
# Optimization of Non-Linear Robust Controller for Robotic Manipulator System

By

**Mahmoud Mohamed**

**Thesis submitted in fulfilment of the requirement for the degree of Doctor of Philosophy**

School of Engineering

Cardiff University

United Kingdom

Cardiff, 2023

# ABSTRACT

This study focuses on the integration of artificial intelligence and knowledge-based systems to enhance the control of a complex multi-link mechanism. The research involves the development and examination of the Robogymnast as a platform for investigating the complexities and difficulties associated with a three-link robot system. By utilising modelling, simulation, and advanced control methods, the objective of the research is to improve the overall performance and manoeuvring capabilities of mechanisms with limited actuation, thereby contributing to the progress of robotics.

A mathematical model of the acrobot movement is constructed using the Lagrange equations, representing the motion of the robotic gymnast. This presents a control challenge due to its nonlinear and multivariable nature. To address this, a discrete-time linear model is proposed that specifically concentrates on the swinging action of the Robogymnast. The system's mathematical model is linearised for the investigation, providing a way to explore the determination of state space within the system. This work proposes and examines an approach to control the triple-link Robogymnast and assess its stability. In this study, a Proportional Integral Derivative (PID) controller is implemented and compared with a Linear Quadratic Regulator (LQR) to evaluate and investigate the Robogymnast system. The study also explores factors influencing the control of swing in the underactuated three-link Robogymnast.

This research also endeavours to enhance the performance of a proposed PID controller by employing two distinct algorithms, the Ant Colony Optimisation (ACO) and the Gravitational Search Algorithm (GSA), to stabilize the triple-link Robogymnast robotic system. These algorithms are utilized to fine-tune the PID controller parameters before its integration with the robot for subsequent stability response evaluation. The primary focus of the study lies in examining the application of a PID controller within a three-link robotic system. The findings indicate that the ACO algorithm with PID succeeds in enhancing the system's performance when contrasted with the GSA using a PID controller. The optimised results of the system demonstrate a significant reduction in overshoot by 95.46%, from 6.386 *p.u* to 0.290 *p.u* for the first joint. The values remain at 1 *p.u* for joints 2 and undergo a minor change of 3%, going from 0.309 *p.u* to 0.300 *p.u* for the third joint. The rise and settling times are also significantly reduced. Importantly, the Integral Time Absolute Error (ITAE) for the first joint is reduced by 87.63%, decreasing from 94.180 to 11.650. As for the second joint, there is a slight reduction of 2.21%, resulting in a value of 0.310. Lastly, the third joint shows a small enhancement of 4.76%, going from 0.021 to 0.020.

This study focused on developing a method for controlling the motion of the Robogymnast system by synchronising the stepper motors. To analyse the system's performance, a simulation was created using MATLAB/Simscape, examining various phases. The simulation and practical implementation of the controllers were carried out using MATLAB® and the STM32F microcontroller. In summary, this research contributes to the field of robotic systems by highlighting the significance of advanced control techniques and optimisation algorithms.

# DEDICATION

*Thank you, ALLAH (SWT), for giving me the strength and ability to complete this thesis.*

********

To

MY MOTHER,

MY FATHER,

MY WIFE AND CHILDREN

MY BROTHERS, SISTERS, AND GRANDFATHER

ALL MY FRIENDS

TO MY BELOVED COUNTRY ‘LIBYA’

# ACKNOWLEDGEMENTS

**Alhamdullilah,** firstly, I want to express my gratitude to ALLAH (SWT), the lord of the worlds for making it possible for me to complete this thesis.

I would like to express my gratitude to my supervisors, Dr Fatih Anayi and Dr Michael Packianather, for providing me with the opportunity to carry out this highly interesting and relevant research, as well as for their help and support throughout the period of this research work.

I am eternally grateful for the financial support from the Ministry of Higher Education and Scientific Research in Libya throughout all my years of PhD study.

I would like to express my gratitude to my mother and father for always being there for me and for their prayers and support. As well as my thanks also go out to my darling wife for endless support and patience during these four years of study.

I would also like to extend my gratitude to all my colleagues at the school of engineering for their support, encouragement, and productive discussions during meetings and seminars. A special thanks to Mr. Andrew Rankmore and the technical team at Cardiff University workshop for their contribution in building the practical aspect of the system.

## LIST OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATION

| | |
|---|---|
| ACROBOT | Acrobatic Robot |
| ACO | Ant Colony Optimisation |
| AI | Artificial Intelligence |
| BA | Bees Algorithm |
| BVP | Boundary Value Problem |
| DoF | Degrees of Freedom |
| DC | Direct Current |
| DLQR | Discrete-time Linear Quadratic Regulator |
| DIP | Double Inverted Pendulum |
| DIP | Downward Initial Position |
| ENN | Elman neural network |
| ECM | Equivalent Centre of Mass |
| GA | Genetic Algorithm |
| GSA | Gravitational Search Algorithm |
| IJC | Independent Joint Control |
| ITAE | Integral Time of Absolute Error |
| LQR | Linear Quadratic Regulator |
| LC | Local Control |
| MOEA | Multi-Objective Evolution Algorithm |
| MDFC | Multiple-prediction Delayed Feedback Control |
| NN | Neural Network |
| PSO | Particle Swarm Optimisation |
| PC | Personal Computer |
| PV | Process Variable |

| | |
|---|---|
| PD | Proportional Derivative |
| PID | Proportional Integral Derivative |
| Robogymnast | Robot gymnast |
| SLS | Selective Laser Sintering |
| SP | Set-Point |
| SG | Speed Gradient |
| TLIPs | Triple Link Inverted Pendulums |
| UTP | Upward Target Position |
| 3 DoF | 3 Degrees of Freedom |

# LIST OF SYMBOLS

| Symbol | Description |
|---|---|
| $O_{sh}$ | Overshoot |
| $T_r$ | Rising time |
| $T_s$ | Settling time |
| $U_{sh}$ | Undershoot |
| $\theta$ | (*Theta*) Angle, joint |
| g | Gravity |
| iter | Iteration |
| L | Lagrange |
| T | Kinetic energy |
| t | Time |
| V | Potential energy |
| s | Seconds |
| *p.u.* | Per unit |

# Chapter 1: Introduction

## 1.1 Introduction

An inverted pendulum is an ideal tool for the experimental study of control theories and an effective model for testing control policies in control engineering. There is strong non-linearity in the triple inverted pendulum, and the system is complex. As a result of the multivariable of such systems, it is difficult to model and control their stabilizing action and swing. Additionally, given the dynamic nature of the structural components in this highly intricate underactuated system, this model becomes particularly beneficial for conducting simulations, making comparative assessments, and optimizing diverse control strategies like the linear quadratic regulator (LQR) and proportional-integral-derivative (PID) controllers [1][2]. The current study focuses on controlling the motion of a manipulator robot with under-actuation. The multi-link robotic system can perform various tasks and has been studied extensively in the field of AI. However, there is still significant potential for further research. The primary objective of this study is to investigate the swinging of a triple-link Robogymnast mechanism, which includes two active-power links and an overhead non-powered link [3][4].

Studying a multi-link robotic system, such as a Robogymnast, holds significant importance in the field of robotics and automation for several reasons. Understanding the dynamics and control of multi-link systems is crucial for optimizing their performance, ensuring safety, and enabling them to perform complex tasks with precision. Such research can lead to advancements in robotic prosthetics, where replicating natural limb movement is

essential for improving the quality of life for individuals with limb disabilities. Moreover, multi-link robots are pivotal in hazardous environments, as they can be used to replace humans in tasks that are dangerous or inaccessible, such as search and rescue missions.

Secondly, the study of multi-link robotic systems contributes to the broader field of robotics and artificial intelligence. These systems require advanced control algorithms and sensor integration, pushing the boundaries of AI research. By developing and fine-tuning these algorithms, researchers can enhance the autonomy and adaptability of robotic systems, making them more capable of handling real-world scenarios.

## 1.2 Motivation

The inverted pendulum is an interesting underactuated system that has been widely studied for its control algorithms. It is often depicted as a single inverted link mounted on a movable cart, which can represent something like a rocket booster during lift off [5]–[8]. However, inverted pendulums can also have multiple links, which adds more degrees of freedom and complexity to the system. These multi-link systems are nonlinear, multivariable, and pose significant challenges in terms of modelling and control [9]–[11].

A three-link Robogymnast is a robotic system that has three links, or segments, connected by joints. It is possible that a Robogymnast with three links could be used in a variety of applications, depending on the specific design and capabilities of the system. Some potential applications could include:

- Rehabilitation: A three-link Robogymnast could potentially be used as a tool for physical therapy or rehabilitation, to help patients recover from injuries or conditions that affect their mobility [3].

- Research: Researchers in the field of robotics can utilize a three-link Robogymnast as a testbed for studying the dynamics of multi-link robotic systems or for developing and testing new control algorithms [3].
- Education: The system could be used as a teaching tool in classrooms or workshops, to help students learn about robotics, mechanics, and control systems.

It is essential to understand that complex (underactuated) multi-link structures provide excellent test environments for evaluating, optimizing, and comparing different control methods. The nonlinearity of these systems makes them suitable for a variety of real-life applications and presents challenging modelling and control challenges. In recent years, it has been utilized extensively for the study of control algorithms. The investigation of such systems will allow researchers to create answers to the motion challenges encountered by injured or disabled people with impaired limbs [3][12]–[14]. Furthermore, the development can involve viewing the robot as a multi-link system that resembles a simplified model of a human standing on two legs [15][16].

This research introduces a rebuilt design with new components for a three-link robot gymnast, called Robogymnast, with two powered joints. Each link represents the symmetry of the human body with two hands and two legs. The system is mounted on a freely rotating high bar. The goal of the robot is to perform dynamic movements to achieve the desired motion. In this case, the robot is meant to mimic a human acrobat hanging from a high bar and attempting to swing motion. However, unlike a human acrobat, the robot's hands are firmly attached to the freely rotating high bar, which is mounted on ball bearings. The rotation of the high bar aids in the swing phase. The research investigates the development and implementation of various control strategies to address this issue. Figure 1.1 shows the actual Robogymnast.

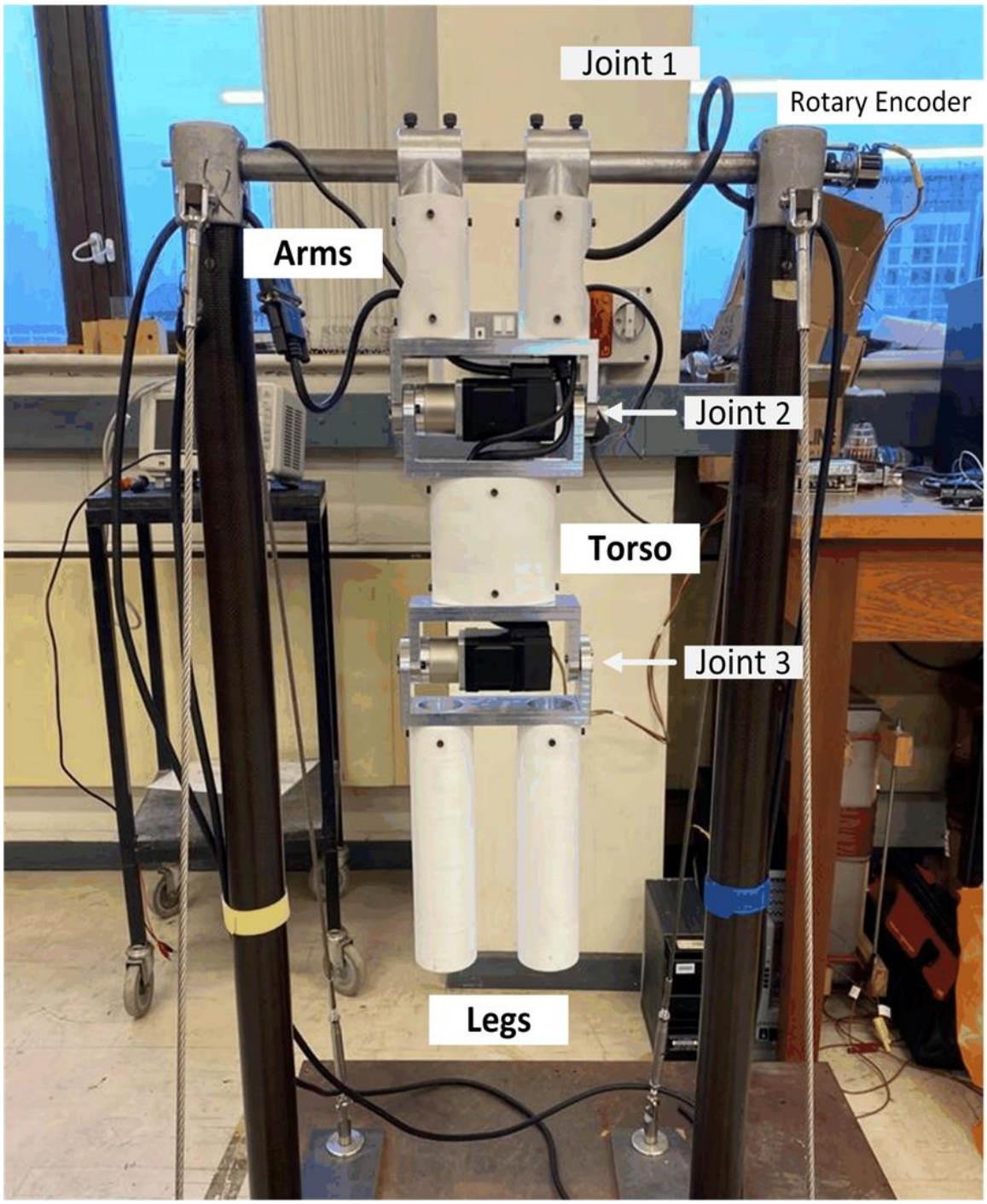


Figure 1.1 Robogymnast body

Exploring this system is driven by a motivation – the development of Robogymnast. This system is equipped with all the essential physical components necessary to achieve the desired motion. Notably, this endeavor represents a pioneering application of a stepper motor for synchronizing the system's motion, marking a novel contribution to this research.

## 1.3 Problem statement

Robotic systems are growing in significance across various sectors, such as manufacturing, healthcare, and entertainment. A specific area of interest involves managing multi-link robotic systems, which can be quite intricate due to the interplay among various links and joints. This complexity issue includes:

1. Dynamics: The dynamic characteristics of multi-link robotic systems may exhibit complexity, particularly when taking into account the interrelations among various links and joints. This may result in difficulties in precisely anticipating the system's movement and effectively regulating its motion.
2. Sensing and feedback: Precisely determining the system's state (for instance, joint angles, velocities, and accelerations) can be challenging due to factors such as measurement noise. Furthermore, controlling the system effectively based on the ascertained state may be demanding, particularly when addressing high-dimensional systems or in situations where the system's dynamics exhibit considerable uncertainty.
3. Real-time constraints: In numerous instances, the robotic system has to rapidly adapt to alterations in its surroundings or to instructions from a human operator. This necessitates the control system to function in real time and be capable of managing swiftly evolving inputs and circumstances.
4. System complexity: multi-link robotic systems can be highly complex, particularly when taking into account the number of links and joints, each joint's degrees of freedom, and the interplay among the links. This may cause challenges in designing and executing efficient control algorithms for the system.
5. Scalability: As the number of links in a multi-link robotic system rises, the control problem's complexity also increases. Consequently, it is critical to develop control

algorithms that are scalable and capable of managing systems with an extensive array of links and degrees of freedom.

## 1.4 Research aim, objectives and contributions

The aim of this research is to rebuild, simulate, and design a controller for an underactuated three-link Robogymnast system after which analyze the outcomes. The main target of the research is to use modelling, simulation, and control of underactuated mechanisms to gain a comprehensive understanding of modern control techniques, optimization and their applications in industry and society.

### 1.4.1 Research Objectives

The aim of this study is accomplished by achieving the following objectives:

- To use a mathematical model and proposed controller techniques, simulate and analyse the stabilisation control system for the Robogymnast.
- To build and operate a comprehensive system comprised of various appropriate components, including the installation of two stepper motors on the 2nd and 3rd joints, a rotary encoder on the top shaft for motion measurement, and the utilization of a contemporary microcontroller for system control.
- To develop a new method for the 'swinging' of a 3-link Robogymnast system mounted on a freely rotating high bar using stepper motors.
- To apply a swarm-based optimization technique to optimize a parameter that affects the control actions applied to the two motors driving the robot simultaneously.
- To select the optimum parameters for the controllers using swarm-based optimization.

- To implement the fine-tuned parameters for initiating the swing process in the real-time robotic mechanism.
- To evaluate and confirm the validity of the simulation model (Simscape) and contrast the suggested control mechanisms via simulation.

### 1.4.2 Contribution to knowledge

The novelty of this work is summarised in the following points:

- Built a triple-link robotic system (Robogymnast) equipped with all necessary physical components needed to produce the system's desired motion. For the first time in such an application, a stepper motor is being utilised to synchronise and examine the motion.
- Implement a Proportional Integral Derivative (PID) and Linear Quadratic Regulator (LQR) controller and perform a comprehensive analysis to assess the robustness of each controller in controlling the operation of the stepper motors in the Robogymnast system.
- Ant Colony Optimization (ACO) algorithm implemented on the acrobot system, which was used to fine-tune the parameters of the proposed controller.

## 1.5 Methodology

To accomplish the previous objectives, the following methodology was utilized:

- Previous work review: As part of the research, a comprehensive survey of the current state of the art was performed in order to identify the primary requirements for controlling complex multi-link mechanisms as well as the associated problems. Furthermore, the scope of this study also includes the implementation and analysis of control methods.

- The Euler-Lagrange approach is applied to derive a mathematical model and dynamic equations for the Robogymnast.
- A contemporary three-link robotic system (Robogymnast) will be constructed and assembled utilizing modern manufacturing technologies, such as aluminium and SLS materials.
- Each link of the Robogymnast system will be synchronised for smooth movement utilizing a stepper motor, a novel approach to this process.
- MATLAB software is utilized to simulate swing control with its associated toolboxes. The parameters are optimized using the ACO algorithm, and the results are implemented in the real system through the microcontroller.
- The candidates for the alternative model are investigated and evaluated in order to determine their suitability. Validation of the developed model is conducted by comparing it with the mathematical model and experimental data.
- The swinging control system is implemented. That will be achieved via STM Microcontroller, and its software Additionally, these tools help to obtain the outcomes data of the system.
- The results from both the simulation and the real-time system are investigated and compared.

## 1.6 Thesis outline

The thesis contents are organized as follows:

Table 1.1 The Layout of thesis

| Chapter No. | Description |
|---|---|
| 1 | This chapter offers an overview of the study, delineates the aim, objectives, outlines contributions, methodology and published works related to the topic. |
| 2 | This chapter delivers a comprehensive and current review of the literature regarding complex multi-link robotic systems, specifically concentrating on stabilization control issues, swing control tactics, and system-related optimization methodologies. Furthermore, it offers an evaluation of each part and provides a succinct summary. |
| 3 | The third chapter introduces the system description and mathematical model steps of the Robogymnast. The overall system is discussed and illustrated using Figures and diagrams. The design of the system (Setup) is discussed, including the components and prototype of the system. |
| 4 | The chapter presents a study on controller design for a non-linear, three-link robotic gymnast named Robogymnast. The focus is on exploring the swing of a triple-link system with two powered links and an unpowered one. Two control techniques, LQR and PID controllers, are examined, with the system modelled and simulated in MATLAB. ITAE was used to evaluate the effectiveness of both controllers, with the PID controller showing superior results. concludes with an analysis of the PID controller's robustness. |
| 5 | This chapter establishes the selected algorithms, ACO and GSA, providing an overview of each one. It also imparts an understanding of the fundamental concepts of both algorithms. Lastly, it compares the ACO with the GSA results to identify the best possible performance of the proposed controller. |
| 6 | This chapter showcases the outcomes of both simulation and real-time performance, followed by a comparative analysis to evaluate the system's performance. This comparison aids in confirming the optimized results for each segment of the multi-link motion system. |
| 7 | The final chapter encapsulates the key elements of this dissertation and sketches out prospective pathways for future research. |

## 1.7 Publications

***Journals*:**

- M. Mohamed, B. Abdul Samad, F. Anayi, M. Packianather, and K. Yahya, “Analysing Various Control Technics for Manipulator Robotic System (Robogymnast),” *Computers, Materials & Continua*, vol. 75, no. 3, pp. 4681–4696, 2023, doi: 10.32604/cmc.2023.035312.
- B. Abdul Samad, M. Mohamed, F. Anayi, and Y. Melikhov, “An Investigation of Various Controller Designs for Multi-Link Robotic System (Robogymnast),” *Knowledge*, vol. 2, no. 3, pp. 465–486, Sep. 2022, doi: 10.3390/knowledge2030028.
- B. Abdul Samad, M. Mohamed and F. Anayi, "Enhanced the Control Strategy of a Triple Link Robotic System (Robogymnast)," in IEEE Access, vol. 11, pp. 31997-32005, 2023, doi: 10.1109/ACCESS.2023.3262190.

***Conferences*:**

- M. Mohamed, F. Anayi, M. Packianather, B. A. Samad, and K. Yahya, “Simulating LQR and PID controllers to stabilise a three-link robotic system,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 2033–2036. doi: 10.1109/ICACITE53722.2022.9823512.
- B. A. Samad, F. Anayi, Y. Melikhov, M. Mohamed and E. Altayef, "Modelling of LQR and Fuzzy-LQR Controllers for Stabilisation of Multi-link Robotic System (Robogymnast)," *2022 8th International Conference on Automation, Robotics and Applications (ICARA)*, 2022, pp. 33-38, doi: 10.1109/ICARA55094.2022.9738577.
- B. A. Samad, M. Mohamed, F. Anayi and Y. Melikhov, "A hybrid Fuzzy approach of different controllers to stabilize a 3-link swinging robotic (Robogymnast)," *2022 2nd International Conference on Advance Computing and Innovative Technologies in*

*Engineering (ICACITE)*, 2022, pp. 2432-2437, doi: 10.1109/ICACITE53722.2022.9823768.

- Abdul Samad, B.; Mohamed, M.; Anayi, F. Motion planning of triple links robotic system, *in Proceedings of the 3rd International Electronic Conference on Applied Sciences*, 1–15 December 2022, MDPI: Basel, Switzerland, doi:10.3390/ASEC2022-13774.
- B. A. Samad, M. Mohamed, and F. Anayi " Triple link robotic system (Robogymnast): A comprehensive overview," *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2023, *in press.*

# Chapter 2: Background Review

## 2.1 Introduction

This chapter reviews the problems and challenges associated with the design and implementation of underactuated systems. The background review conducted in this chapter provides an in-depth comprehension of the current state of the research area, highlighting the key contributions made by different authors. In addition, this chapter also presents a summary of the conventional control and trainable control theories, highlighting their key concepts and how they are applied in the context of underactuated control systems. The emphasis on the swinging problem serves as an example of how these control theories can be applied to solve specific issues in underactuated control systems. Overall, the primary objective of this chapter is to furnish a thorough comprehension of the symbolism, theoretical constructs, and challenges inherent to underactuated systems, as well as the current state of research in this field.

## 2.2 Background

Three-link mechanisms system offers an excellent model to apply within control engineering to test control policies, being an optimal experimental instrument for studying control theories. The triple inverted pendulum is a highly non-linear, open- loop system characterised by its instability. This kind of multi-link system is multivariable and represents a challenging modelling and control problem in terms of swing and stabilisation. At the same time, the

dynamic characteristic of structural features of the high-complexity underactuated mechanism means that it provides a productive approach for testing in terms of comparative evaluation, simulation, and optimisation of diverse control methods, e.g., proportional-integral-derivative (PID) controllers, linear quadratic regulator (LQR) controllers, and other techniques.

A large number of research have examined the behaviour of inverted pendulums [17]. The majority of control experiments in the field of inverted pendulum utilize the rail-cart setup [18]–[20]. However, there is an increasing interest in pendulums that swing, such as the acrobot and Robogymnast, as they have potential uses in the development of walking robots [21]. Previous research has concentrated on various controller designs, with some studies focusing on developing control systems for swinging motion, while others have concentrated on creating control systems for maintaining an upright balance. There have also been efforts to merge the control systems for swinging and balancing [15][22].

## 2.3 Complex multi-link mechanism

A mechanism refers to a series of interconnected parts that facilitate the transfer and alteration of physical movement [23]. Complex multi-link mechanisms are mechanisms that have fewer linkages than the number of degrees of freedom (DoF) they possess. In other words, the number of ways that these mechanisms can move is greater than the number of links connecting the parts [24]. Complex multi-link mechanisms can be referred to as underactuated mechanisms, and they provide advantages such as reduced energy consumption, more efficient use of materials, and less space required when used in different applications [24]. In academia, underactuated mechanisms serve as a valuable experimental platform for assessing and contrasting various control techniques [25]. Controlling underactuated systems is difficult

because many of them cannot be linearized using full-state feedback, particularly around equilibrium points. Furthermore, some of these systems may not be controllable locally over short time periods, which adds to the challenge of controlling them. As a result, the control of underactuated systems is a challenging problem [26]. Control engineering and robotics have recently been concentrating on studying the control of these mechanisms, which has become a significant research topic [27]. The inverted pendulum is a well-known instance of an underactuated mechanism.

The designing and fabricating a kinematic walking machine were conducted in [28], which is a four-legged device that can walk on any surface. The machine is powered by a single motor and each leg is made up of a four-bar mechanism based on Chebyshev's parallel motion equation. The machine's motion is achieved by the rotation of the four-driving links that have a 90° angular difference, causing the legs to move front and back to move the body. Brass bushings are used to reduce friction between rotating parts, and the movement of the legs is traced and compared to theoretical values. Machines and mechanisms have been used since ancient times to reduce human effort and have impacted almost every aspect of human society since the industrial revolution [28].

A pendulum is an object that hangs from a fixed point and swings back and forth due to the force of gravity. It's often used to control movements [29]. Even though they are nonlinear, inverted pendulums are still useful and can now be used to demonstrate concepts in nonlinear control, such as the ability to catch the pendulum and swing. Pendulums are ideal for demonstrating hybrid systems and the management of chaotic systems. The double and triple link pendulums have been extensively studied as a test platform for non-linear control. The authors of [30] suggest that pendulums are well-suited for illustrating these concepts.

## 2.4 Swing control

The issue of the swing of multi-link mechanisms systems with passive joints has been the subject of extensive research by many scientists and engineers. The goal of this research is to develop methods for controlling the motion of these systems, particularly during the swing phase, in order to achieve the desired outcome. The study [3] focused on a complex underactuated multilink mechanism called the Robogymnast, which is a triple-link underactuated pendulum. The aim was to understand its complex nature and challenges in developing its control system. A mathematical model of the robot is derived using Euler-Lagrange equations, and the control system is based on a discrete-time linear model around the downward position. Invasive Weed Optimization (IWO) is used to optimize the swing motion of the robot, and the values obtained from IWO are applied to both simulation and experiment. A novel approach of modelling the Robogymnast using a multi-layered Elman neural network (ENN) is proposed to deal with the complexity and nonlinearity of the system. The results showed that the ENN model provides a better representation of the actual system compared to the mathematical model. The study highlighted the potential of controlling complex underactuated multilink mechanisms through the manipulation of their natural dynamics to design more energy-efficient machines with smooth motions similar to those found in the natural world.

This study [15] involved the construction of a three-link robot gymnast (Robogymnast) powered by two geared DC motors. A mathematical model is derived using Lagrange equations to deal with the control challenges of the robot's motion. The study proposes an approach for designing the control system based on a discrete-time linear model around the upright position of the Robogymnast. A new technique is also proposed to manipulate the frequency and amplitude of sinusoidal signals to study the swinging motion of the Robogymnast. The Bees

Algorithm is used to optimize the control parameters for the swinging motion. Two different control methods are adopted to study the balancing/stabilizing of the Robogymnast in both the downward and upright configurations. A switching mechanism between swinging and balancing algorithm is also proposed to explore the combination between the two motions. The simulation and experimental realization of the controllers are implemented using MATLAB software and the C++ program environment.

The authors of [22] focused on a control method designed to stabilise the large swinging movements of a 3-link horizontal bar gymnastic robot. The authors have expanded on the Multiple-prediction Delayed Feedback Control (MDFC) method, which has previously been proven effective in controlling chaotic systems. This study takes into account the impact of friction at the link joints on the dynamic of the underactuated gymnastics robot, which can produce significant differences compared to a friction-free scenario. The proposed control method includes three types of control inputs, one of which is the MDFC for ensuring asymptotic stability and the remaining two for compensating for friction. The results from numerical simulations and experiments demonstrate the efficacy of this proposed method.

There has been a focus on understanding the challenges presented by passive joints and developing strategies to overcome them to achieve successful swing [31]–[39]. The authors of [40] discussed various attempts to control the movements of the inverted pendulum. The research has proposed a controller that utilizes a combination of a feedforward controller for swinging the acrobot up and a feedback controller for stabilizing it in the upright position.

The study in [31] delves into the problem of controlling an underactuated, three-link robot gymnast, referred to as Robogymnast, in order to replicate the movement of a human acrobat swinging to an inverted position while hanging on a high bar. A unique challenge presented in this task is balancing the robot while it is in an upright position, as its hands are firmly attached

to a freely rotating high bar. The solution employed in this study involves applying varying sinusoidal torques to the robot's shoulder and hip joints, with the amplitude of the torque increasing and frequency decreasing as the swing angle increases. The experiments produced positive results, with Robogymnast successfully swinging from a stable downward position to an inverted one.

In [34] approached the swing problem differently by combining 2nd and 3rd links into a single virtual link and applying a coordinate transformation to the angles of joints 2 and 3. This research article focuses on addressing a specific control issue related to a three-link gymnastic robot that moves in a vertical plane. The robot's first joint is not actuated (passive) while the rest are actuated (active) as shown in Figure 2.1. The problem being addressed is the swing control issue.

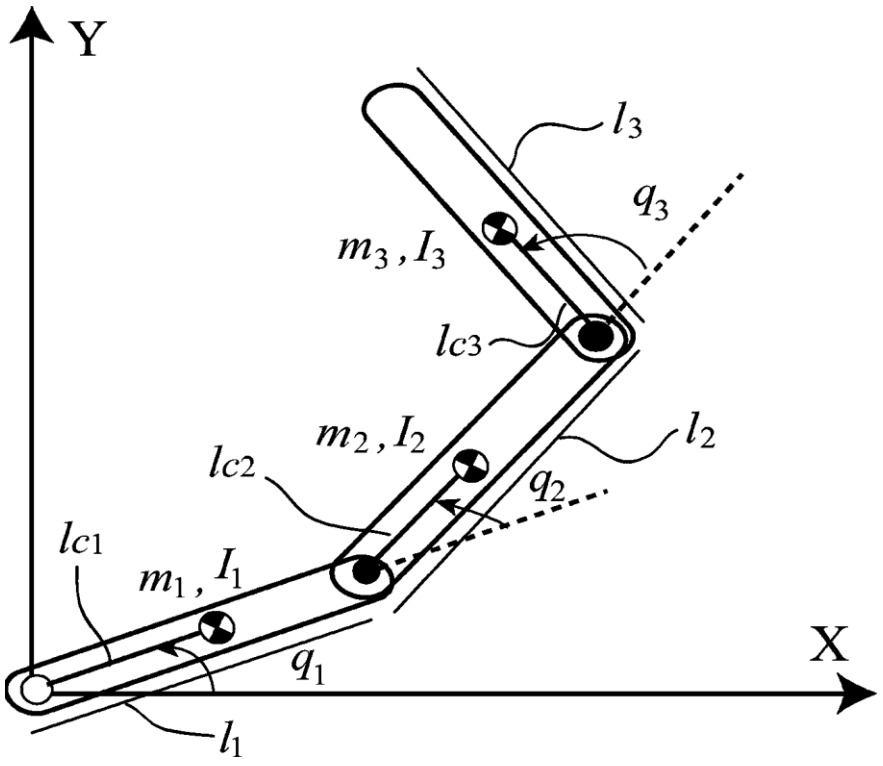


Figure 2.1 Three-link underactuated robot [34]

The study [34] conducted a comprehensive examination of the robot's motion under a specific controller, determining specific requirements for the control parameters to successfully achieve the swing control goal. The study also includes simulation results based on the parameters of a three-link robot that is modelled after a human gymnast, to support the validity of the theoretical findings.

The authors of [35] examined the utilization of energy-based methods for resolving the challenge of the swing, which had been previously attempted. Through an in-depth examination of the swing problem, they accomplished the control objectives by steering clear of singular points in the control law with a higher controller factor than had been utilized in previous efforts.

The properties of basic techniques for swinging an inverted pendulum were discussed in the study [37], and demonstrated that the pendulum's performance is closely linked to the ratio of the pivot's maximum acceleration to the acceleration due to gravity. The study also compared energy-based strategies with the minimum time strategy and found that the minimum time solutions are more robust.

Research in [41] assessed the design and implementation of computer control systems for balancing and controlling the attitude of double and triple-inverted pendulums. A DC motor at the upper joint is used for the double pendulum, and a combination of a DC motor at the middle joint and proportional position control at the upper joint is used for the triple pendulum. The controllers are based on linearized discrete-time models and utilize state feedback through reduced-order state observers. They are designed using MATLAB and implemented in C language on a PC and were found to have satisfactory performance in experiments.

A research discussed the dynamics of a swinging two-bar linkage and to achieve a range of motion patterns. They developed a neural controller to control the system's movement by mimicking the movement of human arms during a harmonic swing. The controller was designed to achieve both small swings and large rotational movements. One advantage of this approach is improved control over the movement of the system. However, the realism of the resulting movements is heavily dependent on the accuracy of the model and the movement controller used [42] .

The authors of [43] investigated a study on controlling a gymnastic robot with three links. They suggested using a neurocontrol system to execute the swinging motion of the 3-degree of freedom manipulator. The controller they proposed utilized a neural network (NN) and a genetic algorithm (GA). To make the robot's movements more similar to those of a human gymnast, the researchers applied constraints to the angles of the joints. Then examined the controller's performance with various swing timings and conducted control simulations. The results of the simulations indicate that the neurocontroller is capable of controlling the system within the specified constraints and swing timings [43].

The research in [44] proposed control method aimed to resolve the problem of driving Robogymnast from a downward to an upright position. It involved applying frequently changing sinusoidal torques to the two motors located at the hip and shoulder joints. Unlike other control methods, this technique did not consider the angular positions or velocities of the robot when generating control signals.

A new swing controller has been developed for a three-link acrobot as illustrated in Figure 2.2, utilizing human simulated intelligence control theory. The swing process has been divided into six phases, based on the analysis of athlete's movements. These phases have been described using sensory-motor intelligence, Bang-Bang and PD controllers have been designed for each

phase. The controller allows the acrobot to swing efficiently and smoothly. Simulation results demonstrate the effectiveness and viability of the proposed approach [45].

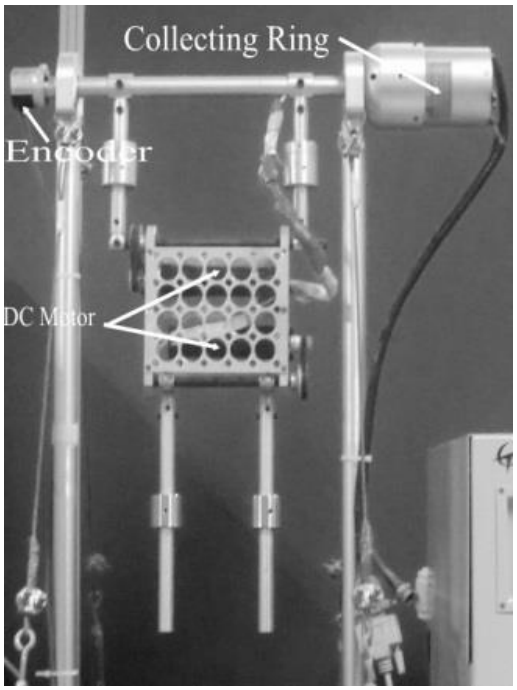


Figure 2.2 Physical Model of 3-link acrobot [45]

The control of a triple pendulum on a cart was discussed in [46] as depicted in Figure 2.3. Utilizing a two-part control system that includes a nonlinear feedforward controller and an optimal feedback controller. The task of moving the pendulum from one point to another is considered as a nonlinear problem with adjustable parameters that are determined from the input and output dynamics. The main emphasis of the research is to demonstrate the practical application of the triple pendulum swing technique. The control method for the swing manoeuvre was successfully applied and evaluated through experiments on a test bench.

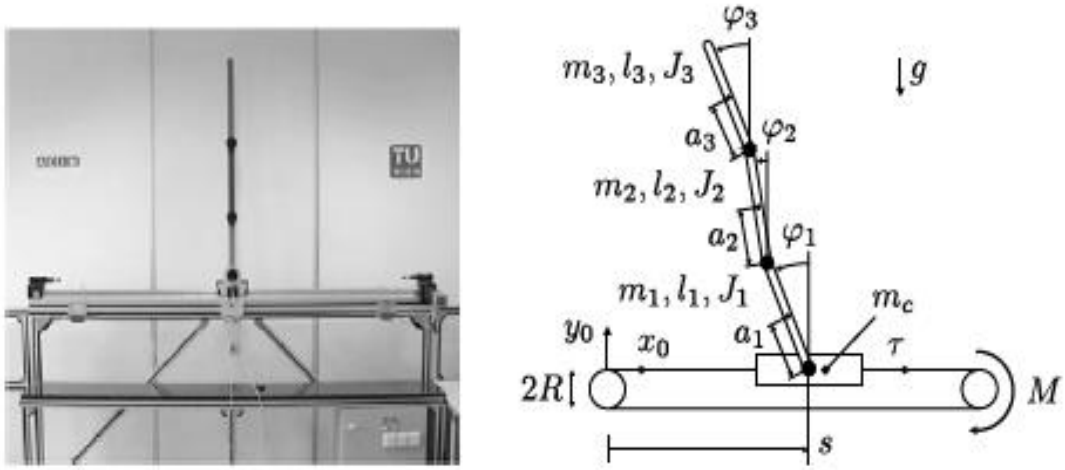


Figure 2.3 Triple pendulum on a cart [46]

The replication of the swing and giant swing motion of underactuated robots using the technique of horizontal bar gymnastics was discussed in [47], focusing on the equivalent centre of mass (ECM) of the robots and the gymnast. The efficient motions of the gymnast's ECM are identified for both swing and giant swing using motion capturing techniques. A partial linearization method is designed to make the ECM of the underactuated robot replicate this motion, which is used to reproduce the gymnast's swing and giant swing motions in the robots. The effectiveness of the proposed controller is demonstrated through numerical simulations.

Research in [48] proposed a control method for the vertical acrobot system to move it from a downward initial position to an upward target position. The method involves framing a trajectory with adjustable parameters to stabilize the active link and adjusting the trajectory parameters to bring the passive link close to the end angle. A PD-based tracking controller is then used to track the planned trajectory, and a stabilization controller is implemented to stabilize the system at the UTP. The method was validated through simulations.

The authors of [49] proposed a control law to stabilize the energy of a two-link gymnastic robot, called acrobot, to a desired value during both swing and giant-swing motions as depicted

in Figure 2.4. The control method involves periodically changing the position of the second link and modulating the amplitude of the periodic input based on the deviation from the desired energy. The study uses the averaging method to analyse the swing and giant-swing motions and derives the averaged equations and energy equations. The proposed control method is shown to effectively control the energy of the real acrobot, and it can be applied to other similar systems. The conclusion highlights the effectiveness of the proposed control approach and its potential application to other systems such as pendulums and pendubot.

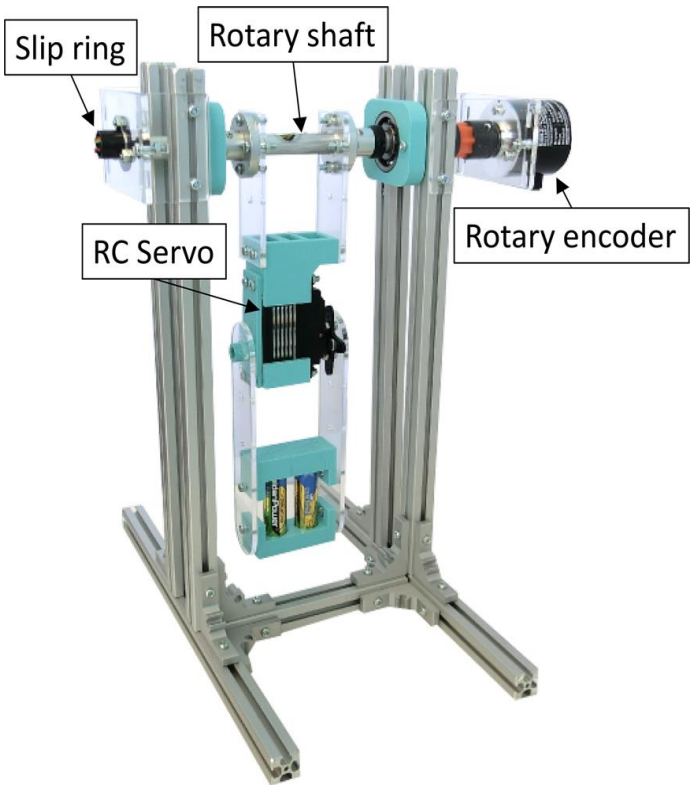


Figure 2.4 Experimental system of acrobot [49]

## 2.5 Optimisation algorithm

In the present context, optimisation refers to the process of identifying the most effective decision among different alternatives in a given problem. This involves finding the optimal

value of a function, considering the constraints or limitations involved [50]. Optimization algorithms are used to compare and iterate different solutions until an optimal or satisfactory result is obtained. Computer-aided design activities now incorporate optimization techniques.

Every optimization problem is composed of fundamental elements that include the following [51]:

- The function that represents the quantity intended to be optimized is referred to as the objective function.
- The value of the objective function is influenced by a group of variables or unknowns.
- A collection of limitations or conditions that limit the possible values that can be assigned to the unknowns is referred to as a set of constraints.

The target of an optimization technique is to allocate values to the unknowns from the allowable domain, in such a way that satisfies the constraints and optimizes the objective function [52].

There are two main categories of optimization techniques in common use: deterministic optimization and stochastic optimization [53]. These optimization techniques are grouped based on the type of algorithm that is utilized to execute the optimization process.

Deterministic optimization techniques employ predefined rules for transitioning from one solution to another. These algorithms heavily rely on linear algebra and typically involve calculating the gradient of the response variable. Deterministic optimization methods are quicker than stochastic optimization since they require fewer evaluations of the response variable to obtain a solution. Nonetheless, deterministic optimization algorithms aim to locate a stationary point in the response variable. As a result, the optimal solution found may be a

local maximum or minimum rather than a global one. Additionally, deterministic algorithms inherently focus on a single objective [54].

Stochastic optimization techniques are similar in nature to probabilistic translation rules and are better suited to addressing problems where the relationship between inputs and outputs is unknown or complex. Stochastic optimization methods are generally categorized as general-purpose approximation search techniques that can be applied to a broad range of optimization problems. They offer a more flexible and adaptable approach to optimization than deterministic methods, making them useful in a variety of applications [55].

Many published studies exist that explain how evolutionary optimization algorithms can be used in designing and improving control systems [56]–[59]. According to [56] presented a new genetic approach to solve the Linear Quadratic Regulator (LQR) problem. The optimal LQR design depends on selecting proper weighting matrices, which is usually done through a trial-and-error process. To eliminate this trial-and-error process, the authors propose using a Genetic Algorithm (GA) to find the best weighting matrices and optimal control gain. The approach is applied to an inverted pendulum system, which is a well-known LQR problem, and the results obtained through computer simulations were satisfactory. The study in [56] concludes that the genetic approach presents a new possibility for finding the proper weighting matrices for LQR design, which is not a trivial problem. This method was implemented on an active suspension system [57]. Research in [59] discussed the application of an improved genetic algorithm to optimize the sliding mode controller for a double inverted pendulum system. The sliding mode controller is designed to stabilize the pendulum at an upright equilibrium position, which is challenging due to the system's nonlinearity. The genetic algorithm is applied to search for the optimal sliding surface and other key parameters to enhance the control system's performance. The improved genetic algorithm effectively improves global convergence and local search capabilities and avoids premature convergence.

The simulation experiments demonstrate that the sliding mode controller optimized by the improved genetic algorithm successfully stabilizes the double-inverted pendulum at the upright position with satisfactory performance [59].

Swarm-based optimization algorithms are becoming increasingly popular and have been utilized by researchers to address a wide range of engineering and manufacturing problems. One such algorithm is the particle swarm optimization PSO, which has been used in recent studies to develop a designed controller for the inverted pendulum system [60]. For example, [61] improved the PSO and used it to optimize the optimal controller, resulting in a successful stability control of a double-inverted pendulum. [62] utilized the PSO optimization technique to adjust the state feedback gains and evaluate the stability and tracking of a solitary inverted pendulum on a cart. By enhancing the Gaussian membership functions of the fuzzy model of a nonlinear issue using the PSO algorithm, [63] successfully extended the algorithm's capabilities. This resulted in the creation of an adaptive fuzzy logic controller that effectively achieved the desired outcome for the cart with the inverted pendulum system. The study in [64] conducted the use of (PSO) metaheuristic and two of its variants (inertia weight and constriction coefficient) to optimize the membership functions of fuzzy control systems for benchmark problems of water tank and inverted pendulum. The authors compare the advantages of each variant in the algorithm and find that PSO can be an efficient method for designing fuzzy control systems. The simulation results demonstrate the potential use of bio-inspired optimization methods in solving fuzzy control problems.

