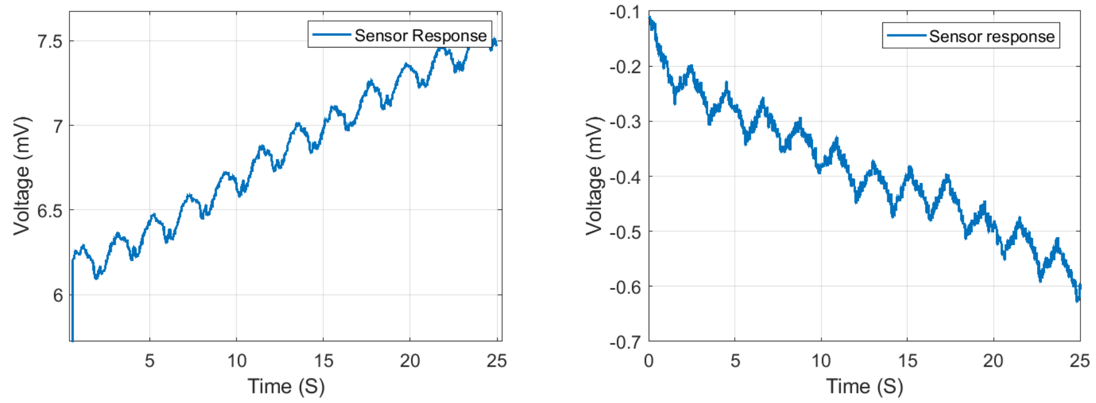
(a) Unfiltered sensor response.

(b) Filtered sensor response.

(c) Sensor response against input torque.

(d) Reference torque and mapped sensor torque.

Figure 6.11: Experiment 2 results for sinusoidal input with a frequency of 1 Hz and an amplitude of $\frac{\pi}{4}$.



(a) Sensor response for experiment 3.

(b) Sensor response for experiment 4.

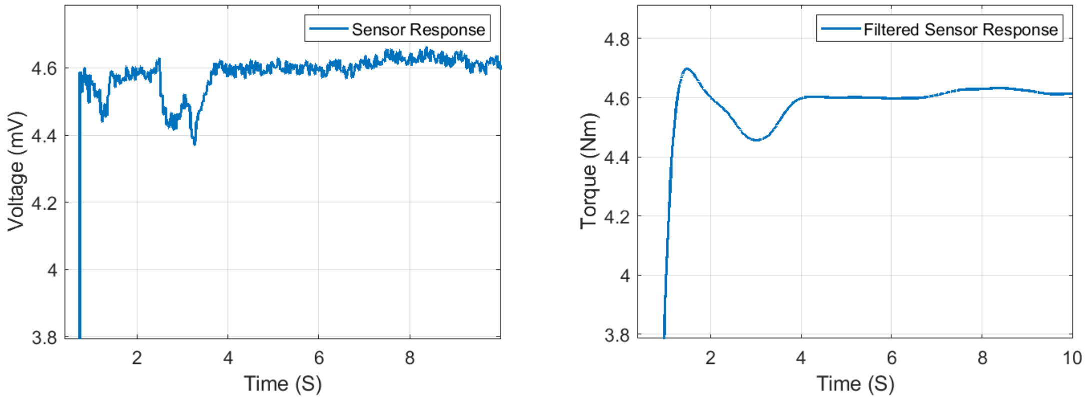
Figure 6.12: Sensor response for the third experiment and fourth experiment

shown in Figure 6.12b, the sensor readings show a drifting trend over time, making the system uncontrollable. Across these four experiments, it becomes apparent that ensuring a reliable response from the sensor is challenging. Without a reliable sensor readout, achieving effective control over the system proves to be challenging.

6.3.3 Static Interaction Control

Same as in the simulation, an external disturbance was introduced to the system during a steady-state condition to assess the impact of external forces and observe the functionality of the interaction control in static interaction scenarios. The introduction of the disturbance led to a deviation in the pendulum angle, maintaining a displacement from the desired angle until the external disturbance was removed. Subsequently, the pendulum smoothly returned to the desired angle without oscillations, demonstrating an anticipated and controlled response as shown in Simulation Figure 5.14.

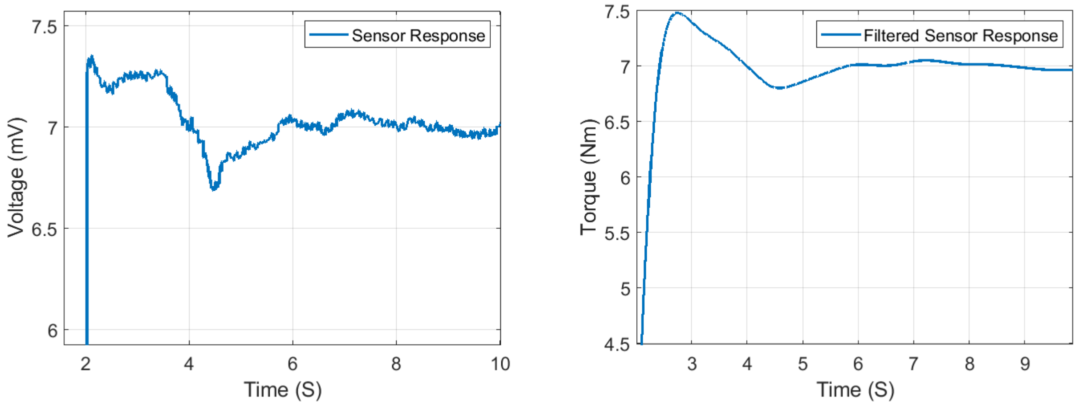
To investigate the impact of stiffness and damping coefficient values on interaction control, the same approach explained in the simulation was also used with the real system. The higher damping and stiffness coefficients led to a more rigid system and the lower values of lower damping and lower stiffness, led the system to a more flexible system that is more responsive to external force and displays faster reactions to disturbances. These results were visually seen with the experimental results.



(a) Unfiltered sensor response

(b) Filtered sensor response

Figure 6.13: Sensor response for static interaction scenario 1.



(a) Unfiltered sensor response

(b) Filtered sensor response

Figure 6.14: Sensor response for static interaction scenario 2.

6.3.4 Dynamic interaction control

To explore the dynamic interaction control, a sinusoidal input with a frequency of 1 Hz and an amplitude of $\frac{\pi}{4}$ was used as the desired trajectory. A disturbance or an interaction was introduced to the system a couple of seconds after the actuation.

The experimental outcomes deviated from the simulated expectations, revealing a distinct system behaviour. The observed behaviour of the system has shown that the system follows the desired trajectory for a couple of seconds and then starts to oscillate and this oscillation increases with time and quickly the system becomes inherently unstable. One possible reason is that the torque error of the torque control loop might become dominant which makes the system unstable and oscillate further away from the desired trajectory. This might be because of the sensor drift.

6.4 Conclusion

In this chapter, the control algorithm developed and simulated in Chapter 5 was implemented in a Mechaduino to investigate the practical applicability of a 3D printed torque sensor in real-world interaction control scenarios. The demonstration setup, comprising various components such as the sensor-integrated pendulum system, Mechaduino, and supporting electronics, was detailed. Experiments were conducted not only to verify the sensor's behavior observed during characterization but also to assess the 3D printed torque sensor's capability for interaction control in static and dynamic scenarios.

The initial experiments, conducted without the introduction of external interaction forces, delivered promising results that aligned well with the expected behaviour observed during characterization, the system outcomes demonstrated the successful performance of the implemented control algorithm in maintaining the desired trajectory.

The interaction control experiments included assessments of both static and dynamic interaction control scenarios. However, when external interactions were introduced, the system's response exhibited challenges, specifically during the dynamic interaction control. The assessments of static interaction control scenarios showcased the system's resilience to external disturbances, with the pendulum smoothly returning to the desired angle. Dynamic interaction control experiments revealed challenges related to sensor response noise and drift, impacting repeatability and complicating torque control. The dynamic interaction control experiments demonstrated that the system initially (for a couple of seconds) follows the desired trajectory but experiences oscillations and vibrations over time potentially might be due to the influence of sensor drift and hysteresis.

Initially, the system encountered challenges related to unstable contacts during its movement. However, the improvements in design successfully addressed these issues. As a result, the impact of unstable contacts on the system has been significantly minimized. It appears that the observed drift and hysteric behaviour in the system, particularly under dynamic conditions, can be attributed to factors other than the previously problematic unstable contacts. This emphasizes the need to address sensor imperfections like hysteresis and drift.

7 Conclusion and Recommendations

7.1 Conclusion

An investigation into a potential control approach for interaction control using a 3D printed sensor is detailed in this work. The primary research question was addressed through the exploration of the following sub-questions.