Research in [65] proposed a new approach to designing weighting matrices for linear quadratic regulator (LQR) using a multi-objective evolution algorithm (MOEA). The approach involves establishing a multi-objective optimization model for LQR weighting matrices and applying MOEA to it, resulting in the simultaneous achievement of multiple performance indexes in the control system. The validity of the proposed approach is confirmed through simulations of a

double-inverted pendulum system, which showed shorter adjusting time and smaller amplitude deviation from steady-state compared to the pole assignment LQR weighting matrices design approach. Therefore, the proposed approach provides a useful method for solving LQR weighting matrices designing problems.

The authors of [66] introduced a novel search algorithm named the Bees Algorithm (BA), which is based on the food foraging habits of honey bees. The algorithm employs a combination of random and neighbourhood searches and can be applied to both functional and combinatorial optimization problems. The researchers utilized the Linear Quadratic Regulator (LQR) approach to regulate the parameters of a fuzzy logic controller that was designed to stabilize an underactuated two-link acrobatic robot. Specifically, they used LQR to calculate the scaling gains required to develop the fuzzy logic controller [31][67].

Another study [68] presented an improved version of the Bees Algorithm, an optimization technique that combines global and local search strategies. The enhancement involves implementing a fuzzy logic system for selecting local search sites, which reduces the number of required parameters. The algorithm is shown to be effective through various optimization problems, including stabilizing a two-link acrobatic robot (ACROBOT) and tuning membership functions for a fuzzy logic system. Kalman filtering is introduced as a faster optimization method to guide worker bees towards optimal search sites. The resulting algorithm is applied to train a neural network for wood defect identification, and the improvements significantly enhance the algorithm's performance while maintaining its robustness.

The studies [69][70] suggested a novel approach for initiating the motion of a Robogymnast with three links and three joints. The method involved supplying energy to the Robogymnast to create a swinging motion, which would cause it to flip beyond the vertical position. This was accomplished by adjusting the control signals that were delivered to the Direct Current (DC)

motors installed at the shoulder and hip joints. To optimize the parameters governing the amplitudes and frequencies of the sinusoidal control signals. To achieve this, the researchers utilized newly developed optimization technique (Bees Algorithm) to tune the parameters of a swing control of a (Robogymnast) [25]. They randomly generated a set of parameter values, from which three were chosen to simulate the behaviour of the Robogymnast during the swing phase. The outcome demonstrated that the swing process of the Robogymnast was successful.

## 2.6 Ant Colony Optimization

The Robogymnast swing optimization problem requires the use of a stochastic optimization population-based search technique due to its random and adaptable nature, which is better suited to the problem's characteristics, such as the unclear relationship between the inputs and outputs. This type of optimization approach offers greater flexibility than deterministic methods and can explore a wider range of possible solutions, making it the preferred choice for this specific problem [3].

ACO stands for Ant Colony Optimization. It is a metaheuristic optimization algorithm inspired by the behaviour of ants when searching for food. Ants deposit pheromones on the ground to indicate favourable paths that other members of the colony should follow. ACO utilizes a similar mechanism for solving optimization problems [71][72]. In Ant Colony Optimization, a group of artificial ants construct solutions to the optimization problem and exchange information on the quality of these solutions through a communication scheme that is similar to the one used by real ants. This optimization technique is inspired by nature and mimics the behaviour of ants to find the optimal solution to a problem [73][74].

This developed metaheuristic algorithm has demonstrated its ability to perform well and adapt to various combinatorial optimization problems. It has been successfully utilized in different scenarios, indicating its resilience and flexibility [75]. ACO algorithms are constructed based on the following concepts [76]:

- For a given problem, every path taken by an ant represents a possible solution.
- As ants follow a path, the amount of pheromone they leave behind is linked to the quality of the solution associated with that path.
- When ants have to choose between multiple paths, those with higher pheromone levels are more likely to be selected, increasing the probability of discovering better solutions.

The three studies: the current study, [3], and [15], all focus on the control of a complex multi-link mechanism, specifically the Robogymnast. However, each takes a distinct approach to address this challenge. In the current study, the emphasis is on leveraging artificial intelligence and knowledge-based systems to enhance control. This is achieved using stepper motors for the first time in such an application. The study uses a PID controller and compares it with LQR for system control. Furthermore, it incorporates optimization techniques such as ACO and GSA algorithms for PID controller parameter tuning, with an emphasis on practical implementation.

In contrast, [3] centers on understanding the complexity of the Robogymnast and proposes an Elman neural network model as an alternative to traditional mathematical models. The study also introduces the concept of manipulating the robot's natural dynamics to design more energy-efficient control strategies, aligning the motion with those found in the natural world. It highlights the potential of neural networks in capturing the system's intricacies. Lastly, in study [15], the Robogymnast is constructed with a focus on control around its upright position. The study employs the Bees Algorithm for optimization and explores a switching mechanism

between swinging and balancing. It introduces a unique approach by manipulating the frequency and amplitude of sinusoidal signals to study the robot's swinging motion. The Bees Algorithm is employed for optimizing control parameters. Each study offers a unique perspective on enhancing the control of the Robogymnast, with optimization techniques, contributing valuable insights to the field of robotics.

## 2.7 Summary

Multi-link robot systems have become increasingly prevalent in various fields such as manufacturing, medical, and research purposes due to their ability to perform complex tasks with high precision and efficiency. This literature review aimed to provide an overview of the current state of research in multi-link robot systems, focusing on key areas such as design, control, sensing and enhancing. This section provided a summary of the different types of multi-link underactuated systems, and the controller methods used to achieve complex locomotion for n-link robot systems. Specifically, swing control, and balancing control have been discussed. The literature suggested that the design of multi-link robot systems is a crucial factor in achieving optimal performance. Studies have focused on optimizing kinematics, dynamics, and actuation systems to improve overall system performance.

The chapter also presented the use of optimization techniques to improve control performance. The insights and findings discussed in this review can provide valuable guidance for future research and development efforts in this area. The next chapter will focus on the Robogymnast system, including its description and the derivation of a mathematical model.

# Chapter 3: Mathematical Model and System Design

## 3.1 Introduction

This Chapter focuses on describing the Robogymnast system and developing its mathematical model. The Robogymnast is classified as a complex multi-link mechanism, specifically a triple-link pendulum. It is considered an underactuated mechanism due to its lack of full actuation, which poses challenges in designing a controller for the Robogymnast [15] [77]. As a result of these difficulties, the main essential challenges with Robogymnast are (i) The determination of the complex mathematical equations of motion, (ii) Build the entire system, (iii) The ability to move from one place to another and (iv) Analysis and control the motion of the system.

In this critical third chapter of the thesis, the study delves into a detailed exploration of the Robogymnast system. This involves a comprehensive description of the system, followed by the formulation of an exhaustive mathematical model that encapsulates its function and behaviour. The mathematical model will act as an accurate tool for understanding, predicting, and controlling the system's response under various conditions. Next, the study transitions into the design of the physical system. This phase includes the meticulous selection and specification of system components, which is informed by the insights gleaned from the mathematical model. The section concludes with an explanation of how these components are

skillfully integrated to create a coherent system capable of achieving the desired functionality. This systematic approach of exploring the system's model and design is not just vital for the successful operation of the system, but also lays a robust foundation for the ensuing chapters of this thesis. As the study progresses, will notice how the mathematical model serves as a springboard for developing an effective control strategy, and the system design will support its practical implementation.

In Section 3.2, an extensive overview of the system is provided, which includes descriptions and physical specifications, as well as a schematic diagram of the Robogymnast. Section 3.3 focuses on deriving and explaining the mathematical model of the system, offering a detailed account of the process from the Euler-Lagrange equation to the state space model. Section 3.4 discusses the setup of the Robogymnast system, detailing all the components used in this research for the execution of the experimental test. Finally, Section 3.5 concludes the chapter by summarizing the main points covered.

## 3.2 System description

The Robogymnast has 3 degrees of freedom, with two of them being actuated and the other one being unactuated. As the system is complex, designing controllers for the Robogymnast requires computer-simulated tests to ensure their effectiveness before they can be implemented on the real system. To achieve this, a mathematical model of the Robogymnast was developed using the Euler-Lagrange equations of motion. This mathematical model serves as a tool to simulate the behaviour of the Robogymnast and test different control strategies [78] [79]. In order to understand the behaviour of the Robogymnast under various conditions, it is necessary to derive the equations of motion. Additionally, these equations are used to simulate the performance of the Robogymnast when feedback control is applied [80].

The design of the three-link robotic system employed in this research is inspired by a common gymnastic movement where a person swings over a high bar using free rotation. A diagram of the system is illustrated in Figure 3.1. The first link is comparable to a gymnast's arms without wrist or elbow joints. The second link merges the gymnast's torso, neck, and head into a single entity, while the third link represents the legs and feet, excluding joints at the knees and ankles. The non-motorized passive joint (Joint 1) in the system is akin to the gymnast's hands, and the motorized second and third joints correspond to the shoulder and hip joints in a human body. The first link consists of two 130mm arms extending from the shaft to the first joint, including a 90mm diameter and 7mm thick Selective Laser Sintering (SLS) material attached to an aluminum part holder that accommodates the first stepper motor. The second link, representing the robot's body, is a 90mm SLS tube also connected to an aluminum holder that houses the second stepper motor. The final link, which represents the lower part of the system resembling the legs, includes two 20cm SLS material tubes which are economical and sturdy.

The Robogymnast, a multi-link rotational robot, is engineered to mimic human movement. The primary concept is to perform a swing and rotations on a high bar through reciprocal motion between the simulated shoulder and hip joints, much like an acrobat's maneuver. The operation of the Robogymnast system is powered by two stepper motors linked to a stepper driver, enabling smooth movements of the system. The control system programming is managed using an STM32F Microcontroller, employing C++ language to translate instructions from the PC-based control system to the robotic system. Each of the links has an assigned sensor; joint 1 is connected to a rotary encoder, while joints 2 and 3 are respectively connected to precision potentiometers 2 and 3 for higher accuracy. These sensors track the angles at each point. A block diagram of the Robogymnast operation system can be seen in Figure 3.2.

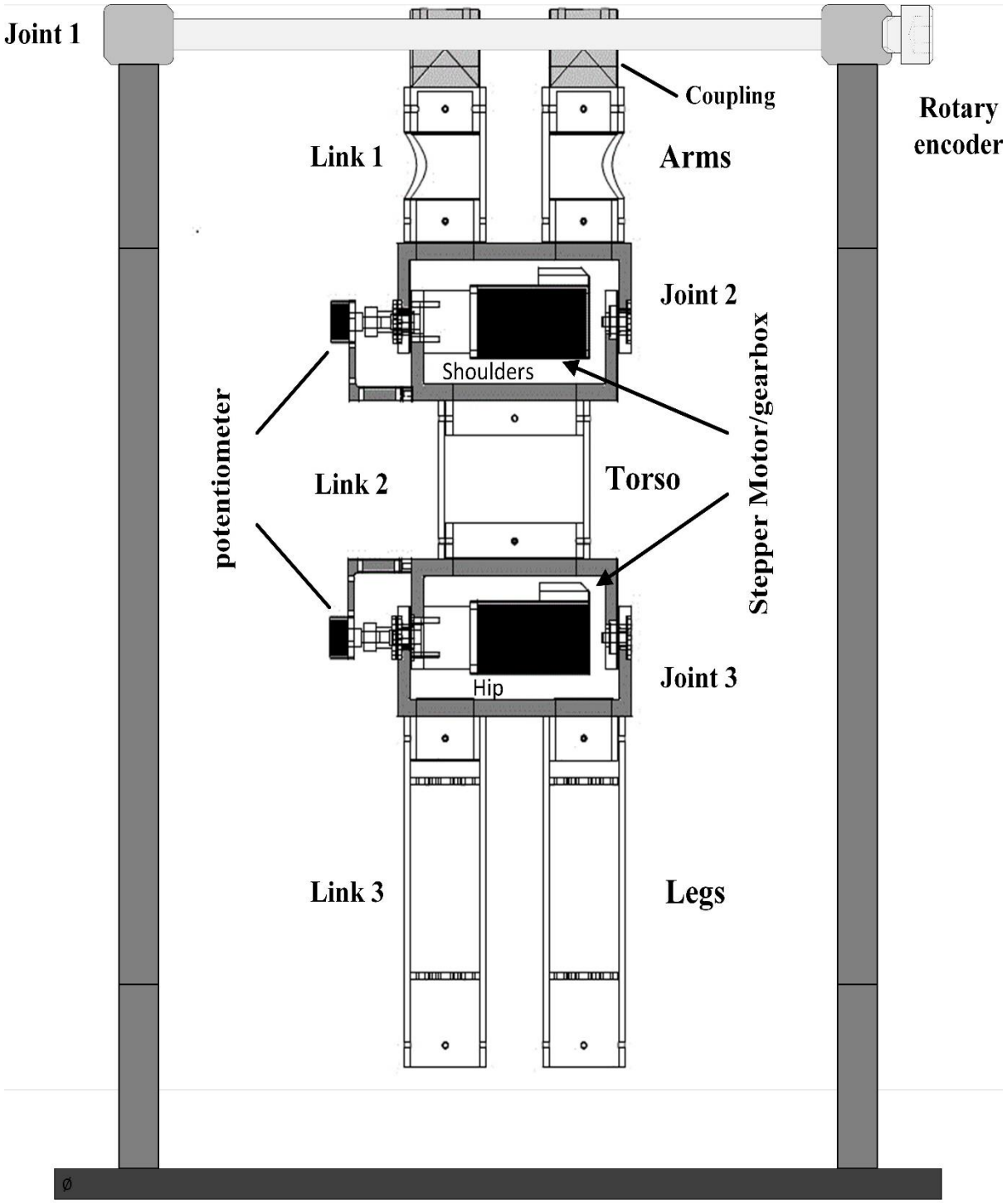


Figure 3.1 Robogymnast diagram

The Robogymnast control is facilitated by a PC outfitted with a suitable STM32F microcontroller. C++ programs are utilized to transfer input/output commands between the PC and the Robogymnast during the experiments. This C++ programming environment is used to transfer commands between the PC and the Robogymnast, which is powered by a 5V DC source. The program includes a code segment for recording experimental data and storing it on a hard disk. The computer controller software comprises a state feedback controller, a discrete integrator, and a motion code.

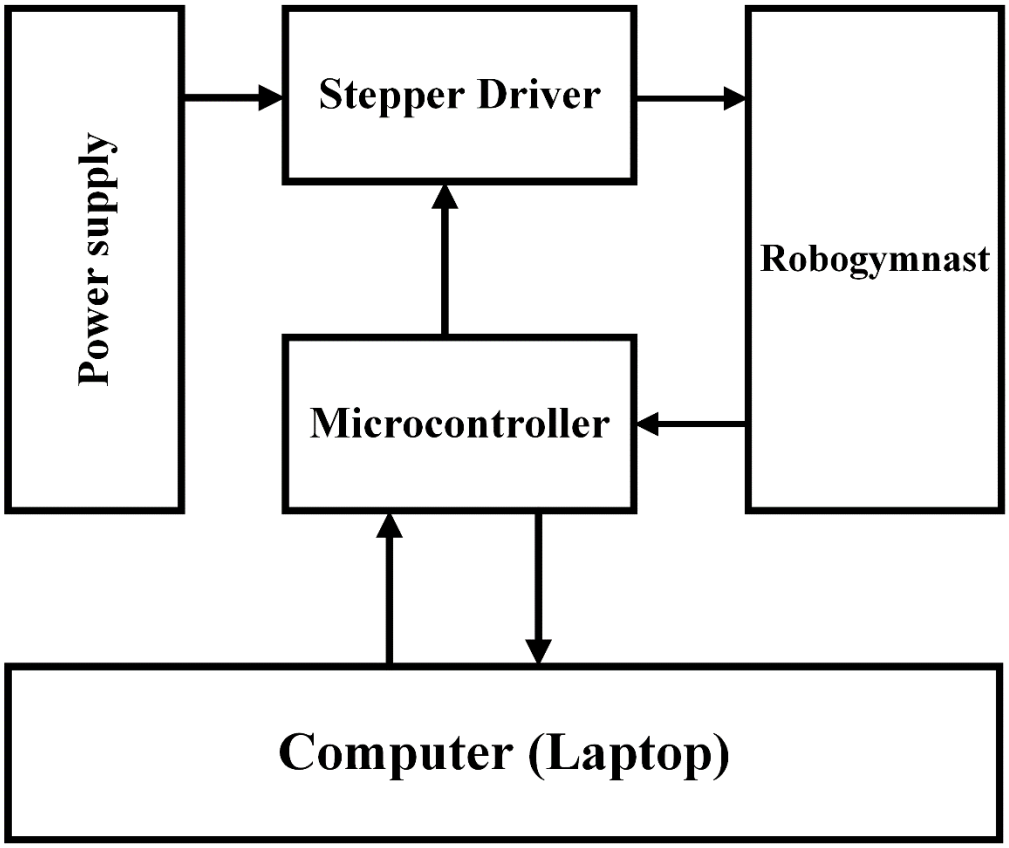


Figure 3.2 Robogymnast operation system

The primary component of each link is constructed from two rigid carbon SLS tubes that are 90 mm in diameter. These tubes are cost-effective, easy to cut, and have a lightweight mass of just 0.085 kg/m. Aluminium sections, 7 mm thick, are attached to both ends of each link to provide a structure for mounting sensors and actuators. Joint 1 features a steel shaft set

on ball bearings. An incremental rotary encoder is installed at one end of the shaft to measure the angle of Link 1. Further details about the encoder can be found in the system design section 3.4.1. Joints 2 and 3 are composed of two parts. The first part includes a combination of a stepper motor and gearbox, with its output shaft linked to the respective link. The second component consists of a potentiometer, which provides the relative angles between adjacent links. The potentiometer is connected to a short steel shaft, which is mounted on ball bearings on both sides. One of the critical aspects of robot design is the selection of actuators. The actuator's angular velocity and torque are vital performance characteristics of the robot, playing a significant role in the choice of actuators to sustain the required motion. The movements of the Robogymnast are mathematically modeled as detailed in the subsequent section.

## 3.3 Mathematical Model

To derive the Euler-Lagrange equations for a specific condition or scenario, one needs to construct the Lagrangian equations of the system. The Lagrangian represents the difference between the potential energy and kinetic energy of the system [81]. By considering the Lagrangian, it can obtain the equations of motion that describe how the system behaves under different conditions. The method of Lagrangian motion has been utilized in previous studies to analyse the dynamics of a mechanical manipulator with a rigid link made up of several components [82]–[84]. The dynamics of the system are represented through mathematical equations obtained using Lagrange's equation method. This methodology has been employed in various other research works to determine the equations of motion for similar systems [85]–[87].

This research focuses on the Robogymnast, which is a three-link system in the vertical plane with two actuators located at the shoulder and hip joints, but no actuator at the hand joint.

To establish the mathematical model for the Robogymnast, Lagrange's mathematical statements are employed, following the approach taken by Eldukhri and Pham in their 2010 study [78]. The design of the Robogymnast was made with the flexibility to accommodate future changes or modifications. These potential modifications may involve upgrading the actuators for stronger ones, adding extra link(s) to increase the degrees of freedom, adjusting the length of the shaft or free rotating bar, incorporating sensors to measure the angle position and angular velocity (such as a rotary encoder) of the second and third links, using angular position sensors (potentiometers) [15][88]. Figure 3.3 illustrates a Schematic representation of Robogymnast in the downward position.

This section presents a mathematical model for the three-link robotic system (Robogymnast) using the Lagrange equations of motion. The Lagrange equations provide a systematic approach to derive the equations of motion of a mechanical system by taking into account the total energy of the system. The model considers the three links of the robot, labeled as the Arm (1st Link), Body (2nd Link), and Leg (3rd Link), represented by L1, L2, and L3, respectively. The angles of rotation for each joint are denoted by $J_1$, $J_2$, and $J_3$.

$L_1$ = length of link 1 $m_1$ = mass of link 1 $J_1$ = joint 1

$L_2$ = length of link 2 $m_2$ = mass of link 2 $J_2$ = joint 2

$L_3$ = length of link 3 $m_3$ = mass of link 3 $J_3$ = joint 3

The measurement of joint angles is crucial in the control and manipulation of robotic systems. In the context of the three-link robotic system of the Robogymnast, the joint angles are represented by $\theta_1$, $\theta_2$, and $\theta_3$, and are related to the corresponding angular orientations $q_1$, $q_2$, and $q_3$. The joint angles describe the orientation of each link with respect to the preceding link

and the base of the system. The relationships between the joint angles and joint displacements are mathematically expressed in equations (3.1), (3.2), and (3.3), respectively:

$$\theta_1 = q_1 \tag{3.1}$$

$$\theta_2 = (q_1 + q_2) \tag{3.2}$$

$$\theta_3 = (q_1 + q_2 + q_3) \tag{3.3}$$

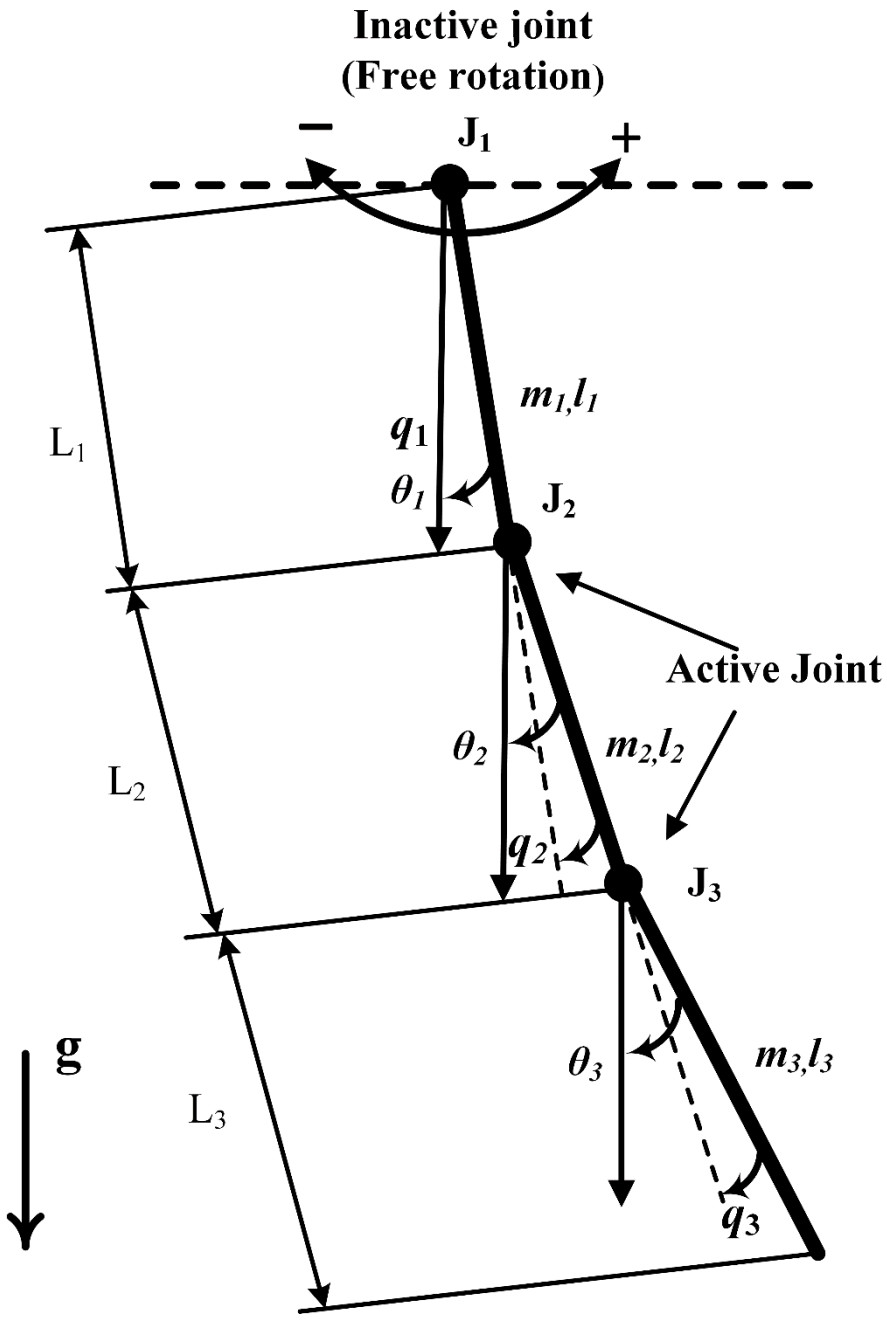


Figure 3.3 Schematic representation of Robogymnast

Equations (3.1), (3.2), and (3.3) calculate joint angles from joint displacements for a robot with three links. Joint angles are essential for control strategies in Robogymnast, derived from joint displacements measured by sensors such as encoders.

The angles of rotation for each joint of the three-link robotic system were formulated in the previous section. In this section, the corresponding angular velocities for each joint are derived, denoted by $\dot{\theta}_1$, $\dot{\theta}_2$, and $\dot{\theta}_3$. The angular velocity of the first joint ($\dot{\theta}_1$) is simply the time derivative of the angle of rotation ($q_1$), as shown in equation (3.4). Similarly, the angular velocity of the second joint ($\dot{\theta}_2$) is the sum of the time derivative of the angle of rotation of the first joint ($q_1$) and the time derivative of the angle of rotation of the second joint ($q_2$), as shown in equation (3.5). Finally, the angular velocity of the third joint ($\dot{\theta}_3$) is the sum of the time derivative of the angle of rotation of the first joint ($q_1$), the time derivative of the angle of rotation of the second joint ($q_2$), and the time derivative of the angle of rotation of the third joint ($q_3$), as shown in equation (3.6). The derived equations for the angular velocities of each joint will be used in the subsequent sections to derive the equations of motion for the system.

$$\dot{\theta}_1 = \dot{q}_1 \tag{3.4}$$

$$\dot{\theta}_2 = \dot{q}_1 + \dot{q}_2 \tag{3.5}$$

$$\dot{\theta}_3 = \dot{q}_1 + \dot{q}_2 + \dot{q}_3 \tag{3.6}$$

In Figure 3.4 axes analysis diagram of Robogymnast is displayed to help understand the steps of the Mathematical model.

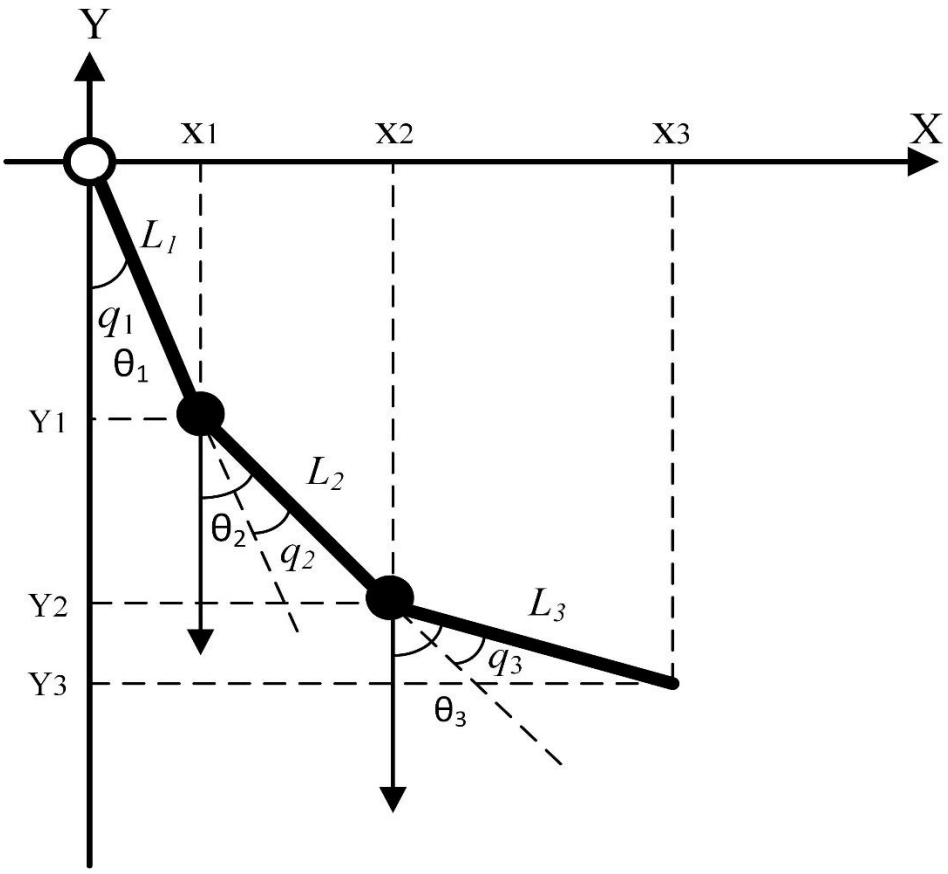


Figure 3.4 Axes analysis diagram of Robogymnast

In addition to considering the angular positions and velocities, it is important to also consider the position of each joint in space. The position of each joint can be expressed as a function of the joint angles and the lengths of the links connecting them. Specifically, the x and y coordinates of each joint can be calculated using trigonometric functions based on the length of the link and the angle of rotation of the joint.

For the three-link robotic system, the position of each joint can be calculated using equations (3.7)-(3.12). Equations (3.7)-(3.9) express the x-coordinates of the joints as a function of the joint angles and link lengths. Equation (3.7) describes the x-coordinate of the first joint, which is determined solely by the angle of the first joint ($q_1$) and the length of the first link ($L_1$). Equation (3.8) expresses the x-coordinate of the second joint as a function of the angles of the first and second joints ($q_1$ and $q_2$) and the lengths of the first and second links ($L_1$

and $L_2$). Finally, equation (3.9) gives the x-coordinate of the third joint in terms of the angles of all three joints ($q_1$, $q_2$, and $q_3$) and the lengths of all three links ($L_1$, $L_2$, and $L_3$).

Equations (3.7) - (3.9) utilize the S notation, where S represents the sine function, to simplify the equations and reduce complexity. Similarly, C is used to denote the cosine function in the position equations. The numbers 1, 2, and 3 following "sine" and "cosine" refer to $\theta_1$, $\theta_2$, and $\theta_3$, respectively.

$$x_1 = L_1 \sin\theta_1 = L_1 \sin q_1 = L_1 S_1 \tag{3.7}$$

$$x_2 = L_1 \sin q_1 + L_2 \sin(q_1 + q_2) = L_1 S_1 + L_2 S_{1+2} \tag{3.8}$$

$$x_3 = L_1 \sin q_1 + L_2 \sin(q_1 + q_2) + L_3 \sin(q_1 + q_2 + q_3) = L_1 S_1 + L_2 S_{1+2} + L_3 S_{1+2+3} \tag{3.9}$$

Equations (3.10)–(3.12) express the y-coordinates of the joints as a function of the joint angles and link lengths.

$$y_1 = -L_1 \cos\theta_1 = L_1 \cos q_1 = -L_1 C_1 \tag{3.10}$$

$$y_2 = -L_1 C_1 - L_2 C_{1+2} \tag{3.11}$$

$$y_3 = -(L_1 C_1 + L_2 C_{1+2} + L_3 C_{1+2+3}) \tag{3.12}$$

Equations (3.10)–(3.12) utilize the C notation, where C represents the cosine function. These equations can be used to determine the position of the end effector (i.e., the tool or hand of the robot) in Cartesian coordinates, which is essential for controlling the robot's movements.

Equations (3.13) - (3.15) represent the velocity in the x-direction.

**Velocities**:

$$\dot{x}_1 = L_1 \dot{\theta}_1 \cos\theta_1 = L_1 \dot{q}_1 \cos q_1 = L_1 \dot{q}_1 C_1 \tag{3.13}$$

$$\dot{x}_2 = L_1\,\dot{\theta}_1\cos\theta_1 + L_2\,\dot{\theta}_2\cos\theta_2 = L_1\,\dot{q}_1\cos q_1 + L_2\,(\dot{q}_1+\dot{q}_2)\cos(q_{1+}q_2) \quad (3.14)$$
$$= L_1\,\dot{q}_1 C_1 + L_2\,(\dot{q}_1+\dot{q}_2)\,C_{1+2}$$
$$\dot{x}_3 = L_1\,\dot{\theta}_1\cos\theta_1 + L_2\,\dot{\theta}_2\cos\theta_2 + L_3\,\dot{\theta}_3\cos\theta_3 = L_1\,\dot{q}_1\cos q_1 + L_2\,(\dot{q}_1+\dot{q}_2)$$
$$\cos(q_{1+}q_2) + L_3\,(\dot{q}_1+\dot{q}_2+\dot{q}_3)\cos(q_{1+}q_{2+}q_3)$$
$$= L_1\,\dot{q}_1 C_1 + L_2\,(\dot{q}_1+\dot{q}_2)\,C_{1+2} + L_3\,(\dot{q}_1+\dot{q}_2+\dot{q}_3)\,C_{1+2+3} \quad (3.15)$$

$$\dot{y}_1 = L_1\,\dot{\theta}_1\sin\theta_1 = L_1\,\dot{q}_1\sin q_1 = L_1\dot{q}_1 s_1 \quad (3.16)$$

$$\dot{y}_2 = L_1\,\dot{\theta}_1\sin\theta_1 + L_2\,\dot{\theta}_2\sin\theta_2 = L_1\,\dot{q}_1\sin q_1 + L_2\,(\dot{q}_1+\dot{q}_2)\sin(q_1+q_1) \quad (3.17)$$
$$= L_1\,\dot{q}_1 s_1 + L_2\,(\dot{q}_1+\dot{q}_2)\,s_{1+2}$$
$$\dot{y}_3 = L_1\,\dot{\theta}_1\sin\theta_1 + L_2\,\dot{\theta}_2\sin\theta_2 + L_3\,\dot{\theta}_3\sin\theta_3 = L_1\,\dot{q}_1\sin q_1 + L_2\,(\dot{q}_1+\dot{q}_2)$$
$$\sin(q_1+q_1) + L_3\,(\dot{q}_1+\dot{q}_2+\dot{q}_3)\sin(q_1+q_2+q_3) =$$
$$= L_1\,\dot{q}_1 s_1 + L_2\,(\dot{q}_1+\dot{q}_2)\,s_{1+2} + L_3\,(\dot{q}_1+\dot{q}_2+\dot{q}_3)\,s_{1+2+3} \quad (3.18)$$

Equations (3.16) to (3.18) represent the velocities in the y-direction of the end effector of the manipulator when one, two, and three links are moving, respectively. These equations are essential in understanding the dynamics of the system and can be used to design control strategies for the manipulator.

### 3.3.1 Compute the Lagrange of the acrobot system:

Equation (3.19) represents the total kinetic energy of the three-link robotic system, which is the sum of the kinetic energies of the individual links. The kinetic energy of each link is proportional to the square of its velocity $v_i$, which can be expressed in terms of the joint velocities $\dot{\theta}_i$. The first term in Equation (3.19) represents the kinetic energy of link 1, while the second and third terms represent the kinetic energies of links 2 and 3, respectively.

The total Kinetic energy (K, E) of the acrobot system:

$$T = \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 + \frac{1}{2}m_3 v_3^2 \quad (3.19)$$

The coefficients $m_i$ represents the masses of the links, and $v_i$ represents the velocity of link $_i$.

In the Lagrangian formulation, the kinetic energy T is expressed as a function of the velocities $v_i$ of the system. In the case of the three-link robotic system, the velocities can be expressed in terms of the joint velocities $\dot{\theta}_i$. Using the equations for the positions $x_i$ and $y_i$ of the links, expressions for the velocities vi can be derived.

For the first link, the velocity $v_1$ is given by:

$$\begin{aligned}\|v_1\|^2 &= \dot{x}_1^2 + \dot{y}_1^2 = (L_1\dot{\theta}_1 \cos\theta_1)^2 + (L_1\dot{\theta}_1 \sin\theta_1)^2 \\ &= L_1^2\,\dot{\theta}_1^2 = L_1^2\,\dot{q}_1^2\end{aligned} \tag{3.20}$$

where $\dot{q}_1$ is the joint velocity of the first link. This expression shows that the velocity of the first link is dependent only on the joint velocity of the first link and the length of the link.

By using similar calculations, the velocities of the second and third links can be expressed in terms of the joint velocities as well. The expression for the velocity $v_2$ of the second link is given by:

$$\begin{aligned}\|v_2\|^2 &= \dot{x}_2^2 + \dot{y}_2^2 = (L_1\dot{\theta}_1 \cos\theta_1 + L_2\dot{\theta}_2 \cos\theta_2)^2 + (L_1\dot{\theta}_1 \sin\theta_1 + L_2\dot{\theta}_2 \sin\theta_2)^2 \\ &= L_1^2\,\dot{q}_1^2 + L_2^2\,(\dot{q}_1 + \dot{q}_2)^2 + 2\,L_1 L_2\,\dot{q}_1\,(\dot{q}_1 + \dot{q}_2)\,C_1\end{aligned} \tag{3.21}$$

where $\dot{q}_1$ and $\dot{q}_2$ are the joint velocities of the first and second links, respectively. Similarly, the expression for the velocity $v_3$ of the third link is given by:

$$\begin{aligned}\|v_3\|^2 &= L_1^2\,\dot{\theta}_1^2 + L_2^2\,\dot{\theta}_2^2 + L_3^2\,\dot{\theta}_3^2 + 2\,[(L_1 L_2\,\dot{\theta}_1\dot{\theta}_2 \cos(\theta_1 - \theta_2) + L_1 L_3\,\dot{\theta}_1\dot{\theta}_3 \\ &\quad \cos(\theta_1 - \theta_3) + L_2 L_3\,\dot{\theta}_2\dot{\theta}_3 \cos(\theta_2 - \theta_3)] \\ &= L_1^2\,\dot{q}_1^2 + L_2^2\,(\dot{q}_1 + \dot{q}_2)^2 + L_3^2(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 + 2\,[(L_1 L_2\,\dot{q}_1(\dot{q}_1 + \dot{q}_2)\,C_1 \\ &\quad + L_1 L_3\,\dot{q}_1(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)\,C_{2+3} + L_2 L_3\,(\dot{q}_1 + \dot{q}_2)\,(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)\,C_3]\end{aligned} \tag{3.22}$$

- **The total kinetic energy of the acrobot system (K.E):**

The total kinetic energy of the acrobot system is a function of the joint velocities of the three-link robotic system. It is expressed in equation (3.20), where $v_i^2$ represents the squared velocity of the ith link. By substituting the expressions for $v_i^2$ in terms of the joint velocities $\dot{\theta}_i$, will obtain the final expression for the kinetic energy in terms of the joint velocities. This expression is useful for analyzing the motion of the acrobot system and understanding the

energy involved in its movement. The equation for the total kinetic energy of the acrobot system also can be expressed as (3.23), is based on a mathematical model that calculates the system's kinetic energy based on the velocities of the masses and their distances from the axis of rotation. The acrobot system consists of three masses attached to a flexible rod, and the equation considers the kinetic energies of each mass as well as the kinetic energy arising from their interactions.

This equation is derived from the Lagrangian equation of motion for the system, which describes the system's behaviour based on the difference between the potential and kinetic energies. By using this equation, it can gain insights into the movement and energy dynamics of the acrobot system.

$$\begin{aligned} T = {} & \frac{1}{2}(m_1 + m_2 + m_3) L_1^2 \dot{q}_1^2 + \frac{1}{2}(m_2 + m_3) L_2^2 (\dot{q}_1 + \dot{q}_2)^2 + \frac{1}{2} m_3 L_3^2 \\ & (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 + (m_2 + m_3) L_1 L_2 \dot{q}_1 (\dot{q}_1 + \dot{q}_2) C_2 + m_3 L_1 L_3 \\ & \dot{q}_1 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3) C_{2+3} + m_3 L_2 L_3 (\dot{q}_1 + \dot{q}_2)(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) C_3 \end{aligned} \tag{3.23}$$

However, in addition to kinetic energy, potential energy also plays a critical role in determining the system's behaviour. The total potential energy of the acrobot system depends on the gravitational potential energy of each mass and the elastic potential energy of the flexible rod connecting them. In the following subsection, the total potential energy expression will be derived and used in conjunction with the kinetic energy expression to derive the Lagrangian equation of motion for the system.

- **The total potential energy of the acrobot system (P.E):**

The total potential energy of the acrobot system is a function of the system's configuration and the gravitational potential energy associated with the position of each mass relative to the ground. The potential energy is stored in the system and can be converted into kinetic energy as the system moves. In the case of the acrobot system, the potential energy depends on the positions of the three masses and their height above the ground.

The total potential energy of Robogymnast system can be expressed as a sum of the potential energies of each mass, as shown in the following equation (3.24):

$$V = m_1 \, g \, y_1 + m_2 \, g \, y_2 + m_3 \, g \, y_3 \quad (3.24)$$

Here, $m_i$ represents the mass of the *i*th mass, g is the acceleration due to gravity, and $y_i$ is the height of the *i*th mass above the ground.

Alternatively, the potential energy can be expressed in terms of the joint angles $\theta_i$ using the law of cosines, which relates the distances between the masses to the joint angles. The resulting expression is:

$$V = -(m_1 + m_2 + m_3) L_1 \, g \, C_1 - (m_2 + m_3) L_2 \, g \, C_{1+2} - m_3 L_3 \, g \, C_{1+2+3} \quad (3.25)$$

Here, $L_i$ represents the length of the *i*th segment of the link, and $\theta_i$ is the angle between the *i*th and $(i + 1)$th segments.

The potential energy term in the Lagrangian equation of motion for the system describes how the system behaves based on the difference between potential and kinetic energies. The potential energy term in the equation arises from the interaction between the masses and the gravitational field. It determines how the system responds to external forces and the equilibrium positions of the masses. In the next section, the discussion will focus on how the Lagrangian equation of motion can be utilized to model the dynamics of the acrobot system and predict its behavior.

- **Lagrangian equation of motion:**

The Lagrangian, denoted by L, is a function that describes the system's behaviour based on the difference between its total kinetic energy and potential energy. For the acrobot system, the Lagrangian is given by the equation (3.26):

$$L = T - V \quad (3.26)$$

In this case, The Lagrangian is given by the expression in equation (3.27):

$$\mathrm{L} = \frac{1}{2}(m_1 + m_2 + m_3)\, L_1^2\, \dot{q}_1^2 + \frac{1}{2}(m_2 + m_3)\, L_2^2\, (\dot{q}_1 + \dot{q}_2)^2 + \frac{1}{2} m_3\, L_3^2\, (\dot{q}_1 + \dot{q}_2 + \dot{q}_3)^2 + (m_2 + m_3)\, L_1 L_2\, \dot{q}_1(\dot{q}_1 + \dot{q}_2)\, C_2 + m_3 L_1\, L_3\, \dot{q}_1(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)\, C_{2+3} + m_3 L_2\, L_3\, \dot{q}_1(\dot{q}_1 + \dot{q}_2)(\dot{q}_1 + \dot{q}_2 + \dot{q}_3)\, C_3 + (m_1 + m_2 + m_3)\, L_1\, \mathrm{g}\, C_1 + (m_2 + m_3)\, L_2 \mathrm{g}\, C_{1+2} + m_3 L_3\, \mathrm{g}\, C_{1+2+3} \quad (3.27)$$

The Lagrangian equation of motion is used to describe the dynamics of the acrobot system. It is given by $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0$, where $i$ = 1, 2, 3. This equation describes how the joint angles and velocities of the system change over time in response to the forces and torques acting on the masses. By solving this equation, the motion of the acrobot system and its stability can be predicted.

The Euler-Lagrange equation is a mathematical tool used to derive the equations of motion for a system described by a Lagrangian. In the context of the Robogymnast system, the equations of motion for the three can be used. The Euler-Lagrange equation is given by:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = 0 \qquad i = 0 \qquad \mathrm{i} = 1, 2, 3 \quad (3.28)$$

where $\tau_i$ represents the generalized force acting on the ith link, $q_i$ represents the generalized coordinate of the *i*th link, and L represents the Lagrangian of the system. The torque applied to joints is set to zero, as shown in equation (3.28). This means that joints are not being acted on by any external force or torque.

To obtain the equation of motion for a specific link, say the first link, the Euler-Lagrange equation is used:

$$\tau_1 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1} = 0 \quad (3.29)$$

Where $\tau_1$ is the generalized force acting on the first link, $q_1$ is the time derivative of the generalized coordinate $q_1$, and $\frac{\partial L}{\partial q_1}$ and $\frac{\partial L}{\partial \dot{q}_1}$ are partial derivatives of the Lagrangian L with respect to $q_1$ and $\dot{q}_1$, respectively. Solving the Euler-Lagrange equation for each of the three links gives us the equations of motion for the acrobot system, which describe how the links

move over time under the influence of the forces acting on them. These equations are essential for studying the dynamics of the system and understanding its behavior.

Start by calculating the partial derivative of the Lagrangian with respect to the joint velocity, which gives us the expression:

$$\frac{\partial L}{\partial \dot{q}_1} = ( m_1 + m_2 + m_3 ) L_1^2 \dot{q}_1 + ( m_2 + m_3 ) L_2^2 (\dot{q}_1 + \dot{q}_2) + m_3 L_3^2 (\dot{q}_1 + \dot{q}_2 + \dot{q}_3) + (m_2 + m_3) L_1 L_2 (2\dot{q}_1 + \dot{q}_2) C_2 + m_3 L_1 L_3 (2\dot{q}_1 + \dot{q}_2 + \dot{q}_3) C_{2+3} + m_3 L_2 L_3 (2\dot{q}_1 + 2\dot{q}_2 + \dot{q}_3) C_3 \quad (3.30)$$

Next, will take the time derivative of this expression, which gives us the rate of change of the partial derivative of the Lagrangian with respect to the joint velocity:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) = ( m_1 + m_2 + m_3 ) L_1^2 \ddot{q}_1 + (m_2 + m_3 ) L_2^2 (\ddot{q}_1 + \ddot{q}_2) + m_3 L_3^2 (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) + (m_2 + m_3) L_1 L_2 (2\ddot{q}_1 + \ddot{q}_2) C_2 - (m_2 + m_3) L_1 L_2 (2\dot{q}_1 + \dot{q}_2)\dot{q}_2 S_2 + m_3 L_1 L_3 (2\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) C_{2+3} - m_3 L_1 L_3 (2\dot{q}_1 + \dot{q}_2 + \dot{q}_3)(\dot{q}_2 + \dot{q}_3) S_{2+3} + m_3 L_2 L_3 (2\ddot{q}_1 + 2\ddot{q}_2 + \ddot{q}_3) C_3 - m_3 L_2 L_3 (2\dot{q}_1 + \dot{q}_2 + \dot{q}_3)\dot{q}_3 S_3 \quad (3.31)$$

To represent the dynamics of the acrobot system and describe the acceleration of the joints in terms of the system's mass, geometry, and applied forces, the partial derivative of the Lagrangian with respect to the joint position was subtracted, which resulted in the equation.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_1}\right) - \frac{\partial L}{\partial q_1} = ( m_1 + m_2 + m_3 ) L_1^2 \ddot{q}_1 + (m_2 + m_3) L_2^2 (\ddot{q}_1 + \ddot{q}_2) + m_3 L_3^2 (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) + (m_2 + m_3 ) L_1 L_2 (2\ddot{q}_1 + \ddot{q}_2) C_2 - (m_2 + m_3 ) L_1 L_2 (2\dot{q}_1 + \dot{q}_2) \dot{q}_2 S_2 + m_3 L_1 L_3 (2\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) C_{2+3} - m_3 L_1 L_3 (2\dot{q}_1 + \dot{q}_2 + \dot{q}_3)(\dot{q}_2 + \dot{q}_3) S_{2+3} + m_3 L_2 L_3 (2\ddot{q}_1 + 2\ddot{q}_2 + \ddot{q}_3) C_3 - m_3 L_2 L_3 (2\dot{q}_1 + \dot{q}_2 + \dot{q}_3)\dot{q}_3 S_3 + ( m_1 + m_2 + m_3 ) L_1 g S_1 + (m_2 + m_3) L_2 g S_{1+2} + m_3 L_3 g S_{1+2+3} \quad (3.32)$$

Similarly, in order to simplify the system's dynamics and focus on the behavior of the remaining joints (2nd and 3rd), set the torque applied to the second and third joints to zero, as illustrated in the equations. This means that no external force or torque is acting on the second and third joints. After doing this, repeat the same process with the other joints to derive the final equations.

$$\tau_2 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_2}\right) - \frac{\partial L}{\partial q_2} = 0 \quad (3.33)$$

$$\tau_3 = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_3}\right) - \frac{\partial L}{\partial q_3} = 0 \quad (3.34)$$

In conclusion, the equations for the total kinetic energy and potential energy of a three-link acrobot system have been derived. Also derived the Lagrangian equation of motion, which allows us to study the dynamics of the system and understand how it behaves under the influence of forces and torques. The Euler-Lagrange equation was then used to obtain the equations of motion for each of the three links, which describe how the links move over time under the influence of the forces acting on them. Also, the expressions that have been derived are useful for analyzing the motion of the acrobot system and understanding the energy involved in its movement. These equations can be used to predict the motion of the system and its stability, which is important for developing control strategies for the system.

The numerical model of the Robogymnast is calculated by substituting the values of the parameters given in Table 3.1.

Table 3.1 Robogymnast Parameters

| **Parameters** | **Symbol** | **Mean Values** |
|---|---|---|
| Length of the first link | $L_1$ | 0.16 m |
| Length of the second link | $L_2$ | 0.18 m |
| Length of the third link | $L_3$ | 0.24 m |
| Weight of the first link | $m_1$ | 1.2 kg |
| Weight of the second link | $m_2$ | 1.2 kg |
| Weight of the third link | $m_3$ | 0.5 kg |
| Angles between pole 1,2,3 | $\theta$ | $\theta_1$, $\theta_2$, $\theta_3$ (rad) |
| Initial value of the angles | $q_1$, $q_2$, $q_3$ | 0 (rad) |
| Gravity | g | $9.81 m/s^2$ |

The Robogymnast system in its vertical position was modelled in a linear, continuous-time, state-space manner via MATLAB and its various toolbox, along with further M-files. For this reason, the system has to be linearized as in the following subsection.

### 3.3.2 Linearised model of the system

The Robogymnast system is nonlinear in the state-space form.

$$\dot{x} = \mathrm{f}(\mathrm{x}, \mathrm{u}) \tag{3.35}$$

The state vector of the acrobot, denoted as $\dot{x}$, is defined as $x^T = [q_1 \quad q_2 \quad q_3 \quad \dot{q_1} \quad \dot{q_2} \quad \dot{q_3}]$. The control input, represented by $u$ is equal to $\tau$ ($\mathrm{u} = \tau$) which is the applied input vector. it is considered a scalar because there is only one actuator that provides torque input to the system. The expression for the angular acceleration, $\ddot{q} = [\ddot{q}_1 \ ; \ddot{q}_2 \ ; \ddot{q}_3]^T$ , was obtained by solving for $\ddot{\theta}$ in equation (3.36).

$$\ddot{q} = \mathrm{M}^{-1}[\tau - \mathrm{C}(q, \dot{q}) - \mathrm{G}(q)] \tag{3.36}$$

Equation (3.36) is of the form $\dot{\mathrm{x}} = \mathrm{f}(\mathrm{x}, \mathrm{u})$. The term $\dot{\mathrm{x}}$ is a $2 \times 1$ matrix, which contains nonlinear elements. In these forms the first two elements of $\dot{\mathrm{x}}$ are just the last two elements of x, also, G is the gravity vector. M represents the mass matrix. In the context of mechanical systems, it represents a matrix that describes the distribution of masses in the system and their influence on the system's dynamics. "$\mathrm{C}(\mathrm{q}, \dot{\mathrm{q}})$" represents the vector of Coriolis and centrifugal forces. These forces arise in rotating and moving systems and are dependent on both the configuration variables (q) and their rates of change ($\dot{\mathrm{q}}$). The equation you provided is a representation of the system's dynamics in a linearised form, often used in control theory and modelling. It relates the joint accelerations ($\ddot{\mathrm{q}}$) to the applied torques ($\tau$) and the effects of mass distribution (M), centrifugal forces (C), and gravitational forces (G) on the system. To linearize the last two elements about an operating point vector (OP), Taylor's expansion is used:

$$\delta\dot{\mathrm{x}}(\mathrm{t}) = \left(\frac{\partial \mathrm{f}(\mathrm{x}, \mathrm{u})}{\partial \mathrm{x}}\right)_{\mathrm{x=OP,u=0}} \delta\mathrm{x}(\mathrm{t}) + \left(\frac{\partial \mathrm{f}(\mathrm{x}, \mathrm{u})}{\partial \mathrm{u}}\right)_{\mathrm{x=OP,u=0}} \delta\mathrm{u}(\mathrm{t}) \tag{3.37}$$

$\dot{\mathrm{x}}(\mathrm{t})$ represents a small deviation of the states from the operating point. The coefficients of $\delta\mathrm{x}(\mathrm{t})$ and $\delta\mathrm{u}(\mathrm{t})$ are denoted as A and B, respectively, and are evaluated at the operating point.

Therefore, the system's linear state-space model is represented by the following equation system:

$$\dot{x} = Ax + Bu \tag{3.38}$$

$$y = Cx + Du \tag{3.39}$$

$\dot{x}$ is the state vector of the system, *y* is the output vector.

Table 3.1 provides the nomenclature and values for the parameters. In this context, A, B, C, and D represent matrices used for state-space modelling. The controllers are designed to stabilize pendulum links when they are vertically aligned downwards, while also minimizing vibrations. The stable equilibrium point corresponding to the states of the links is defined by $\theta_1 = \theta_2 = \theta_3 = 0$. The state-space representation of the Robogymnast equations can be written as follows:

$$A = \begin{bmatrix} 0_3 & I_3 \\ A_{21} & A_{22} \end{bmatrix}$$

Where:

$$0_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$A_{21} = \begin{bmatrix} 0 & 2.6825 & -0.0657 \\ 0 & 29.2751 & -15.8236 \\ 0 & -57.5286 & 247.5924 \end{bmatrix}$$

$$A_{22} = \begin{bmatrix} -0.0286 & -0.0083 & 0.0284 \\ -0.0391 & -0.1957 & 1.2358 \\ 0.0589 & 1.4085 & -18.0527 \end{bmatrix}$$

The numerical model of the Robogymnast was computed using MATLAB/toolbox to determine A. Furthermore, the state-space matrices for the Robogymnast, represented by A, B, C, and D, are illustrated as follows.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2.6825 & -0.0657 & -0.0286 & -0.0083 & 0.0284 \\ 0 & 29.2751 & -15.8236 & -0.0391 & -0.1957 & 1.2358 \\ 0 & -57.5286 & 247.5924 & 0.0589 & 1.4085 & -18.0527 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.0314 \\ 1.6582 \\ -2.4837 \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad D = [0]$$

The discrete-time model of the Robogymnast is obtained by discretizing the equations (3.38) and (3.39). These equations are then implemented in the MATLAB command window to incorporate the mathematical model matrices into the simulation. This process is followed in order to obtain the results.

### 3.3.3 The ability to manage and monitor the state-space

In the field of control theory, two significant concepts are the controllability and observability of a system. These represent the capacity to regulate an external input and gauge the system's internal state, respectively. Controllability assesses the extent to which a given actuator setup can regulate the system's entire states, while observability evaluates the sensor setup's capability to provide all the data required to estimate the system's entire states. Traditional control theory provides tests for controllability and observability that rely on the rank deficiency of the corresponding matrices. These computations are typically carried out using MATLAB.

A system is deemed controllable if its controllability matrix is of full rank, and it's observable if its observability matrix is of full rank. However, this response might not suffice for practical engineering dilemmas, where more quantifiable data is often required [89].

```
Co = ctrb(sys.A,sys.B);
rank (Co)
ans = 6
'It is controllable'
```

Observability pertains to the capacity to discern the internal state of a system based on its output data. A system is considered observable when you can determine its internal states uniquely from its output data.

```
Ob = obsv(sys.A, sys.C);
rank(Ob)
ans = 6
'It is observable'
```

## 3.4 System Design

This section delves into the configuration of the Robogymnast system, detailing all the apparatus deployed in this study for conducting experimental trials. The test apparatus and data collection techniques are also outlined. Additionally, the method of data compilation is explicated. The section also introduces the multi-link motion signals of the Robogymnast that will be employed for result detection and subsequent optimization via chosen algorithms in forthcoming chapters.

### 3.4.1 Physical structure of Robogymnast

The physical structure of the robot mirrors a human gymnast, equipped with two arms, two legs, and a torso. This design allows the system to execute a broad array of gymnastic actions, including swings and manipulations. The robot's body is constructed using SLS material, an acronym for Selective Laser Sintering (SLS). This 3D printing technology employs a laser to bond tiny particles of material into a solid entity. SLS is frequently utilized to produce intricate, high-quality components with excellent precision and robustness. Furthermore, aluminium is used in the joint sections. The robot's body needs to sustain balance and modify its movements in real-time to ensure stability during gymnastic activities. This can be accomplished using sensors that identify the robot's positioning and alignment, along with sophisticated control algorithms that allow for necessary movement adjustments. Additionally, sensors like encoders and potentiometers are used to perceive the robot's environment and system position. These sensors offer feedback to the multi-link control system, thereby enabling it to make necessary adjustments to its movements.

In conclusion, the design of the Robogymnast body is a multifaceted and demanding endeavor, necessitating expertise in materials science, robotics, biomechanics, and control theory. However, by combining the correct technologies and design elements, it's feasible to create a robot capable of executing an extensive array of gymnastics movements with precision. Figure 3.5 displays the Robogymnast body itself, and the system's components are depicted and discussed in this chapter.

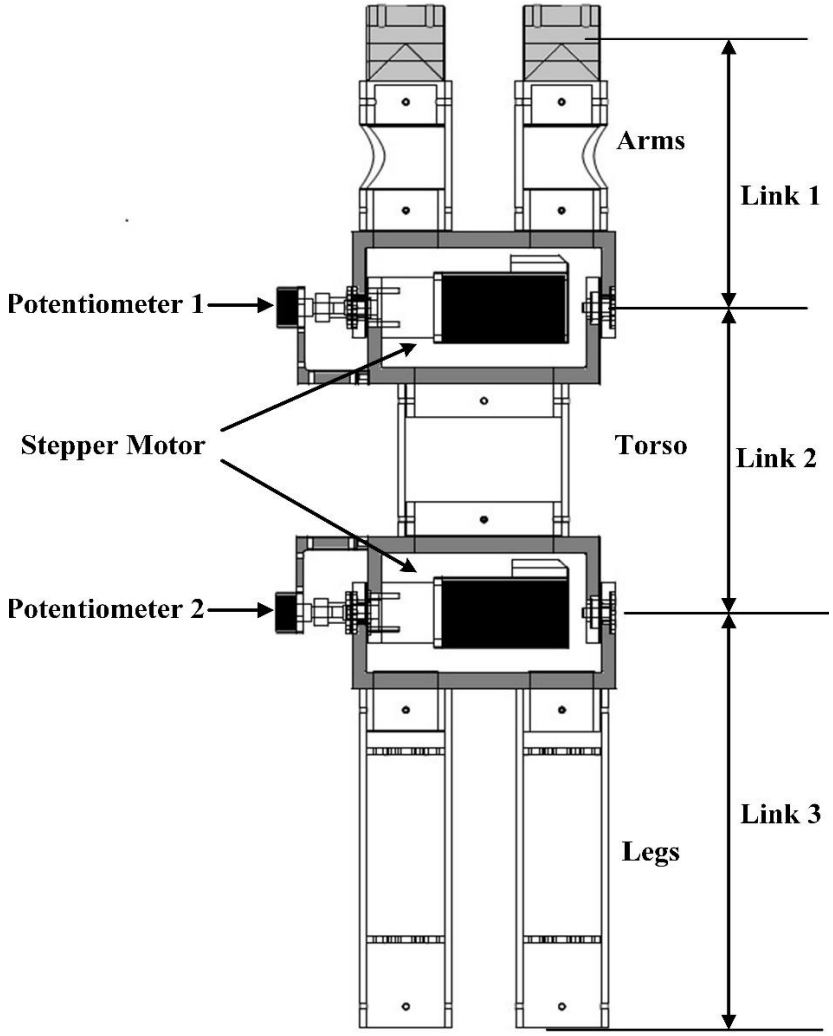


Figure 3.5 Initial Design of Robogymnast

The following subsections demonstrate the components used to build the entire acrobot system such as the actuators and sensors.

- **Stepper motor:**

Stepper motors operate by using a toothed rotor and a group of magnets that are drawn to numerous field windings, enabling the motor's position to be managed in distinct steps. The advantage of stepper motors is that they don't need closed-loop feedback, but they can skip steps under intense shock loads, a common occurrence in the joints of legged robots. While introducing closed-loop feedback could rectify this, stepper motors are often quite hefty and

exhibit low transparency [90]. As shown in Figure 3.6, the stepper motor is coupled with an encoder. Additionally, the stepper motor's monitoring test illustrated in Figure 3.7 can be utilized to confirm the system's preliminary motion.

Figure 3.6 Stepper Motor

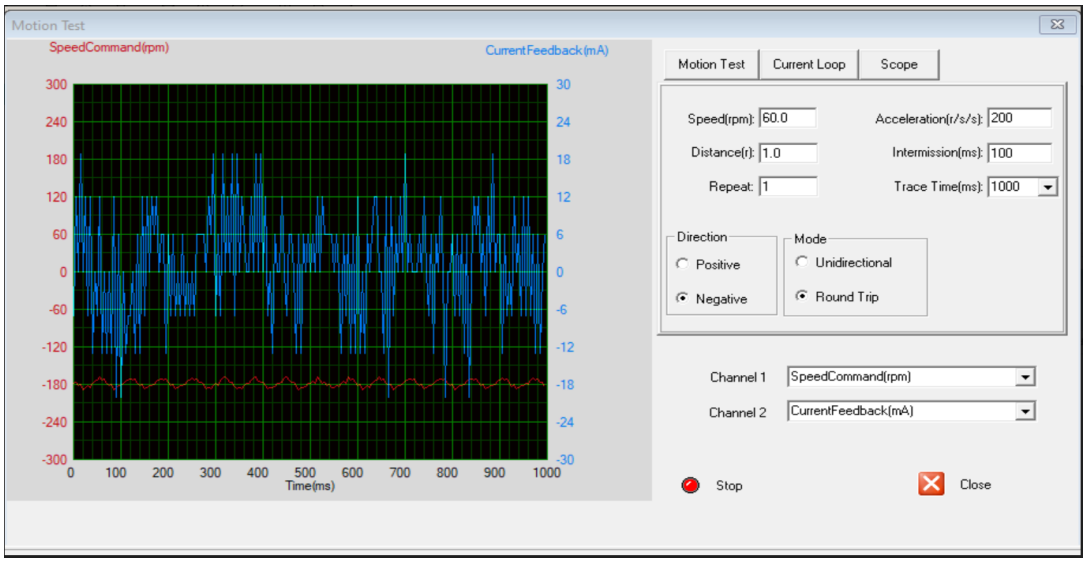


Figure 3.7 Monitoring Test for the Stepper Motor

### 3.4.2 Robogymnast Operation System

The Robogymnast system's operation encompasses a multitude of various elements working together to allow the robot to execute gymnastic motions accurately and efficiently. This subsection provides more insight into the integral components of a Robogymnast system, specifically, the operational system components that manoeuvre and regulate the Robogymnast. The principal elements of this system include the motor driver, which controls and transmits instructions to the stepper motor, and the microcontroller, which is used to program the entire system and connect all parts concurrently. The microcontroller and the motor drivers are displayed in Figure 3.8.

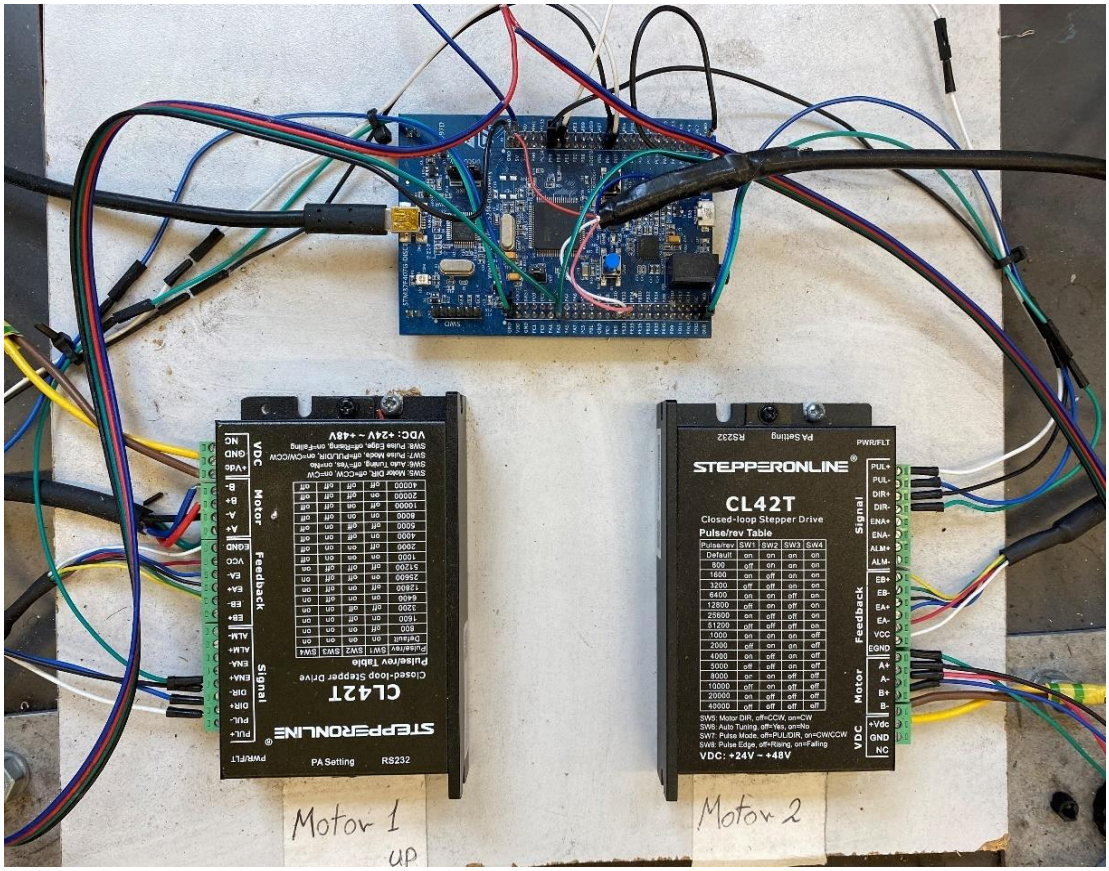


Figure 3.8 STM32 Stepper driver

- **Microcontroller (STM-32F):**

As illustrated in Figure 3.9, a microcontroller is a compact computer housed on a single integrated circuit, designed to govern specific devices or processes. It generally includes a central processing unit (CPU), memory, and various specialized hardware components that facilitate specific functions. The STM-32F microcontroller, a product of STMicroelectronics, is a microcontroller family. It's built on the ARM Cortex-M processor architecture and intended for use in embedded systems and applications demanding real-time control, low power usage, and high performance [91].

The STM32F427xx and STM32F429xx devices are built on the high-performance Arm® Cortex®-M4 32-bit RISC core, which operates at a frequency of up to 180 MHz. The STM32F4 devices feature high-speed embedded memories, including Flash memory of up to 2 Mbytes, up to 256 Kbytes of SRAM, up to 4 Kbytes of backup SRAM, and a broad array of enhanced I/O and peripherals linked to two APB buses.


Figure 3.9 STM-32 Microcontroller [91]

These characteristics make the STM32F427xx and STM32F429xx microcontrollers appropriate for a diverse array of applications:

- Motor drive and application control
- Home audio appliance
- Alarm systems, video intercom, and HVAC
- Medical equipment
- Printers, and scanners
- Industrial applications: PLC, inverters, circuit breakers

- **Personal Computer (PC):**

As Shown in Figure 3.10. Lenovo ThinkPad T430 PC Intel (R) Core (TM) i5-3230M CPU @ 2.60GHz (4 CPUs), 2.6GHz, 8192MB RAM is used to program the system, In addition to programming the system, the Dell PC is also used to write the STM32F programming code, which is the code that will be loaded onto the STM32F microcontroller. The STM32F microcontroller is a type of embedded system that is commonly used in a variety of applications, from simple sensor readings to complex control systems.


Figure 3.10 Lenovo ThinkPad T430 laptop

This laptop also delivers the 5V power required by the STM32F microcontroller. This power supply is crucial for the microcontroller's optimal functioning, as it needs a consistent and dependable power source to operate accurately. Overall, employing a robust and competent computer, such as the Lenovo ThinkPad PC with an Intel Core i5-3230M processor, is vital for the effective programming and running of the STM32F microcontroller system.

## 3.5 Overview of the entire System

This section illustrates the connection of system components as shown in Figure 3.11. Subsequently, the final representation of the multi-link robotics system (Robogymnast), designed and constructed at Cardiff University, is shown in Figure 3.12.

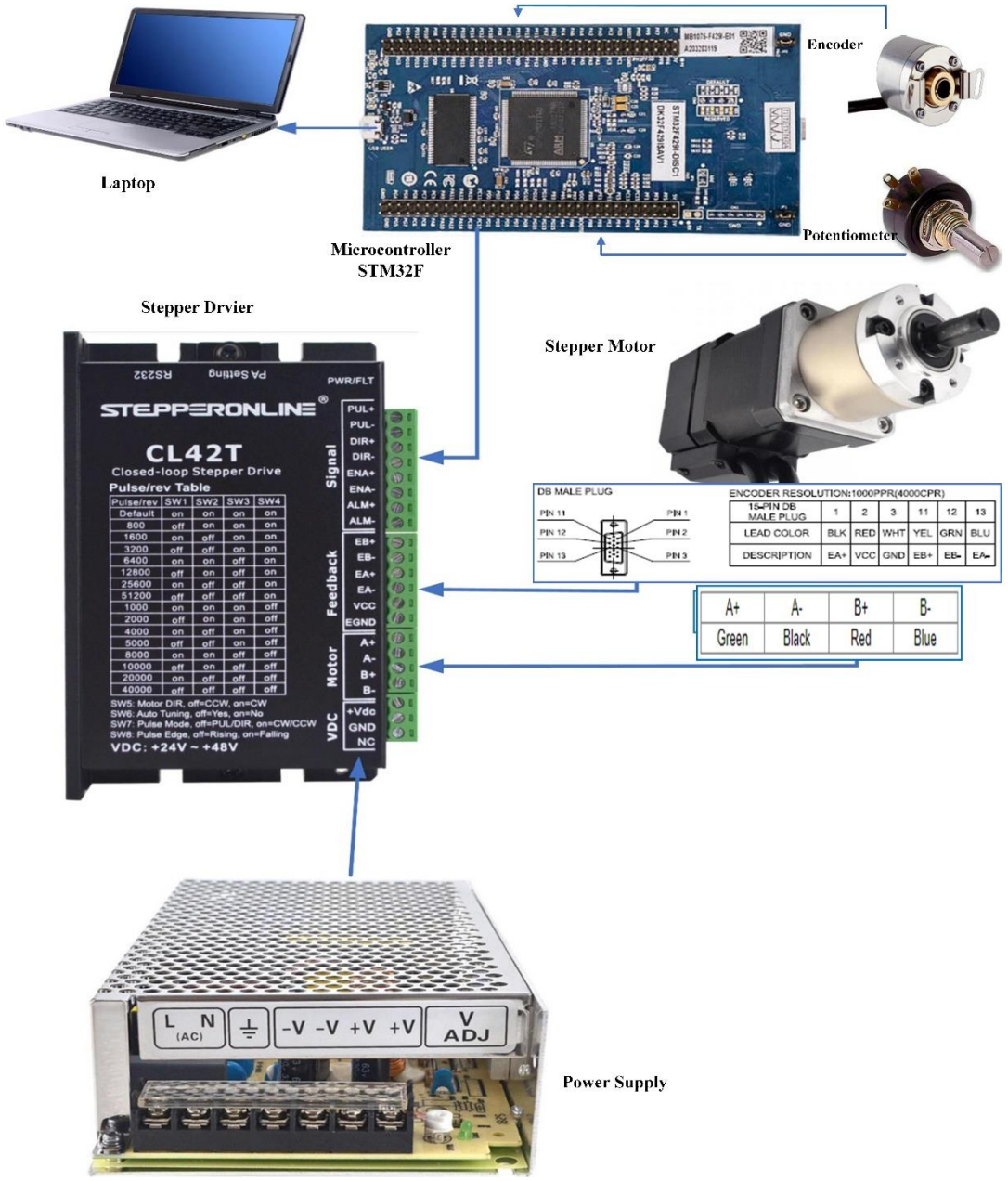


Figure 3.11 The connection of system components

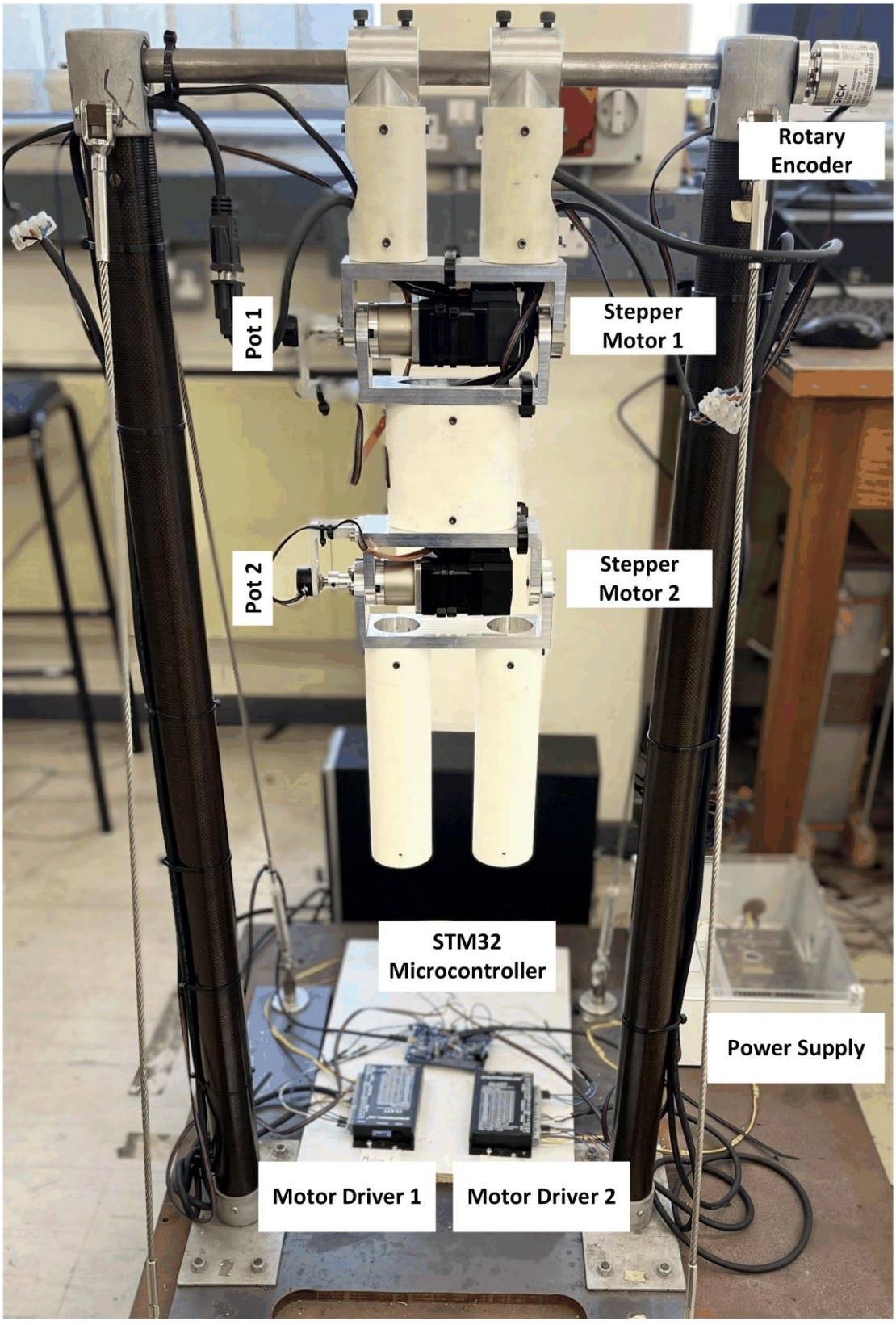


Figure 3.12 Robogymnast System

## 3.6 Summary

This chapter has offered a comprehensive description of the Robogymnast system and its mathematical modelling. The target of the work outlined in this chapter was to calculate and design a straightforward mathematical model for the triple-link robotic system (Robogymnast). A mathematical model of the Robogymnast was derived using the Euler-Lagrange approach, which describes the system's dynamics. The linearised equations of motion and their state-space representations were subsequently introduced. The system's structure and the composition of its components have been detailed. An in-depth overview of the entire Robogymnast system and its constituent parts was provided, along with a depiction and explanation of the system's process flow. Chapter 4 will involve the design and examination of swing controllers for the Robogymnast. It will present the results and discuss each controller, followed by a comparison of their respective outcomes.

# Chapter 4: Swinging Controller

## 4.1 Introduction

The control of swinging motion is a key element of the Robogymnast system, aiming to mimic the movement of human gymnasts. This controller utilizes algorithms, motors and sensors to govern the swinging motion of the robot's body, enabling it to perform intricate manoeuvres. The controller's main objective is to ensure that the system responds as required and meets specific performance criteria, such as stability, tracking, and disturbance rejection. A well-designed controller can improve the efficiency, safety, and reliability of various systems in a wide range of applications, including robotics, and manufacturing. Effective controller design requires a deep understanding of the system's behaviour, modelling techniques, and control theory [92]–[94].

This chapter addresses the swing problem of a non-linear, three-link robotic gymnast, also known as Robogymnast. Some control techniques were employed to investigate the swing process of this type of inverted pendulum mechanism. The swing problem has been studied in previous works, including those studies [31][95][96]. The three-link mechanism is an excellent experimental tool for studying control theories and testing control policies in control engineering. It is also an effective model for swing testing. However, the triple inverted pendulum system has strong non-linearity and instability. These characteristics make it challenging to model and control the system. Additionally, the multivariable nature of such systems adds further complexity. Due to the dynamic nature of the structural components of

this underactuated system, it is an ideal model for simulations, comparative evaluations, and optimization of various control approaches such as linear quadratic regulators (LQR) and proportional-integral-derivative (PID) controllers [97]–[99].

As part of the present study, problems related to controlling the movement of a manipulator robot featuring underactuation are addressed. Several different task types can be addressed by the multi-link robotic system, which has been extensively investigated within AI studies but represents significant scope for further investigation. In this study, the primary objective was to investigate swinging for a triple-link Robogymnast mechanism consisting of two active-power links and an overhead non-powered link.

This chapter will investigate the control methods applied within the study in detail. This begins with a discussion of the function of the LQR control system and how this was implemented for the parameters of the robotic system, with a similar discussion following for the PID control system. Modelling is done through MATLAB and is utilised for the simulation and implementation of this system, with the results then being provided and comparisons are made. Furthermore, the chapter assesses how robustly the developed controller performs, as well as its capacity to respond to unanticipated disturbance from outside the system and compares this to the other controller types applied [100].

## 4.2 LQR Controller

The linear quadratic regulators (LQR) controller has long been applied for high-performance, high-stability closed-loop systems, being used for effective control of feedback gain. The multivariate nature of LQR allows for simultaneous control of displacement angles across the 3-link inverted pendulum [3][101]. LQR was chosen based on its capacity to deal with significant disturbance events and keep systems stable with no reductions in operational performance [98][102]–[104].

In state feedback control (SFC), simplifications are made for equations for poles of the system, placed relative to K as the gain matrix, as well as state variables. Through this approach, the poles of closed-loop systems can be placed anywhere that is desirable. On the other hand, when it comes to feedback control of outputs, the process involves multiplying the feedback components using the state feedback gain matrix. Then, a comparison is made with reference values for inputs. State feedback control (SFC) is primarily used for calculating the gain matrix [105].

LQR controllers are frequently utilized for this aim, and optimally for these controllers, K matrix parameters would include cost function (J) to optimize states, x(t), and system control signal u(t) [106]. In which Q represents a constant symmetry positive matrix, with matrix R representing a matrix. Optimization of control is achieved through the application of the following equation to calculate P and K:

$$\mathrm{u(t)} = -\mathrm{K\,x(t)} \tag{4.1}$$

$$\mathrm{J} = \frac{1}{2}\int_0^{\infty} (\mathrm{x^t\,Qx} + \mathrm{u^t\,Ru})\,\mathrm{dt} \tag{4.2}$$

$$U = R^{-1}\,B^T\,PX = -KX \tag{4.3}$$

The algebraic Riccati equation is used to determine K and P values.

$$A^T + PA - PBR^{-1}\,B^T P + Q = 0 \tag{4.4}$$

$$K = R^{-1}\,B^T P = [K_1\,K_2\,K_3\,\ldots,K_6] \tag{4.5}$$

Using MATLAB, K value has been obtained.

K = [0.2581 22.789 -507.886 0.940 -12.250 -19.480]

In this part, LQR is executed using MATLAB/Simulink, as shown in Figure 4.1. Employing LQR with MATLAB/Simulink entails calculating the optimal gain matrix through MATLAB's LQR function, followed by utilizing the LQR block in Simulink to apply the gain to the system input and produce the controlled output. This method offers an effective and adaptable means to develop and simulate LQR controllers for an extensive array of control systems.

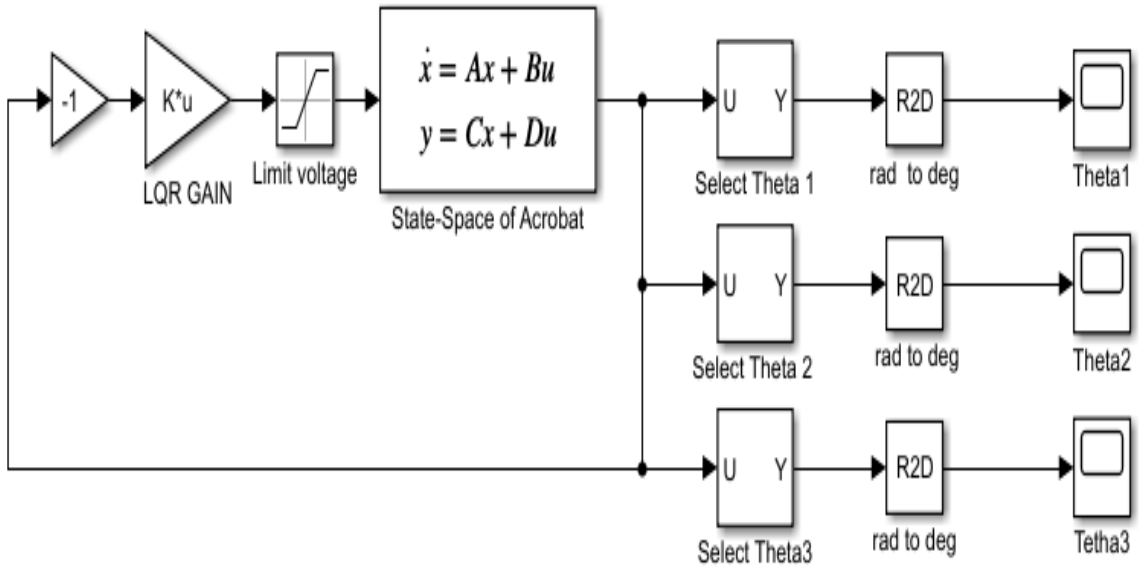


Figure 4.1 LQR controller simulation model

Table 4.1 LQR performance

| **Theta** | **Controller** | **Overshoot** $O_{sh}\ (pu)$ | **Undershoot** $U_{sh}\ (pu)$ | **Settling time** $T_s\ (s)$ | **Rising time** $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | LQR | 8.02 | -5.69 | 15.519 | 0.335 |
| $\theta_2$ | LQR | 1.03 | -1.32 | 4.914 | 0.075 |
| $\theta_3$ | LQR | 0.25 | -0.41 | 3.331 | 0.050 |

**LQR results:**

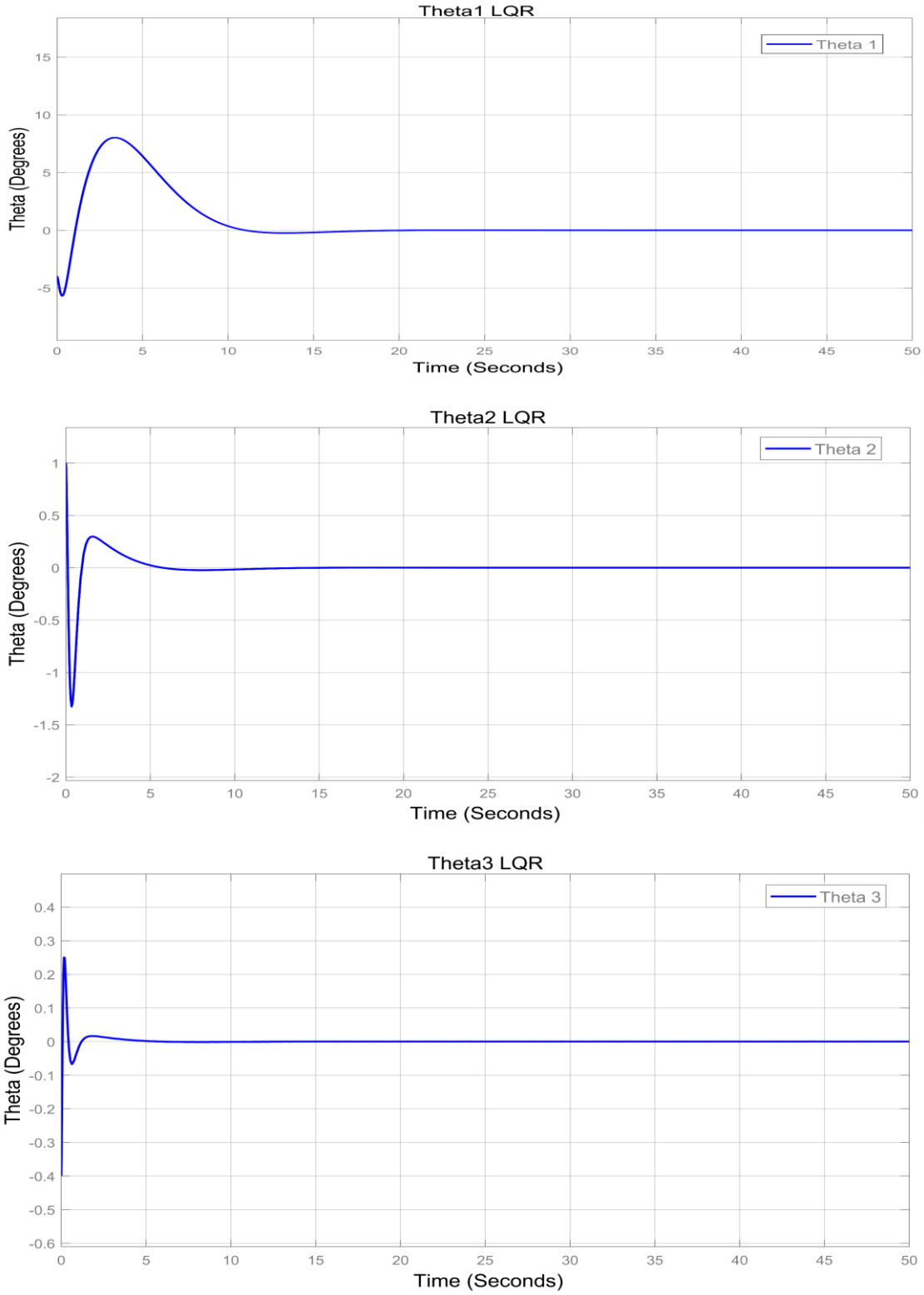


Figure 4.2 The step response of the system (LQR)

The provided Table 4.1 shows the performance of the system with three different joints, represented by Theta 1, Theta 2, and Theta 3. Each joint is controlled by an LQR controller, and the table shows the overshoot, undershoot, settling time, and rising time for each joint.

From the table, it is evident that the performance of the joints varies significantly. Theta 1 has the highest overshoot and settling time, indicating that it takes longer for the joint to stabilize and oscillates more before reaching the steady state. On the other hand, Theta 3 has the lowest overshoot, settling time, and rising time, indicating that it stabilizes faster and has fewer oscillations. It is also interesting to note that Theta 2 has the lowest overshoot and undershoot values, indicating that it has the fastest response among the three joints. Nonetheless, it demonstrates a slightly prolonged settling time in comparison to Theta 3, possibly attributed to the compromise between precision and a slower rate of convergence, Figure 4.3 represents LQR performance comparison.

Analysing how various joints in the system perform when controlled by the same controller underscores the significance of meticulously choosing and adjusting control parameters for each joint within a multi-link system.