How can an interaction control strategy be developed for achieving stable, safe operation with a 3D printed torque sensor?

1. What are the existing interaction control strategies applicable to robotic systems in achieving stable and safe operation?

A comprehensive review of the literature reveals various interaction control strategies. Passive compliance control relies on mechanical devices such as springs and damping mechanisms to limit interaction forces, providing robust control but with limitations in responsiveness. Active compliance control integrates force feedback into the system, allowing the robot to actively respond to external forces and maintain precise position control effectively. Each strategy has its advantages and limitations, making them suitable for different applications and scenarios. Direct force control explicitly commands specific forces at the end-effector, valuable for tasks requiring precise force application, while indirect force control, represented by impedance control, establishes a dynamic relationship between end-effector motion and interaction forces, offering flexibility and adaptability. Impedance control is one of the most used interaction control strategies to achieve stable and safe operation within robotic systems.

2. Are there specific methods that have been successfully applied with 3D printed torque sensors?, What is the most suitable interaction control method to use with a 3D printed torque sensor?

The literature review indicates a gap in evidence regarding interaction control strategies specifically tailored for 3D printed torque sensors. Although the search scope does not reveal such strategies in existing literature, insights from related studies on force/torque-based impedance control and position-based impedance control provide a foundation for exploring the potential of 3D printed torque sensors.

The literature review suggests that the most suitable interaction control method for a system involving a 3D printed torque sensor could be torque-based impedance control. This is a cascade control system which has an inner torque loop with the use of a torque sensor and an outer impedance loop. This method is well-suited for tasks where the robot needs to interact with the environment or humans, emphasizing force sensing and compliance. Torque-based impedance control is adaptable to uncertain or dynamic environments, making it a good fit for applications involving torque sensors.

3. What performance metrics should be considered during simulations to evaluate stability and safety?

The impedance gains are important parameters in torque-based impedance control as they significantly influence how the system responds to external forces. These gains determine the stiffness and damping characteristics of the system, playing a role in shaping its behaviour. Higher stiffness provides more resistance to deformations induced by external forces, while damping affects how quickly the system dissipates energy. Tuning these gains is an important aspect of optimizing the control strategy. By systematically varying the impedance gains during simulations, observed the impact on the system's responsiveness, stability, and overall performance. Analyzing the system's reaction to different impedance gains provided valuable insights into finding an optimal balance that ensures both stable and compliant behaviour in response to external interactions.

4. How can the chosen interaction control method be practically implemented in a real setup using 3D printed torque sensor?

The practical implementation of the chosen interaction control method involves integrating the 3D printed torque sensor into the real setup. A 1 DOF single pendulum system with an integrated torque sensor was selected as the mechanical system. The control algorithm of the torque-based impedance control system was developed in Simulink, and translated into executable code and loaded onto a Mechaduino. The control architecture comprises a cascade system with an inner torque loop utilizing feedback from the 3D printed torque sensor and an outer impedance loop regulating system compliance. A Wheatstone bridge configuration was used to read the torque measurements from the sensor.

In conclusion, the torque-based impedance control algorithm, designed to explore the capabilities of a 3D-printed torque sensor for interaction control, demonstrated promising outcomes when the sensor was assumed to be ideal. However, the transition to a real setup revealed results that did not align with expectations. The presence of a non-ideal sensor introduced complexities that demand careful consideration for achieving stable and safe operation.

A notable difference was seen when comparing results from characterizations, where the sensor exhibited a closely linear response, to those from the demonstration setup, where nonlinearity became more apparent. Consequently, it is apparent that the sensor representation linear fit derived from characterizations cannot always ensure stable operations in the demonstration setup.Furthermore, during the experiments, efforts were made to improve the sensor response by applying filters to remove high-frequency noise. However, it became evident that this measure alone was insufficient to achieve stable operation.

7.2 Discussion and Recommendation

This study aimed to investigate the applicability of 3D printed torque sensors in control-focused applications through an exploration of interaction control techniques. The thesis discussed various interaction control strategies and simulated the most fitting control approach for interaction control using 3D printed sensors. Finally, this strategy was implemented in a tangible demonstration setup using the developed torque sensor.

In the initial phase, modifications were made to the initial sensor design to address stability concerns in the electrical connection between the sensor and wires. The characterization process revealed key sensor characteristics like non-linearity, hysteresis, drift and repeatability. A linear fit was derived from the sensor response to represent sensor behaviour. A 1-DOF pendulum system with integrated sensors was used for practical implementation, exploring both dynamic and static interaction controls to validate the proposed strategy's effectiveness.

As mentioned before, the simulation of the torque-based impedance controller has demonstrated promising results but under the assumption of an ideal torque source. However, the real-world torque sensor introduces non-linearities, primarily due to drift, creep, and hysteresis. Addressing and compensating for these non-linear behaviours in the sensor response would significantly improve the usability of the torque sensor in the control system. Therefore, future work should focus on developing robust strategies to effectively compensate for nonlinearities, ensuring the reliable integration of the torque sensor into the control system. Possible avenues for future work could also include implementing advanced signal processing techniques to filter out sensor noise, exploring calibration methods to reduce drift, and investigating hysteresis compensation algorithms. By focusing on these aspects, the goal is to improve the accuracy and reliability of the control system, making it more robust in practical applications.

Moreover, it is evident that the sensor characterization yielded better results compared to those obtained in the experimental control setup. This discrepancy can be attributed to several factors. During the characterization phase, data acquisition was executed using a high-performance Data Acquisition System at a notably higher frequency. In contrast, the experimental setup employed a manually designed half-bridge configuration but at a lower frequency.

Additionally, the conditions under which the sensor was characterized differed from those in the experimental setup. In the characterization phase, the sensor underwent controlled compression and elongation within a stable environment. The sensor was securely fixed horizontally to a plate, limiting its movement to one degree of freedom (1DOF) with stable connections. Conversely, in the experimental setup, the sensor was positioned vertically, and the connections were subject to movement along with the pendulum's oscillations.

These variations in the experimental conditions undoubtedly contribute to significant differences in the measurements obtained. Therefore one suggestion, it is advisable to characterize the sensor's behaviour while explicitly considering real-world setups, given their potentially greater impact compared to controlled characterization environments.

Moreover, the simulation relied on an idealized sensor model, overlooking the complexity present in the actual sensor's dynamic behaviour. Future research could explore the development of a mathematical model that accurately represents the dynamics of the torque sensor. This would bring the simulation closer to reality, enabling a more reliable representation of the sensor's characteristics and behaviour.

In Mechaduino, where computational power is limited, the utilisation of digital filters may present challenges due to their computational demands. Implementing analog filters, such as low-pass filters, directly in the bridge configuration can be an efficient alternative.

A Appendix 1

A.1 RP Manipulator

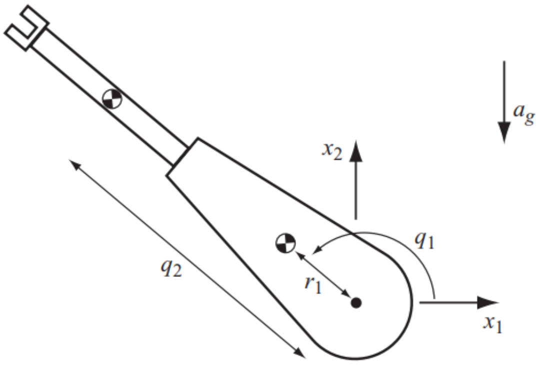


Figure A.1: A RP manipulator [40, pp. 351]

Let's examine an example, figure A.1 depicts a robotic arm composed of an RP (revolute-prismatic) configuration, featuring one revolute joint (q_1) and one prismatic joint (q_2) representing the angle and distance respectively. In this setup, q_1 signifies the angle of the first joint concerning the world frame's x_1 axis, while $q2 > 0$, represents the distance from the first joint to the centre of mass of the second link. The first link has a centre of mass at a distance r_1 from the first joint, with mass m_1 and inertia I_1. Similarly, the second link, has a mass of m_2 and an inertia of I_2. The gravitational force, denoted by a_g acts in the $-x_2$ direction of the world frame [40]. Considering the links individually, the kinetic energy of the first link can be represented as follows:

$$T_1(q,\dot{q}) = \frac{1}{2}m_1 {v_1}^2 + \frac{1}{2}I_1 {\omega_1}^2 \tag{A.1}$$

Here, v_1 denotes the linear velocity of the centre of mass of the link, and ω_1 represents the angular velocity of the link:

$$v_1 = r_1 \dot{q}_1 \tag{A.2}$$

$$\omega_1 = \dot{q}_1 \tag{A.3}$$

Substituting A.2 and A.3 in A.1 brings:

$$T_1(q,\dot{q}) = \frac{1}{2}m_1 {r_1}^2 \dot{q}_1^2 + \frac{1}{2}I_1 \dot{q}_1^2 \tag{A.4}$$