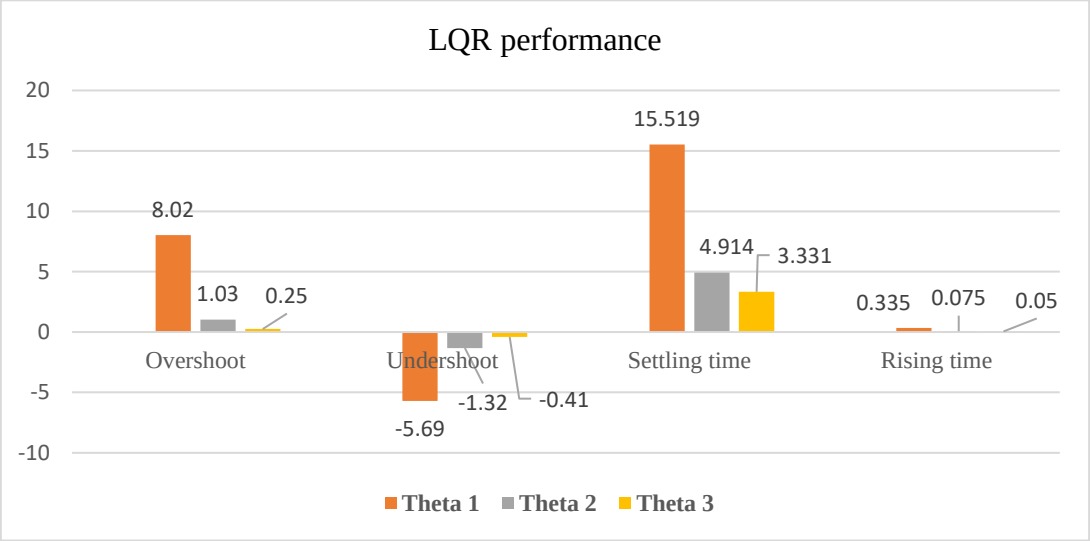


Figure 4.3 LQR performance comparison for Theta 1, 2 and 3

## 4.3 PID Controller

Employed-feedback proportional integral derivative (PID) controllers are frequently applied in industry and in different situations in which control must be continuously modulated [5]. In a PID controller, analysis and measurement of error occur based on the target set-point (SP) differential and process variable (PV), with real-time adaptations based on proportional (P), integral (I), and derivative (D). Practically, this leads to control functions being adjusted automatically with a high degree of accuracy and responsiveness. The PID algorithm of the controller increases system capacity, returning measured outputs to targeted inputs while minimizing deferral error [107]. PID controllers have the distinct feature that they utilize three control types, with proportional, integral, and derivative effects on their outputs, to optimize control and ensure it is as efficient as possible. Calculation and measurement of PID controller

outputs use proportional, integral, and derivative terms [15][99]. With the output u(t), the following describes the PID controller:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{4.6}$$

The simulation of movements similar to those in gymnastics was conducted, and the results of this process were compared with results using other controllers. The PID controller can be shown through the equation which follows:

$$C = K\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{4.7}$$

Wherein $K$ provides proportional gains and $T_i$ is integral time, with $T_d$ representing derivative time. If a controller relied solely on PID, this would result in infinitely continuing high-

frequency gains, which would be both detrimental and impossible. Therefore, low-pass filters must be used with PID controllers [108].

$$F = \frac{1}{(sT_f + 1)^2} \tag{4.8}$$

MATLAB/Simulink is used to implement the PID controller as shown in Figure 4.4. Utilizing PID with MATLAB/Simulink requires employing the PID function in MATLAB to determine the optimal gain matrix, followed by implementing the PID block in Simulink to apply the gain to the system input and produce the regulated output.

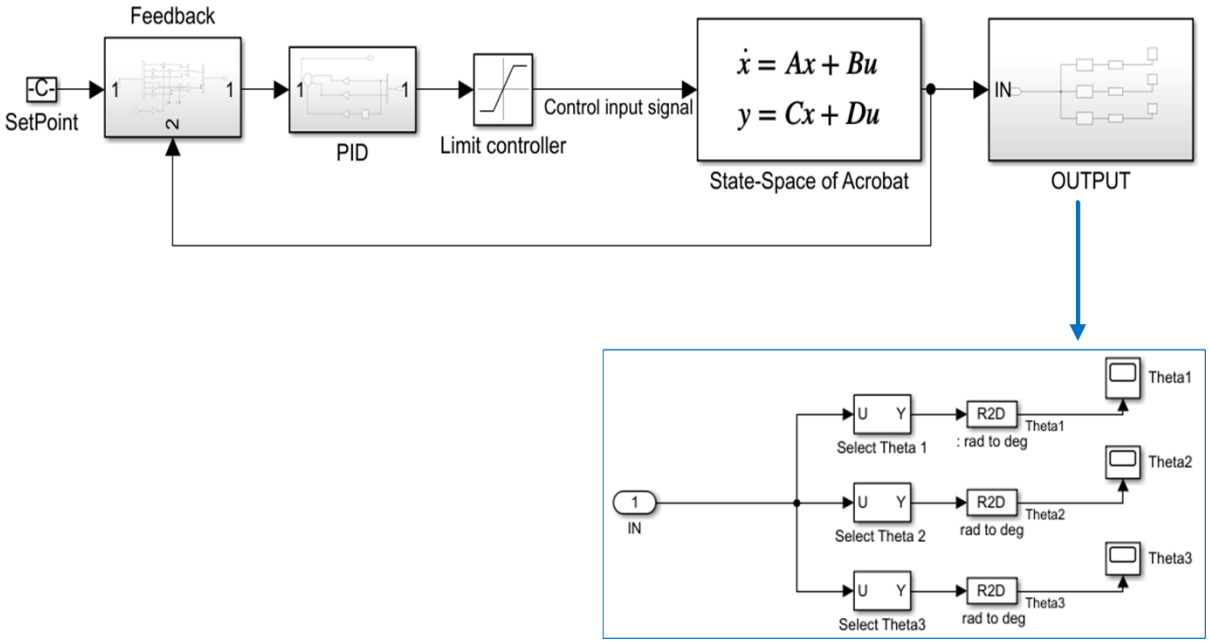


Figure 4.4 PID controller simulation model

Table 4.2 PID performance

| Theta | Controller | Overshoot $O_{sh}\ (pu)$ | Undershoot $U_{sh}\ (pu)$ | Settling time $T_s\ (s)$ | Rising time $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\boldsymbol{\theta_1}$ | PID | 6.386 | -5.444 | 24.106 | 0.444 |
| $\boldsymbol{\theta_2}$ | PID | 1 | -0.998 | 13.587 | 0.067 |
| $\boldsymbol{\theta_3}$ | PID | 0.309 | -0.40 | 3.806 | 0.040 |

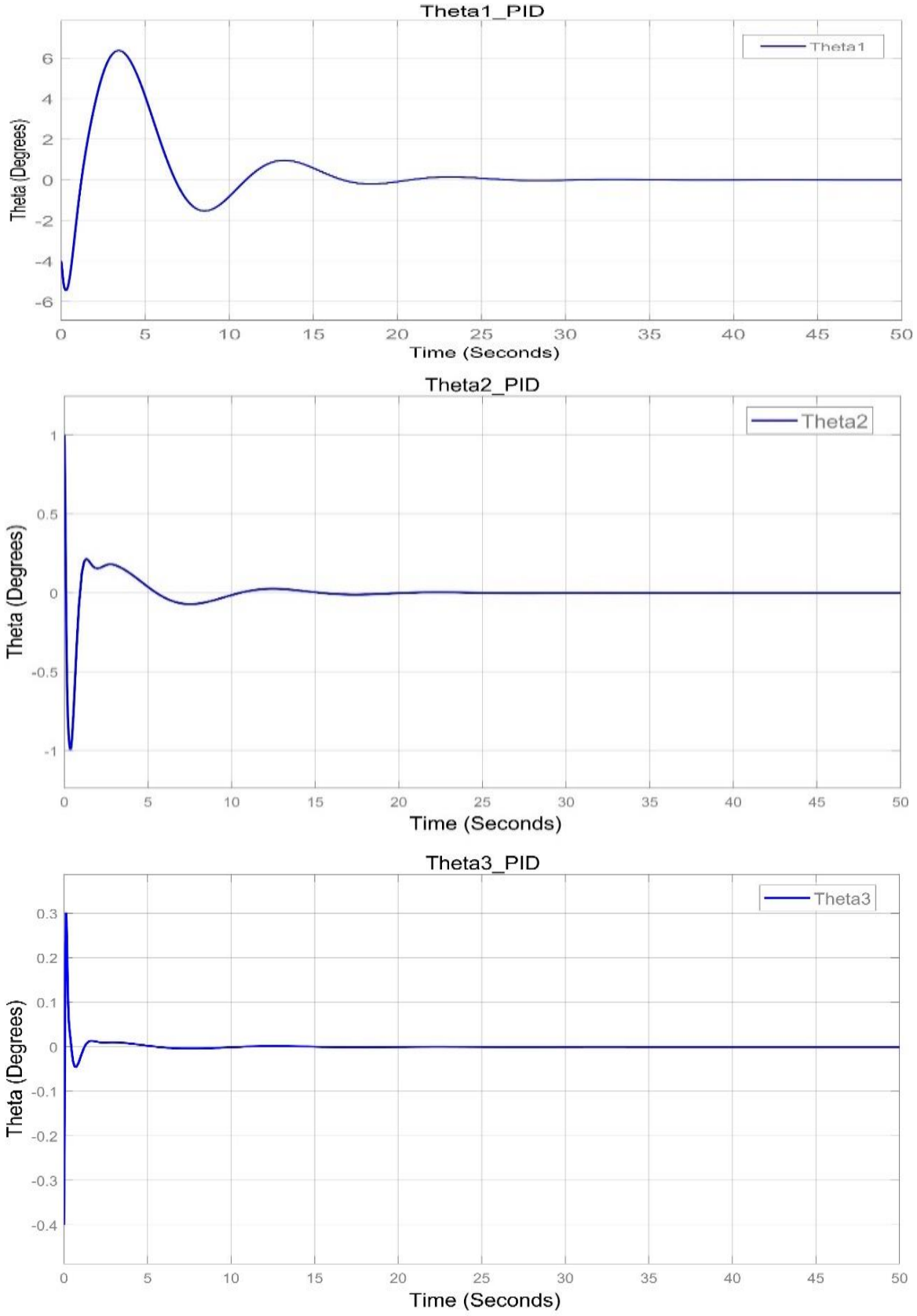


Figure 4.5 The step response of the system (PID)

Table 4.2 presents the performance analysis of a PID (Proportional, Integral, Derivative) controller applied to three different joints of a system, represented by theta values ($\theta_1$, $\theta_2$ and $\theta_3$). Each row of the table corresponds to a specific joint and its performance metrics when using a PID controller. The performance metrics evaluated for each joint include overshoot, undershoot, settling time, and rising time. Here's a brief explanation of each performance metric with the provided numbers for each joint:

In control theory, overshoot is the phenomenon of a signal exceeding its desired value. For joint $\theta_1$, the overshoot is 6.386, while for joint $\theta_2$ is 1, and for joint $\theta_3$ = 0.309. The different overshoot values indicate that the controller response varies across the joints, with joint $\theta_1$ experiencing the most aggressive response.

Undershoot: The amount by which the system output falls below its final steady-state value before reaching it. The undershoot for joint $\theta_1$, is -5.444, for joint $\theta_2$ is -0.998, and for joint $\theta_3$= -0.40. Similar to overshoot, the varying undershoots values suggest that the controller response differs for each joint.

Settling time: The time it takes for the system to stabilize within a specified error band of the final steady-state value. The settling times for joints 1, 2 and 3 are 24.106 seconds, 13.587 seconds, and 3.806 seconds, respectively. The different settling times indicate that the controller stabilizes each joint at different rates, with joint 3 achieving stability the fastest.

Rise time: The time it takes for the system output to go from 10% to 90% of its final steady-state value. The rise times for Theta 1, 2, and 3 are 0.444 seconds, 0.067 seconds, and 0.040 seconds, respectively. The varying rise times indicate that the controller brings each joint to its desired state at different speeds, with theta 3 reaching its state the fastest. Figure 4.6 illustrates PID performance.

In summary, the table shows that the performance metrics of the PID controller vary across the three different joints of the system. Joint 1 experience the most aggressive response, while joint 3 achieves the fastest settling and rise times.

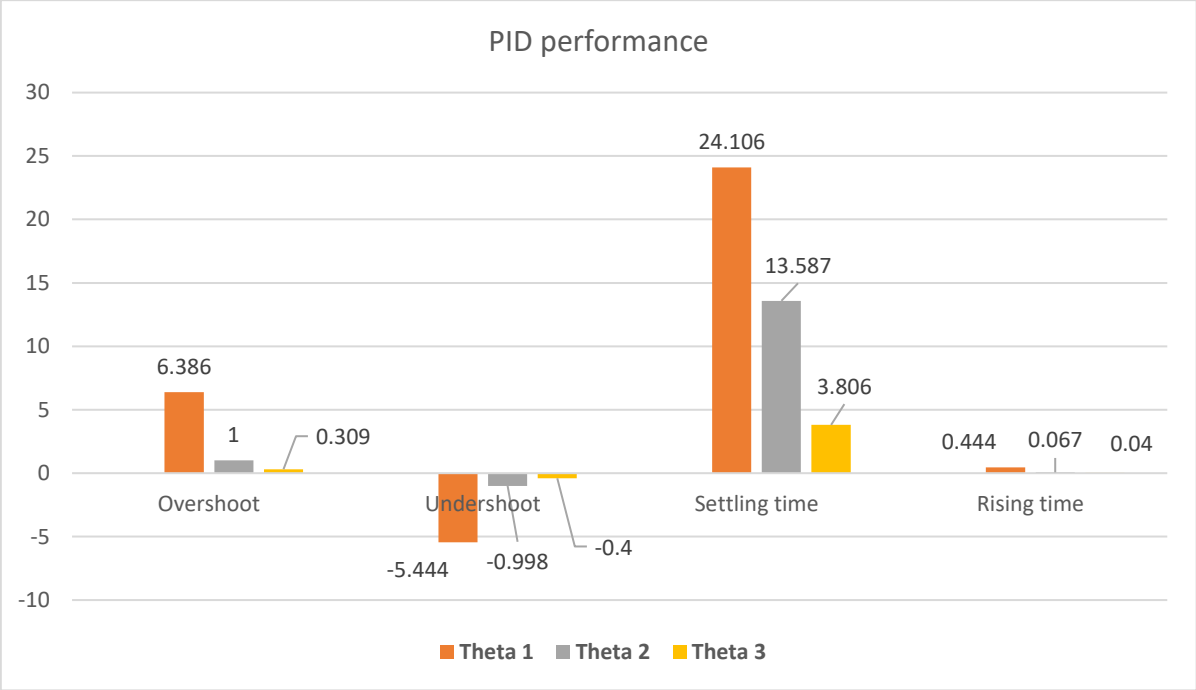


Figure 4.6 PID performance comparison for Theta 1, 2 and 3

## 4.4 Evaluating the performance of LQR and PID controllers

In the rapidly evolving world of robotics, control systems play a vital role in ensuring stability and precision in the movements of robotic systems. Robogymnast systems, which aim to mimic the agility and dexterity of human gymnasts, require appropriate controllers to execute complex motions. Two prominent controllers employed in such systems are the Linear Quadratic Regulator (LQR) and the Proportional-Integral-Derivative (PID) controller. This section focuses on comparing the performance of LQR and PID controllers specifically in the context of Robogymnast systems. By comparing their underlying principles, applicability,

strengths, and weaknesses. The comparison between both controller's step responses are shown in Figure 4.7.

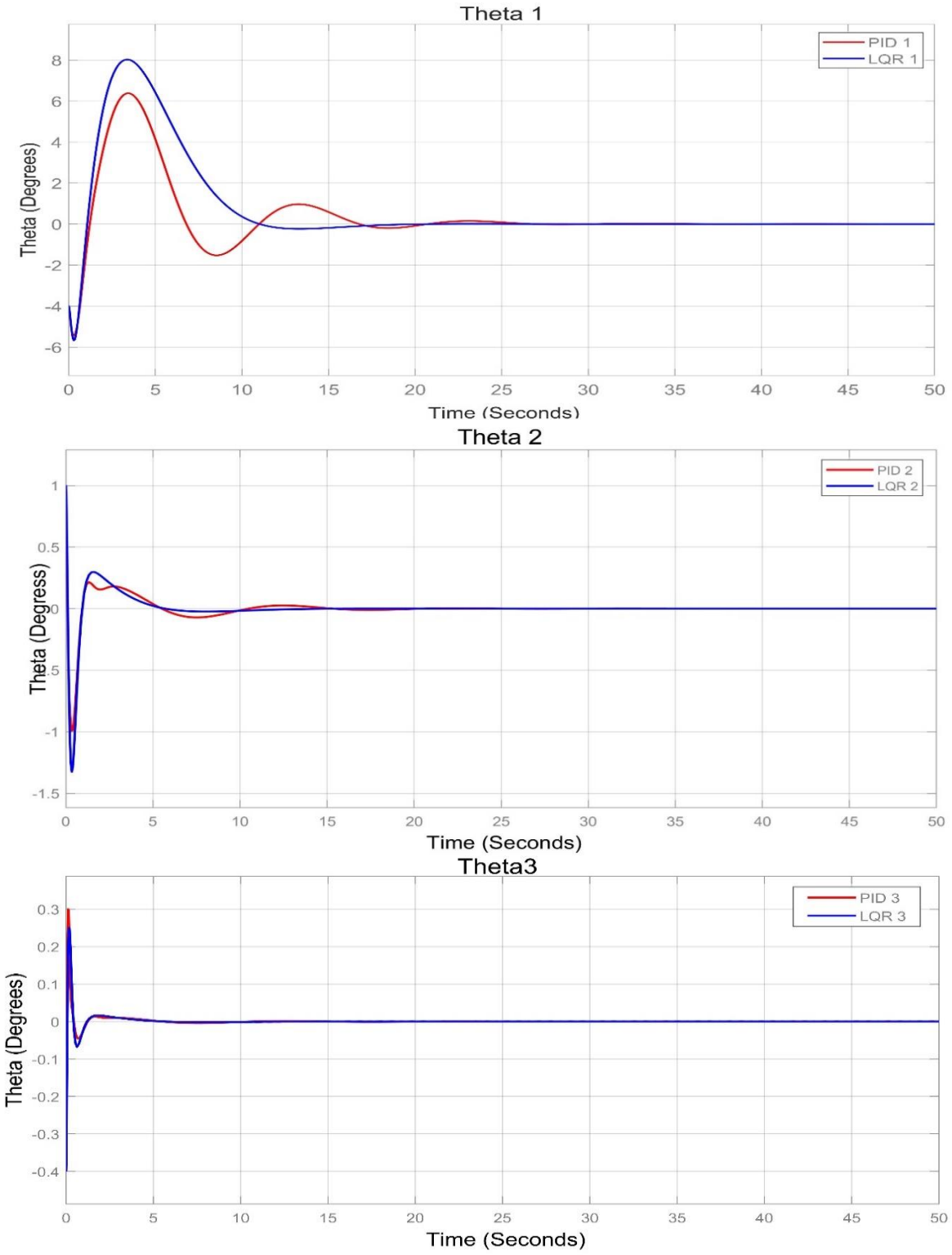


Figure 4.7 Step response of PID and LQR controllers

Table 4.3 Comparison of performance between LQR and PID outcomes

| Theta | Controller | Overshoot $O_{sh}\ (pu)$ | Undershoot $U_{sh}\ (pu)$ | Settling time $T_s\ (s)$ | Rising time $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | LQR | 8.02 | -5.69 | 15.519 | 0.335 |
| | PID | 6.386 | -5.444 | 24.106 | 0.444 |
| $\theta_2$ | LQR | 1.03 | -1.32 | 4.914 | 0.075 |
| | PID | 1 | -0.998 | 13.587 | 0.067 |
| $\theta_3$ | LQR | 0.25 | -0.41 | 3.331 | 0.050 |
| | PID | 0.309 | -0.40 | 3.806 | 0.040 |

Table 4.3 represents a comparison between the performance of Linear Quadratic Regulator (LQR) and Proportional-Integral-Derivative (PID) controllers for controlling different joints ($\theta_1$, $\theta_2$and $\theta_3$) of a robotic system. The performance metrics evaluated are the same as in the previous subsection.

In summary, the PID controller generally outperforms the LQR controller in terms of overshoot, undershoot, and rising time but the differences are not substantial. In terms of settling time, LQR is better than PID. Based on these results, the choice between LQR and PID controllers for a specific joint should be carefully considered, taking into account the importance of each performance metric in the context of the overall system. Figure 4.8 gives more illustration of all results.

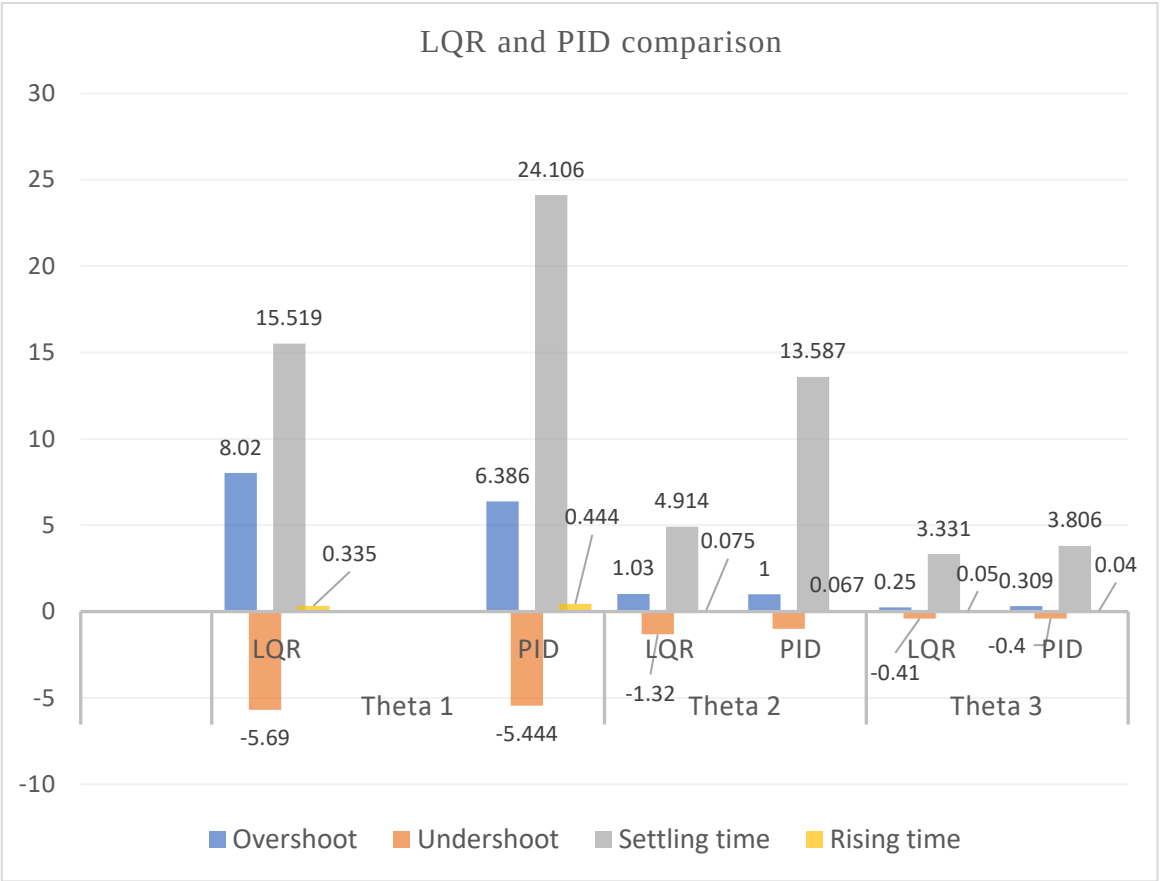


Figure 4.8 Comparison of performance between LQR and PID

## 4.5 Integral time of absolute error (ITAE)

Integral Time of Absolute Error (ITAE) is a performance measure used to evaluate the effectiveness of control systems, particularly in the context of tuning controllers in control engineering [109]. Researchers have found ITAE to be an effective criterion in designing and optimizing control systems [110]. The performance metric, (ITAE) has been employed to evaluate the efficiency of both Linear Quadratic Regulator (LQR) and Proportional-Integral-Derivative (PID) controllers when applied to the Robogymnast system. This assessment is achieved by comparing the ITAE values for each respective controller.

Table 4.4 ITAE values of LQR and PID controller

| Theta | Controller | ITAE |
|---|---|---|
| $\boldsymbol{\theta_1}$ | LQR | 159.7 |
| | PID | 94.180 |
| $\boldsymbol{\theta_2}$ | LQR | 0.322 |
| | PID | 0.317 |
| $\boldsymbol{\theta_3}$ | LQR | 0.022 |
| | PID | 0.021 |

Table 4.4 shows the (ITAE) values for a multi-joint system with three different joints, represented by Theta 1, Theta 2, and Theta 3. The table compares the ITAE values obtained by applying two different controllers, LQR and PID, to each joint. Upon analysing the table, it is apparent that the ITAE values differ significantly for each joint and controller combination. For instance, the ITAE value for Theta 1 with the LQR controller is significantly higher than the ITAE value for the same joint with the PID controller, suggesting that the PID controller is more effective at controlling this joint. On the other hand, the ITAE values for Theta 2 and Theta 3 with both controllers are very similar, indicating that both controllers perform equally the same for these joints.

Moreover, the ITAE values themselves indicate the level of error in each joint's response, where a lower ITAE value suggests a better-controlled joint. Notably, the ITAE values for Theta 2 and Theta 3 are relatively low for both controllers, indicating that these joints have good control. However, the ITAE value for Theta 1 is relatively high. It is clear that PID controller provides better performance for this system.

Overall, the table provides a useful comparison of the performance of two different controllers on different joints in a multi-joint system. The comparison highlights the importance of selecting and tuning controllers for individual joints in a multi-joint system.

## 4.6 Robustness investigation for PID controller

To ascertain the robustness of the PID controller, this section conducts a comprehensive analysis of the parametric uncertainty within the three-link system and its implications on system stabilization, employing a multi-scenario approach. A diverse range of potential parametric system conditions are examined. Initially, individual testbed parameters are modified in isolation, followed by simultaneous adjustments of multiple parameters, with increments and decrements of 15% and 30% from the baseline values. This systematic exploration facilitates a thorough understanding of the controller's robustness under various scenarios and contributes to its refinement for improved performance.

### 4.6.1 Results:

#### 4.6.1.1 Case 1: Original Value

Table 4.5 PID performance Case 1

| Theta | Controller | $O_{sh}\ (pu)$ | $U_{sh}\ (pu)$ | $T_s\ (s)$ | $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | PID | 6.386 | -5.444 | 24.106 | 0.444 |
| $\theta_2$ | PID | 1.000 | -0.998 | 13.587 | 0.067 |
| $\theta_3$ | PID | 0.309 | -0.400 | 3.806 | 0.040 |

As shown above, Figure 4.9 displays the controller in the original value where the Robogymnast parameters are constant.

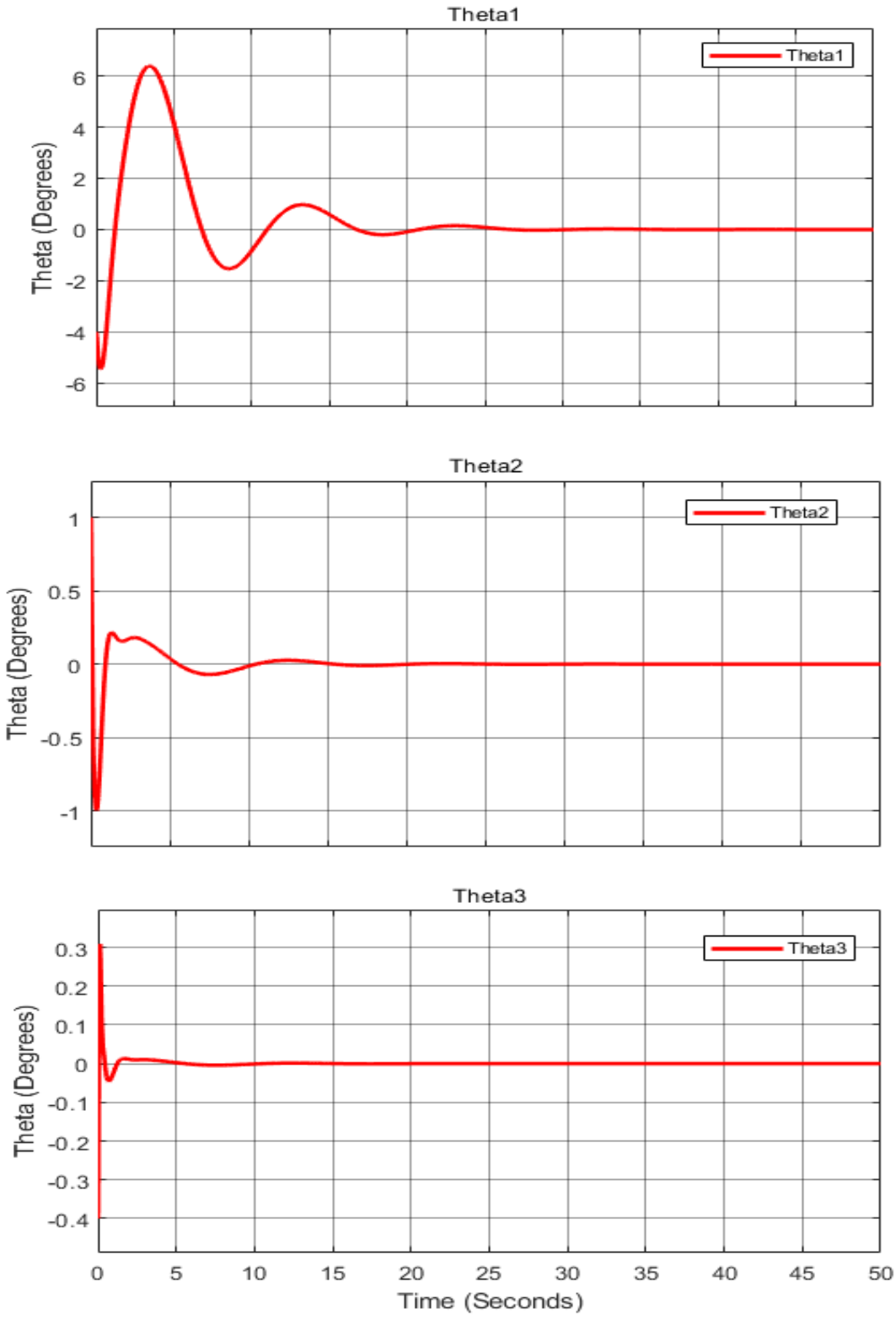


Figure 4.9 The system response of the upper, middle and lower joint of Robogymnast in Case 1

### 4.6.1.2 Case 2: (+15%)

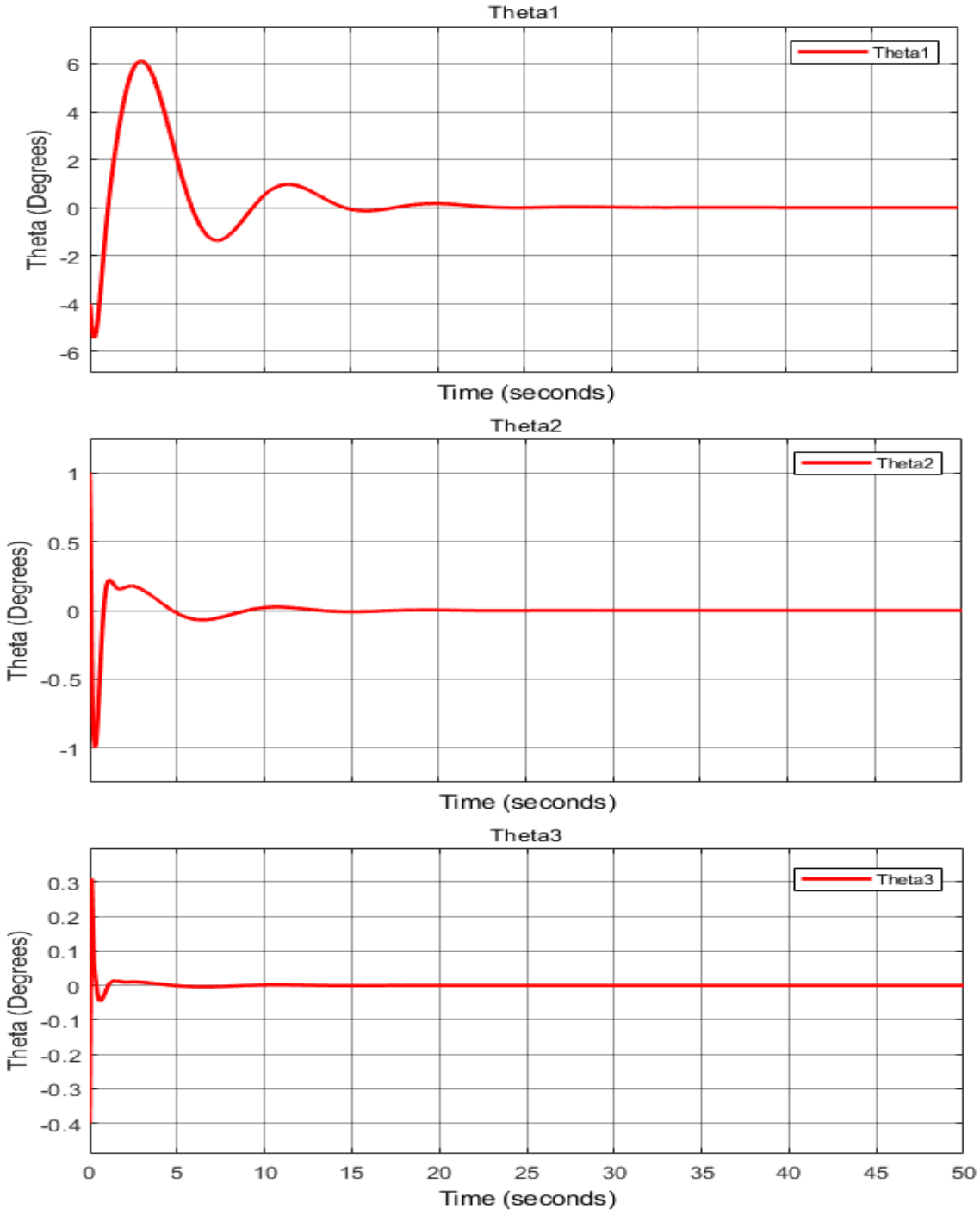


Figure 4.10 The system response of Theta1, Theta2 and Theta3 in Case 2

In Table 4.6, a full outcome of PID performance, case 2 is presented.

Table 4.6 PID performance in Case 2

| Theta | Controller | $O_{sh}\ (pu)$ | $U_{sh}\ (pu)$ | $T_s\ (s)$ | $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | PID | 6.114 | -5.445 | 21.123 | 0.388 |
| $\theta_2$ | PID | 1 | -0.999 | 11.594 | 0.058 |
| $\theta_3$ | PID | 0.309 | -0.400 | 3.238 | 0.035 |

#### 4.6.1.3 Case 3: (+30%)

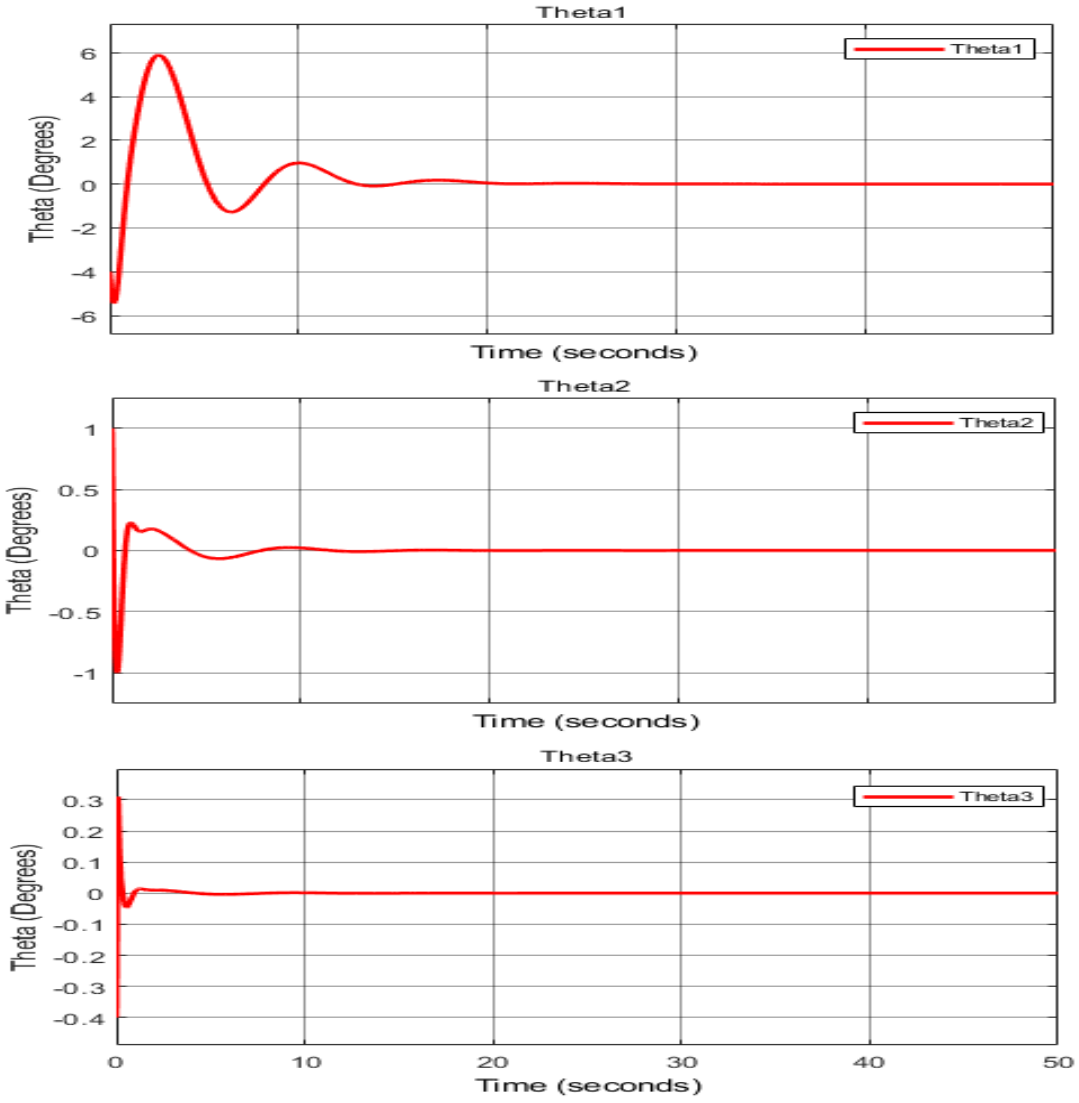


Figure 4.11 The system response of the 1st, 2nd and 3rd joint of Robogymnast in Case 3

Table 4.7 The performance of PID controller in Case 3 (+30%)

| Theta | Controller | $O_{sh}\ (pu)$ | $U_{sh}\ (pu)$ | $T_s\ (s)$ | $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | PID | 5.901 | -5.447 | 18.892 | 0.345 |
| $\theta_2$ | PID | 1 | -1.001 | 10.108 | 0.051 |
| $\theta_3$ | PID | 0.310 | -0.400 | 2.817 | 0.030 |

As shown, Table 4.7 presents the controller performance in the case 3 value where the Robogymnast parameters are added (+30%) to the original value.

#### 4.6.1.4 Case 4: (-15%)

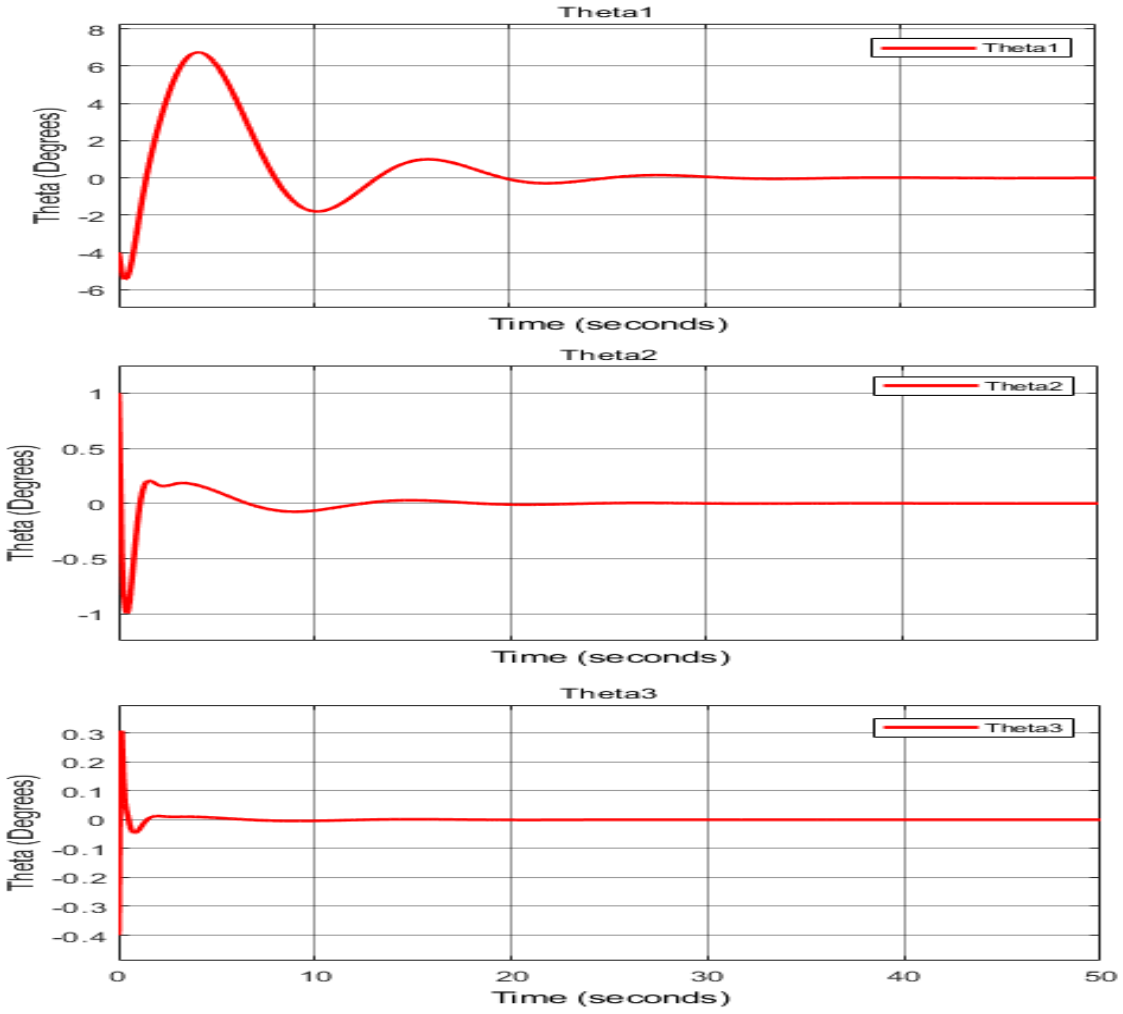


Figure 4.12 The system response of joint-1, joint-2 and joint-3 in Case 4

Table 4.8 PID performance in Case 4 (-15%)

| Theta | Controller | $O_{sh}\ (pu)$ | $U_{sh}\ (pu)$ | $T_s\ (s)$ | $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | PID | 6.749 | -5.441 | 28.388 | 0.518 |
| $\theta_2$ | PID | 1 | -0.995 | 16.392 | 0.079 |
| $\theta_3$ | PID | 0.308 | -0.400 | 4.612 | 0.047 |

### 4.6.1.5 Case 5: (-30%)

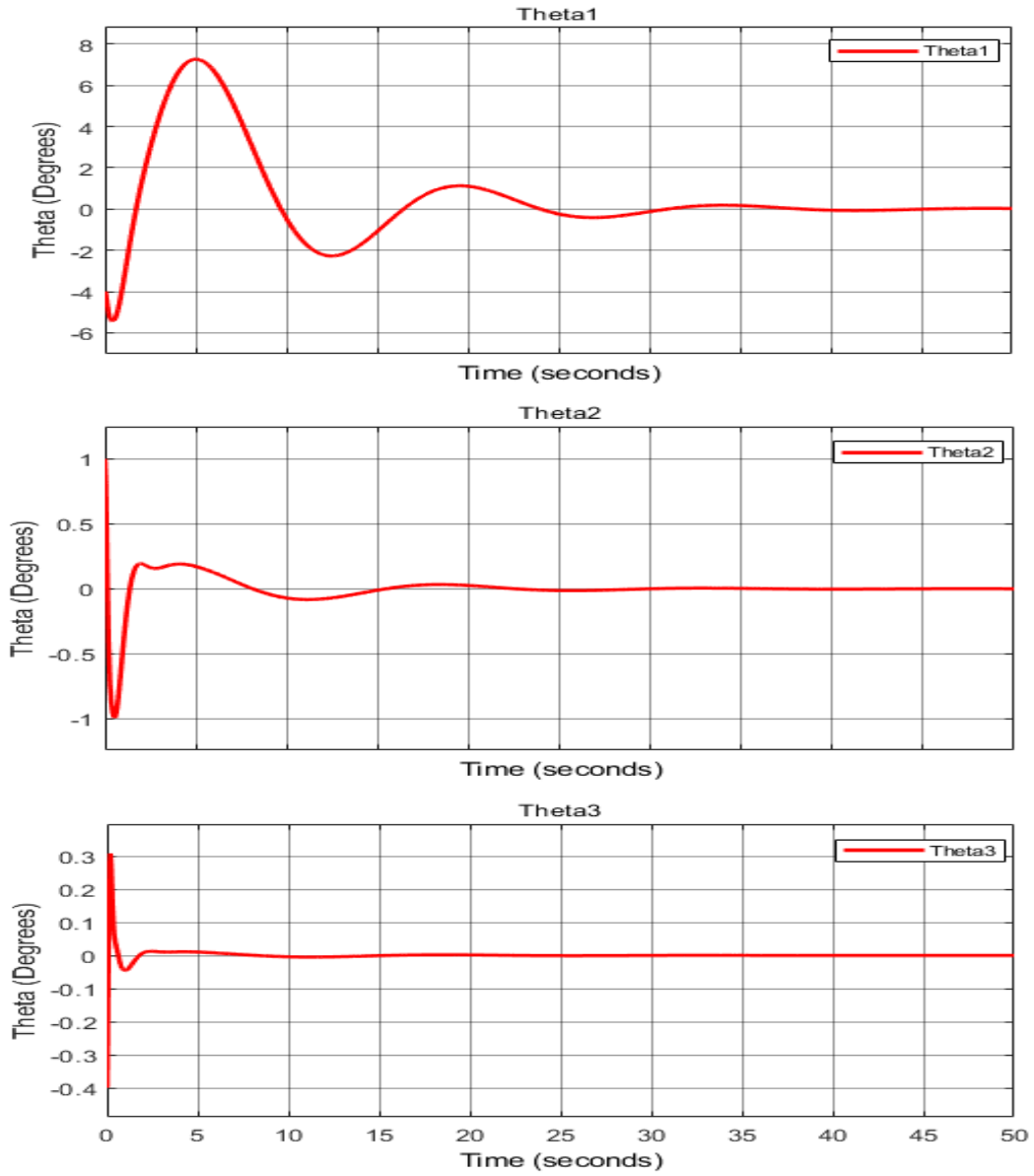


Figure 4.13 The system response of the first, second and third joint in Case 5

Finally, in Table 4.9 a full result of case 5, the PID performance is illustrated to validate the system stability.