The potential energy of the first link is given by:

$$V_1(q) = m_1 a_g r_1 sin(q_1) \tag{A.5}$$

The kinetic energy of the second link can be described by expression:

$$T_2(q,\dot{q}) = \frac{1}{2}m_2{v_2}^2 + \frac{1}{2}I_2{\omega_2}^2 \tag{A.6}$$

With v_2 denoting the linear velocity of the centre of mass of the link, and ω_2 representing the angular velocity of the link.

$$v_1 = \sqrt{{\dot{q}_2}^2 + (q_2\dot{q}_1)^2} \tag{A.7}$$

$$\omega_2 = \dot{q}_1 \tag{A.8}$$

Substituting A.7 and A.8 in A.6 results in:

$$T_2(q,\dot{q}) = \frac{1}{2}m_2\left({\dot{q}_2}^2 + (q_2\dot{q}_1)^2\right) + \frac{1}{2}I_2\dot{q}_1^2 \tag{A.9}$$

The potential energy of the second link is given by:

$$V_2(q) = m_2 a_g q_2 sin(q_1) \tag{A.10}$$

The Lagrangian equation for the whole system is given by:

$$L = T_1 - V_1 + T_2 - V_2 \tag{A.11}$$

$$L = \frac{1}{2}\left(\left(I_1 + I_2 + m_1{r_1}^2 + m_2{q_2}^2\right){\dot{q}_1}^2 + m_2{\dot{q}_2}^2\right) - a_g sin(q_1)(m_1r_1 + m_2q_2) \tag{A.12}$$

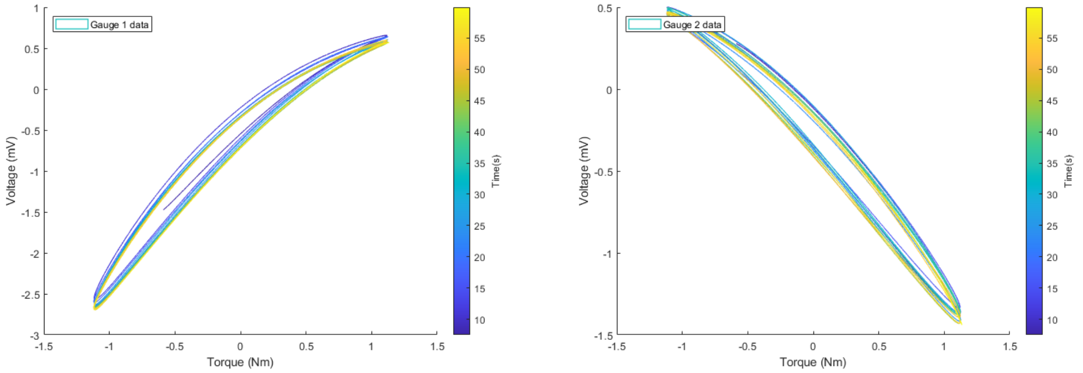
Using the Euler-Lagrange equations:

$$u_1 = \left(I_1 + I_2 + m_1{r_1}^2 + m_2{q_2}^2\right)\ddot{q}_1 + 2m_2q_2\dot{q}_1\dot{q}_2 + a_g cos(q_1)(m_1r_1 + m_2q_2) \tag{A.13}$$

$$u_2 = m_2\ddot{q}_2 - m_2q_2{\dot{q}_1}^2 + a_g cos(q_1)m_2 \tag{A.14}$$

B Appendix 2

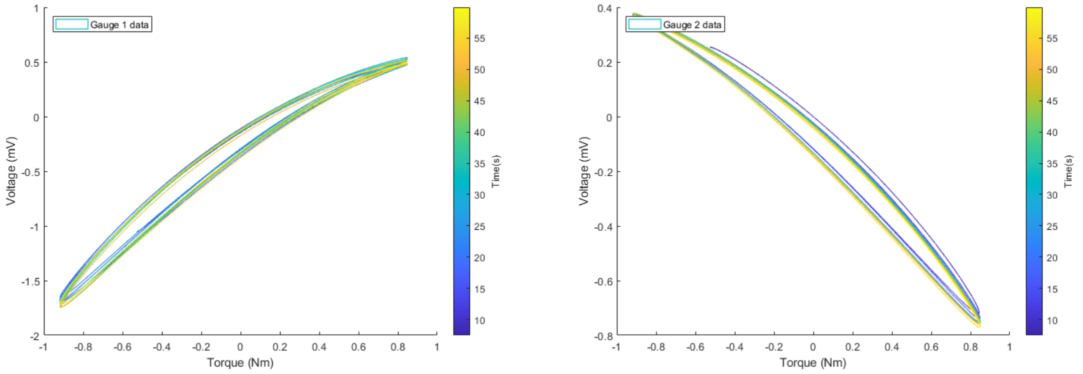
B.1 Sensor Characterisation



(a) Response of gauge 1.

(b) Response of gauge 2.

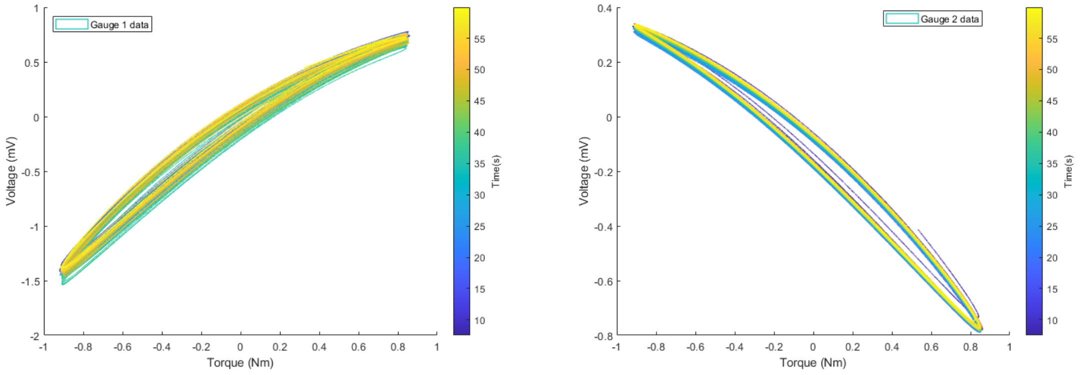
Figure B.1: Individual gauge response for 10 N force at 0.2 Hz sinusoidal excitation.



(a) Response of gauge 1.

(b) Response of gauge 2.

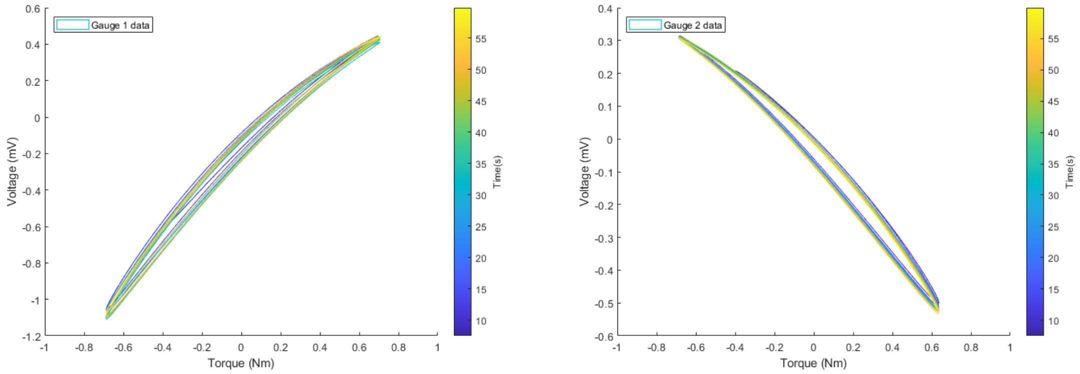
Figure B.2: Individual gauge response for 8 N force at 0.2 Hz sinusoidal excitation.



(a) Response of gauge 1.

(b) Response of gauge 2.

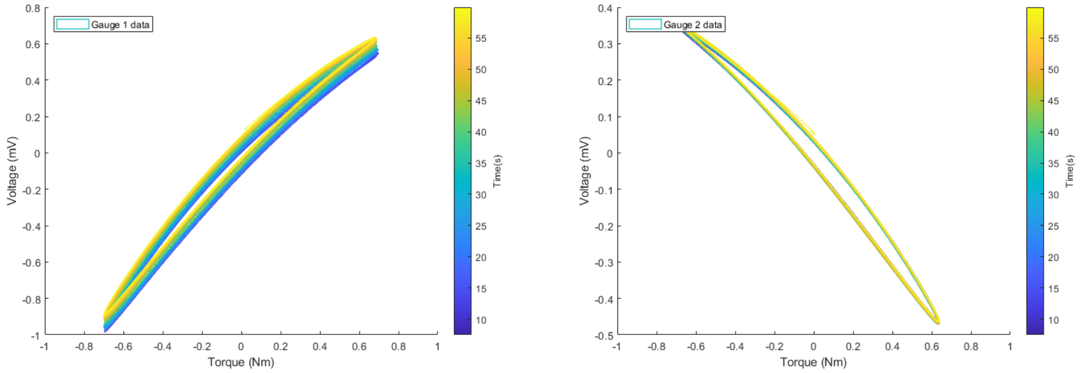
Figure B.3: Individual gauge response for 8 N force at 1 Hz sinusoidal excitation.