Table 4.9 The performance of PID controller in Case 5 (-30%)

| Theta | Controller | $O_{sh}\ (pu)$ | $U_{sh}\ (pu)$ | $T_s\ (s)$ | $T_r\ (s)$ |
|---|---|---|---|---|---|
| $\theta_1$ | PID | 7.264 | -5.439 | 34.983 | 0.626 |
| $\theta_2$ | PID | 1 | -0.992 | 20.586 | 0.096 |
| $\theta_3$ | PID | 0.308 | -0.400 | 5.836 | 0.057 |

### 4.6.2 Comparison of Integral Time Absolute Error

This section presents the Integral Time of Absolute Error (ITAE) for each case investigated, including the original values, as well as those with a variation of (±15%) and (±30%) in the PID controller. The ITAE is a crucial performance index used in control system design, optimization, and tuning. Comparing the ITAE across multiple cases is significant. This process involves evaluating and contrasting the ITAE values of different control strategies or scenarios within the Robogymnast environment to determine their relative effectiveness. The following Table 4.10 illustrates the different five cases values.

Table 4.10 ITAE values for all cases

| Theta | Case 1 | Case 2 | Case 3 | Case 4 | Case 5 |
|---|---|---|---|---|---|
| $\theta_1$ | 94.180 | 80.890 | 70.940 | 112 | 139.600 |
| $\theta_2$ | 0.317 | 0.240 | 0.188 | 0.434 | 0.698 |
| $\theta_3$ | 0.021 | 0.016 | 0.012 | 0.029 | 0.047 |

## 4.7 Summary

This chapter discussed the swinging controller design for a non-linear, three-link robotic gymnast, known as Robogymnast. The main objective of the study was to investigate the swinging of a triple-link mechanism consisting of two active-power links and an overhead non-powered link. The chapter focuses on two control techniques: linear-quadratic-regulators (LQR) and proportional-integral-derivative (PID) controllers. Modelling was conducted using MATLAB, which was utilized for the simulation and implementation of the system. The performance of LQR and PID controllers was compared for different joints (Theta 1, Theta 2, and Theta 3) of the robotic system. The performance metrics evaluated were overshoot, undershoot, settling time, and rising time. It was observed that the PID controller generally outperforms the LQR controller. The Integral Time of Absolute Error (ITAE) was used to evaluate the efficiency of both LQR and PID controllers when applied to the Robogymnast system. It was found that the PID controller provided better performance for the system based on the ITAE values.

Finally, the robustness of the PID controller was investigated by conducting a comprehensive analysis within the three-link system under various scenarios. The PID controller demonstrated stable performance and good robustness under different conditions. Chapter 5 will delve into the optimization techniques applied to the Robogymnast system. It will clarify the results derived from each method, followed by a discussion and comparison of their respective outcomes.

# Chapter 5: Optimisation Techniques

## 5.1 Introduction

Optimization techniques have become an integral part of various domains, including control engineering, due to their ability to improve the performance of complex systems and solve challenging real-world problems. These techniques aim to find the best possible solution to a given problem by minimizing or maximizing an objective function, subject to a set of constraints [111]. Optimization in control refers to the process of finding the optimal solution to a control problem, it involves identifying the best possible outcome based on certain criteria or constraints, which is typically defined as a set of goals or objectives that need to be achieved while minimizing some measure of performance or cost. This can involve designing controllers or control systems that are able to achieve the desired performance while minimizing the use of resources such as energy or computational resources or optimizing the operation of a control system to achieve the desired performance in real-time. Optimization in control can be achieved through the use of various optimization algorithms and techniques [112].

In this chapter, the focus is on the implementation of two prominent optimization algorithms, Ant Colony Optimization (ACO) and Gravitational Search Algorithm (GSA), within the current system. The goal is to identify optimal solutions, improve overall efficiency, and draw a comparison between the performances of these two techniques. ACO, inspired by the foraging behaviour of ants [113]. Conversely, GSA, a physics-based optimization

algorithm, is based on the concept of gravitational force and mass interactions, enabling efficient exploration of the search [114]. By integrating the stochastic search mechanism of ACO with the balanced exploration and exploitation capabilities of GSA, the aim is to enhance the optimization process's robustness and convergence speed in the current system. Additionally, a performance comparison of these algorithms will be conducted to identify their respective strengths and weaknesses. This comparison will provide valuable insights into the potential effects of applying these algorithms.

## 5.2 Optimisation Method

In order to address an optimisation issue, a specific optimal formulation method is required. A single optimal formulation approach cannot be applied to all design problems, as the objective functions and related parameters vary across problems. The primary goal of the formulation process is to create a mathematical model representing the optimal design problem, which is then solved using an appropriate optimization algorithm. This algorithm requires the optimization problem to be presented in a specific format [115] [116].

The initial phase of developing an algorithm entail recognizing the necessity and objective of optimization. Following this, components such as design variables, constraints, objective functions, variable bounds, and algorithms are chosen specifically for the given problem. A detailed explanation of these steps is provided below.

- Decision variables

  In the context of optimization challenges, decision variables, also known as design variables, represent the undetermined elements that must be identified through problem-

solving efforts. Notably, the optimization simulation's performance and expediency are dependent on the number of decision variables to a large extent [117].

- Constraints

  Once the design variables have been determined, it is essential to establish the constraints or restrictions for the problem at hand. Constraints define the connection between design variables and other factors to satisfy the demands of a physical phenomenon or resource limitations [118]. Examples of such constraints include the battery state of charge in electric vehicles and battery storage, voltage limits in distribution networks, and thermal capacities of distribution network cables. Constraints can be expressed as equalities (=) or inequalities (less than or equal to ≤, or greater than or equal to ≥). As noted in [115], the majority of constraints in design issues are typically inequality-based.

- Objective function

  The third step in optimal problem formulation is the objective function, which is a crucial aspect in operations research, decision making, and optimization problems. The objective function serves as a mathematical representation of the goal or target that needs to be achieved. In other words, it is a function that quantifies the performance of a decision variable or a set of decision variables in the context of a problem [119][120]. The objective function can either be maximized or minimized depending on the nature of the problem. For example, in a profit maximization problem, the objective function represents the total profit that needs to be maximized, while in a cost minimization problem, the objective function represents the total cost that needs to be minimized [121]. In conclusion, the objective function is a critical component in optimal problem formulation. It represents the

goal to be achieved through the optimization process, and its careful definition is essential for obtaining the correct and optimal solution.

- Variable bounds

  The subsequent step involves establishing variable bounds by setting minimum and maximum limits on design variables. While certain algorithms do not require this information, others rely on it. These bounds indicate that solution points must fall within the specified range [122].

### 5.2.1 Review of optimization algorithms

An algorithm is typically considered a series of instructions that a computer can interpret, featuring clear and unambiguous meaning. It consistently requires input and generates output. A visual representation of an algorithm can be seen in Figure 5.1 [123].

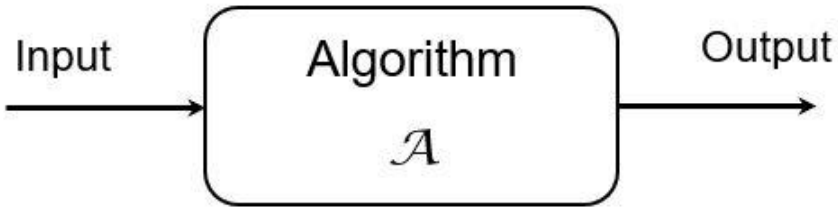


Figure 5.1 Graphical representation of an algorithm [123]

In mathematical contexts, the process of producing a series of solutions for a specific issue is referred to as an iterative method, with an algorithm being a distinct type of iterative method. An optimization algorithm serves to determine an optimized solution for a given function. For instance, considering a function f(x), the optimized solution would correspond to the x value at which f(x) is minimized or maximized, given certain constraints on x.

## 5.3 ACO Algorithm

The Ant Colony Optimization (ACO) algorithm is a powerful and innovative computational technique inspired by the foraging behaviour of natural ant colonies. Initially proposed by Dorigo et al. in 1991 [124]. This swarm intelligence-based approach has been widely applied to tackle complex combinatorial optimization problems [125][126]. Ants exhibit sophisticated collective behaviour when searching for food sources, efficiently exploiting their environment and adapting to changes. This behaviour is driven by their ability to communicate indirectly through pheromone trails [127]. The ACO algorithm models this processes by simulating artificial ants that traverse a graph representation of a problem, depositing pheromone trails that influence the decisions of subsequent ants [128]. Over time, these pheromone trails guide the ants towards increasingly optimal solutions [129]. Researchers have found that ACO algorithms demonstrate excellent performance in solving complex optimization problems, often surpassing traditional optimization techniques [130].

The Ant Colony Optimization algorithm represents a promising and versatile optimization technique that has found numerous applications in a wide range of problem domains. By leveraging the principles of swarm intelligence and the remarkable foraging behaviour of ants, the ACO algorithm has proven to be an effective and robust approach to combinatorial optimization [131][132].

The flow chart of Ant Colony Optimization algorithm consists of several stages as illustrated in the following Figure 5.2.

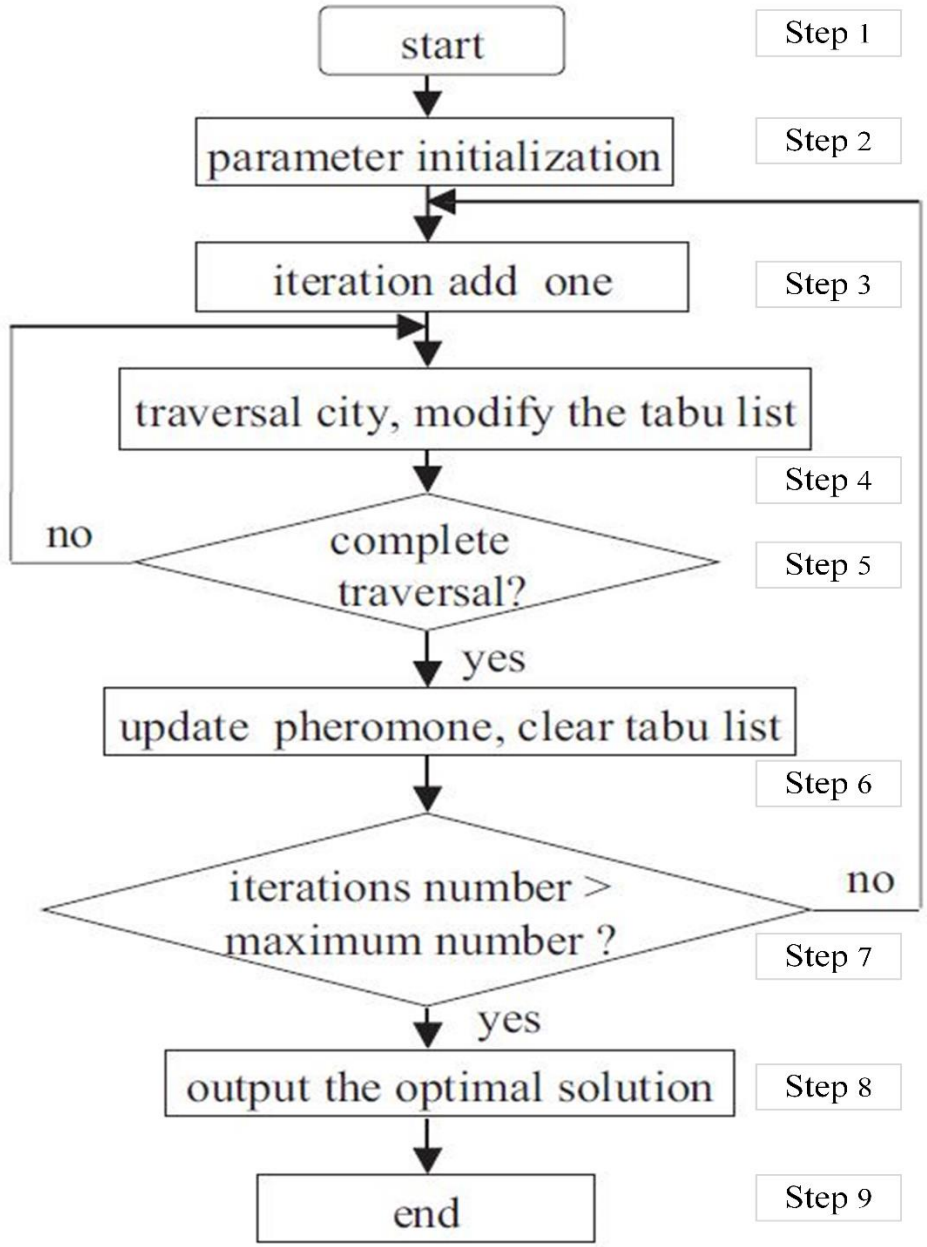


Figure 5.2 Ant Colony Optimisation algorithm flow chart [133]

First, the problem is defined, and the parameters are set. Then, a set of artificial ants is created and placed on the starting point. Each ant moves through the problem space by selecting a path based on pheromone trails left by other ants. As the ants move, they deposit pheromones on their paths, which attract other ants to follow the same path. This process continues until all ants have completed their journey and a solution is found [133].

Next, the quality of each solution is evaluated using a fitness function. The best solutions are then used to update the pheromone trails in order to reinforce good paths and discourage bad ones [133].

Finally, the algorithm checks if a stopping criterion has been met (e.g., maximum number of iterations reached or desired level of convergence achieved). If not, it repeats the process from step 3 until a satisfactory solution is found [134].

### 5.3.1 The implementation of ACO on Robogymnast system

Implementing Ant Colony Optimization (ACO) on the Robogymnast system to minimize Integral Time Absolute Error (ITAE) in the controller involves a series of steps. Here is a step-by-step guide for implementing ACO on the Robogymnast system:

1. Define the problem and model the system: The first step is to define the problem and create a mathematical model of the Robogymnast system. You need to identify the system's parameters and dynamics to design the controller. This can be achieved through the system identification process [135].

2. Formulate the optimization problem: Define the ITAE as the objective function to minimize. The ITAE is an integral performance measure that evaluates the quality of control by weighting the error according to time. Formally, the ITAE can be defined in equation (5.1) [136]:

   $$ITAE = \int t * |e(t)| \, dt \quad (5.1)$$

   Where: t represents time, and e(t) is the error signal at time t.

3. Initialize the ACO algorithm: Set the initial values for the algorithm parameters, such as the number of ants (m), pheromone evaporation rate (ρ), pheromone constant (Q),

exploration parameter (α), and exploitation parameter (β). These parameters can be fine-tuned according to the specific problem to obtain the best performance [137].

4. Construct solutions: In each iteration, the ants construct solutions by moving through the search space, probabilistically selecting the next controller parameters based on the pheromone levels and heuristic information [137].

5. Evaluate solutions: Calculate the ITAE for each solution generated by the ants. This will allow you to determine the quality of each solution and update the pheromone trails accordingly [138].

6. Update pheromone trails: Increase the pheromone level on the path that corresponds to better solutions (i.e., lower ITAE values), while decreasing it for the worse solutions. This will guide the ants towards better solutions in the subsequent iterations [138].

7. Termination criteria: Check whether the termination criteria are met, such as reaching a maximum number of iterations, convergence of the solutions, or reaching a satisfactory ITAE value. If the criteria are not met, return to step 4 and continue the process [137].

8. Extract the best solution: Once the termination criteria are met, extract the best solution found during the iterations. This solution represents the optimal controller parameters (P, I and D) values that minimize the ITAE for the Robogymnast system [136].

9. Implement the controller: Use the optimal controller parameters obtained from the ACO to design the controller for the Robogymnast system. Verify the performance of the controller in terms of minimizing the ITAE and ensuring satisfactory system response.

The mathematical model of the Ant Colony Optimization (ACO) algorithm for the PID controller optimization problem is summarised in the following steps:

Step 1: Define the problem and model the system dynamics:

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

$\dot{x}$ is the state vector of the system, y is the output vector.

Step 2: Formulate the optimization problem

Objective function:

$$ITAE = \int t * |e(t)| \, dt$$

Step 3: Initialize the ACO algorithm Parameters:

Number of ants (m), Pheromone evaporation rate (ρ), Pheromone constant (Q), Exploration parameter (α) and Exploitation parameter (β).

Step 4: Construct solutions

In each iteration, ants construct solutions by probabilistically selecting PID controller parameters P, I, and D:

$$u_{ant}(t) = P_{ant} \cdot e(t) + I_{ant} \int_0^t e(\tau)\, d\tau + D_{ant} \frac{d}{dt} e(t)$$

Step 5: Evaluate solutions

Calculate ITAE for each ant's solution:

$$\text{ITAE}_{\text{ant}} = \int_0^T t \ .|e_{ant}(t)|\ dt$$

Where T is the final time.

Step 6: Update pheromone trails

Update pheromone levels based on solution quality:

$$\Delta\tau_{ant} = \frac{Q}{\text{ITAE}_{ant}}$$

Step 7: Termination criteria

Check termination criteria, such as maximum iterations or convergence.

Step 8: Extract the best solution

After termination, select the ant that produced the best solution:

Best_ant = arg min($\text{ITAE}_{\text{ant}}$)

Step 9: Implement the controller

Use control parameters from the best ant to design the PID controller:

$P_{optimal} = P_{\text{Best\_ant}}$

$I_{optimal} = I_{\text{Best\_ant}}$

$D_{optimal} = D_{\text{Best\_ant}}$

These values of PID controller are implemented in the system dynamics to control the Robogymnast.

The main Ant Colony Optimization (ACO) algorithm parameters given by the following Table 5.1 [137].

Table 5.1 ACO parameters

| Parameter | Symbol | Description |
|---|---|---|
| Number of ants | m | The number of artificial ants used in the algorithm. |
| Pheromone level | $\tau$ | Represents the attractiveness of a particular solution component, based on previous experience. |
| Pheromone evaporation rate | $\rho$ | The rate at which pheromone evaporates, controlling the balance between exploration and exploitation. ($0 < \rho \leq 1$) |
| Pheromone constant | Q | A constant used in the pheromone update equation to adjust the intensity of pheromone deposited. |
| Exploration parameter | $\alpha$ | Controls the influence of pheromone levels on the ants' decision-making process. |
| Exploitation parameter | $\beta$ | Controls the influence of heuristic information on the ants' decision-making process. |
| Heuristic information | $\eta$ | Provides problem-specific guidance to ants based on the local characteristics of the search space. |

These parameters are critical for the performance of the ACO algorithm and may need to be fine-tuned according to the specific optimization problem being solved. Proper tuning of these parameters can lead to more efficient search processes and better optimization results.

### 5.3.2 ACO results

This work was implemented in MATLAB version (2022a) installed on Intel (R) Core (TM) i5-8500 CPU @ 3.00GHz computer, the ACO algorithm code was programmed in (.m files), and the Robogymnast system model was simulated in the MATLAB Simulink

environment the m MATLAB file (the code of the algorithm) is calling the Simulink file where the acrobot system model is simulated. Figure 5.3 shows The Convergence graph of ACO.

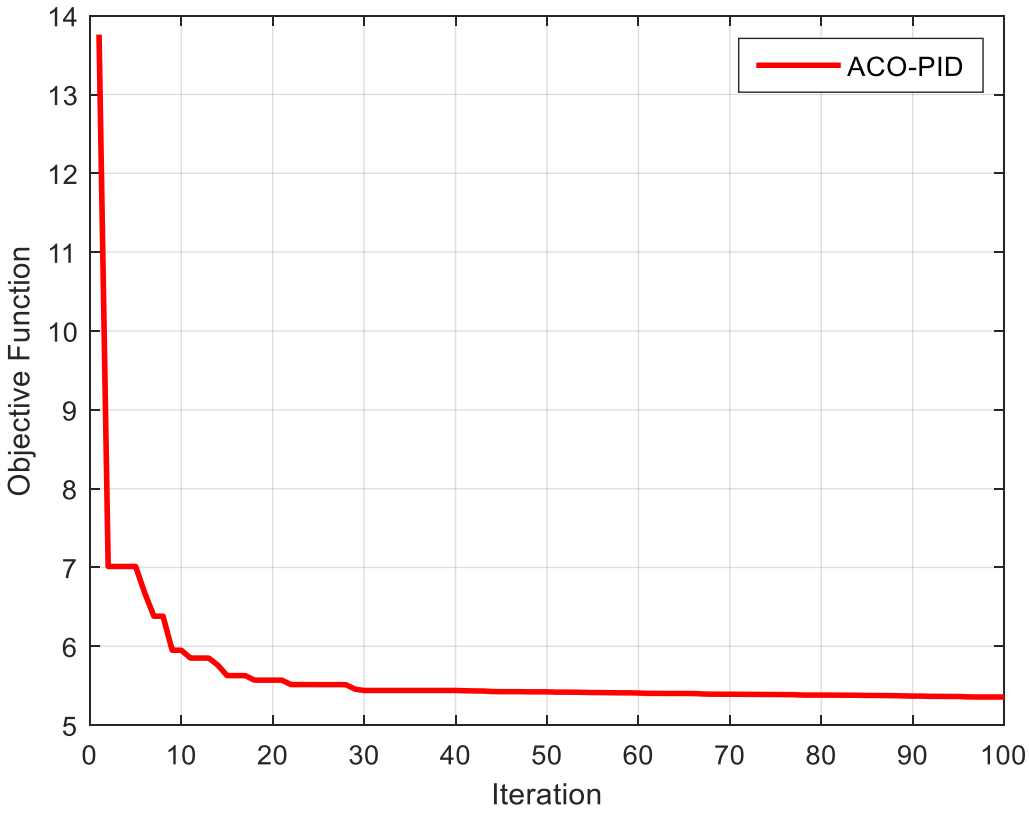


Figure 5.3 The Convergence graph of ACO algorithm

Table 5.2 ACO results parameters

| No. Iteration | No. Population |
|---|---|
| 100 | 40 |

Figure 5.3 shows the convergence graph of the ACO algorithm which provides a visual representation of how the algorithm's performance evolves over iterations. The convergence graph of the ACO algorithm illustrates the progress of the algorithm over a series of iterations. The x-axis shows the number of iterations, while the y-axis represents the best objective function value achieved at each iteration. The graph demonstrates the algorithm's ability to

gradually improve its performance as it explores the solution space. Table 5.2 shows ACO optimised parameters and Figure 5.4 display PID optimised response for each joint.

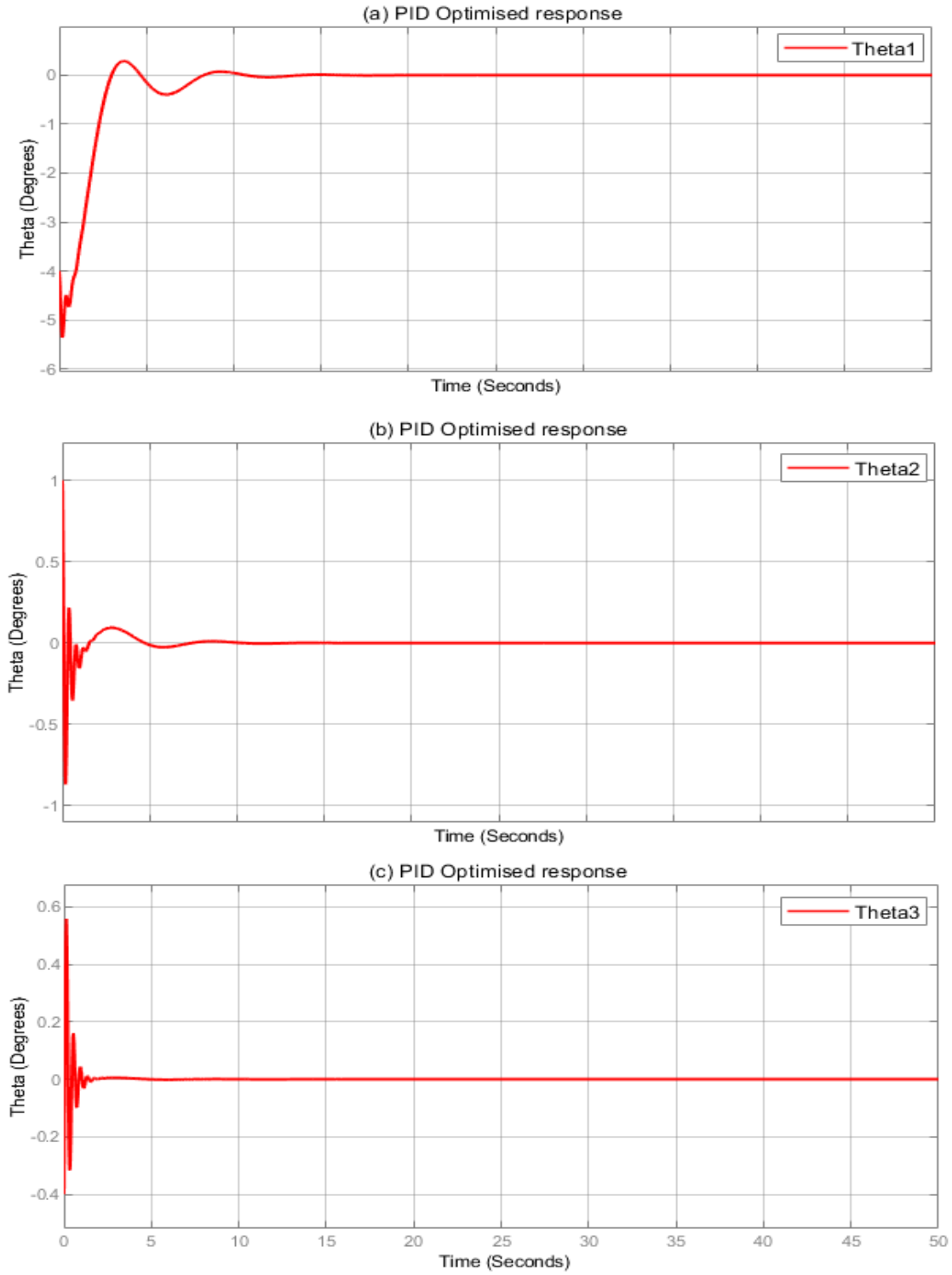


Figure 5.4 (a) Optimised PID-ACO system response for 1st joint; (b) PID-ACO system response for 2nd joint; (c) PID-ACO system response for 3rd joint of Robogymnast

Table 5.3 PID-ACO optimised performance

| Symbol | Controller | Overshoot $O_{sh}\ (pu)$ | Undershoot $U_{sh}\ (pu)$ | Settling time $T_s\ (s)$ | Rising time $T_r\ (s)$ | ITAE |
|---|---|---|---|---|---|---|
| $\theta_1$ | PID-ACO | 0.290 | -5.360 | 7.734 | 0.182 | 11.650 |
| $\theta_2$ | PID-ACO | 1 | -0.868 | 6.404 | 0.041 | 0.310 |
| $\theta_3$ | PID-ACO | 0.300 | -0.400 | 1.363 | 0.033 | 0.020 |

Table 5.3 presents the performance metrics of a PID-ACO controller applied to the three joints (Theta1, Theta2 and Theta3). The metrics evaluated include overshoot, undershoot, settling time, rising time, and ITAE.

For joint 1, the PID-ACO controller exhibited an overshoot of 0.290 *p.u*, an undershoot of -5.360 *p.u*, a settling time of 7.734 second, a rising time of 0.182 second, and an ITAE of 11.650. In the case of 2nd joint, the controller displayed an overshoot of 1 *p.u*, an undershoot of -0.868 *p.u*, a settling time of 6.404 second, a rising time of 0.041 second, and an ITAE of 0.310. Lastly, for 3rd joint, the optimized PID performance demonstrated an overshoot of 0.300 *p.u*, an undershoot of -0.400 *p.u*, a settling time of 1.363 second, a rising time of 0.033 second, and an ITAE of 0.020.

The results indicate that the PID-ACO controller's performance varied across the three joints. In terms of overshoot, joint 2 exhibited the highest value, while joint 1 had the most significant undershoot. Furthermore, Theta1 had the longest settling and rising times, whereas Theta3 showed the shortest values for these metrics. Theta1 also had the highest ITAE, which suggests that its overall performance might be less desirable compared to the other Thetas.

In conclusion, this analysis demonstrates that the optimised performance of the PID controller is highly dependent on the specific joint (Theta) it is applied to. Further investigation

could focus on determining the factors that influence these variations and exploring ways to optimize the controller's performance for each joint.

## 5.4 Gravitational Search Algorithm

The Gravitational Search Algorithm (GSA) is a nature-inspired optimization algorithm that has been increasingly utilised to address complex optimization problems across a variety of fields. First introduced by Rashedi, Nezamabadi-pour, and Saryazdi in 2009 [139]. GSA is grounded in the principles of Newtonian physics, specifically the laws of gravitation and mass interactions. Since its foundation, the algorithm has been extensively studied and modified, leading to a multiplicity of applications in diverse disciplines, such as engineering design, feature selection, power system optimization, and scheduling problems [140]–[143].

The primary aim of GSA is to efficiently search the solution space of complex optimization problems and locate the global optimum or near-optimal solutions. In GSA, a population of agents, representing potential solutions to the optimization problem, is initialized. Each agent is assigned a mass, which is determined by the fitness of the corresponding solution. The agents are then attracted to each other based on their gravitational forces, which are influenced by their respective masses and the distance between them. Throughout the iterative search process, the gravitational forces between the agents are calculated, and the sites of the agents are updated accordingly. As the search progresses, agents with better fitness values, or higher masses, exert stronger gravitational forces on the others, causing the population to converge towards the global optimum [139][140]. The main flowchart of GSA is shown in Figure 5.5.

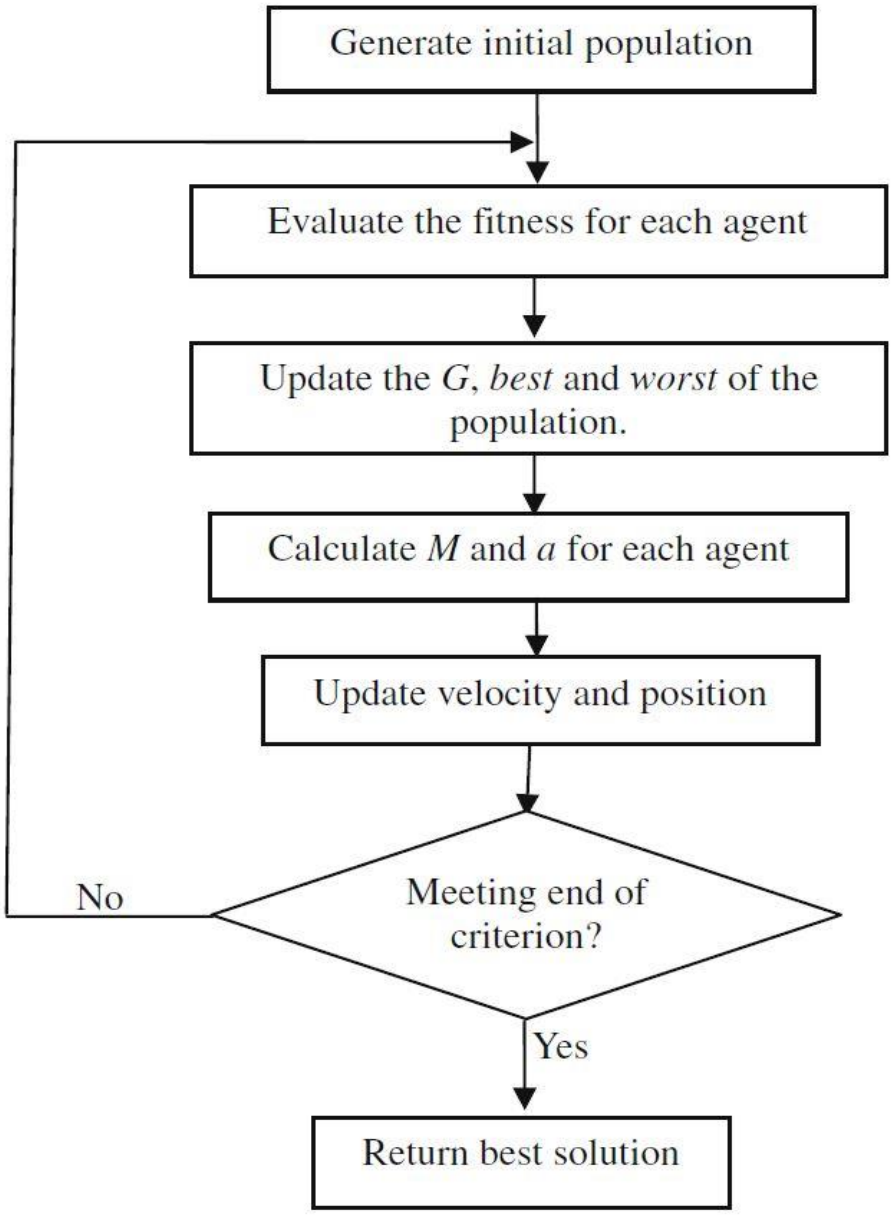


Figure 5.5 The flowchart of GSA [139]

In this segment, it provides a succinct overview of the Gravitational Search Algorithm (GSA) through several key steps [139]:

- Initialization: Set algorithm parameters, create an initial population of agents with random positions, and calculate their fitness values and masses.

- Iteration loop: perform the following steps for each iteration until the termination criteria are met:
- Calculate the gravitational constant G(t) for the current iteration.
- For each agent, compute the total gravitational force from other agents, calculate acceleration, and update the agent's velocity and position.
- Evaluate the fitness values of agents with their updated positions and update their masses accordingly.
- Termination: Check the termination criteria (e.g., the maximum number of iterations reached, or a predefined error threshold is met). If the termination criteria are satisfied, obtain the best solution found by the GSA.

### 5.4.1 The applications of GSA

In the field of control engineering, the Gravitational Search Algorithm (GSA) has been applied to various control field problems, including tuning of controllers, identification of systems, and control optimization. Here are some applications of GSA in the control field:

- Image Processing: GSA is used in feature detection and pattern recognition in images, which is a significant part of computer vision tasks. This can be applied in areas like facial recognition, object detection, and medical imaging [144].
- Fuzzy Logic Controller Design. Research in [145] proposed a hybrid algorithm combining Particle Swarm Optimization PSO and GSA to design a fuzzy logic controller. The results show that the hybrid algorithm outperforms the individual optimization algorithms.

- Power flow for distribution networks [146]. The study was conducted a GSA-based method for solving the optimal power flow (OPF) problem in distribution networks with distributed generation (DG) units. The OPF problem is a nonlinear optimization challenge with equality and inequality constraints. The goals are to minimize fuel costs for DG units, reduce power loss in the network, and achieve simultaneous minimization of both fuel costs and power loss. The results are compared to those from a genetic algorithm, demonstrating the effectiveness and robustness of the GSA approach.

### 5.4.2 The implementation of GSA on Robogymnast system

Implementing the Gravitational Search Algorithm (GSA) on the Robogymnast system to minimize the Integral of Time Absolute Error (ITAE) in the controller involves several steps [139][147][148]. Here are outlines of the steps and equations involved in GSA for optimizing PID controller parameters to minimize ITAE in the Robogymnast system:

1. Problem formulation: Define the problem as minimizing the ITAE of the Robogymnast controller. The controlled system is represented by the state-space equations:

   $$\dot{x} = Ax + Bu$$

   $$y = Cx + Du$$

2. Define the fitness function: Create a fitness function that evaluates the ITAE for a given set of controller parameters. The fitness function is defined as the ITAE of controller parameters $K_p$, $K_i$ and $K_d$:

   $$ITAE = \int t * |e(t)| \, dt$$

3. Initialize the GSA parameters: Choose the initial values for the GSA parameters, such as the number of agents (N), search space dimensions D, where D=3 for PID parameters $K_p$,

$K_i$ and $K_d$, and iteration limits. Initialize the positions of the agents (candidate solutions) randomly within the search space [139].

4. Calculate the fitness of each agent: Evaluate the ITAE for each agent's position in the search space using the defined fitness function. The fitness values represent the agents' masses, with better-performing agents having higher masses [139]

5. Calculate gravitational force and acceleration: Calculate the gravitational force $F_i$ acting on each agent *i* based on the agent masses and the inverse square law of gravity:

$$F_i = G \,.\frac{m_i m_j}{r_{ij}^2}$$

Where $m_i$ and $m_j$ are the masses of agents $i$ and $j$, and $r_{ij}$ is the distance between them.

Calculate the acceleration of each agent $a_i$ under the influence of the gravitational forces from other agents [139].

$$a_i = \frac{F_i}{m_i}$$

6. Update the agents' positions: Modify the agents' positions based on their accelerations and velocities using the following equations for PID parameters $K_p$, $K_i$ and $K_d$:

$$K_{p_i}\,(t+1) = K_{p_i}(t) + v_{p_i}(t+1).\Delta t$$
$$K_{i_i}\,(t+1) = K_{i_i}(t) + v_{i_i}(t+1).\Delta t$$
$$K_{d_i}\,(t+1) = K_{d_i}(t) + v_{d_i}(t+1).\Delta t$$

Where $v_{p_i}(t)$, $v_{i_i}(t)$, and $v_{d_i}(t)$ are the velocities of agent $i$ for PID parameters, at time t.

7. Termination criteria: Repeat steps 4-6 until a predefined termination criterion is met, such as reaching a maximum number of iterations or a satisfactory level of ITAE reduction.

8. Extract the optimal solution: Identify the agent with the best fitness value (lowest ITAE) and extract its PID parameters $K_{p_{opt}}$, $K_{i_{opt}}$, and $K_{d_{opt}}$ from the search space. These values represent the optimal controller parameters for minimizing ITAE in the Robogymnast system.

9. Implement the optimal controller parameters obtained: Apply the optimal PID controller parameters ($K_{p_{opt}}$, $K_{i_{opt}}$, and $K_{d_{opt}}$) to the Robogymnast system to achieve improved performance with minimized ITAE value.

It is important to note that the (GSA) has been modified and enhanced by various researchers to improve its performance for specific applications. Here is Table 5.4 of the most common GSA parameters along with their descriptions [139].

Table 5.4 GSA parameters

| Parameter | Description |
|---|---|
| N | Population size (number of agents) |
| $\mathbf{G_0}$ | Initial gravitational constant |
| α | Gravitational constant decay factor (decreases gravitational constant over time) |
| $K_{best}$ | Number of best agents that exert a gravitational force on other agents |
| ε | Small constant used to avoid division by zero in the gravitational force calculations |
| iteration_max | Maximum number of iterations (stopping criterion) |

### 5.4.3 GSA results

This work was implemented in MATLAB version (2022a) installed on Intel (R) Core (TM) i5-8500 CPU @ 3.00GHz computer, the GSA code was programmed in (.m files), and the Robogymnast system model was simulated in the MATLAB Simulink environment (the .m

MATLAB file (the code of the algorithm) is calling the Simulink file where the acrobot system model is simulated.

Figure 5.6 displays the convergence graph of the GSA which provides a visual representation of how the algorithm's performance evolves over iterations. The convergence graph illustrates the progress of the algorithm over a series of iterations. The x-axis shows the number of iterations, while the y-axis represents the best objective function value achieved at each iteration. The figure illustrates that the GSA optimisation minimise the ITAE value over the iteration. Table 5.5 shows some parameter values of the algorithm.

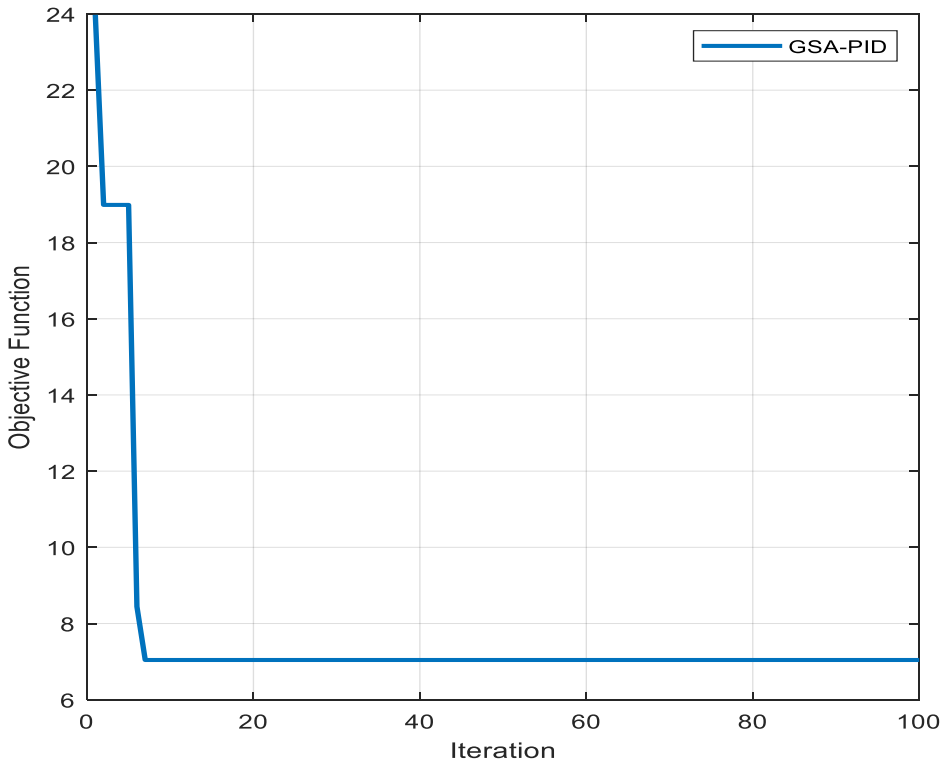


Figure 5.6 The Convergence graph of GSA

Table 5.5 GSA parameters

| No. Iteration | No. Population |
|---|---|
| 100 | 50 |

Figure 5.7 displays the optimised PID-GSA system response for joints 1, 2 and 3 of the Robogymnast. After which Table 5.6 presents the performance of PID-GSA controller applied to the three joints (Theta1, Theta2 and Theta3).