(**a**) Response of gauge 1.

(**b**) Response of gauge 2.

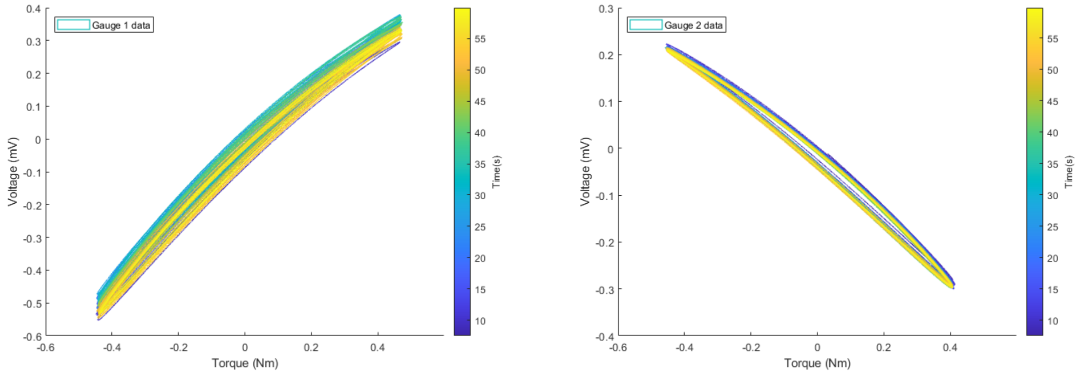
Figure B.4: Individual gauge response for 6 N force at 0.2 Hz sinusoidal excitation.



(**a**) Response of gauge 1.

(**b**) Response of gauge 2.

Figure B.5: Individual gauge response for 6 N force at 1 Hz sinusoidal excitation.

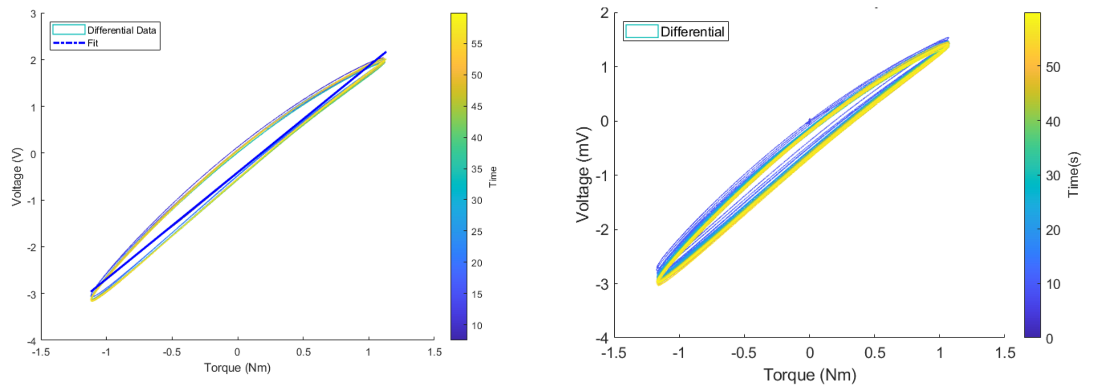


(**a**) Response of gauge 1.

(**b**) Response of gauge 2.

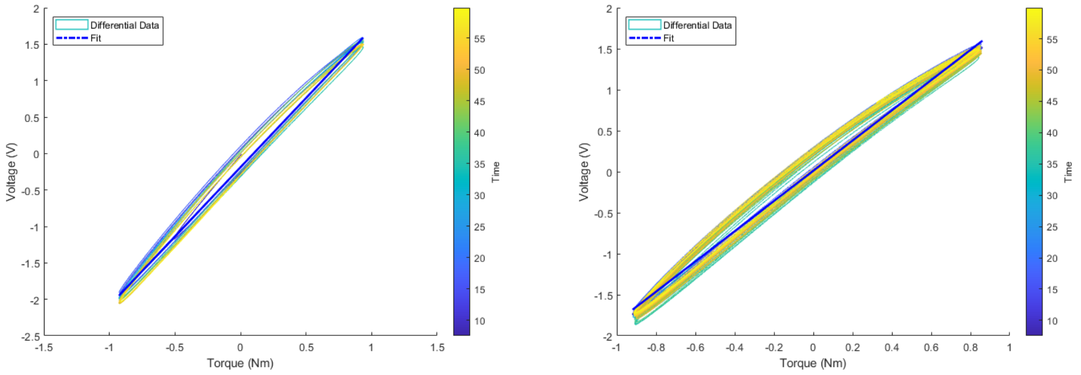
Figure B.6: Individual gauge response for 4 N force at 1 Hz sinusoidal excitation.

B.1.1 Differential response



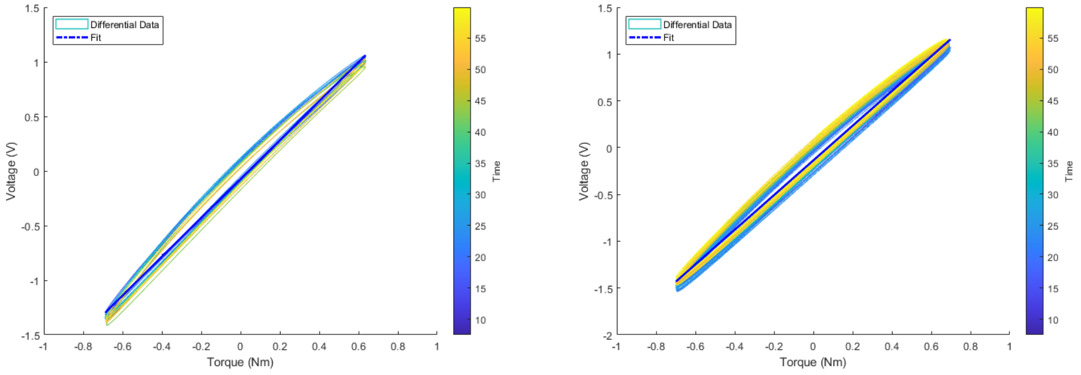
(a) Differential response at 0.2 Hz sinusoidal excitation. **(b)** Differential response at 1 Hz sinusoidal excitation.

Figure B.7: Differential response for 10 N force with linear fits.



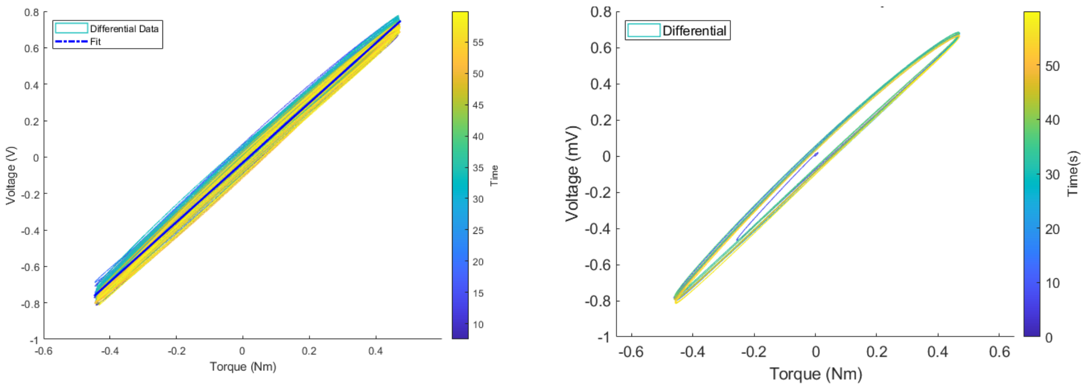
(a) Differential response at 0.2 Hz sinusoidal excitation. **(b)** Differential response at 1 Hz sinusoidal excitation.

Figure B.8: Differential response for 8 N force with linear fits.



(a) Differential response at 0.2 Hz sinusoidal excitation. **(b)** Differential response at 1 Hz sinusoidal excitation.

Figure B.9: Differential response for 6 N force with linear fits.



(a) Differential response at 0.2 Hz sinusoidal excitation. **(b)** Differential response at 1 Hz sinusoidal excitation.