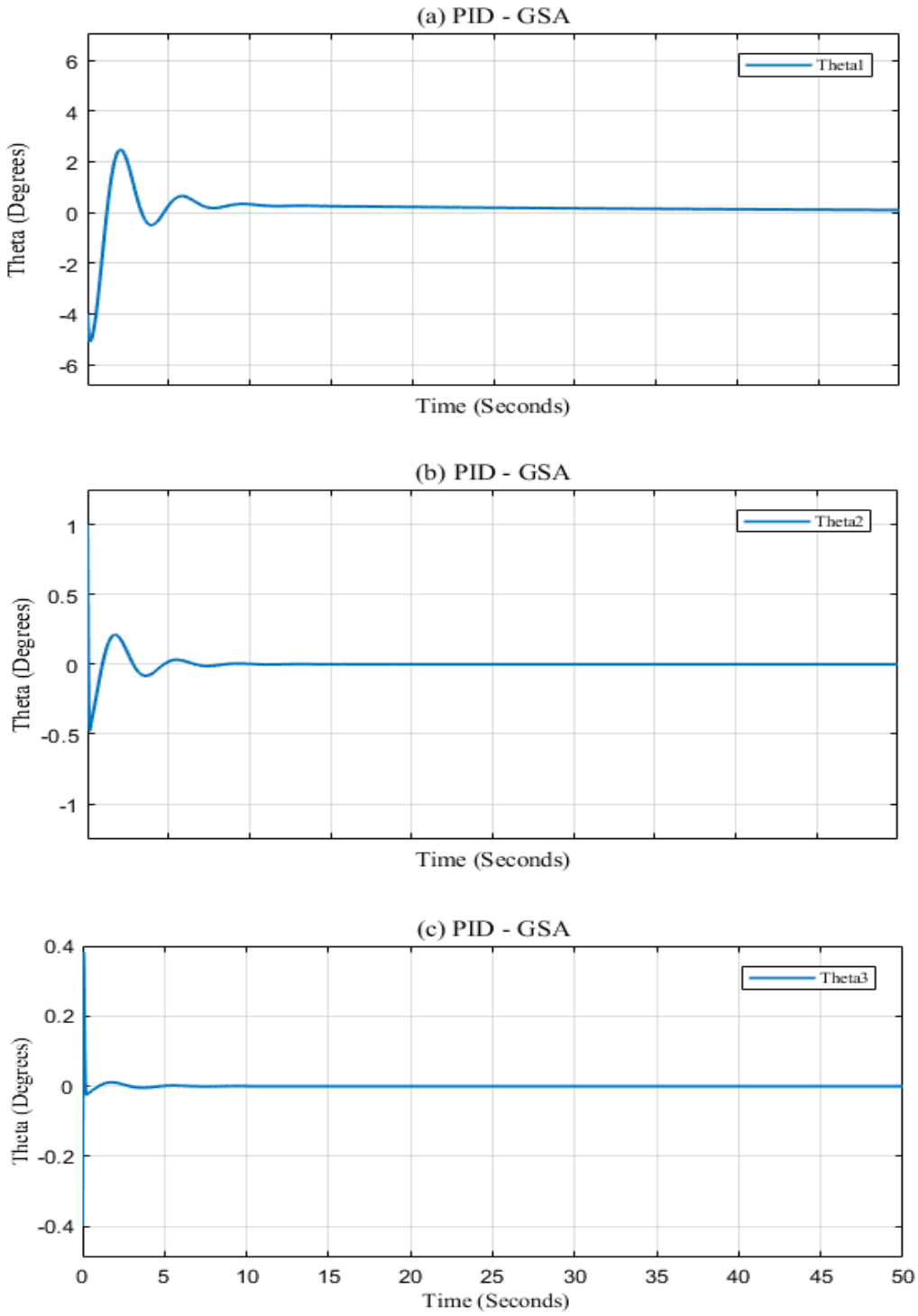


Figure 5.7 (a) Optimised PID-GSA system response for 1st joint; (b) PID-GSA system response for 2nd joint; (c) PID-GSA system response for 3rd joint of Robogymnast

Table 5.6 PID-GSA performance

| Symbol | Controller | Overshoot $O_{sh}\ (pu)$ | Undershoot $U_{sh}\ (pu)$ | Settling time $T_s\ (s)$ | Rising time $T_r\ (s)$ | ITAE |
|---|---|---|---|---|---|---|
| $\theta_1$ | PID-GSA | 2.478 | -5.107 | 8.250 | 0.016 | 72.60 |
| $\theta_2$ | PID-GSA | 1 | -0.480 | 7.076 | 0.038 | 0.392 |
| $\theta_3$ | PID-GSA | 0.383 | -0.400 | 2.250 | 0.016 | 0.026 |

Table 5.6 presents that for Theta 1, the PID-GSA controller exhibited an overshoot of 2.478 *p.u*, an undershoot of -5.107 *p.u*, a settling time of 8.250 seconds, a rising time of 0.0168 seconds, and an ITAE of 72.60. In the case of 2nd joint, the controller displayed an overshoot of 1 *p.u*, an undershoot of -0.480 *p.u*, a settling time of 7.076 seconds, a rising time of 0.0380 seconds, and an ITAE of 0.392. Lastly, for the 3rd joint, the optimized PID performance demonstrated an overshoot of 0.3836 *p.u*, an undershoot of -0.400 *p.u*, a settling time of 2.250 seconds, a rising time of 0.016 seconds, and an ITAE of 0.026.

The results indicate that the PID-GSA controller's performance varied across the three joints. In terms of overshoot, joint 1 exhibited the highest value, while joint 3 had the lowest undershoot value. Furthermore, Theta1 had the longest settling time, whereas Theta3 showed the shortest value of settling time. regarding the riding time, Thetas 1 and 3 have the same value which is less that Theta 2. The ITAE values indicate the accuracy of the system's response. The lower the ITAE, the more accurately the system tracks the desired output. In this matter, Theta 3 has the lowest ITAE value 0.026, signifying the highest performance in terms of tracking accuracy.

## 5.5 Optimisation comparison

In this subsection a comparison between GSA and ACO algorithm is demonstrated. The optimised response for each joint is shown in Figure 5.8.

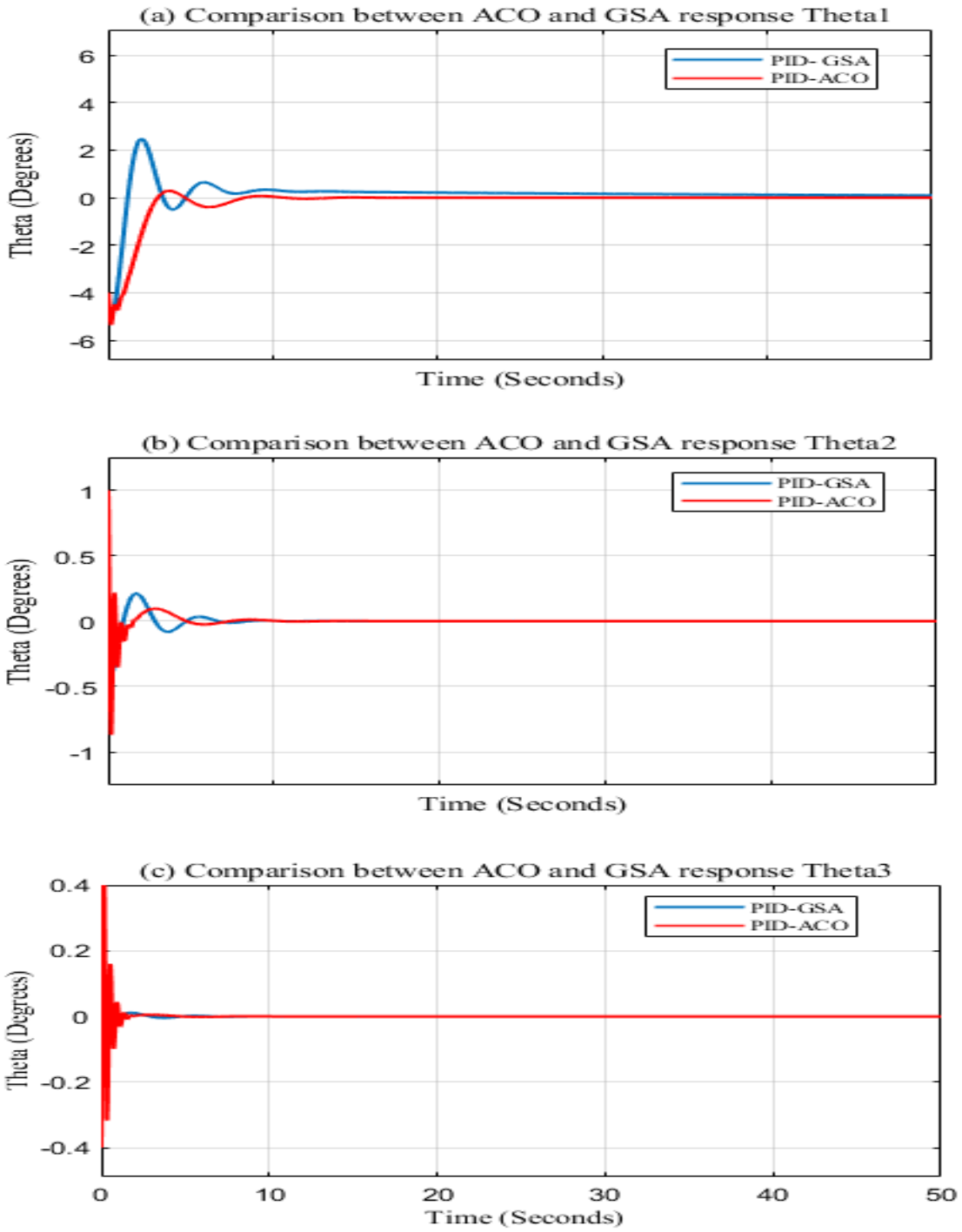


Figure 5.8 (a) Optimised PID for ACO and GSA system response for 1st joint; (b) The ACO and GSA system response for 2nd joint; (c) The ACO and GSA system response for 3rd joint of Robogymnast.

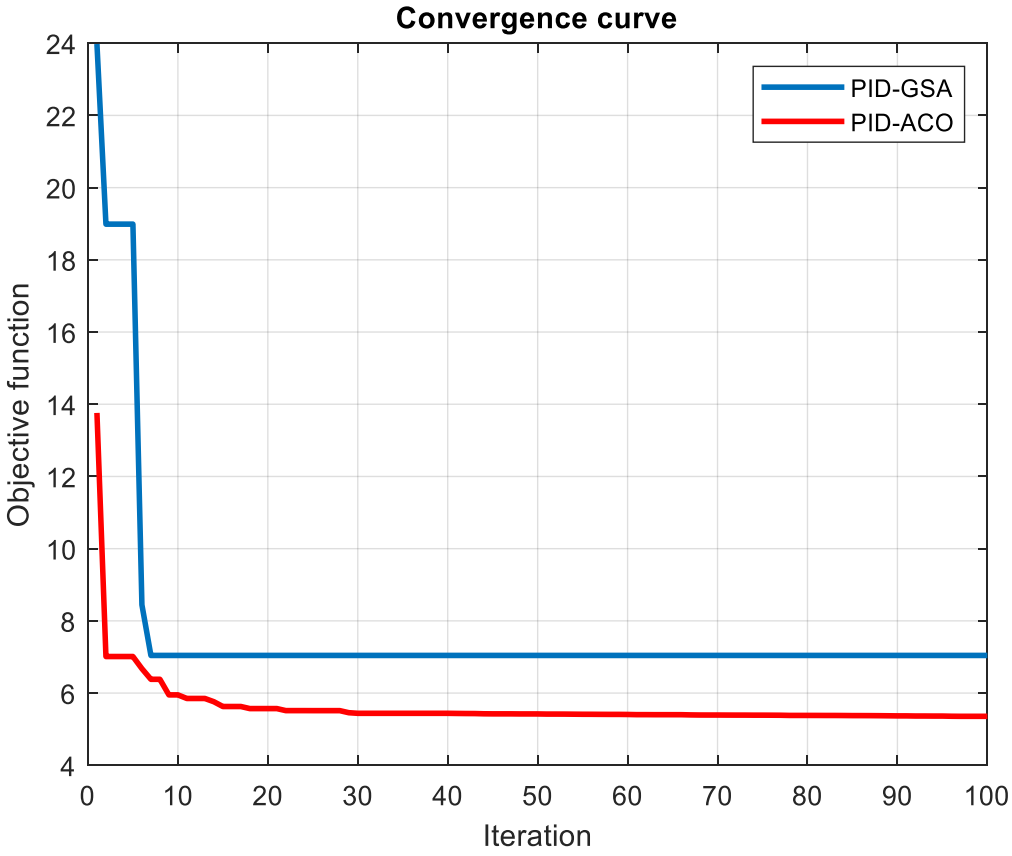


Figure 5.9 ACO and GSA Convergence graph

Ant Colony Optimization (ACO) excels in global search due to its distributed computation, which uses multiple agents to concurrently explore the search space [137]. The algorithm maintains a balance between exploration and exploitation through a probabilistic transition rule, aiding in its global search capabilities. The fading of pheromone trails over time encourages new path exploration and, along with the algorithm's memory of the best solution, guides the search process. The algorithm's inherent parallelism and scalability allow it to tackle large, complex problems effectively [72] [75]. The convergence quality of any algorithm is determined by how fast the algorithm reaches the optimal solution and how accurately the solution is found. The full outcomes of the comparison between ACO and GSA are presented in Table 5.7. The results from the comparison highlight the performance of the ACO algorithm and GSA when applied to the Robogymnast controller for optimizing the values of Theta 1, 2,

and 3. The efficacy of these algorithms was determined by their ability to find the optimal solutions for these parameters, which are crucial for achieving ideal performance in the Robogymnast system.

Table 5.7 Comparison between ACO and GSA response

| Symbol | Controller | $O_{sh}$ ($pu$) | $U_{sh}$ ($pu$) | $T_s$ (s) | $T_r$ (s) | ITAE |
|---|---|---|---|---|---|---|
| $\theta_1$ | PID- ACO | 0.290 | -5.360 | 7.734 | 0.182 | 11.650 |
| | PID- GSA | 2.478 | -5.107 | 8.250 | 0.016 | 72.60 |
| $\theta_2$ | PID- ACO | 1 | -0.868 | 6.404 | 0.041 | 0.310 |
| | PID- GSA | 1 | -0.480 | 7.076 | 0.038 | 0.392 |
| $\theta_3$ | PID- ACO | 0.300 | -0.400 | 1.363 | 0.033 | 0.020 |
| | PID- GSA | 0.383 | -0.400 | 2.250 | 0.016 | 0.026 |

ACO algorithm shows remarkable results, demonstrating its robustness in exploring the solution space. The optimization of the Theta parameters by ACO consistently converged towards the optimal solution, suggesting that the algorithm is capable of effectively managing the complexity of the problem space. The ACO algorithm's inherent ability to balance exploration and exploitation might have contributed to this outcome, as it allows for a broad search of the solution space while also exploiting promising areas. In this work, the main factor considered to evaluate the controller's performance is ITAE. Subsequently, a lower value of ITAE indicates better performance of the optimisation technique.

Table 5.7 presents a comparative analysis of two algorithms ACO and GSA, both used in a PID controller setting to optimize the parameters of Thetas 1, 2 and 3. The performance is evaluated based on overshoot, undershoot, settling time, rise time, and ITAE. In the case of Theta 1, the PID-ACO controller outperforms PID-GSA, as evident from the lower values of

overshoot, settling time, and ITAE. Notably, the PID-ACO controller has a significantly lower ITAE 11.650 compared with 72.60 for GSA, indicating superior overall performance. The optimization of Theta 2 reveals a closer competition between the two algorithms. Both controllers produce a similar rise time, but PID-ACO still manages to output a lower overshoot, settling time, and ITAE. The negligible difference in undershoot suggests that both controllers perform equally in minimizing the system undershoot. When optimizing Theta 3, the PID-ACO controller again outperforms PID-GSA. Although the overshoot and undershoot are similar, PID-ACO demonstrates a faster settling time and slightly lower ITAE. However, PID-GSA exhibits a faster rise time.

In summary, across all three parameters, PID-ACO consistently demonstrates superior performance, particularly with a lower ITAE, suggesting it achieves a more optimal balance of error over time. While both algorithms bring unique strengths, the ACO-based PID controller appears to offer a more robust and balanced performance for this application in the Robogymnast controller.

## 5.6 Summary

This chapter provides a detailed description of various algorithms that will be used to optimise the proposed acrobot system, including an overview, mechanism, and applications of the ACO and GSA algorithms. It is clear from this chapter that both algorithms have been successfully implemented across different application domains. The benefits of these algorithms make them appealing to researchers, who utilize them to solve a broad range of optimization problems. Therefore, in this work, the ACO algorithm is proposed to fine-tune the parameters of the optimal PID controller, which is suggested to optimize the triple-link

robotic system. Furthermore, the ACO algorithm is also employed to compare the response of the system.

To sum up, the results presented suggest that the PID-ACO controller proves successful in proposing the system with various values of the joints (thetas) for analysis. The incorporation of the ACO optimisation algorithm with the PID controller contributes to desirable performance regarding overshoot, undershoot, rising time, settling time, and crucially, the ITAE parameter which is a performance measure used to evaluate the behavior of a control system. In conclusion, the PID controller and ACO algorithm show efficacy when contrasted with the PID-GSA optimization technique. It is evident that the ITAE values are reduced with the PID-ACO controller. The forthcoming chapter will delve into the practical optimized outcomes of the Robogymnast system, comparing them in depth with a MATLAB/Simscape simulation model.

# Chapter 6: Simulation and Experimental Results

## 6.1 Introduction

The Robogymnast was selected to embody a complicated, underactuated multi-link mechanical system, with the intention of assessing and comparing control systems that utilize various methodologies [149]. The design of a control system with underactuated comes with its own set of difficulties, primarily because complete state feedback linearization around a fixed equilibrium point is often unattainable for such mechanisms. Additionally, these types of systems frequently do not offer small-time local controllability (STLC) [150]. This has spurred substantial research interest in the development of underactuated systems within the disciplines of control engineering and robotics. acrobot systems incorporate an element that swings freely from a fixed point, held in place by gravitational forces. This type of mechanism is often used in work involving the regulation of movement, and it can be used to demonstrate both hybrid and chaotic systems [11]. The problem of balancing triple-inverted pendulum systems presents a significant challenge in robotics, primarily due to their structural resemblance and balance factors similar to that of the human body. The acrobat robotic system that emulates human acrobatic activity and takes on the form of an inverted pendulum, is intentionally designed with inherent instability and underactuation. This makes the robot an ideal subject for both theoretical and practical work concerning non-linear controls. The acrobot controllers

examined were primarily based on state variable feedback, as well as proportional-integral-derivative and linear-quadratic regulation methodologies [151][152].

## 6.2 Design of Robogymnast model

The Robogymnast model signifies an intricate robotic system engineered to autonomously execute a range of gymnastic activities. The construction of this model entails the incorporation of diverse mechanical, electrical, and control elements. Mechanical constituents consist of the body, appendages, joints, and actuation devices. Simultaneously, electrical parts encompass sensors, motors, and control units. The control mechanism bears the responsibility of managing sensory data, creating control directives, and supervising the motion of the robot [31]. The overall structure of the Robogymnast model is constructed based on the principles of mechanics and robotics. The model aims to mimic the capabilities and movements of a human gymnast, all while incorporating advanced robotic technologies. The model can perform a wide range of gymnastic maneuvers, including motion planning and swinging. To ensure the robust performance of the Robogymnast model, several design factors need to be considered. These include the balance and weight of the robot, the precision of the sensors, the range of motion of the joints, and the responsiveness of the control system. By optimizing these design considerations, the Robogymnast model is capable of performing complex gymnastic tasks with impressive efficiency and accuracy [153].

This section of the study consists of two primary elements. Initially, a Simscape simulation model is devised to mimic the movement of the system. Following that, the findings derived from the optimized hardware design are exhibited and scrutinized to confirm the swinging motion of the system.

### 6.2.1 Simscape model design

Simscape is a part of the MATLAB and Simulink product families, specifically designed for modelling, simulating, and analysing dynamic systems. It provides a platform for engineers to represent their physical systems as schematic diagrams and use these for simulation and analysis purposes. It offers multiple domains, such as mechanical, electrical, hydraulic, thermal, etc., enabling the user to model real-world physical systems. These different domain systems can be coupled, allowing users to create multi-domain models. Simscape allows engineers to employ fundamental physical principles when creating their models, such as conservation of energy or Newton's laws of motion, which makes the tool valuable for tasks like system-level understanding, component sizing, control design, and testing [154].

This subsection delves into the specifics of modelling and controlling a humanoid robot using MATLAB/Simscape. The robot's design aim was to emulate human-like gymnastic movements. Consequently, the robot is equipped with joints mirroring those present in the human anatomy. The upper link represents the arms, the middle link symbolizes the torso, while the lower link is analogous to the legs.

In reality, the acrobot experiences the force of gravitational acceleration (g). It comprises three links and three-point masses. Figure 6.1 presents the layout of the acrobot, offering a visual depiction of the robot's design. The application of the Simscape MATLAB model in crafting and controlling the robot has demonstrated considerable success in replicating human-like motion and actions. This makes it a precious resource for prospective investigations and advancements in this area.

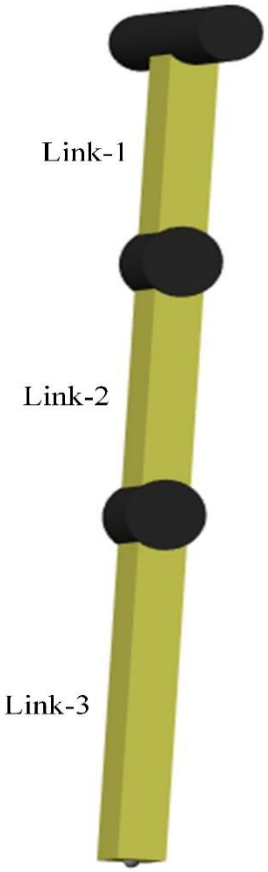


Figure 6.1 Triple-link robotic MATLAB Model

Furthermore, Table 6.1 illustrates the specific parameters of the actual Robogymnast system. These parameters are implemented in the simulation model to validate the real system. The table provides essential details about the physical system, including the lengths and masses of the links, among other factors. Such parameters are crucial for accurately modelling the dynamics of the Robogymnast system and for comparing the model's performance against real-world data.

Table 6.1 Robogymnast Simscape parameters

| **Parameters** | **Link 1** | **Link 2** | **Link 3** |
|---|---|---|---|
| Length | $l_1$= 0.16 m | $l_2$= 0.180 m | $l_3$ = 0.245 m |
| Mass | $m_1$ = 1.2 kg | $m_2$= 1.2 kg | $m_3$ = 0.5 kg |
| Theta | $\theta_1$ = 0 | $\theta_2$ = 0 | $\theta_3$ = 0 |

The parameters outlined in Table 6.1 are crucial in validating the performance of the simulation model against real-world data. By accurately simulating the dynamics of the Robogymnast system, designers can use the simulation model to predict the behaviour of the actual system under different conditions. These predictions can then be compared with the actual data collected from the real system to determine the accuracy of the simulation model. This iterative process is essential in ensuring that the simulation model can effectively mimic the behaviour of the physical system, thus enabling designers to test and improve the system's performance.

Figure 6.2 illustrates the design of a triple-link robotic system created to simulate the Robogymnast. This design was developed using the MATLAB/Multibody toolbox. The design process involved the use of the MATLAB/Multibody toolbox, which facilitated the development of an exceptionally accurate simulation model. The Multibody toolbox figure offered a three pair-blocks, each pair-block represents the joint and link (Theta1, $L_1$), (Theta2, $L_2$) and (Theta3, $L_3$) of the system respectively. The modelling tools and features made it possible to create a realistic and dynamic simulation of the robotic system.

MATLAB's Multibody tool refers to Simscape Multibody, a part of the MATLAB/Simscape family of products. Simscape Multibody extends Simscape with the ability to easily model rigid body mechanical systems in 2D and 3D. With Simscape Multibody, you can model mechanical systems, such as vehicle suspension systems, robot manipulators, and aircraft landing gear. This tool allows for the formulation and integration of equations of motion for mechanical systems directly within the Simulink environment, and it simplifies the process of creating complex models of multibody systems [155]. The MATLAB/Multibody model of the Robogymnast is shown in Figure 6.2. In this model, the initial values for the Robogymnast joint positions are set to $\theta=0°$.

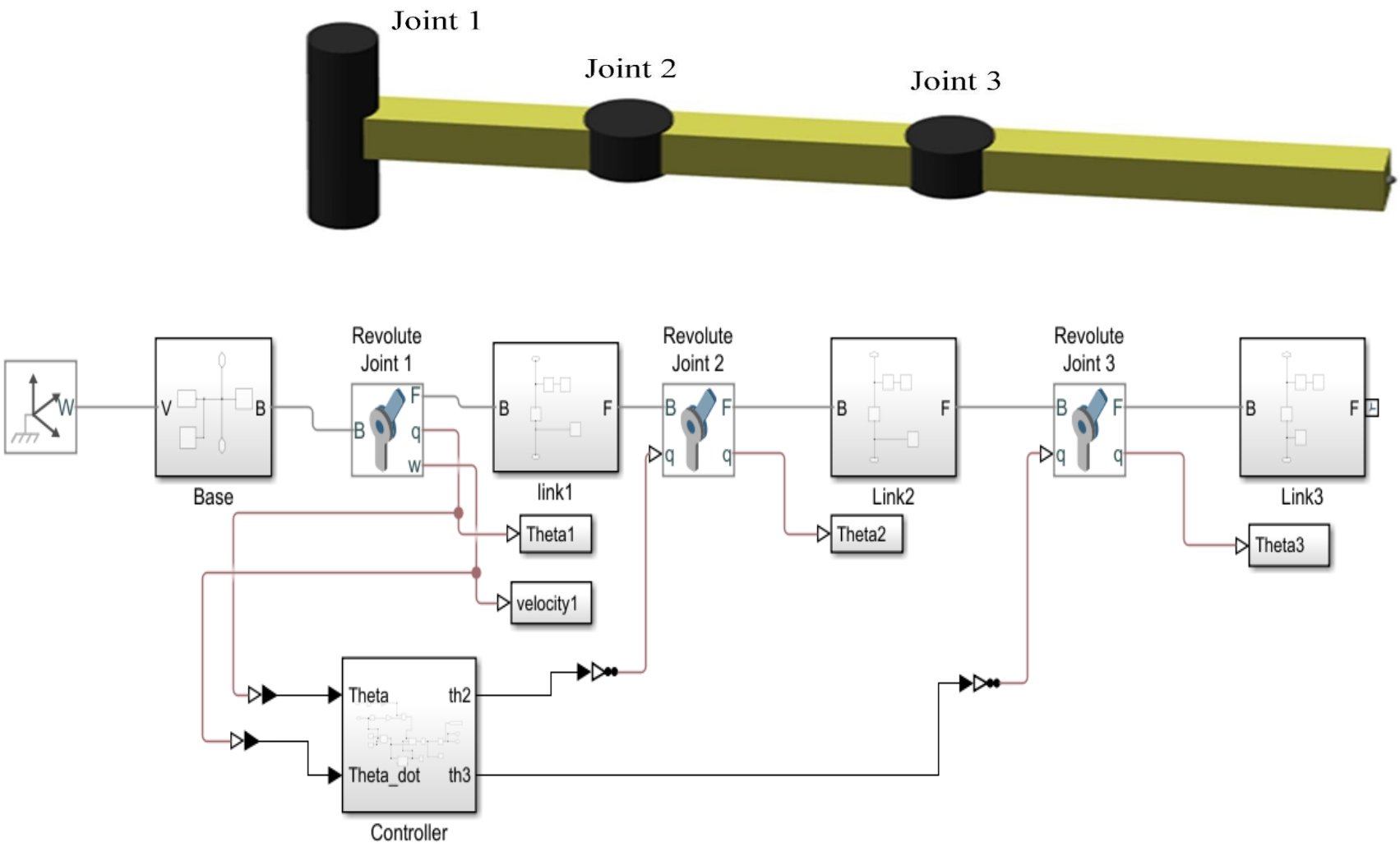


Figure 6.2 The design of the triple link robotic system using MATLAB/Simscape

The parameters used in the MATLAB/Multibody model are precisely identical to those of the actual physical system. This aids in achieving a high degree of correlation between the simulation results and the real-world performance of the Robogymnast.

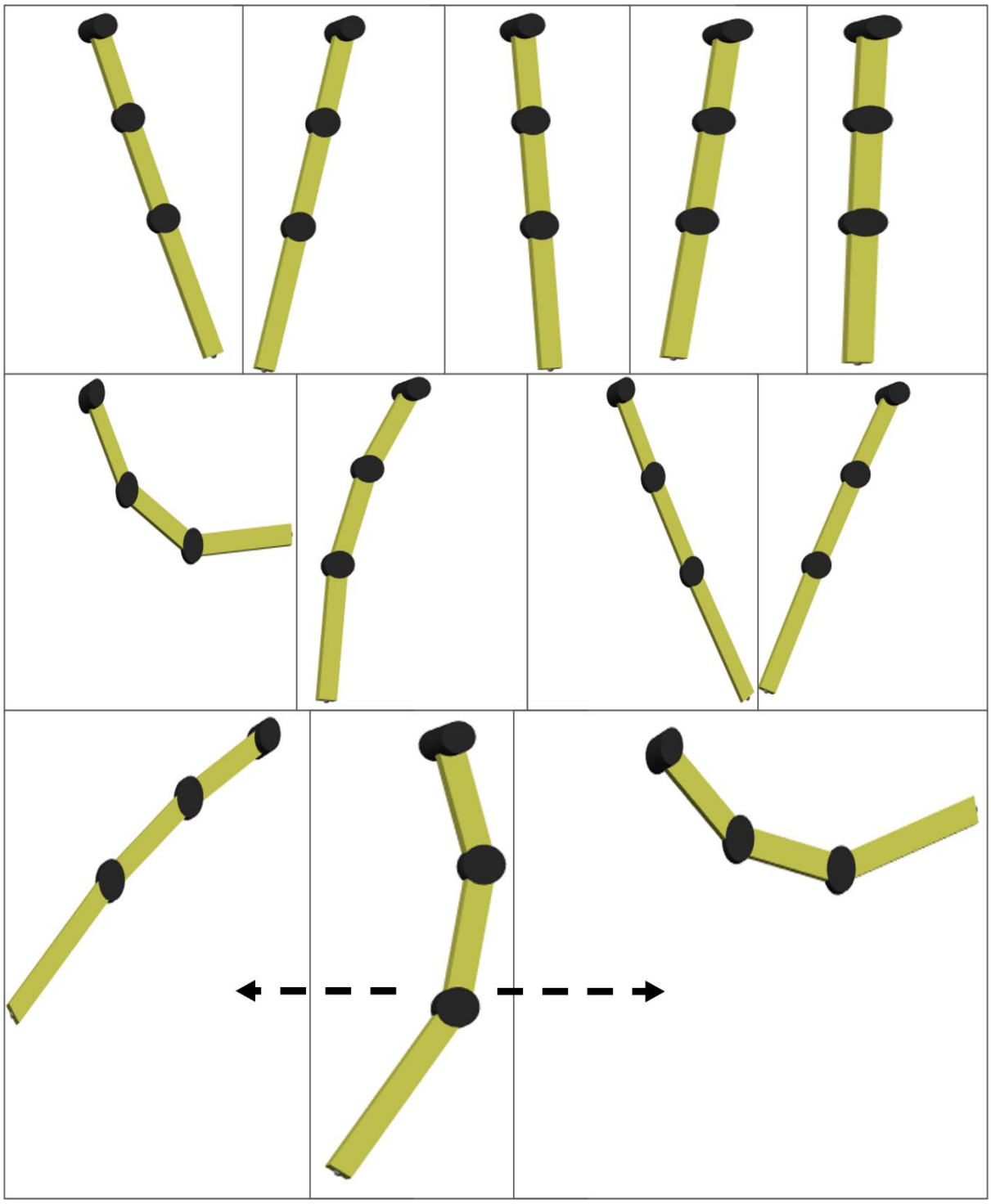

Figure 6.3 Slides show MATLAB Simulink Robogymnast Motion

Figure 6.3 illustrates various motion stages of the triple-link robotic system simulation model within the MATLAB/Simscape environment, displayed concurrently. This visual depiction assists in understanding the intricate movements and dynamic transitions of the system throughout different stages of its operation.

### 6.2.2 Simulation outcomes

MATLAB Simulink and Simscape provide an invaluable platform for the simulation and development of a triple-link robotic system. Using these tools, engineers can construct a detailed model of the robot's structure and its control systems. Simscape facilitates the creation of physical models by providing a variety of predefined components, such as bodies, joints, and actuators, that can be connected to form a multi-body dynamic system. The physical connections correspond directly to the actual physical structure of the robot, providing an intuitive and realistic modelling framework. Once the physical model is created in Simscape, it can be integrated with a control system designed in Simulink [156]. This control system may include various algorithms and strategies to achieve desired performance, such as performing certain maneuvers. By connecting the control system to the Simscape model, the engineer can simulate the closed-loop system and see how it behaves under different conditions. These simulations can be run iteratively, allowing the engineer to tweak parameters, adjust the control strategy, or modify the robot's design, and then quickly see the effects of these changes. It provides an efficient and cost-effective means to experiment with different designs and control strategies before implementing them on the actual robot [157].

The simulation model was executed with a sampling time of 1ms. Figure 6.4 illustrates the output response signal from the triple-link robotic system, also known as Robogymnast, which was modeled and simulated using MATLAB Simulink/Simscape. The operations of this

system are managed by a controller, which is optimized through the Ant Colony Optimization Proportional-Integral-Derivative (ACO-PID) method.

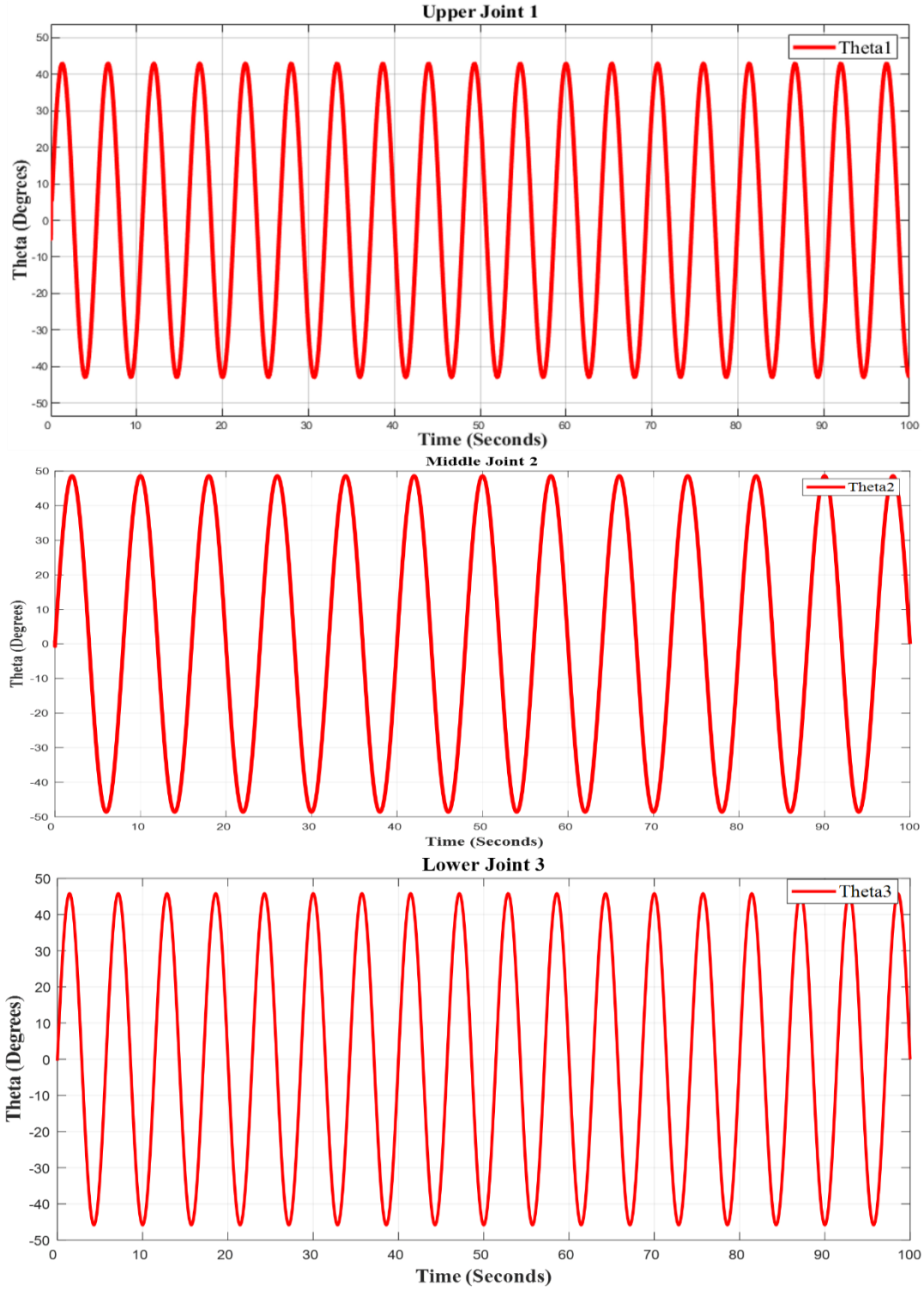


Figure 6.4 The simulation output of triple link robotic system using MATLAB Simscape

## 6.3 Practical results

In the practical results section, the real-world performance of the triple-link robotic system is thoroughly assessed. This includes a discussion and analysis of data collected from the physical implementation of the system. Any challenges encountered during the testing phase, such as mechanical failures or control system errors, are reported and examined. Ultimately, the practical results offer valuable insights into the system's actual performance, laying the foundation for future improvements and optimizations of the robotic system.

The work is applied to a triple-link Robogymnast mechanism, designed to emulate the gymnastic action of swinging to freely rotate over a high bar. Figure 6.5 presents a diagram detailing the components of the operating system, which is powered by two stepper motors. Each motor is managed by a stepper driver control to ensure smooth movement. The control system programming is conducted through the STM32 microcontroller, utilizing the C++ language to facilitate communication between the robotic system, the control system, and the PC. Every link is equipped with its own sensor; link 1 is connected to a rotary encoder, while links 2 and 3 are connected to potentiometers with high precision resolution, respectively. This setup enables the detection of absolute angles at every position.

### 6.3.1 Motion data

The Python program is used to manage the motion data from a triple-link Robogymnast, utilizing suitable sensors or motion capture mechanisms. These tools are capable of monitoring the real-time position and alignment of the robot's links, and the obtained data is subsequently processed through the Python program.

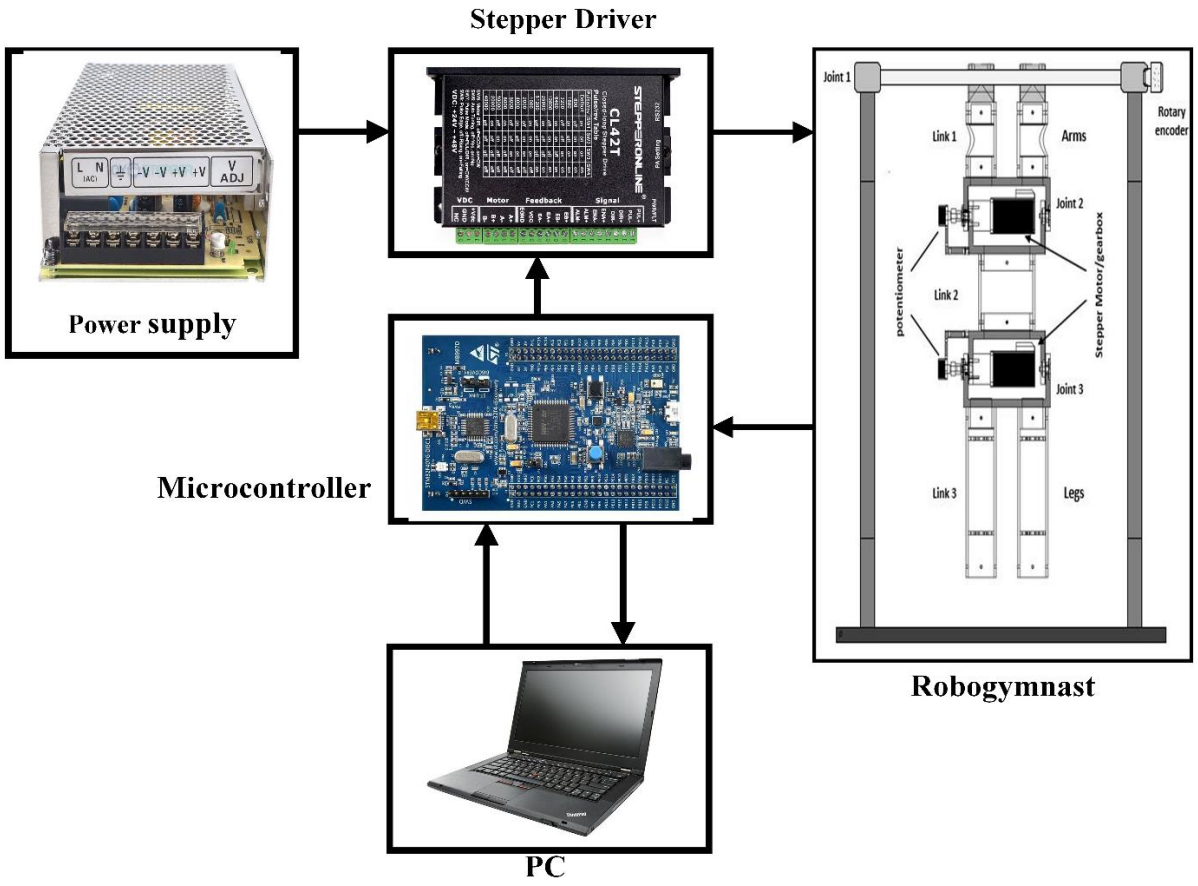


Figure 6.5 Robogymnast operation System

Data collection is meticulously handled through a Python application, underlining the critical role of precise data acquisition and examination for yielding trustworthy simulation or visualization outcomes. The STM32 microcontroller, programmed with the C++ language, manages the control system. Individual sensors are assigned to each link for efficient tracking: a rotary encoder is linked to the first, while potentiometers are connected to the second and third links, enabling the detection of absolute angles in every position. This emphasizes the requirement of a well-constructed Python program to ensure accurate data collection and processing, thereby ensuring superior data quality.

In this part, the spotlight is on the optimized results garnered through the deployment of the Proportional-Integral-Derivative (PID) control method, which was fine-tuned using Ant

Colony Optimization (ACO). The objective was to enhance the motion control of a multi-link system. Figure 6.6 shows the outcomes of the practical motion of the triple-link system.

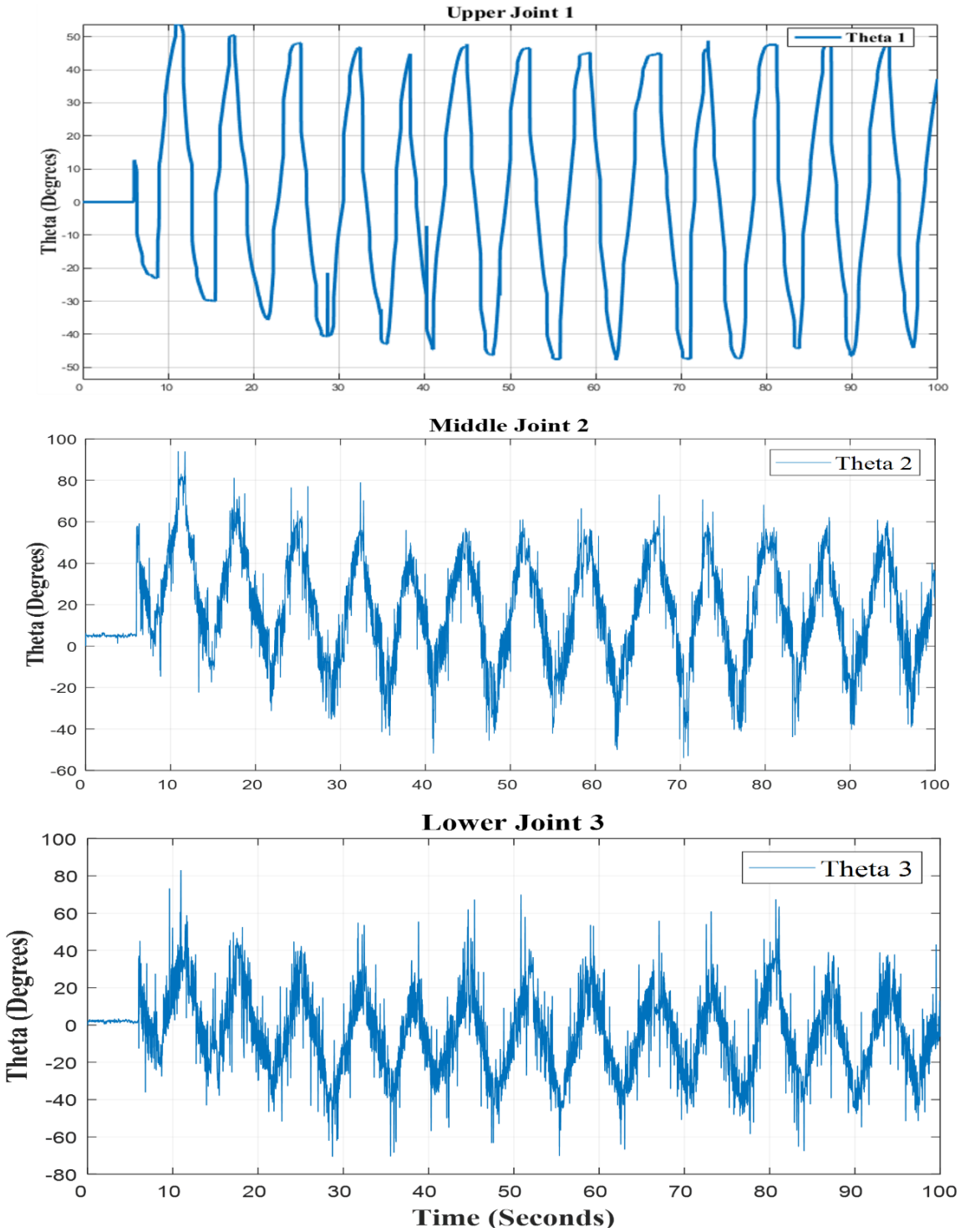


Figure 6.6 The results of practical motion of the triple link system (Robogymnast)

In order to determine the most effective motion parameters for each link, the enhanced parameters were implemented on the system using an STM32 microcontroller. This required coding the microcontroller with the optimized parameters and utilizing it to regulate the system's movement. The STM32 microcontroller is a robust and flexible platform that is extensively employed in control systems and applications related to robotics. The combination of PID and ACO facilitated the optimization of the system's control parameters. The starting parameters for motion served as the foundation, and PID was employed to craft the optimal controller for the system. Subsequently, ACO was utilized to refine the controller's parameters, aiming to achieve the best possible performance. Figure 6.7 displays some movement stages of Robogymnast's practical aspect in Cardiff university laboratories.