Figure B.10: Differential response for 4 N force with linear fits.

B.1.2 Linear fits comparison

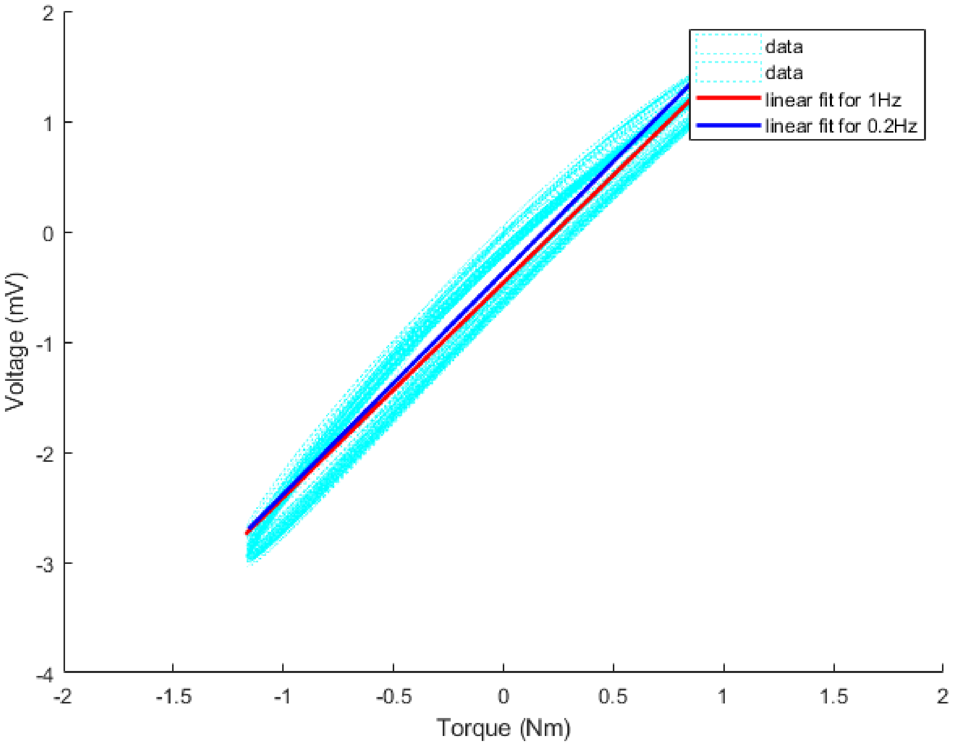


Figure B.11: Linear fit for 10N input force at 1Hz and 0.2Hz frequencies.

The fit for data set with 10 N force input at 1 Hz is,

$$y = 1.9470x - 0.461 \tag{B.1}$$

The fit for data set with 10 N force input at 0.2 Hz is,

$$y = 2.010x - 0.3692 \tag{B.2}$$

The fit for data set with 8 N force input at 1 Hz is,

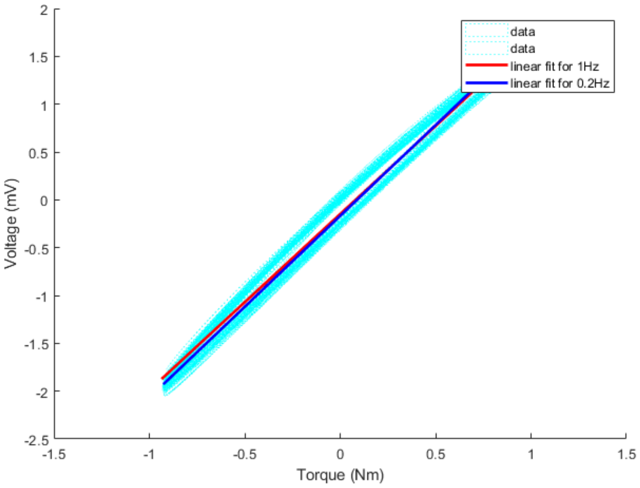


Figure B.12: Linear fit for 8N input force at 1Hz and 0.2Hz frequencies.

$$y = 1.8422x - 0.1488 \tag{B.3}$$

The fit for data set with 8 N force input at 0.2 Hz is,

$$y = 1.8998 - 0.1679 \tag{B.4}$$

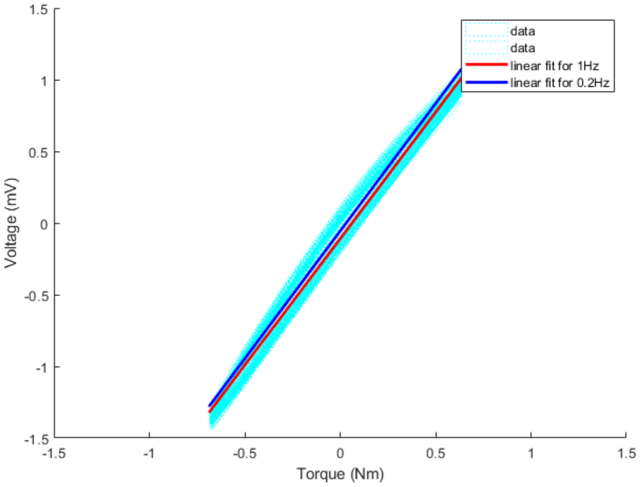


Figure B.13: Linear fit for 6N input force at 1Hz and 0.2Hz frequencies.

The fit for data set with 6 N force input at 1 Hz is,

$$y = 1.7645 - 0.1095 \tag{B.5}$$

The fit for data set with 6 N force input at 0.2 Hz is,

$$y = 1.78.. - 0.0564 \tag{B.6}$$

The fit for data set with 4 N force input at 1 Hz is,

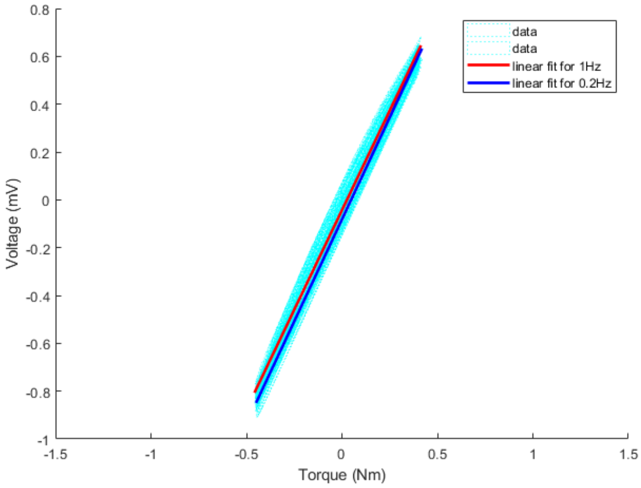


Figure B.14: Linear fit for 4N input force at 1Hz and 0.2Hz frequencies.

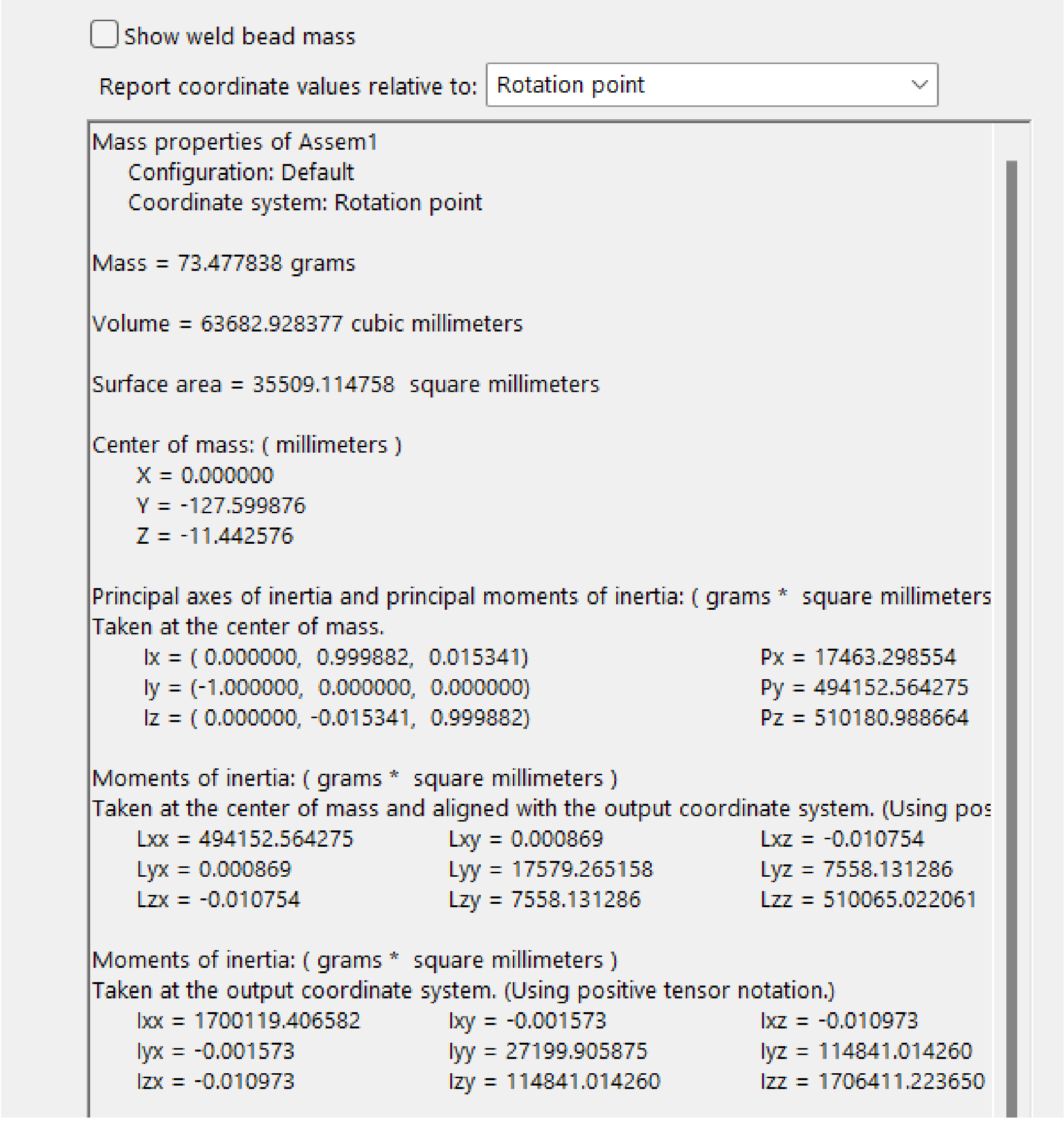
$$y = 1.6756 - 0.045 \tag{B.7}$$

The fit for data set with 4 N force input at 0.2 Hz is,

$$y = 1.6950 - 0.0822 \tag{B.8}$$

C Appendix 3

C.1 Mass properties of the CAD model



☐ Show weld bead mass

Report coordinate values relative to: Rotation point

```
Mass properties of Assem1
     Configuration: Default
     Coordinate system: Rotation point

Mass = 73.477838 grams

Volume = 63682.928377 cubic millimeters

Surface area = 35509.114758  square millimeters

Center of mass: ( millimeters )
      X = 0.000000
      Y = -127.599876
      Z = -11.442576

Principal axes of inertia and principal moments of inertia: ( grams *  square millimeters
Taken at the center of mass.
       Ix = ( 0.000000,  0.999882,  0.015341)                  Px = 17463.298554
       Iy = (-1.000000,  0.000000,  0.000000)                  Py = 494152.564275
       Iz = ( 0.000000, -0.015341,  0.999882)                  Pz = 510180.988664

Moments of inertia: ( grams *  square millimeters )
Taken at the center of mass and aligned with the output coordinate system. (Using pos
      Lxx = 494152.564275        Lxy = 0.000869               Lxz = -0.010754
      Lyx = 0.000869             Lyy = 17579.265158           Lyz = 7558.131286
      Lzx = -0.010754            Lzy = 7558.131286            Lzz = 510065.022061

Moments of inertia: ( grams *  square millimeters )
Taken at the output coordinate system. (Using positive tensor notation.)
      Ixx = 1700119.406582       Ixy = -0.001573              Ixz = -0.010973
      Iyx = -0.001573            Iyy = 27199.905875           Iyz = 114841.014260
      Izx = -0.010973            Izy = 114841.014260          Izz = 1706411.223650
```

Figure C.1: Mass properties of the CAD model of the pendulum system with bolted weights [74].

C.2 Plant simulation - MATLAB

```
% parameters
I = 0.0015;
m = 0.0730 ;
t_ext = 0 ;
B = 0.01;
g = 9.81;
l = 0.128;
T=0.0;

syms a theta(t)
```

```
eqn = T == ((m*(l^2)+I)*diff(theta,2)) +m*g*l*sin(theta)+
   t_ext;


eqn = isolate(eqn,diff(theta,2));

syms x
approx = taylor(sin(x),x,'Order',2);
approx = subs(approx,x,theta(t));

eqnLinear = subs(eqn,sin(theta(t)),approx);

syms theta_0 theta_t0 omega_0
theta_t = diff(theta);
cond = [theta(0) == theta_0, theta_t(0) == theta_t0];
thetaSol(t) = dsolve(eqnLinear,cond);

omega_0Value = sqrt(g/l);
TT = 2*pi/omega_0Value;

theta_0Value  = pi/2; % Solution only valid for small
   angles.
theta_t0Value = 0;      % Initially at rest.

vars   = [omega_0      theta_0      theta_t0];
values = [omega_0Value theta_0Value theta_t0Value];
thetaSolPlot = subs(thetaSol,vars,values);

figure
fplot(thetaSolPlot(t*TT), [0 8]);
grid on;
title('Response of the plant to zero input torque');
xlabel('time(seconds');
ylabel('angle(rad)');

figure
grid on
x_pos = sin(thetaSolPlot);
y_pos = -cos(thetaSolPlot);
fanimator(@fplot,x_pos,y_pos,'ko','MarkerFaceColor','k','
   AnimationRange',[0 10*TT]);
hold on;
fanimator(@(t) plot([0 x_pos(t)],[0 y_pos(t)],'k-','
   LineWidth',4),'AnimationRange',[0 10*TT]);
fanimator(@(t) text(-0.3,0.3,"Timer: "+num2str(t,2)+" s"),
   'AnimationRange',[0 10*TT]);
playAnimation;
```

D Appendix 4

D.1 Controller.cpp

```
//Contains TC5 Controller definition
//The main control loop is executed by the TC5 timer interrupt:

#include <SPI.h>

#include "State.h"
#include "Utils.h"
#include "Parameters.h"

extern double V;  //voltage variable
extern double xyz; //placeholder variable

// Function to calculate torque using inverse dynamics model
double inverseDynamics(double q, double q_ddot) {
  // Constants and parameters
  double I = 0.0015;
  double m = 0.0730;
  double g = 9.81;
  double l = 0.128;
  double torque = (m * (l * l) + I) * q_ddot + m * g * l * sin(q);
  return torque;
}

// Function to convert sensor voltage to torque
double voltagetototorque(double v_read) {
  double cal_torque = (v_read-0.60842)/0.3796; // Calibration formula
  for torque calculation
  //double cal_torque = (v_read);
  return cal_torque;
}

void TC5_Handler() {  // gets called with FPID frequency

  static int print_counter = 0;  //this is used by step response

  if (TC5->COUNT16.INTFLAG.bit.OVF == 1) {  // A counter overflow
  caused the interrupt

    TEST1_HIGH();  //digitalWrite(3, HIGH);  //Fast Write to
    Digital 3 for debugging

    y = lookup[readEncoder()];  //read encoder and lookup corrected
    angle in calibration lookup table

    if ((y - y_1) < -180.0) wrap_count += 1;  //Check if we've rotated
    more than a full revolution (have we "wrapped" around from 359
    degrees to 0 or ffrom 0 to 359?)
    else if ((y - y_1) > 180.0) wrap_count -= 1;

    yw = (y + (360.0 * wrap_count));  //yw is the wrapped angle
    (can exceed one revolution)

    e = (r - yw);
```

```
ITerm += (pKi * e);  //Integral wind up limit
if (ITerm > 150.0) ITerm = 150.0;
else if (ITerm < -150.0) ITerm = -150.0;

DTerm = pLPFa * DTerm - pLPFb * pKd * (yw - yw_1);

//u = (pKp * e) + ITerm + DTerm;
u = (pKp * e) + DTerm; // Calculate control effort (PD impedance
controller)

y_1 = y;  //copy current value of y to previous value (y_1) for
next control cycle before PA angle added


if (u > 0)         //Depending on direction we want to apply torque,
add or subtract a phase angle of PA for max effective torque.
PA should be equal to one full step angle: if the excitation
angle is the same as the current position, we would not move!
{                  //You can experiment with "Phase Advance" by increasing PA when operating a
  y += PA;         //update phase excitation angle
  if (u > uMAX)    // limit control effort
    u = uMAX;      //saturation limits max current command
} else {
  y -= PA;          //update phase excitation angle
  if (u < -uMAX)    // limit control effort
    u = -uMAX;      //saturation limits max current command
}

U = abs(u);  //

if (abs(e) < 0.1) ledPin_HIGH();  // turn on LED if error is
less than 0.1
else ledPin_LOW();                //digitalWrite(ledPin, LOW);


//output(-y, round(U));    // update phase currents
int mappedValue = map(round(u), inputMin, inputMax, outputMin, outputMax);
//SerialUSB.println(mappedValue);

double impedanceTorque = mappedValue * 0.0023888888; //the controller output u is a vlaue be
 //this converted to torque using motor constant
double r_Rad = r * (PI / 180);
double feedforwrdTorque = inverseDynamics(r_Rad, 0, 0);
//SerialUSB.println(feedforwrdTorque);
//SerialUSB.print("  |  ");
//SerialUSB.print(impedanceTorque);
//SerialUSB.print("  |  ");

double referenceTorque = impedanceTorque + feedforwrdTorque;


double feedbackTorque = voltagetotorque(V); //real sensor
reading mapping to torque

 //SerialUSB.print("feedbackTorqueSensor");
 //SerialUSB.print(feedbackTorqueSensor,10);   //this seciton for validating the sensor respo
// SerialUSB.print("  |  ");
```

```
    //SerialUSB.print(impedanceTorque,10);
    //SerialUSB.print("  |  ");
    SerialUSB.print(feedforwrdTorque,10);
    SerialUSB.print("␣␣|␣");
    SerialUSB.println(V,10);

    e2 = (referenceTorque - feedbackTorque);
    // SerialUSB.print("  |  ");
    // SerialUSB.println(e2);
    //SerialUSB.println(torqueError);


    //PI torque controller

    ITerm2 += (tKi * e2);  //Integral wind up limit
    if (ITerm2 > 150.0) ITerm2 = 150.0;
    else if (ITerm2 < -150.0) ITerm2 = -150.0;

    DTerm2 = DTerm2 - pKd * (feedbackTorque - feedbackTorque_1);

    u2 = (tKp * e2) + ITerm2;

    feedbackTorque_1 = feedbackTorque;  //copy current value of y to
    previous value (y_1) for next control cycle before PA angle added

    U2 = abs(u2);               //
    output(-y, round(U2));      // update phase currents
                                // e_3 = e_2;     //copy current values to
                                previous values for next control cycle
    //e_2 = e_1;    //these past values can be useful for more complex controllers/filters.
  Uncomment as necessary
    //e_1 = e;
    // u_3 = u_2;
    //u_2 = u_1;
    //u_1 = u;
    yw_1 = yw;
    //y_1 = y;

    if (print_yw == true) {  //for step resonse... still under development
      print_counter += 1;
      if (print_counter >= 5) {             // print position every 5th loop (every time is t
        SerialUSB.println(int(yw * 1024));  //*1024 allows us to print ints instead of floats
        print_counter = 0;
      }
    }
    TC5->COUNT16.INTFLAG.bit.OVF = 1;  // writing a one clears the flag ovf flag
    TEST1_LOW();                       //for testing the control loop timing
  }

  //if (e2 > 0.28){
  // U2 = 0;
  //U2 = m * g * l * sin(y*(PI/180));
  //output(-y, round(U2));
  //}
}
```