Figure 6.7 Robogymnast Motion

In summary, the employment of PID with ACO tuning served as an effective method for optimizing the control parameters of a multi-link system. The STM32 microcontroller played a crucial role in implementing the refined parameters on the system and assessing the resulting performance enhancements. This methodology can be utilized broadly across various control systems and robotic applications, leading to improved performance and yielding optimized results.

Table 6.2 Robogymnast Motion Results

| Symbol | Parameters | Average (Degrees) | Max Point (Degrees) |
|---|---|---|---|
| $\theta 1$ | Joint 1 (free rotate) upper | (-42) to 45 | (−48) to 55 |
| $\theta 2$ | Joint 2 (Motor 1) middle | (−35) to 60 | (−58) to 94 |
| $\theta 3$ | Joint 3 (Motor 2) lower | (−50) to 55 | (−71) to 82 |

Table 6.2 illustrates the physical movements of the Robogymnast, a three-link robotic system. This table utilizes three symbols, specifically $\theta_1$, $\theta_2$, and $\theta_3$, to signify the degrees of the system's angular motions. The table provides parameters that shed light on the physical characteristics of each system link, including average movement and the maximum attainable point. These parameters are essential for comprehending the conduct of this mechanical system.

Figures 6.6 and 6.7, along with Table 6.2, provide detailed information on each link's motion scope in a triple-link robotic system and outline the actuation mode utilized for each link. Notably, the first link employs a free-rotate joint, whereas motors control the second and third links. The average and peak points of each link's range of movement are provided in degrees, giving an insight into each link's flexibility and precision of movement. For instance, link 1 exhibits an average motion range from -42 to 45 degrees and a maximum motion range

from -48 to 55 degrees. This implies that joint 1 possesses flexibility and can traverse a reasonably wide motion range.

For the second joint, the average motion range is from -35 to 60 degrees, and the maximum span is from -58 to 94 degrees. This demonstrates that link 2 possesses greater flexibility than previously indicated, with an extended maximum range. This might imply that the motor managing link 2 can offer more meticulous control over its motion than what was found in the preceding joint. Lastly, joint 3 displays an average range of motion from -50 to 55 degrees and a maximum range from -71 to 82 degrees, a range that is similar to that of the previous joint.

In summary, this table imparts critical insights for comprehending the abilities and constraints of a three-link robotic system. The amplified range of motion for the 2nd and 3rd joints indicates that the system might be capable of executing more intricate and exact maneuvers. This data is beneficial in architecting and programming the system for certain applications, as well as identifying and resolving any challenges that might occur during its operation.

Figures 6.7 and 6.8 illustrate the synchronized movement of the Robogymnast system, which is a fundamental element of its operational effectiveness. These figures display the synchronized operation of the robot's three links. The seamless interplay of movements in the Robogymnast system is pivotal for its effective performance. This synchronized operation is accomplished through meticulous design and regulation of the robot's joints and actuators. The side view of the Robogymnast's motion gives a clear representation of the robotic system's operational intricacies. This perspective enables one to visualize how the different links interact and work together to achieve the desired movement.

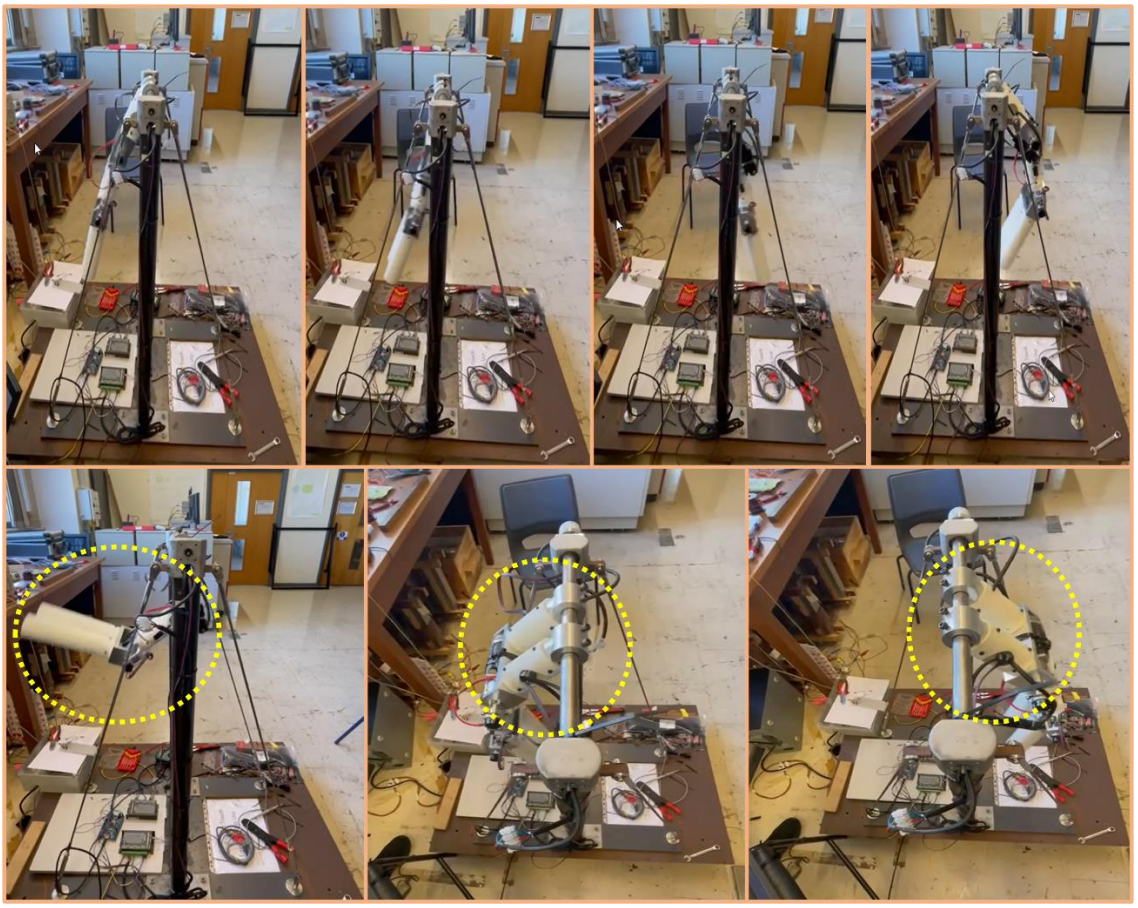

Figure 6.8 Side view of the Robogymnast motion

The previous figures serve as instrumental tools in the display of the Robogymnast system's synchronized motion. These visual depictions facilitate a deeper understanding of the robot's movements, enabling researchers and developers to extract performance insights and pinpoint potential enhancement areas. The Robogymnast system's synchronized motion, a critical operational aspect, necessitates meticulous design and control to guarantee precise, efficient movement.

Table 6.3 provides detailed information on the dynamics of a triple-link robotic system in motion. The table presents positional and angular coordinates of the system at various time points. This data can be used to analyze the system's motion, identifying patterns or trends that can aid in optimizing the robot's design and performance. It's important to note that the data

shown represents a random sampling from a larger dataset. As such, the information in the table reflects only a portion of the complete dataset. Despite this limitation, the table provides significant insights into the dynamic behaviour of the robotic system during motion.

Table 6.3 Random data of triple link movement represented in degrees.

| No. | Time (s) | Joint1 ($\theta_1$) (Degree) | Joint2 ($\theta_2$) (Degree) | Joint3 ($\theta_3$) (Degree) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 2 | 3 | 12.53 | 48.07 | 36.99 |
| 3 | 6 | -14.41 | 27.29 | 10.11 |
| 4 | 9 | -20.65 | 26.10 | -15.24 |
| 5 | 12 | 18.86 | 24.25 | 2.79 |
| 6 | 15 | 34.30 | 49.26 | 5.45 |
| 7 | 18 | -18.89 | 31.89 | -13.35 |
| 8 | 21 | 15.38 | 16.43 | 10.33 |
| 9 | 24 | 40.25 | 26.46 | 30.06 |
| 10 | 27 | -29.23 | 11.77 | -22.51 |
| 12 | 30 | 7.70 | 23.53 | -16.27 |
| 13 | 33 | 49.74 | 64.13 | 20.25 |
| 14 | 36 | -11.98 | 31.35 | -7.76 |
| 15 | 39 | -19.89 | 23.77 | -12.15 |
| 16 | 42 | -35.18 | -9.71 | -27.60 |
| 17 | 45 | 5.36 | 15.35 | -15.89 |
| 18 | 48 | 41.74 | 56.48 | 44.64 |
| 19 | 51 | -11.04 | 27.59 | -8.25 |
| 20 | 54 | -46.28 | -31.92 | -27.98 |
| 21 | 57 | 47.81 | 46.16 | 26.97 |
| 22 | 60 | 25.42 | 21.68 | -5.27 |

## 6.4 Comparison of Simulation and Practical results for Robogymnast

In the comparison section, the performance of the triple-link robotic system, as simulated using MATLAB Simulink/Simscape, is compared against the real-world experimental data. This comparison provides a tangible assessment of how accurately the simulation mimics real-world behaviour and performance. The insights gleaned from this comparative study ultimately guide the enhancement of the simulation model, contributing to more accurate predictions and efficient designs in future iterations of the robotic system.

Figure 6.9 presents alongside analysis of the MATLAB Simulink model and the real-world performance of the Robogymnast across the three joints $\theta_1$, $\theta_2$ and $\theta_3$. Each individual subplot showcases a different joint comparison. Subplot (a) shows a parallel between the Simulink model and the real system for $\theta_1$. The red line represents the output signal from the Simulink model, while the blue line corresponds to the real system's output. The plot reveals a close approximation between the simulated and real-time data, with only slight disparities between the two. This signifies that the Simulink model is effective in predicting the real system's behavior for $\theta_1$.

Subplot (b) presents a comparison between the simulation and the real system for $\theta_2$. The figure reveals a more significant deviation between the simulated and real-time data as compared to subplot (a). Despite this difference, the general direction of the two sets of data remains congruent, both demonstrating a similar oscillation pattern. This resemblance implies that the Simulink model presents a reliable and satisfactory depiction of the real system's behavior for $\theta_2$. Subplot (c) offers a comparative analysis between the simulated model and the real system for $\theta_3$. The graph displays a wider discrepancy between the simulated and real-time data in contrast to the findings in subplots (a) and (b). Despite the larger deviation, the

general behaviour of the two signals is still closely matched, as both lines display similar behavior patterns. This suggests that while there might be slight variances, the Simulink model continues to be a reliable predictor of the real system's behavior for $\theta_3$.

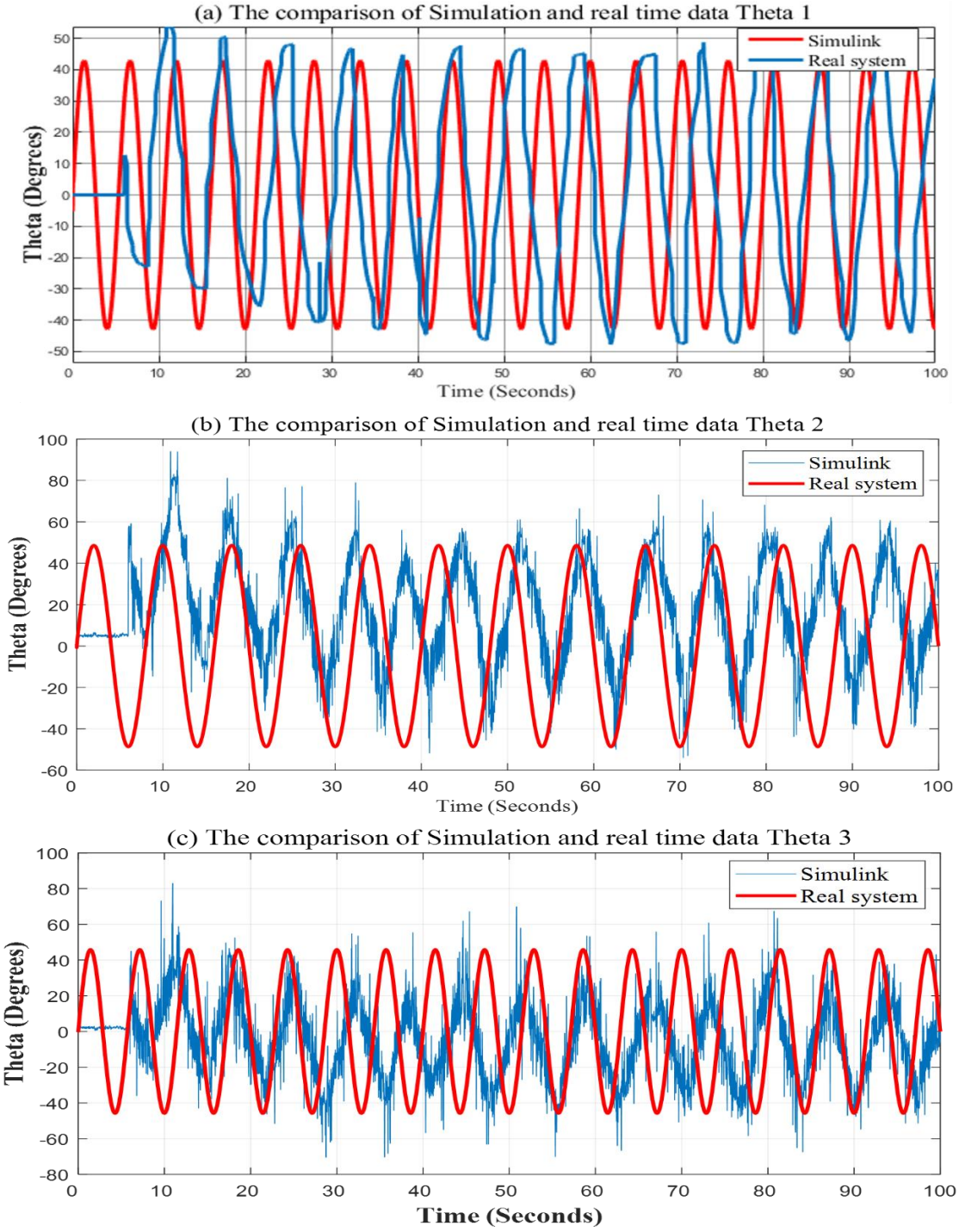


Figure 6.9 (a) Comparison of Simulation and real system for ($\theta_1$); (b) comparison of Simulation and real system of Robogymnast ($\theta_2$); (c) comparison of Simulation and real system of Robogymnast ($\theta_3$)

To conclude, Figure 6.9 demonstrates the high degree of accuracy of the Simulink model in reflecting the real system's behaviour for all three Joints $\theta_1$, $\theta_2$ and $\theta_3$, despite slight variations in precision across the Joints. The comparative analysis between the simulated and real-time data of the Robogymnast affirms that utilizing both simulation and empirical testing can provide an extensive comprehension of the robot's performance. This integrated approach can be instrumental in predicting the system's behaviour, thereby aiding in optimizing its operation. The simulation model allows for testing and analysis in a controlled environment before conducting physical tests on the real robot, which can save time and resources.

### 6.4.1 Convergence between Simulation and Experimental Outcomes of the System

The similarity percentage is a metric that quantifies the level of agreement between a MATLAB/Simulink simulation and the real-time system performance. This percentage reflects the degree of correlation between the outcomes of the simulation and the results derived from actual experiments or real-world system operations. A higher similarity percentage represents a more accurate simulation that closely mirrors the behavior of the real-time system. Conversely, a lower percentage denotes a larger discrepancy between the simulated and actual system results. Figure 6.10 illustrates the similarity percentage, elucidating the degree of congruence between the simulated and experimental results for the triple-link system.

Table 6.4 showcases the similarity percentages associated with the joint angles of the triple-link robotic system. This data can be employed to evaluate the precision of the MATLAB simulation. The similarity percentages for joints $\theta_1$ and $\theta_2$ surpass 70%, indicating that the behavior of these joints in the actual system is effectively captured by the MATLAB simulation. Consequently, the outcomes of the simulation can be regarded as both reliable and accurate for these specific components of the system.

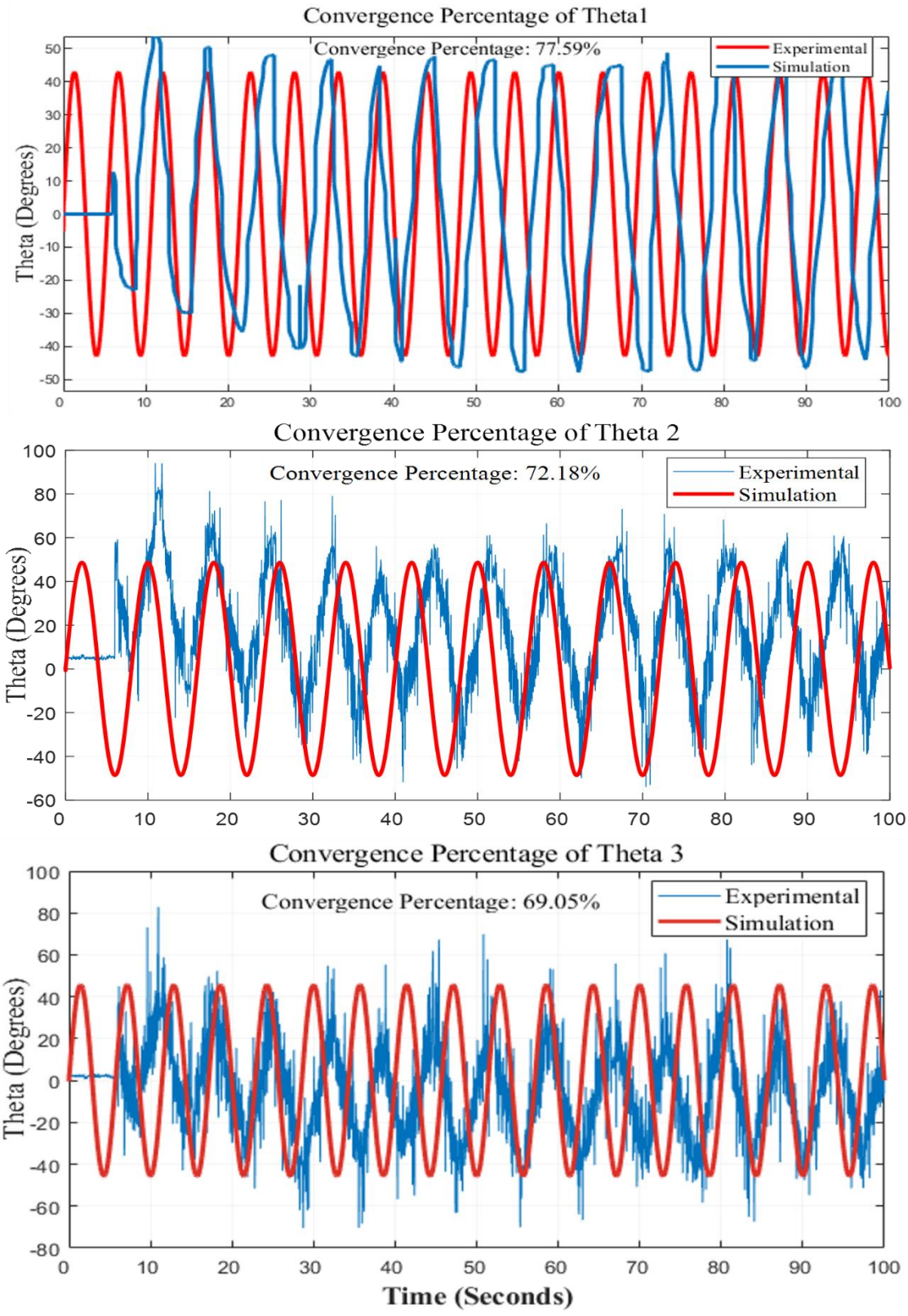


Figure 6.10 Similarity percentage between Simulation and Experimental outcomes of triple link system

Table 6.4 convergence percentages for each joint of the system

| Symbol | convergence percentage |
|---|---|
| $\boldsymbol{\theta_1}$ | 77.59% |
| $\boldsymbol{\theta_2}$ | 72.18% |
| $\boldsymbol{\theta_3}$ | 69.05% |

The similarity percentage for 3rd joint $\theta_3$ is slightly lower, recorded at 69.05%. This implies that the experimental results for this joint may be influenced by factors such as mass and actuator dynamics. These elements could generate disparities between the simulated behavior and the actual system dynamics, yielding slightly less accurate results. Moreover, it is important to acknowledge that these similarity percentages can vary based on specific experimental setups and conditions.

In conclusion, while similarity percentages offer a valuable measure of simulation accuracy, additional factors like the quality of the physical model and data input must also be accounted for. The similarity percentages for the triple-link robotic system imply that the MATLAB simulation is reliable and satisfactory for 1st and 2nd joints but exhibits slightly lower accuracy for the 3rd joint.

## 6.5 Summary

This chapter discussed the modelling and control of a triple-link robotic system, designed to mimic human-like gymnastic movements, utilizing MATLAB/Simulink and Simscape. The robot's structure comprises upper, middle, and lower links, representing the arms, torso, and legs respectively. These software tools enable engineers to create intricate models of the robot's structure and control systems using predefined components. Further, the chapter evaluates the robot's real-world performance, discussing the challenges encountered

during the testing phase, thereby providing insights into potential improvements and optimizations. The performance of the simulated robot is juxtaposed with real-world experimental data to gauge the simulation's accuracy. A metric known as the similarity percentage is introduced in the chapter, quantifying the level of agreement between the simulation and the real-world performance. A graphical representation of this metric illustrates the similarity between the simulated and experimental results. The similarity percentages for each joint angle suggest that the Simulink simulation is relatively precise for joints $\theta_1$ and $\theta_2$, but less so for joint $\theta_3$. The findings of this study have the potential to inform the design and implementation of motion planning algorithms for robotic systems in industrial applications.

# Chapter 7: Conclusions, limitations and future work

This chapter summaries the conclusions, and some limitations of this research. Additionally, it provides suggestions for future work that help to improve the system. The research has effectively fulfilled the predetermined objectives, and the ensuing findings substantiate the efficacy of the proposed methodology.

## 7.1 Conclusions

The purpose of this research was to comprehend and regulate the movement of a 3-DOF robot known as Robogymnast. This system is complicated and operates as a highly nonlinear dynamical entity. In order to meet various motion requirements, it is crucial to consider the dynamics of the complete system and determine appropriate control measures. However, one of the challenges with this type of dynamic system is the difficulty in creating a model that accurately captures its nonlinear behavior. Because of the intricate and nonlinear nature of the dynamics involved, the mathematical model and dynamic equations were derived by adopting a linear approach. A continuous-time model of the proposed structure of the Robogymnast was obtained using the Euler-Lagrange approach, and subsequently, it was linearized in a stable configuration.

In conclusion, this work has successfully achieved its objectives, focusing primarily on examining and exploring the swing control issue for the 3-DoF robot (Robogymnast). To

achieve the best dynamic performance, many measures have been undertaken, encompassing the following aspects:

- The controller:

This research showcases the modelling and simulation of a PID controller's application to stabilize a robotic gymnast system using MATLAB/Simulink. In this investigation, a PID controller was designed and compared to a well-known LQR control technique for the Robogymnast. Concerning the first joint ($\theta_1$), ITAE values indicate that the PID performance is 41.02% superior to the LQR controller. Theta2 ($\theta_2$) results reveal only a slight difference of 1.55%, favouring the PID controller. Similarly, for the 3rd joint ($\theta_3$), the PID performance is 4.54% better than the LQR controller. Initial values of variables within the acrobot system were determined through mathematical modelling, followed by the development of a detailed model to simulate robotic manipulation using the PID controller. Key variables were identified, and calculations for overshoot, undershoot, settling, and rise times were performed. The system's dynamic performance was evaluated, and stability and robustness calculations were carried out for both the PID and LQR controllers. Various scenarios were compared, in which variables were adjusted by different values. The findings reveal that the proposed PID controller outperforms the conventional LQR controller for the Robogymnast. In summary, this study explores the control of a three-link robotic system's swing position and suggests that the proposed controller is effective. The chosen controller can be further developed to incorporate optimized algorithms for future research.

- Optimisation:

In summary, this study implemented a Robogymnast system using a PID controller and improved the system's performance by utilizing various algorithms in MATLAB/SIMULINK

for optimization. The objective was to develop a PID controller and assess its effectiveness within the Robogymnast system. The simulation model incorporated the PID controller, and primary system parameters, such as rise time, settling time, overshoot, undershoot, and ITAE, were analyzed. The optimization method proposed has notably diminished parameters such as ITAE to nearly zero for the second and third joints, and to 11.65 p.u. for the first joint, a significant improvement from the 94.180 p.u. observed in the unoptimized case. Moreover, it lowered other system parameters. The system's dynamic performance was also evaluated. The results indicate that the ACO algorithm in conjunction with the PID controller delivers satisfactory outcomes for the three-link robotic system. The ITAE values for both algorithms demonstrate that PID-ACO surpasses PID-GSA by 83.95% for Theta1 ($\theta_1$). In the case of the second joint ($\theta_2$), ACO exhibits better performance by 20.91%. Lastly, for the lower joint ($\theta_3$), PID-ACO outperforms PID-GSA by 23.07%, resulting in a more favorable outcome.

- The experimental part:

In this study, a planned motion for swing in a 3-link robotic system is presented. The setup of the system is described in detail, including the interconnections of all components. The study offers an in-depth investigation into the development of motion planning for a triple-link robotic system, with a special emphasis on comparing a Simscape MATLAB model with the Robogymnast system. The triple-link robotic system is highly regarded in industrial applications due to its precision in performing complex tasks. The motion planning approach proposed in this study integrates inverse kinematics and trajectory planning techniques to generate optimized motion trajectories for the system. The algorithm's parameters are then implemented using PID-ACO on this system to assess the effectiveness of the approach.

The research provides an exhaustive approach to motion planning in triple-link robotic systems, marking as a contribution to the robotics field. The outcomes of the study hold promise in guiding the creation and implementation of motion planning algorithms within industrial robotic systems. Furthermore, the research underscores the value of optimization methods in enhancing the effectiveness of these algorithms. The study employs a PID controller to improve stability, response time, and the overall efficiency of the Robogymnast system.

A comparison has been conducted between the PID controller and ACO algorithm and another controller and optimization algorithm. The results exhibit a marked superiority in performance metrics, solidifying their effectiveness. The methodology of the research incorporates simulation modelling to examine the key parameters and dynamic performance of the system. The study reveals that the similarity percentages for the first and second joints surpass 72%. This finding implies that the Simulink simulation produce precise results for these joints. However, the third joint's similarity percentage is slightly lower, standing at 69.05%. This discrepancy may indicate that the simulation's accuracy could be influenced by specific elements like mass and actuation. Through these outcomes, the research offers perspectives on the design and optimization of robotic systems, leveraging advanced control methodologies and optimization algorithms.

## 7.2 Limitations

Triple-link robotic systems have some notable drawbacks. Firstly, they can exhibit instability when performing complex tasks or functioning at elevated velocities. Secondly, as the number of links increases, the complexity of their control algorithms escalates correspondingly. This increased complexity can present difficulties in programming and

executing movements with precision. Lastly, their structural design tends to make them susceptible to wear and tear, which can progressively impair their precision and overall performance.

Furthermore, stepper motors, used in this system, are not as efficient as other motor types like AC induction motors or DC motors, particularly for tasks that require torque and speed for operating the acrobot effectively. This limitation arises because the maximum rotational speed and torque of stepper motors are constrained by the steps per revolution and the motor design. Besides, stepper motors may encounter issues of resonance and vibration at specific operating frequencies, which can have adverse impacts on their precision and overall performance. Moreover, operating at high speeds can lead to potential damage, further undermining their effectiveness and durability.

## 7.3 Future work

Having analysed and evaluated the simulation and experimental outcomes detailed in this study, the subsequent potential avenues for further work are outlined below:

- The use of machine learning methods, including neural networks or reinforcement learning, can be a beneficial way to improve the control and efficiency of three-link robotic systems.
- A possible approach to enhance the motion and stability of a triple-link pendulum system could involve using of high-performance motors, such as servo motors or DC brushless motors, to maximize the performance of a Robogymnast.
- The enhancement of triple-link mechanism systems' performance could potentially be realized by exploring advanced control methods, including adaptive control or optimal control.

- For enhancing the precision and stability of the system, it can be beneficial for researchers to investigate the utilization of highly accurate sensors, such as absolute rotary encoders.
- An additional way to enhance the system might be to integrate machine learning algorithms that allow the robot to learn from its errors, thereby improving its performance progressively. This could entail the use of reinforcement learning or similar methods to fine-tune the robot's actions based on feedback received from sensors and other data sources.

[1] H. A. Ismail, M. S. Packianather, R. I. Grosvenor, and E. E. Eldhukri, “The application of IWO in LQR controller design for the Robogymnast,” *IntelliSys 2015 - Proceedings of 2015 SAI Intelligent Systems Conference*, vol. 2, pp. 274–279, 2015, doi: 10.1109/IntelliSys.2015.7361154.

[2] N. S. Bhangal, “Design and Performance of LQR and LQR based Fuzzy Controller for Double Inverted Pendulum System,” Journal of Image and Graphics, vol. 1, no. 3, pp. 143–146, 2013, doi: 10.12720/joig.1.3.143-146.

[3] H. A. Ismail, “INTELLIGENT MODEL-BASED CONTROL OF COMPLEX MULTI-LINK MECHANISMS A thesis submitted to Cardiff University in the candidature for the,” no. December, 2016.

[4] M. Mohamed, B. Abdul Samad, F. Anayi, M. Packianather, and K. Yahya, “Analysing Various Control Technics for Manipulator Robotic System (Robogymnast),” *Computers, Materials & Continua*, vol. 75, no. 3, pp. 4681–4696, 2023, doi: 10.32604/cmc.2023.035312.

[5] Z. Lin, A. Saberi, M. Gutmann, and Y. A. Shamash, “Linear controller for an inverted pendulum having restricted travel: A high-and-low gain approach,” *Automatica*, vol. 32, no. 6, pp. 933–937, 1996, doi: 10.1016/0005-1098(96)00006-4.

[6] J. Yi, N. Yubazaki, and K. Hirota, “Upswing and stabilization control of inverted pendulum and cart system by the SIRMs dynamically connected fuzzy inference model,” *IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 139–152, 1999, doi: 10.1109/fuzzy.1999.793273.

[7] W. Zhong and H. Röck, “Energy and passivity based control of the double inverted pendulum on a cart,” *IEEE Conference on Control Applications - Proceedings*, pp. 896–901, 2001, doi: 10.1109/cca.2001.973983.

[8] Z.-J. L. Zhi-Hao Xu, Dong-Ming Jin, “USING LEARNING SAMPLES TO CONSTRUCT FUZZY LOGIC SYSTEMS WITH THE APPLICATION TO INVERTED PENDULUM CONTROL,” vol. Proceeding, no. 4-5 November, pp. 1085–1088, 2002.

[9] M. Bugeja, “Non-linear swing-up and stabilizing control of an inverted pendulum system,” *IEEE Region 8 EUROCON 2003: Computer as a Tool - Proceedings*, vol. B, pp. 437–441, 2003, doi: 10.1109/EURCON.2003.1248235.

[10] A. Yamada, S. Yamakawa, and H. Fujimoto, “Switching control for inverted pendulum system based on energy modification,” *Proceedings of the SICE Annual Conference*, no. 2, pp. 973–978, 2004.

[11] N. Muškinja and B. Tovornik, “Swinging up and stabilization of a real inverted pendulum,” *IEEE Transactions on Industrial Electronics*, vol. 53, no. 2, pp. 631–639, 2006, doi: 10.1109/TIE.2006.870667.

[12] B. Stephens, “Integral control of humanoid balance,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 4020–4027, 2007, doi: 10.1109/IROS.2007.4399407.

[13] T. Tsuji and K. Ohnishi, “A control of biped robot which applies inverted pendulum mode with virtual supporting point,” *International Workshop on Advanced Motion Control, AMC*, pp. 478–483, 2002, doi: 10.1109/amc.2002.1026967.

[14] F. Gubina, H. Hemaml, and R. B. Mcghee, “On the Dynamic Stability of Biped Locomotion,” *IEEE Trans Biomed Eng*, vol. BME-21, no. 2, pp. 102–108, 1974, doi: 10.1109/TBME.1974.324294.

[15] H. G. Kamil, “Intelligent Model-Based Control of Complex Three-Link Mechanisms,” *thesis submitted to Cardiff University*, no. December, 2015.

[16] M. Doi, Y. Hasegawa, and T. Fukuda, “Passive dynamic autonomous control of bipedal walking,” *Proceedings of the 2004 International Symposium on Micro-NanoMechatronics and Human Science, MHS2004; The Fourth Symposium “Micro-NanoMechatronics for and Information-Based Society” The 21st Century*, pp. 151–156, 2004, doi: 10.1109/mhs.2004.1421293.

[17] C. H. Chiu, “Self-tuning output recurrent cerebellar model articulation controller for a wheeled inverted pendulum control,” *Neural Comput Appl*, vol. 19, no. 8, pp. 1153–1164, 2010, doi: 10.1007/s00521-009-0335-2.

[18] P. Jaiwat and T. Ohtsuka, “Real-Time Swing-up of Double Inverted Pendulum by Nonlinear Model Predictive Control,” *5th International Symposium on Advanced Control of Industrial Processes*, pp. 290–295, 2014.

[19] C. W. Anderson, “Learning to control an inverted pendulum with connectionist networks.,” *Proceedings of the American Control Conference*, vol. 88 pt 1–3, no. April, pp. 2294–2298, 1988, doi: 10.23919/acc.1988.4790107.

[20] X. Xiong and Z. Wan, “The simulation of double inverted pendulum control based on particle swarm optimization LQR algorithm,” in *Proceedings 2010 IEEE International Conference on Software Engineering and Service Sciences, ICSESS 2010*, 2010, pp. 253–256. doi: 10.1109/ICSESS.2010.5552427.

[21] G. DeJong and M. W. Spong, “Swinging up the acrobot: an example of intelligent control,” *Proceedings of the American Control Conference*, vol. 2, pp. 2158–2162, 1994, doi: 10.1109/acc.1994.752458.

[22] D. LIU and H. YAMAURA, “Giant Swing Motion Control of 3-link Gymnastic Robot with Friction around an Underactuated Joint,” *Journal of System Design and Dynamics*, vol. 5, no. 5, pp. 925–936, 2011, doi: 10.1299/jsdd.5.925.

[23] A. V. S. M. Z. Kolovsky , A. N. Evgrafov , Yu. A. Semenov, *Advanced Theory of Mechanisms and Machines*. Springer Science & Business Media, 2012. doi: 10.1007/978-3-540-46516-4.

[24] J. E. S. Uicker, John J. Gordon R. Pennock, *Theory of Machines and Mechanisms*, vol. 13, no. 1. OXFORD UNIVERSITY PRESS, 2003.

[25] E. E. Eldukhri and D. T. Pham, “Autonomous swing-up control of a three-link robot gymnast,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 224, no. 7, pp. 825–833, 2010, doi: 10.1243/09596518JSCE948.

[26] X. Lai, A. Zhang, J. She, and M. Wu, “Motion control of underactuated three-link gymnast robot based on combination of energy and posture,” *IET Control Theory and Applications*, vol. 5, no. 13, pp. 1484–1493, 2011, doi: 10.1049/iet-cta.2010.0210.

[27] S. Takashiro and N. Yoshihiko, “Analysis and Control of Underactuated Mechanisms via the Averaging Method,” *Proc. of 2nd Asian Control Conference*, vol. 1, pp. 273–276, 1997.

[28] K. Manickavelan, B. Singh, and N. Sellappan, “Design, Fabrication and Analysis of Four Bar Walking Machine Based on Chebyshev’s Parallel Motion Mechanism.,” *European International Journal of Science and Technology*, vol. 3, no. 8, pp. 65–73, 2014.

[29] “Miriam Webster’s Collegiate Encyclopedia. Miriam Webster.”, [Online]. Available: https://www.merriam-webster.com/dictionary/pendulum

[30] K. J. Åström and K. Furuta, “Swinging up a pendulum by energy control,” *Automatica*, vol. 36, no. 2, pp. 287–295, 2000, doi: 10.1016/S0005-1098(99)00140-5.

[31] E. E. Eldukhri and D. T. Pham, “Autonomous swing-up control of a three-link robot gymnast,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 224, no. 7, pp. 825–833, Nov. 2010, doi: 10.1243/09596518JSCE948.

[32] I. Fantoni, R. Lozano, and M. W. Spong, “Energy based control of the Pendubot,” *IEEE Trans Automat Contr*, vol. 45, no. 4, pp. 725–729, 2000, doi: 10.1109/9.847110.

[33] A. Bradshaw and J. Shao, “Swing-up control of inverted pendulum systems,” *Robotica*, vol. 14, no. 4, pp. 397–405, 1996, doi: 10.1017/s0263574700019792.

[34] X. Xin and M. Kaneda, “Design and analysis of swing-up control for a 3-link gymnastic robot with passive first joint,” *Proceedings of the IEEE Conference on Decision and Control*, vol. 23, no. 6, pp. 1923–1928, 2007, doi: 10.1109/CDC.2007.4434298.

[35] X. Xin, S. Tanaka, J. H. She, and T. Yamasaki, “Revisiting energy-based swing-up control for the Pendubot,” *Proceedings of the IEEE International Conference on Control Applications*, pp. 1576–1581, 2010, doi: 10.1109/CCA.2010.5611172.

[36] K. Furuta and M. Iwase, “Swing-up time analysis of pendulum,” *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 52, no. 3, pp. 153–163, 2004.

[37] K. J. Åström and K. Furuta, “Swinging up a pendulum by energy control,” *Automatica*, vol. 36, no. 2, pp. 287–295, 2000, doi: 10.1016/S0005-1098(99)00140-5.

[38] C. C. Chung and J. Hauser, “Control of a Swinging Pendulum,” *Automatica*, vol. 31, no. 6, 1990.

[39] F. Gordillo, J. A. Acosta, and J. Aracil, “A new swing-up law for the Furuta pendulum,” *Int J Control*, vol. 76, no. 8, pp. 836–844, 2003, doi: 10.1080/0020717031000116506.

[40] M. Yamakita, M. Iwashiro, Y. Sugahara, and K. Furuta, “Robust swing up control of double pendulum,” *Proceedings of the American Control Conference*, vol. 1, pp. 290–295, 1995, doi: 10.1109/acc.1995.529255.

[41] G. A. Medrano-Cerda, E. E. Eldukhri, and M. Cetin, “Balancing and Attitude Control of Double and Triple Inverted Pendulums,” *Transactions of the Institute of Measurement and Control*, vol. 17, no. 3, pp. 143–154, 1995, doi: 10.1177/014233129501700306.

[42] Q. Han, Z. Qin, X. Yang, and B. Wen, “Rhythmic swing motions of a two-link robot with a neural controller,” *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 2, pp. 335–342, 2007.

[43] S. C. Duong, H. Kinjo, E. Uezato, and T. Yamamoto, “Intelligent control of a three-DOF planar underactuated manipulator,” *Artif Life Robot*, vol. 14, no. 2, pp. 284–288, 2009, doi: 10.1007/s10015-009-0674-1.

[44] E. E. Eldukhri and D. T. Pham, “Autonomous swing-up control of a three-link robot gymnast,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 224, no. 7, pp. 825–833, 2010, doi: 10.1243/09596518JSCE948.

[45] F. Xue, Z. Hou, Y. Hu, and C. Liu, “Human simulated intelligence motion control for a three-link acrobot,” *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, no. 60905053, pp. 844–848, 2011, doi: 10.1109/WCICA.2011.5970634.

[46] T. Glück, A. Eder, and A. Kugi, “Swing-up control of a triple pendulum on a cart with experimental validation,” *Automatica*, vol. 49, no. 3, pp. 801–808, 2013, doi: 10.1016/j.automatica.2012.12.006.

[47] T. Henmi, M. Akiyama, and T. Yamamoto, “Motion control of underactuated linkage robot based on gymnastic skill,” *Electrical Engineering in Japan (English translation of Denki Gakkai Ronbunshi)*, vol. 206, no. 1, pp. 42–50, 2019, doi: 10.1002/eej.23142.

[48] L. Wang, S. Chen, P. Zhang, J. She, and X. Lai, “A simple control strategy based on trajectory planning for vertical acrobot,” *Actuators*, vol. 10, no. 12, pp. 1–12, 2021, doi: 10.3390/act10120308.

[49] Y. Nishiki, H. Kajiwara, and M. Aoyagi, “Control of swing-up and giant-swing motions of Acrobot based on periodic input,” *Nonlinear Dyn*, vol. 108, no. 3, pp. 2297–2308, 2022, doi: 10.1007/s11071-022-07312-x.

[50] C. Mohan and K. Deep, *Optimization Techniques*. New Age International (UK) Ltd, 2009.

[51] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*. Hoboken, N.J: Wiley, 2005.

[52] T. Jia and D. H. Long, “Optimization of Evacuation Route for Urban Traffic Congestion based on Particle Swarm Optimization,” in *2020 IEEE International Conference on Industrial Application of Artificial Intelligence, IAAI 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 351–356. doi: 10.1109/IAAI51705.2020.9332842.

[53] D. P. Heyman and M. J. Sobel, *Stochastic models in operations research: stochastic optimization*, vol. 2. Courier Corporation, 2004.

[54] M. Cavazzuti, “Optimization methods: from theory to design,” 2013. [Online]. Available: https://www.researchgate.net/publication/261750290

[55] G. C. Onwubolu and B. V. Babu, *New Optimization Techniques in Engineering*, vol. 141. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-39930-8.

[56] K.-J. Ha and H.-M. Kim, “A Genetic Approach to the Attitude Control of an Inverted Pendulum System,” 1997.

[57] G. Zhang, Z. Fang, and L. Shu, “GALQR Optimal Control Method and Applying in the Active Suspension System,” Paris, France, 2007.

[58] S. Mobaieen, S. Mobaieen, B. Mohamady, H. Ghorbani, and A. Rabii, “Optimal Control Design Using Evolutionary Algorithms with Application to an Aircraft Landing System,” *J. Basic. Appl. Sci. Res*, vol. 2, no. 2, pp. 1876–1882, 2012, [Online]. Available: www.textroad.com

[59] H. Wei, Q. Qian, H. Qiang, H. Qiaoli, Z. Yixin, and X. Lin, “Optimization of sliding mode controller for double inverted pendulum based on genetic algorithm,” in *2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics, ISSCAA 2008*, 2008. doi: 10.1109/ISSCAA.2008.4776281.

[60] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, IEEE, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

[61] X. Xiong and Z. Wan, “The simulation of double inverted pendulum control based on particle swarm optimization LQR algorithm,” *Proceedings 2010 IEEE International Conference on Software Engineering and Service Sciences, ICSESS 2010*, pp. 253–256, 2010, doi: 10.1109/ICSESS.2010.5552427.

[62] M. I. Solihin, Wahyudi, and R. Akmeliawati, “Self-erecting inverted pendulum employing PSO for stabilizing and tracking controller,” in *Proceedings of 2009 5th International Colloquium on Signal Processing and Its Applications, CSPA 2009*, 2009, pp. 63–68. doi: 10.1109/CSPA.2009.5069190.

[63] S. B. Chandra Debnath, P. Chandra Shill, and K. Murase, “Particle Swarm Optimization Based Adaptive Strategy for Tuning of Fuzzy Logic Controller,” *International Journal of Artificial Intelligence & Applications*, vol. 4, no. 1, pp. 37–50, Jan. 2013, doi: 10.5121/ijaia.2013.4104.

[64] R. Fierro, O. Castillo, F. Valdez, and P. Melin, “Optimization of Membership Functions for the Fuzzy Controllers of the Water Tank and Inverted Pendulum with Differents PSO Variants,” *TELKOMNIKA*, vol. 11, no. 4, pp. 1693–6930, 2013, doi: 10.12928/TELKOMNIKA.v11i4.1639.

[65] Y. Li, J. Liu, and Y. Wang, “Design approach of weighting matrices for LQR based on multi-objective evolution algorithm,” in *Proceedings of the 2008 IEEE International*

*Conference on Information and Automation, ICIA 2008*, 2008, pp. 1188–1192. doi: 10.1109/ICINFA.2008.4608180.

[66] D. T. Pham *et al.*, “Intelligent Production Machines and Systems The Bees Algorithm-A Novel Tool for Complex Optimisation Problems.”

[67] M. W. Spong and D. J. Block, “Pendubot: a mechatronic system for control research and education,” in *Proceedings of the IEEE Conference on Decision and Control*, IEEE, 1995, pp. 555–556. doi: 10.1109/cdc.1995.478951.

[68] A. H. Darwish, “CAERDYg> Enhanced Bees Algorithm with Fuzzy Logic and Kalman Filtering,” 2009.