E Appendix 5

E.1 Use of AI in Education

During the preparation of this work the author(s) used ChatGPT 3.5 and Grammarly in order to improve my academic writing. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the work."

List of tools

1. ChatGPT 3.5
2. Grammarly

Bibliography

[1] R. Heeg, “Control of a system based on a 3d printed torque sensor,” Master’s thesis, University of Twente, 2023.

[2] M. R. Khosravani and T. Reinicke, “3d-printed sensors: Current progress and future challenges,” *Sensors and Actuators A: physical*, vol. 305, p. 111916, 2020. DOI: 10.1016/j.sna.2020.111916.

[3] M. Liu, Q. Zhang, Y. Shao, C. Liu, and Y. Zhao, “Research of a novel 3d printed strain gauge type force sensor,” *Micromachines*, vol. 10, no. 1, p. 20, 2019. DOI: 10.3390/mi10010020.

[4] A. Dijkshoorn, P. Werkman, M. Welleweerd, G. Wolterink, B. Eijking, J. Delamare, R. Sanders, and G. J. Krijnen, “Embedded sensing: Integrating sensors in 3-d printed structures,” *Journal of Sensors and Sensor Systems*, vol. 7, no. 1, pp. 169–181, 2018. DOI: 10.5194/jsss-7-169-2018.

[5] M. Vaezi, S. Chianrabutra, B. Mellor, and S. Yang, “Multiple material additive manufacturing–part 1: a review: this review paper covers a decade of research on multiple material additive manufacturing technologies which can produce complex geometry parts with different materials,” *Virtual and Physical Prototyping*, vol. 8, no. 1, pp. 19–50, 2013. DOI: 10.1080/17452759.2013.778175.

[6] Y. Jiang, M. N. Islam, R. He, X. Huang, P.-F. Cao, R. C. Advincula, N. Dahotre, P. Dong, H. F. Wu, and W. Choi, “Recent advances in 3d printed sensors: materials, design, and manufacturing,” *Advanced Materials Technologies*, vol. 8, no. 2, p. 2200492, 2023. DOI: 10.1002/admt.202200492.

[7] M. Schouten, G. Wolterink, A. Dijkshoorn, D. Kosmas, S. Stramigioli, and G. Krijnen, “A review of extrusion-based 3d printing for the fabrication of electro-and biomechanical sensors,” *IEEE sensors journal*, vol. 21, no. 11, pp. 12900–12912, 2020. DOI: 10.1109/jsen.2020.3042436.

[8] A. Fayazi, N. Pariz, A. Karimpour, and S. H. Hosseinnia, “Robust position-based impedance control of lightweight single-link flexible robots interacting with the unknown environment via a fractional-order sliding mode controller,” *Robotica*, vol. 36, no. 12, pp. 1920–1942, 2018. DOI: 10.1017/s0263574718000802.

[9] R. Kelly, V. Santibánez, and A. Lorıa, *Control of robot manipulators in joint space*. Springer Science & Business Media, 2005. DOI: 10.1007/b135572.

[10] N. Hogan, “Impedance control - An approach to manipulation. I - Theory. II - Implementation. III - Applications,” *ASME Journal of Dynamic Systems and Measurement Control B*, vol. 107, pp. 1–24, mar 1985.

[11] S. Jung, T. C. Hsia, and R. G. Bonitz, “Force tracking impedance control of robot manipulators under unknown environment,” *IEEE Transactions on Control Systems Technology*, vol. 12, no. 3, pp. 474–483, 2004. DOI: 10.1109/tcst.2004.824320.

[12] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, “An atlas of physical human–robot interaction,” *Mechanism and Machine Theory*, vol. 43, no. 3, pp. 253–270, 2008. DOI: 10.1016/j.mechmachtheory.2007.03.003.

[13] J. De Schutter, H. Bruyninckx, W.-H. Zhu, and M. W. Spong, "Force control: A bird's eye view," in *Control problems in robotics and automation*, pp. 1–17, Springer, 1998. DOI: 10.1007/BFb0015073.

[14] D. E. Whitney, "Historical perspective and state of the art in robot force control," *The International Journal of Robotics Research*, vol. 6, no. 1, pp. 3–14, 1987.

[15] C.-C. Cheah and D. Wang, "Learning impedance control for robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 3, pp. 452–465, 1998.

[16] A. Lay-Ekuakille and S. C. Mukhopadhyay, *Wearable and autonomous biomedical devices and systems for smart environment*. Springer, 2010.

[17] J. de Gea Fernández, *Predictive Context-based Adaptive Compliance for Interaction Control of Robot Manipulators*. PhD thesis, Universität Bremen, 2011.

[18] L. Villani and J. De Schutter, "Force control," *Springer handbook of robotics*, pp. 195–220, 2016.

[19] M. Vukobratovic, *Dynamics and robust control of robot-environment interaction*, vol. 2. World Scientific, 2009.

[20] J. Pratt, B. Krupp, and C. Morse, "Series elastic actuators for high fidelity force control," *Industrial Robot: An International Journal*, vol. 29, no. 3, pp. 234–241, 2002.

[21] D. Whitney, "What is the remote centre compliance (rcc) and what can it do?," in *Proceedings of the 9th International Symposium on Industrial Robots, 1979*, 1979.

[22] Wikipedia contributors, "Remote center compliance." `https://en.wikipedia.org/wiki/Remote_center_compliance`.

[23] T. Lefebvre, J. Xiao, H. Bruyninckx, and G. De Gersem, "Active compliant motion: a survey," *Advanced Robotics*, vol. 19, no. 5, pp. 479–499, 2005.

[24] S. Chiaverini, B. Siciliano, and L. Villani, "A survey of robot interaction control schemes with experimental comparison," *IEEE/ASME Transactions on mechatronics*, vol. 4, no. 3, pp. 273–285, 1999.

[25] H. F. Al-Shuka, S. Leonhardt, W.-H. Zhu, R. Song, C. Ding, Y. Li, *et al.*, "Active impedance control of bioinspired motion robotic manipulators: An overview," *Applied bionics and biomechanics*, vol. 2018, 2018.

[26] D. A. Lawrence, "Impedance control stability properties in common implementations," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pp. 1185–1190, IEEE, 1988.

[27] D. E. Whitney, "Force feedback control of manipulator fine motions," 1977.

[28] M. Ali, *Impedance Control of Redundant Manipulators*. Doctor of science in technology, Tampere University of Technology, November 2011.

[29] F. Bruni, F. Caccavale, C. Natale, and L. Villani, "Experiments of impedance control on an industrial robot manipulator with joint friction," in *Proceeding of the 1996 IEEE International Conference on Control Applications IEEE International Conference on Control Applications held together with IEEE International Symposium on Intelligent Contro*, pp. 205–210, IEEE, 1996.

[30] C. R. Carignan and D. L. Akin, "Achieving impedance objectives in robot teleoperation," in *Proceedings of International Conference on Robotics and Automation*, vol. 4, pp. 3487–3492, IEEE, 1997.

[31] D. Matko, R. Kamnik, and T. Bajd, "Adaptive impedance force control of an industrial manipulator," in *ISIE'99. Proceedings of the IEEE International Symposium on Industrial Electronics (Cat. No. 99TH8465)*, vol. 1, pp. 129–133, IEEE, 1999.

[32] A. A. Goldenberg, "Implementation of force and impedance control in robot manipulators," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pp. 1626–1632, IEEE, 1988.

[33] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *1980 19th IEEE conference on decision and control including the symposium on adaptive processes*, pp. 95–100, IEEE, 1980.

[34] P. K. Khosla and T. Kanade, "Experimental evaluation of nonlinear feedback and feedforward control schemes for manipulators," *The International Journal of Robotics Research*, vol. 7, no. 1, pp. 18–28, 1988.

[35] A. Renfrew, "Introduction to robotics: Mechanics and control," *International Journal of Electrical Engineering & Education*, vol. 41, no. 4, p. 388, 2004.

[36] A. Dietrich, *Whole-body impedance control of wheeled humanoid robots*, vol. 116. Springer, 2016.

[37] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4, pp. 1047–1054, IEEE, 1987.

[38] N. Sprague, "Coordinate frames." https://www.academia.edu/38172552/Frames, 2016. Visited on 2023-08-25.

[39] V. A. Deshpande and A. B. Verma, "Dynamics of robot manipulators: a review," *International Journal of Engineering Research and Technology*, vol. 3, pp. 603–606, 2010.

[40] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of robot motion: theory, algorithms, and implementations*. MIT press, 2005.

[41] Tropical Labs, "Mechaduino manual." https://tropical-labs.com/wp-content/uploads/2018/11/Mechaduino-Manual-0.1.3.pdf, 2018.

[42] Y. Lou, J. Wei, and S. Song, "Design and optimization of a joint torque sensor for robot collision detection," *IEEE Sensors Journal*, vol. 19, no. 16, pp. 6618–6627, 2019.

[43] D. Kosmas, "Model-based hysteresis compensation and control with 3d printed lousy sensors," Master's thesis, University of Twente, 2020.

[44] M. Schouten, B. Prakken, R. Sanders, and G. Krijnen, "Linearisation of a 3d printed flexible tactile sensor based on piezoresistive sensing," in *2019 IEEE SENSORS*, pp. 1–4, IEEE, 2019.

[45] A. Asia, "Is 3d printing environmentally friendly?." https://additive3dasia.com/news/is-3d-printing-environmentally-friendly/.

[46] Replique, "Guide to sustainable 3d printing materials." https://replique.io/2022/04/26/guide-to-sustainable-3d-printing-materials/, April 26 2022.

[47] B. O'Neill, "Optimal pla bed temperature and print temperature settings." https://www.wevolver.com/article/pla-bed-temperature-print-temperature-settings, Novemebr 23, 2023.

[48] Filament2Print, "Tds conductive pla." https://filament2print.com/gb/index.php?controller=attachment&id_attachment=94.

[49] 3DPrint, "Diabase engineering unveils the h-series 3d printer." https://3dprint.com/220069/diabase-engineering-h-series/, 2018.

[50] P. Research, "Prusaslicer documentation." https://github.com/prusa3d/PrusaSlicer. PrusaSlicer Documentation.

[51] C. Abeykoon, P. Sri-Amphorn, and A. Fernando, "Optimization of fused deposition modeling parameters for improved pla and abs 3d printed structures," *International Journal of Lightweight Materials and Manufacture*, vol. 3, no. 3, pp. 284–297, 2020.

[52] R. Narang and D. Chhabra, "Analysis of process parameters of fused deposition modeling (fdm) technique," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 3, no. 10, pp. 41–48, 2017.

[53] DEWESoft, "Dewe-43a technical reference manual." https://d36j349d8rqm96.cloudfront.net/3/6/Dewesoft-DEWE-43A-Manual-EN.pdf, 2023.

[54] S. M. C. Actuators, "Lca series." https://www.smac-mca.com/products/linear-actuators/lca-series.

[55] Technovolt, "Technovolt acquisition system." https://www.technovolt.ro/en/aquisition-system. DEWE 43A.

[56] SMAC Corporation, "Smac lcc-10 product manual." http://pgqfm-mgqub.servertrust.com/v/vspfiles/assets/images/LCC-10%20SMAC%20Product%20Manual%20Rev1%200.pdf, 2011.

[57] SMAC Moving Coil Actuators, *Embedded motion control library*. INGENIA-CAT, S.L., 2013. Revision 1.3 (Firmware version 2.0).

[58] M. Schouten, "Smac actuator matlab drivers." https://github.com/martijnschouten/lab-equipment-matlab-drivers/tree/master/SMAC%20actuator, 2019.

[59] DEWESoft, "Strain measurement using strain gage sensors." https://training.dewesoft.com/online/course/strain-measurement-using-strain-gage-sensors.

[60] "Matlab documentation for filtfilt." https://nl.mathworks.com/help/signal/ref/filtfilt.html. Zero-phase digital filtering.

[61] T. B. Cunha, *Hydraulic Compliance Control of the Quadruped Robot HyQ*. Doctor of philosophy (ph.d.), University of Genoa, Italy and Istituto Italiano di Tecnologia (IIT), March 2013.

[62] A. Heins, "Lagrangian mechanics in three ways." https://adamheins.com/blog/lagrangian-mechanics-three-ways, 2021.

[63] S. Echeandia and P. M. Wensing, "Numerical methods to compute the coriolis matrix and christoffel symbols for rigid-body systems," *Journal of Computational and Nonlinear Dynamics*, vol. 16, no. 9, p. 091004, 2021.

[64] G. Meinsma, “Nine lessons on control systems,” 2019. Notes for UTwente course "Robust Control".

[65] W. S. Levine, *The Control Handbook (three volume set)*. CRC press, 2018.

[66] J. Buchli, M. Kalakrishnan, M. Mistry, P. Pastor, and S. Schaal, “Compliant quadruped locomotion over rough terrain,” in *2009 IEEE/RSJ international conference on Intelligent robots and systems*, pp. 814–820, IEEE, 2009.

[67] F. Petit and A. Albu-Schäffer, “State feedback damping control for a multi dof variable stiffness robot arm,” in *2011 IEEE international conference on robotics and automation*, pp. 5561–5567, IEEE, 2011.

[68] M. Focchi, *Strategies To Improve the Impedance Control Performance of a Quadruped Robot*. Doctoral dissertation, University of Genoa, Italy and Istituto Italiano di Tecnologia (IIT), April 2013.

[69] M. Focchi, G. A. Medrano-Cerda, T. Boaventura, M. Frigerio, C. Semini, J. Buchli, and D. G. Caldwell, “Robot impedance control and passivity analysis with inner torque and velocity feedback loops,” *Control Theory and Technology*, vol. 14, pp. 97–112, 2016.

[70] T. Labs, “Mechaduino: Powerful open-source industrial servo motor.” https://tropical-labs.com/wp-content/uploads/2018/10/MechaduinoManual0.1.2.pdf.

[71] S. Electronics, “Load cell amplifier hx711 breakout hookup guide.” https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide/all, Accessed: 2023.

[72] Tropical Labs, “Mechaduino 0.2 Servo.” https://tropical-labs.com/shop/mechaduino-0-2-servo/, 2023.

[73] N. I. Corporation, “Strain gauge measurement – a tutorial.” http://elektron.pol.lublin.pl/elekp/ap_notes/ni_an078_strain_gauge_meas.pdf, 1998.

[74] “Solidworks.” https://www.solidworks.com/.