[69] H. G. Kamil, E. E. Eldukhri, M. S. Packianather, H. G. Kamil, and P. A. Uk, “OPTIMISATION OF SWING-UP CONTROL PARAMETERS FOR A ROBOT GYMNAST USING THE BEES ALGORITHM Intelligent robotised physiotherapy View project Innovative production machines and systems View project OPTIMISATION OF SWING-UP CONTROL PARAMETERS FOR A ROBOT GYMNAST USING THE BEES ALGORITHM.” [Online]. Available: https://www.researchgate.net/publication/273647238

[70] E. E. Eldukhri and H. G. Kamil, “Optimisation of swing-up control parameters for a robot gymnast using the Bees Algorithm,” *J Intell Manuf*, vol. 26, no. 5, pp. 1039–1047, Oct. 2015, doi: 10.1007/s10845-013-0848-5.

[71] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Comput Intell Mag*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[72] M. Dorigo and G. Di Caro, “The Ant Colony Optimization Meta_Heuristic,” *New Ideas in Optimization*, pp. 11–32, 1999.

[73] A. C. Kakas *et al.*, “Ant Colony Optimization,” in *Encyclopedia of Machine Learning*, Boston, MA: Springer US, 2011, pp. 36–39. doi: 10.1007/978-0-387-30164-8_22.

[74] D. O. and S. A., “Swarm Intelligence from Natural to Artificial Systems: Ant Colony Optimization,” *International Journal on Applications of Graph Theory In wireless Ad Hoc Networks And sensor Networks*, vol. 8, no. 1, pp. 9–17, Mar. 2016, doi: 10.5121/jgraphoc.2016.8102.

[75] M. Dorigo, G. Di Caro, and L. M. Gambardella, “Ant Algorithms for Discrete Optimization.” [Online]. Available: http://direct.mit.edu/artl/article-pdf/5/2/137/1661667/106454699568728.pdf

[76] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, “Data mining with an ant colony optimization algorithm,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, Aug. 2002, doi: 10.1109/TEVC.2002.802452.

[77] H. Azizi and B. Ismail, “INTELLIGENT MODEL-BASED CONTROL OF COMPLEX MULTI-LINK MECHANISMS A thesis submitted to Cardiff University in the candidature for the,” 2016.

[78] E. E. Eldukhri and D. T. Pham, “Autonomous swing-up control of a three-link robot gymnast,” *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 224, no. 7, pp. 825–833, 2010, doi: 10.1243/09596518JSCE948.

[79] M. W. Spong, “Swing up control of the acrobot,” in *Proceedings - IEEE International Conference on Robotics and Automation*, Publ by IEEE, 1994, pp. 2356–2361. doi: 10.1109/robot.1994.350934.

[80] H. Hemami and R. L. Farnsworth, “Postural and Gait Stability of a Planar Five Link Biped by Simulation,” *IEEE Trans Automat Contr*, vol. 22, no. 3, pp. 452–458, 1977, doi: 10.1109/TAC.1977.1101513.

[81] M. W. Spong, S. Hutchinson, and M. Vidyasagar, “Robot Modeling and Control First Edition,” 2006.

[82] V.I. Arnold, “Mathematical Methods of Classical Mechanics,” no. Second Edition, 1989.

[83] “Robot Modeling and Control,” *Industrial Robot: An International Journal*, vol. 33, no. 5, pp. 403–403, Sep. 2006, doi: 10.1108/ir.2006.33.5.403.1.

[84] J. J. Craig, *Introduction to Robotics Mechanics and Control Third Edition*. 1986.

[85] K. Furut, T. Ochiai, and N. Ono, “Attitude control of a triple inverted pendulum,” *Int J Control*, vol. 39, no. 6, pp. 1351–1365, 1984.

[86] G. A. Medrano-Cerda, E. E. Eldukhri, and M. Cetin, “Balancing and Attitude Control of Double and Triple Inverted Pendulums,” *Transactions of the Institute of Measurement and Control*, vol. 17, no. 3, pp. 143–154, 1995, doi: 10.1177/014233129501700306.

[87] K. G. Eltohamy and C. Y. Kuo, “Nonlinear optimal control of a triple link inverted pendulum with single control input,” *Int J Control*, vol. 69, no. 2, pp. 239–256, Jan. 1998, doi: 10.1080/002071798222811.

[88] M. Mohamed, F. Anayi, M. Packianather, B. A. Samad, and K. Yahya, “Simulating LQR and PID controllers to stabilise a three-link robotic system,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 2033–2036. doi: 10.1109/ICACITE53722.2022.9823512.

[89] A. Preumont, “Controllability and Observability,” in *Vibration Control of Active Structures: An Introduction Third Edition*, A. Preumont, Ed., Dordrecht: Springer Netherlands, 2011, pp. 275–297. doi: 10.1007/978-94-007-2033-6_12.

[90] C. N. Aithal, P. Ishwarya, S. Sneha, C. N. Yashvardhan, and K. V. Suresh, “Design of a Bipedal Robot,” in *Lecture Notes in Electrical Engineering*, Springer Science and Business Media Deutschland GmbH, 2021, pp. 55–67. doi: 10.1007/978-981-16-0443-0_5.

[91] “https://stm32f4-discovery.net/stm32f429-discovery/.”

[92] R. C. Dorf Robert H Bishop and R. C. Dorf Robert H Bishop Prentice Hall Upper Saddle River Boston Columbus San Francisco New York Indianapolis London Toronto Sydney Singapore Tokyo Montreal Dubai Madrid Hong Kong Mexico City Munich Paris Amsterdam Cape Town, “MODERN CONTROL SYSTEMS SOLUTION MANUAL,” 2011. [Online]. Available: www.elsolucionario.org

[93] Y. Zennir, E. Guechi, and R. Bendib, “Robust fractional multi-controller design of inverted pendulum system,” in *2016 20th International Conference on System Theory, Control and Computing, ICSTCC 2016 - Joint Conference of SINTES 20, SACCS 16, SIMSIS 20 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2016, pp. 277–282. doi: 10.1109/ICSTCC.2016.7790678.

[94] F. Zhao, Z.-P. Jiang, and T. Liu, “Event-triggered adaptive optimal control using output feedback: An adaptive dynamic programming approach; Event-triggered adaptive optimal control using output feedback: An adaptive dynamic programming approach,” 2019.

[95] X. Xin and M. Kaneda, “The swing up control for the acrobot based on energy control approach,” in *Proceedings of the IEEE Conference on Decision and Control*, 2002, pp. 3261–3266. doi: 10.1109/cdc.2002.1184374.

[96] M. W. Spong, “The Swing Up Control Problem For The Acrobot,” *IEEE Control Syst*, 1995, doi: 10.1109/37.341864.

[97] N. S. Bhangal, “Design and Performance of LQR and LQR based Fuzzy Controller for Double Inverted Pendulum System,” *Journal of Image and Graphics*, vol. 1, no. 3, pp. 143–146, 2013, doi: 10.12720/joig.1.3.143-146.

[98] R. Mohammadi Asl, A. Mahdoudi, E. Pourabdollah, and G. Klančar, “Combined PID and LQR controller using optimized fuzzy rules,” *Soft comput*, vol. 23, no. 13, pp. 5143–5155, Jul. 2019, doi: 10.1007/s00500-018-3180-3.

[99] V. Bakırcıoğlu, M. A. Şen, and M. Kalyoncu, “Optimization of PID controller based on The Bees Algorithm for one leg of a quadruped robot; Optimization of PID controller based on The Bees Algorithm for one leg of a quadruped robot,” 2016, doi: 10.1051/C.

[100] B. Abdul samad, M. Mohamed, F. Anayi, and Y. Melikhov, “An Investigation of Various Controller Designs for Multi-Link Robotic System (Robogymnast),” *Knowledge*, vol. 2, no. 3, pp. 465–486, Sep. 2022, doi: 10.3390/knowledge2030028.

[101] E. Vinodh Kumar and J. Jerome, “Robust LQR controller design for stabilizing and trajectory tracking of inverted pendulum,” in *Procedia Engineering*, Elsevier Ltd, 2013, pp. 169–178. doi: 10.1016/j.proeng.2013.09.088.

[102] R. A. Paz, “The Design of the PID Controller,” 2001.

[103] P. Boscariol, L. Scalera, and A. Gasparetto, “Nonlinear control of multibody flexible mechanisms: A model-free approach,” *Applied Sciences (Switzerland)*, vol. 11, no. 3, pp. 1–14, Feb. 2021, doi: 10.3390/app11031082.

[104] B. A. Samad, M. Mohamed, F. Anavi, and Y. Melikhov, “A hybrid Fuzzy approach of different controllers to stabilize a 3-link swinging robotic (Robogymnast),” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 2432–2437. doi: 10.1109/ICACITE53722.2022.9823768.

[105] K. J. Åström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2021.

[106] H. G. Kamil, E. E. Eldukhri, and M. S. Packianather, “Balancing Control of Robot Gymnast Based on Discrete-Time Linear Quadratic Regulator Technique,” in *Proceedings - 2nd International Conference on Artificial Intelligence, Modelling, and Simulation, AIMS 2014*, Institute of Electrical and Electronics Engineers Inc., May 2014, pp. 137–142. doi: 10.1109/AIMS.2014.38.

[107] H. Purnawan, Mardlijah, and E. B. Purwanto, “Design of linear quadratic regulator (LQR) control system for flight stability of LSU-05,” in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Sep. 2017. doi: 10.1088/1742-6596/890/1/012056.

[108] P. Nordfeldt and T. Hägglund, “Decoupler and PID controller design of TITO systems,” *J Process Control*, vol. 16, no. 9, pp. 923–936, Oct. 2006, doi: 10.1016/j.jprocont.2006.06.002.

[109] D. Graham and R. C. Lathrop, “The synthesis of ‘optimum’ transient response: Criteria and standard forms,” *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 72, no. 5, pp. 273–288, Jul. 2013, doi: 10.1109/tai.1953.6371346.

[110] S. P. Bhattacharyya, “The art of control engineering: K. Dutton, S. Thompson and B. Barraclough; Addison-Wesley, Longman, 1997, ISBN: 0-201-17545-2.,” *Automatica*, vol. 37, pp. 961–964, Mar. 2001.

[111] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*. John Wiley & Sons, 2013.

[112] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, “Model Predictive Control: Theory, Computation, and Design 2nd Edition.” [Online]. Available: http://www.nobhillpublishing.com

[113] M. Dorigo, “Ant Colony Optimization,” 2004. [Online]. Available: https://www.researchgate.net/publication/36146886

[114] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” *Inf Sci (N Y)*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: 10.1016/j.ins.2009.03.004.

[115] S. Waqas and P. Strachan, “Development of an Optimisation Algorithm for Auto-sizing Capacity of Renewable and Low Carbon Energy Systems,” 2011.

[116] A. Sani Hassan, “Management of Distributed Energy Resources in Energy Systems THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING,” 2016.

[117] S. S. Rao, “Engineering Optimization: Theory and Practice,” 2009.

[118] J. Bisschop, “AIMMS Modeling Guide - Formulating Optimization Models.” [Online]. Available: www.aimms.com.

[119] G. B. Dantzig, “Linear Programming and Extensions,” 1963.

[120] S. Skarvelis-Kazakos, “Emissions of Aggregated Micro-generators THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY,” 2011.

[121] S. M. Ross, *Introduction to probability models*. Academic press, 2014.

[122] D. Cinquegrana and E. Iuliano, “Investigation of adaptive design variables bounds in dimensionality reduction for aerodynamic shape optimization,” *Comput Fluids*, vol. 174, pp. 89–109, Sep. 2018, doi: 10.1016/j.compfluid.2018.07.012.

[123] A. K. Hartmann and H. Rieger, “Optimization Algorithms in Physics,” 2001.

[124] M. Dorigo and A. Colorni, “The Ant System: Optimization by a colony of cooperating agents,” 1996.

[125] M. Dorigo and L. M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” 1997.

[126] B. Bullnheimer, R. F. Hartl, and C. Strauss, “An improved Ant System algorithm for the Vehicle Routing Problem.”

[127] B. Hölldobler and E. O. Wilson, *The ants*. Harvard University Press, 1990.

[128] M. Dorigo and T. Stützle, *Ant Colony Optimization*. The MIT Press, 2004. doi: 10.7551/mitpress/1290.001.0001.

[129] E. Bonabeau, M. Dorigo, G. Theraulaz, and G. Theraulaz, *Swarm intelligence: from natural to artificial systems*, no. 1. Oxford university press, 1999.

[130] B. Bullnheimer, R. F. Hartl, and C. Strauss, “An improved Ant System algorithm for the Vehicle Routing Problem.”

[131] M. Dorigo and G. Di Caro, “The ant colony optimization Meta-Heuristic,” *Universite Libre de Bruxelles*, 1999.

[132] L. Maria Gambardella and M. Dorigo, “AN ANT COLONY SYSTEM HYBRIDIZED WITH A NEW LOCAL SEARCH FOR THE SEQUENTIAL ORDERING PROBLEM ACCEPTED FOR PUBLICATION IN INFORMS JOURNAL ON COMPUTING.”

[133] D. Zeng *et al.*, “An improved ant colony optimization algorithm based on dynamically adjusting ant number,” in *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest*, 2012, pp. 2039–2043. doi: 10.1109/ROBIO.2012.6491268.

[134] K. Ghoseiri and B. Nadjari, “An ant colony optimization algorithm for the bi-objective shortest path problem,” *Applied Soft Computing Journal*, vol. 10, no. 4, pp. 1237–1246, 2010, doi: 10.1016/j.asoc.2009.09.014.

[135] L. Ljung, *Signal Analysis and Prediction*. Boston, MA: Birkhäuser Boston, 1998. doi: 10.1007/978-1-4612-1768-8.

[136] F. G. Martins, “Tuning PID Controllers using the ITAE Criterion*,” 2005. [Online]. Available: www.mathworks.com

[137] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996, doi: 10.1109/3477.484436.

[138] M. Dorigo, “Optimization, learning and natural algorithms,” *Ph. D. Thesis, Politecnico di Milano*, 1992.

[139] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” *Inf Sci (N Y)*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009, doi: 10.1016/j.ins.2009.03.004.

[140] A. Kaveh and S. Talatahari, “A novel heuristic optimization method: Charged system search,” *Acta Mech*, vol. 213, no. 3–4, pp. 267–289, 2010, doi: 10.1007/s00707-009-0270-4.

[141] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-Verse Optimizer: a nature-inspired algorithm for global optimization,” *Neural Comput Appl*, vol. 27, no. 2, pp. 495–513, Feb. 2016, doi: 10.1007/s00521-015-1870-7.

[142] M. F. AlHajri, M. R. AlRashidi, and M. E. El-Hawary, “Hybrid particle swarm optimization approach for optimal distribution generation sizing and allocation in distribution systems,” in *Canadian Conference on Electrical and Computer Engineering*, 2007, pp. 1290–1293. doi: 10.1109/CCECE.2007.328.

[143] A. Zhang, G. Sun, Z. Wang, and Y. Yao, “A hybrid genetic algorithm and gravitational search algorithm for global optimization,” *Neural Network World*, vol. 25, no. 1, pp. 53–73, 2015, doi: 10.14311/NNW.2015.25.003.

[144] F. Ahmed Parvez, P. Gera, G. Pradeepini, and U. N. Dulhare, “Applications of Gravitational Search Algorithm (GSA)”, [Online]. Available: https://www.researchgate.net/publication/316991040

[145] S. Mirjalili and S. Z. M. Hashim, “A new hybrid PSOGSA algorithm for function optimization,” in *Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application*, 2010, pp. 374–377. doi: 10.1109/ICCIA.2010.6141614.

[146] J. Radosavljević, M. Jevtić, N. Arsić, and D. Klimenta, “Optimal power flow for distribution networks using gravitational search algorithm,” *Electrical Engineering*, vol. 96, no. 4, pp. 335–345, Dec. 2014, doi: 10.1007/s00202-014-0302-5.

[147] R. K. Sahu, S. Panda, and S. Padhan, “Optimal gravitational search algorithm for automatic generation control of interconnected power systems,” *Ain Shams Engineering Journal*, vol. 5, no. 3, pp. 721–733, 2014, doi: 10.1016/j.asej.2014.02.004.

[148] U. K. Rout, R. K. Sahu, and S. Panda, “Gravitational search algorithm based Automatic Generation Control for interconnected power system,” *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2013*, pp. 558–563, 2013, doi: 10.1109/ICCPCT.2013.6528998.

[149] H. A. Ismail, E. E. Eldhukhri, and M. S. Packianather, “Invasive weed optimization of swing-up control parameters for robot gymnast,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 88–93, 2014, doi: 10.1109/AIM.2014.6878052.

[150] X. Lai, A. Zhang, J. She, and M. Wu, “Motion control of underactuated three-link gymnast robot based on combination of energy and posture,” *IET Control Theory and Applications*, vol. 5, no. 13, pp. 1484–1493, Sep. 2011, doi: 10.1049/iet-cta.2010.0210.

[151] B. Abdul Samad, M. Mohamed, and F. Anayi, “Motion Planning of a Triple-Link Robotic System,” MDPI AG, Feb. 2023, p. 48. doi: 10.3390/asec2022-13774.

[152] K. Lakshmi, “Design of robust energy control for cartinverted pendulum,” *International Journal of Engineering and Technology*, vol. 4, no. 1, pp. 66–76, 2007.

[153] M. Mohamed, F. Anayi, M. Packianather, B. A. Samad, and K. Yahya, “Simulating LQR and PID controllers to stabilise a three-link robotic system,” in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 2033–2036. doi: 10.1109/ICACITE53722.2022.9823512.

[154] MATLAB/Simscape, “https://uk.mathworks.com/products/simscape.html.”

[155] Simscape Multibody, “https://uk.mathworks.com/products/simscape-multibody.html.” Accessed: May 19, 2023. [Online]. Available: https://uk.mathworks.com/products/simscape-multibody.html

[156] Design. Simulate. Deploy., “https://uk.mathworks.com/products/simulink.html.” Accessed: May 20, 2023. [Online]. Available: https://uk.mathworks.com/products/simulink.html

[157] Simscape Model and simulate multidomain physical systems, “https://uk.mathworks.com/products/simscape.html.”

# Appendices

## A1. The system motion code

```
/* USER CODE BEGIN Header */
 **********************************
  * @file           : main.c
  * @brief          : Main program body
  * @attention
  * <h2><center>&copy; Copyright (c) 2021 STMicroelectronics.
  * All rights reserved.</center></h2>
  * This software component is licensed by ST under Ultimate Liberty license
  * SLA0044, the "License"; You may not use this file except in compliance with
/* USER CODE END Header */
/* Includes ------------------------------*/
#include "main.h"
#include "usb_host.h"
/* Private includes ----------------------*/
/* USER CODE BEGIN Includes */
#define MAX_TETHA1 4500
#define MIN_TETHA1 -4500
#define MAX_TETHA2 5000
#define MIN_TETHA2 -5000
#define DEG2PULS 50
#define m_speed1 30
#define m_speed2 35
#define pos_param 0.1
#define vel_param 3
#define mot1_dir_cw HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5 , GPIO_PIN_RESET)
#define mot1_dir_ccw HAL_GPIO_WritePin(GPIOB,GPIO_PIN_5 , GPIO_PIN_SET)
#define mot2_dir_cw HAL_GPIO_WritePin(GPIOE,GPIO_PIN_6 , GPIO_PIN_RESET)
#define mot2_dir_ccw HAL_GPIO_WritePin(GPIOE,GPIO_PIN_6 , GPIO_PIN_SET)
#define ledg_on HAL_GPIO_WritePin(GPIOD,GPIO_PIN_12 , GPIO_PIN_SET)
#define ledg_of HAL_GPIO_WritePin(GPIOD,GPIO_PIN_12 , GPIO_PIN_RESET)
#define ledy_on HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13 , GPIO_PIN_SET)
#define ledy_of HAL_GPIO_WritePin(GPIOD,GPIO_PIN_13 , GPIO_PIN_RESET)
#define ledr_on HAL_GPIO_WritePin(GPIOD,GPIO_PIN_14 , GPIO_PIN_SET)
#define ledr_of HAL_GPIO_WritePin(GPIOD,GPIO_PIN_14 , GPIO_PIN_RESET)
#define ledb_on HAL_GPIO_WritePin(GPIOD,GPIO_PIN_15 , GPIO_PIN_SET)
#define ledb_of HAL_GPIO_WritePin(GPIOD,GPIO_PIN_15 , GPIO_PIN_RESET)
#define  th2_cal_gin -0.0597
#define  th2_cal_angl_ofst 2000
#define  th2_cal_ofst 1.328
#define  th3_cal_gin  0.0542
#define  th3_cal_angl_ofst 2130
#define  th3_cal_ofst 1.1191
#define  delta_time 0.001
double theta_real=0,theta=0,theta_old1=0,theta_old2=0,theta_old3=0 , enc_real=0,msin=0 ;
double theta_polar=0,vel_sum=0,vel1=0,vel2=0,vel3=0,vel_old1=0,vel_old2=0,vel_old3=0, vel_polar=0, vel_p=0  ;
double theta1=0, theta2=0, theta3=0, timm=0,
thetas[30],vels[30],vels_sum=0,velocity1=0,thetas_sum=0,
theta1_old=0,delta_theta1=0;
int t3_sp=0,t9_sp=0,sp_st3=0,sp_st9=0;
int t3_st=0,t9_st=0,enc=0,enc_old=0,delta_enc=0,enc2=0;
int t3_sp_cn=0,t9_sp_cn=0,t3_sp_cn2=0,t9_sp_cn2=0;
int dir1=1,dir2=1,m_dir=1;
int cycle_time=0, cycle_time_old=0, ii=0;
 GPIO_PinState t3_state,t9_state; // Variable to store the state of the button
 uint16_t angl1, angl2,  angl3 ;
 int robo_data[20];
 unsigned int mot1_sp=m_speed1,mot2_sp=m_speed2;
 void acrobat_robot_int(){
            ledb_of;
 }
 // motor(motor, direction , speed)
```

## Appendices

```
60.  // for select the motor : 1= motor 1, 2=motor2
61.  // for select the motor direction :
62. 1=CW , -1=CCW
63.  void motor(int mot,int dir,int pos) // Motor function
64.  {
65.            if (mot==1)
66.            {
67.  if (dir==1)//******* **********
68. Direction          mot1_dir_cw;
69.  else if (dir==-1)
70.            mot1_dir_ccw;
71.  // motor1
72.   //boButton_state = HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin);
      t3_st = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4);
if((t3_st==1)&&(sp_st3==0))
73.  { if(dir==1)
74. t3_sp_cn++;
75.  else if (dir==-1)
76.  t3_sp_cn--;
77.  sp_st3=1;
78.  ledg_on;
79.  }
80.  else if(t3_st==0)
81.  { sp_st3=0;
82. ledg_of;}
83.  //*****************
84. if( t3_sp_cn>(pos))
85. t3_sp_cn--;
86.  if( t3_sp_cn<(-pos))
87.  t3_sp_cn++;
88.  }
89.  else if (mot==2)
90.  {
91.  if (dir==1)
     //********************************* Direction
92.  mot2_dir_cw;
93. else if (dir==-1)
94.  mot2_dir_ccw;
95.  //^^^^^^^^^^motor2
96.  t9_st = HAL_GPIO_ReadPin(GPIOE, GPIO_PIN_5);
97.  if((t9_st==1)&&(sp_st9==0))
98.   {
99.   if(dir==1)
100. t9_sp_cn++;
101. else if (dir==-1)
102. t9_sp_cn--;
103. sp_st9=1;
104. ledr_on;
105. }
106. else if(t9_st==0)
107. {
108. sp_st9=0;
109. ledr_of;}
110. //******************
111. if( t9_sp_cn>(pos))
112. t9_sp_cn--;
113. if( t9_sp_cn<(-pos))
114. t9_sp_cn++;
115. }
116. }
117. //*****************************
118. void angulear_pos(){
119.//*********************************
     ***Theta1
120. enc_real=enc;
121. theta_real=(enc_real/4096)*360;
122. theta=theta_real;
123. if ((theta_real>180)&&(theta_real<=360))
124. theta=theta_real-360;
125.
126. theta_old3=theta_old2;
127. theta_old2=theta_old1;
128. theta_old1=theta;
129.theta_polar=(theta_old1+theta_old2+theta_old3)/3;
130. theta1=theta_polar;
131. vel1=(theta_old1-theta_old2)/delta_time;
132. vel2=(theta_old1-theta_old3)/(delta_time*2);
133. vel2=(theta_old2-theta_old3)/delta_time;
134. vel_sum=(vel1+vel2+vel3)/3;
135. vel_old3=vel_old2;
136. vel_old2=vel_old1;
137. vel_old1=vel_sum;
138. l_polar=(vel_old1+vel_old2+vel_old3)/3;
139.thetas_sum=0;
140. for(ii=0;ii<29;ii++){
141. thetas[ii]=thetas[ii+1];
142. thetas_sum+=thetas[ii];
143.}
144.thetas[29]=theta_polar;
145.thetas_sum+=thetas[29];
146.theta1=thetas_sum/30;
147.vels_sum=0;
148.for(ii=0;ii<29;ii++){
149. vels[ii]=vels[ii+1];
150. vels_sum+=vels[ii];
151.}
152.vels[29]=vel_polar;
153.vels_sum+=vels[29];
```

```
154.velocity1=vels_sum/30;
155. //**********Theta2 & Theta3
156. // Calibartion
157. theta2=th2_cal_gin*(angl2-
     th2_cal_angl_ofst) + th2_cal_ofst;
158. theta3=th3_cal_gin*(angl3-
     th3_cal_angl_ofst) + th3_cal_ofst;
159. }
160. void motor_dir()
161. {
162.  if (m_dir==1){
163.     motor(1,1,vel_p);// motor1
164.     motor(2,-1,vel_p);// motor2de
165.  }
166.  else if (m_dir==-1){
167.      motor(1,-1,vel_p);// motor1
168.      motor(2,1,vel_p);// motor2de
169.  }
170. }
171. //Algorithm ACO
172. void acrobat()
173. {
174.    delta_enc=enc-enc_old;
175.    delta_theta1=theta1-theta1_old;
176.    theta1_old=theta1;
177.    if (delta_enc<0)
178.    {
179.         m_dir=1;
180.                ledy_on;
181.                       ledb_of;
182.                       HAL_Delay(1);
183.    }
184.    else if (delta_enc>0)
185.    {
186.          m_dir=-1;
187.                ledy_of;
188.                       ledb_on;
189.                       HAL_Delay(1);
190.    }
191. }
192. void conv_data()
193. {
194.             timm+=(delta_time*200);
195.             cycle_time_old=cycle_time;
196.             cycle_time=timm;
197.             if(((-360 <= theta_polar) ||
     (theta_polar <= 360)) && ((-360 <=
     theta2) || (theta2<= 360)) && ((-360 <=
     theta3) || (theta3<= 360)) ){
198.             robo_data[0]=cycle_time;
199.             robo_data[1]=theta_polar*100;
200.             robo_data[2]=theta2*100;
201.             robo_data[3]=theta3*100;
202.
     printf("%d,%d,%d,%d\n",robo_data[0],ro
     bo_data[1],robo_data[2],robo_data[3]);
203.             }
204. void vel_select()
205. {
206.              if(cycle_time >2000){
207.                                  mot1_sp=35;
208.                                  mot2_sp=40;
209.              }
210.              /*
211.              msin
     =1+(sin(theta_polar*0.0174533))*pos_par
     am;
      vel_p=(velocity1*0.01)*vel_param;
212.if(vel_p<0) vel_p*=-1;
213.             mot1_sp=10+vel_p;
214.             mot2_sp=15+vel_p;
215.             */
216. }
217./* USER CODE END Includes */
218./* Private typedef -----------------------------
     */
219./* USER CODE BEGIN PID */
220./* USER CODE END PID */
221./* Private define ------------------------------
     */
222./* USER CODE BEGIN PID */
223./* USER CODE END PID */
224./* Private macro -------------------------------
     */
225.int _write(int file, char *ptr, int len)
226.  /* Reset of all peripherals, Initializes the
     Flash interface and the Systick. */
227.  HAL_Init();
228.  /* USER CODE BEGIN Init */
229.  /* USER CODE END Init */
230.  /* Configure the system clock */
231.  SystemClock_Config();
232.  /* USER CODE BEGIN SysInit */
233.  /* USER CODE END SysInit */
234.  /* Initialize all configured peripherals */
235.  /* USER CODE BEGIN WHILE */
236.  acrobat_robot_int();
237.  HAL_TIM_Encoder_Start(&htim1,
238.  while (1)
239.  {
240.  //**************Acrobat robot
241.              //  dir1=1;
242.              //  dir2=-1;
243.   //vel_select();
244.   acrobat();
245.   motor_dir();
246.   angulear_pos();
```

```
247.   conv_data();
248.            htim3.Instance->PSC=mot1_sp;
249.            htim9.Instance->PSC=mot2_sp;
250.            enc_old=enc;
251.            enc=TIM1->CNT;
252.            enc2=__HAL_TIM_GET_COU
    NTER(&htim1);
253.            // Get ADC value
254.            HAL_ADC_Start(&hadc1);
255.            HAL_ADC_PollForConversion(
    &hadc1, HAL_MAX_DELAY);
256.            angl1 =
    HAL_ADC_GetValue(&hadc1);
257.            HAL_ADC_Start(&hadc2);
258.            HAL_ADC_PollForConversion(
    &hadc2, HAL_MAX_DELAY);
259.            angl2 =
    HAL_ADC_GetValue(&hadc2);
260.            HAL_ADC_Start(&hadc3);
261.            HAL_ADC_PollForConversion(
    &hadc3, HAL_MAX_DELAY);
262.            angl3 =
    HAL_ADC_GetValue(&hadc3);
263.    /* USER CODE END WHILE */
264.    MX_USB_HOST_Process();
265.    /* USER CODE BEGIN 3 */
266.  }
267.  /* USER CODE END 3 */
268.}
269./**
270.  * @brief System Clock Configuration
271.  * @retval None
272.  */
273.void SystemClock_Config(void)
274.{
275.  RCC_OscInitTypeDef
    RCC_OscInitStruct = {0};
276.  RCC_ClkInitTypeDef
    RCC_ClkInitStruct = {0};
277.  RCC_PeriphCLKInitTypeDef
    PeriphClkInitStruct = {0};
278.  /** Configure the main internal regulator
    output voltage
279.  */
280.    Error_Handler();
281.  }
282. PeriphClkInitStruct.PeriphClockSelection
    = RCC_PERIPHCLK_I2S;
283.  PeriphClkInitStruct.PLLI2S.PLLI2SN =
    192;
284.  PeriphClkInitStruct.PLLI2S.PLLI2SR =
    2;
285.  if
    (HAL_RCCEx_PeriphCLKConfig(&Perip
    hClkInitStruct) != HAL_OK)
286.  {
287.    Error_Handler();
288.  }
289.}
290./**
291.  * @brief ADC1 Initialization Function
292.  * @param None
293.  * @retval None
294.  */
295.static void MX_ADC1_Init(void)
296.{
297.  /* USER CODE BEGIN ADC1_Init 0 */
298.  /* USER CODE END ADC1_Init 0 */
299.  ADC_ChannelConfTypeDef sConfig =
    {0};
300.  /** Configure for the selected ADC
    regular channel its corresponding rank in
    the sequencer and its sample time.
301.  */
302.  sConfig.Channel = ADC_CHANNEL_2;
303.  sConfig.Rank = 1;
304.  sConfig.SamplingTime =
    ADC_SAMPLETIME_3CYCLES;
305.  if (HAL_ADC_ConfigChannel(&hadc2,
    &sConfig) != HAL_OK)
306.  {
307.    Error_Handler();
308.  }
309.  /* USER CODE BEGIN ADC2_Init 2 */
310.  /* USER CODE END ADC2_Init 2 */
311.}/**
312.  * @brief ADC3 Initialization Function
313.  * @param None
314.    if (HAL_ADC_Init(&hadc3) !=
    HAL_OK)
315.  /** Configure for the selected ADC
    regular channel its corresponding rank in
    the sequencer and its sample time.
316.  */
317.  sConfig.Channel = ADC_CHANNEL_3;
318.  sConfig.Rank = 1;
319.  sConfig.SamplingTime =
    ADC_SAMPLETIME_3CYCLES;
320.  if (HAL_ADC_ConfigChannel(&hadc3,
    &sConfig) != HAL_OK)
321.  {
322.    Error_Handler();
323.  /* USER CODE BEGIN I2S3_Init 0 */
324.  /* USER CODE END I2S3_Init 0 */
325.  /* USER CODE BEGIN I2S3_Init 1 */
326.  /* USER CODE END I2S3_Init 1 */
```

```
  hi2s3.Instance = SPI3;
  hi2s3.Init.Mode = I2S_MODE_MASTER_TX;
  hi2s3.Init.Standard = I2S_STANDARD_PHILIPS;
  hi2s3.Init.DataFormat = I2S_DATAFORMAT_16B;
  hi2s3.Init.MCLKOutput = I2S_MCLKOUTPUT_ENABLE;
  hi2s3.Init.AudioFreq = I2S_AUDIOFREQ_96K;
  hi2s3.Init.CPOL = I2S_CPOL_LOW;
  hi2s3.Init.ClockSource = I2S_CLOCK_PLL;
  hi2s3.Init.FullDuplexMode = I2S_FULLDUPLEXMODE_DISABLE;
  if (HAL_I2S_Init(&hi2s3) != HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_Encoder_Init(&htim1, &sConfig) != HAL_OK)
  {
GPIO pin : PDM_OUT_Pin */
  GPIO_InitStruct.Pin = PDM_OUT_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
HAL_GPIO_Init(PDM_OUT_GPIO_Port, &GPIO_InitStruct);
  /*Configure GPIO pin : B1_Pin */
  GPIO_InitStruct.Pin = B1_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_EVT_RISING;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(B1_GPIO_Port, &GPIO_InitStruct);
  /*Configure GPIO pin : BOOT1_Pin */
  GPIO_InitStruct.Pin = BOOT1_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  HAL_GPIO_Init(BOOT1_GPIO_Port, &GPIO_InitStruct);
  /*Configure GPIO pin : CLK_IN_Pin */
  GPIO_InitStruct.Pin = CLK_IN_Pin;
  GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  GPIO_InitStruct.Alternate = GPIO_AF5_SPI2;
  HAL_GPIO_Init(CLK_IN_GPIO_Port, &GPIO_InitStruct);
  /*Configure GPIO pins : LD4_Pin LD3_Pin LD5_Pin LD6_Pin
  Audio_RST_Pin */
  GPIO_InitStruct.Pin = LD4_Pin|LD3_Pin|LD5_Pin|LD6_Pin
HAL_GPIO_Init(OTG_FS_OverCurrent_GPIO_Port, &GPIO_InitStruct);
  /*Configure GPIO pin : PB5 */
  GPIO_InitStruct.Pin = GPIO_PIN_5;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);
/* USER CODE BEGIN 4 */
/* USER CODE END 4 */
/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
void Error_Handler(void)
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  /* USER CODE END Error_Handler_Debug */
}
#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
```

```
399.  /* User can add his own implementation
      to report the file name and line number,
400.     ex: printf("Wrong parameters value:
      file %s on line %d\r\n", file, line) */
401.  /* USER CODE END 6 */
402.}
403.#endif /* USE_FULL_ASSERT */
404./************************ (C)
      COPYRIGHT STMicroelectronics
      *****END OF FILE****/
```

## A2. Controllability and Observability MATLAB code

```
A=[     0         0         0         1         0         0
        0         0         0         0         1         0
        0         0         0         0         0         1
        0    2.6835   -0.0657   -0.0286   -0.0083    0.0284
        0   29.2751  -15.8236   -0.0391   -0.1957    1.2358
        0  -57.5286  247.5924    0.0589    1.4085  -18.0527
   ];

B=[0 0 0 1.0314 1.6582 -2.4837];

C=[ 1 0 0 0 0 0;
    0 1 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 1 0 0;
    0 0 0 0 1 0;
    0 0 0 0 0 1];

D = zeros(6,1);

Co = ctrb(A,B);
% controbillty matrix

Ob = obsv (A,C);
% Observability matrix

if  rank (Co) == size (A,1)
   'It is controllable'
else
  'It is not Controllable'
end


if  rank (Ob) == size (A,1)
   'It is observable'
else
  'It is not observable'
end
```

## A3. Convergence percentage MATLAB code:

```
G = load('Figure1.fig', '-mat');
h  = findobj(gca, 'Type', 'line');
x1 = get(h(1), 'XData')
y1 = get(h(1), 'YData')
x2 = get(h(2), 'XData')
y2 = get(h(2), 'YData')

diff 1= x1 - y1;
diff 2= x2 – y2;

abs_diff = abs(diff1);
mean_diff = mean(abs_diff1);
convergence = mean(diff1);
 % Calculate the convergence percentage
convergence_percentage = (1 - sum(abs_diff1) / sum(abs(diff1))) * 100;
fprintf('Convergence Percentage: %0.2f%%\n', convergence_percentage);


% Display the convergence percentage on the figure
text(5, 8, sprintf('Convergence Percentage: %.2f%%', convergence_percentage))
```

## A4. Rotary Encoder (1st joint):

The SICK TTL DBS36 Incremental encoder offers a sturdy, compact, and simple-to-install solution for numerous speed and position tasks. It generates 1024 PPR at its output, and its 1.5m cable can be arranged either axially or radially, based on the installation requirements, as depicted in Figure A4.1. The encoder features an 8mm hollow shaft, which can be downsized to 6mm using the collet accessory 2013656. The encoder's stator coupling is designed with slots to allow for flexible mounting on a Pitch Circle Diameter (PCD) between 42-46 mm. The encoder can operate within a wide temperature spectrum of -20 to +85 degrees Celsius and is IP65-protected, guaranteeing reliable functionality even in rigorous operating conditions. It operates on a current of 50 mA and a supply voltage ranging from 4.5 to 5.5V.

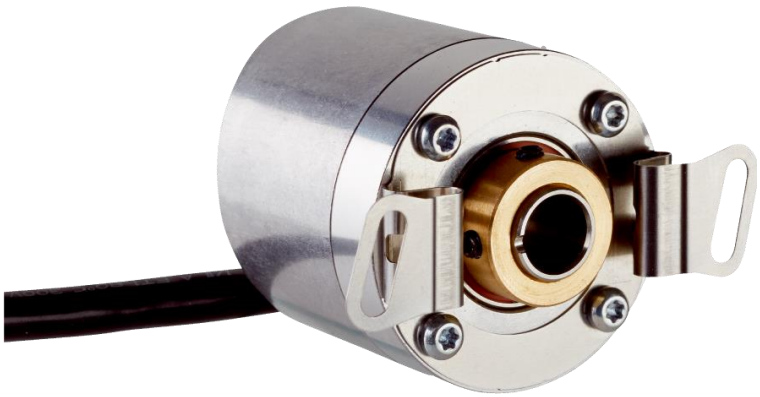

Figure A4. 1 Rotary Encoder

- **Potentiometer:**

The Rotary potentiometer is depicted in Figure A4.2. SP22E-10K is a High-Precision Single Turn Potentiometer that features a conductive plastic element, gold-plated terminals, a high-temperature thermoplastic housing, and a stainless-steel shaft. It has an electrical angle of 320 ±5° and a mechanical angle of the same degree.

Figure A4. 2 Rotary potentiometer

The encoder and potentiometers are employed to collect accurate data, which is subsequently transmitted to the microcontroller. This data is displayed as output on a computer via the STM32F Microcontroller, as shown in Figure A4. 3. To facilitate this, the rotary encoder and potentiometers are connected to the suitable pins on the microcontroller.

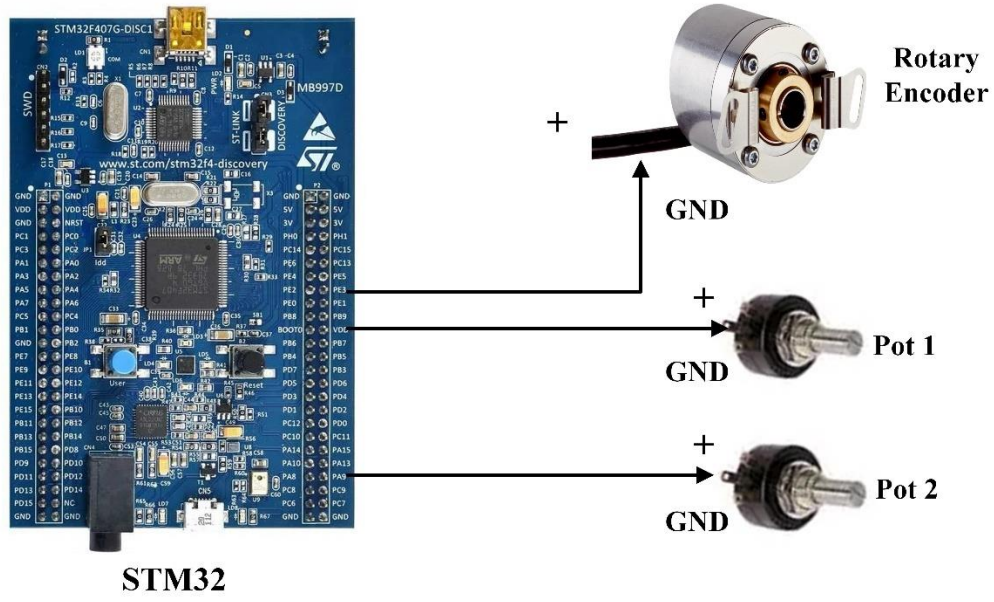


Figure A4. 3 STM32F- Encoders Connection

## Appendices

### A5. Stepper motor driver:

As stated in [117] and depicted in Figure A5, CL42T is a closed-loop stepper driver engineered to tackle the issue of step losses in open-loop stepper control systems. This enhances system dependability with a minimal raise in cost. CL42T leverages sophisticated control algorithms derived from decades of expertise in stepper and servo controls. It's remarkably reliable, cost-effective, and excels in various industrial applications such as CNC, medical, electronics, and packaging. The CL42T can drive 2-phase NEMA11, 14, and 17 stepper motors with incremental encoders, but the encoder resolution must be a 1000-line. In comparison to traditional open-loop stepper systems, the CL42T's closed-loop system can eradicate the risk of lost steps, execute real-time position error corrections, and doesn't necessitate torque reservation (it enables 100% torque implementation). Furthermore, it can power the driven stepper motor with decreased heating, reduced noise, and lower vibration.

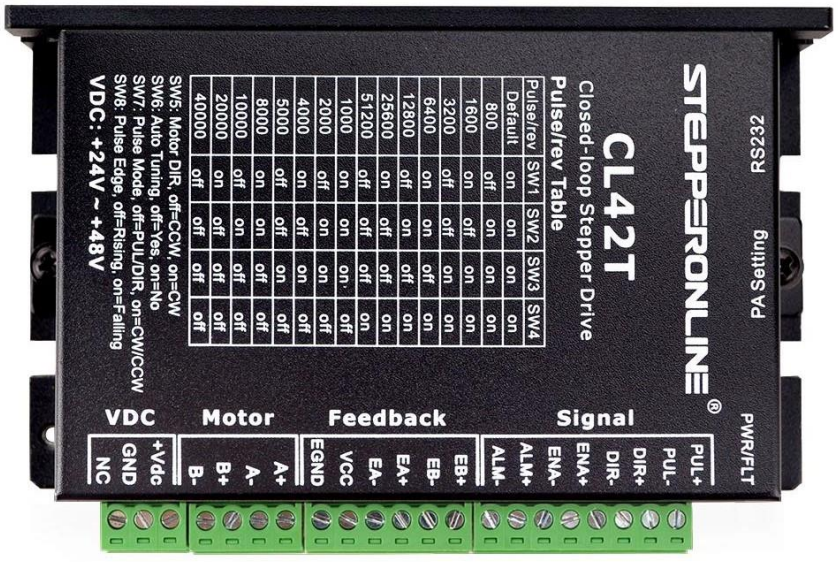


Figure A5 Stepper Driver 1

## Appendices

Table A5 Specifications of Stepper Driver

| Key Features: | Electrical Specifications | Operating Environment |
|---|---|---|
| No loss of step | Output Peak Current: 0~3 A | |
| No torque reservation | Input Voltage: +24~48VDC (Typical 24VDC) | Cooling: Natural Cooling or Forced cooling |
| No hunting or overshooting | Logic Signal Current: 7~16mA (Typical 10mA) | Environment: Avoid dust, oil fog and corrosive gases |
| No tuning for easy setup | Pulse Input Frequency: 0~200kHz | Ambient Temperature: 0℃ – 65℃ |
| 24-48VDC supply voltage, max 3A output current | Pulse Width: 2.5μS | Operating Temperature: 0℃ – 50℃ |
| Max 200 kHz input frequency | Isolation Resistance: 500MΩ | Vibration: 10-50Hz / 0.15mm |
| 15 micro step settings of 800-51,200 via DIP switches, or 200-51,200 via software (increase by 200) | | Storage Temperature: -20℃ – 65℃ |
| Protections for over voltage, over current and position following error | | |

## Appendices

### A6. Power supply:

The Professional 150W 48V 3.1A Switching CNC Power Supply is appropriate for a broad spectrum of applications in Industrial Automation and CNC Stepper/Servo Systems. It is operable with either a 115V or a 230V power supply, with the option selectable via a switch, as depicted in Figure A6.

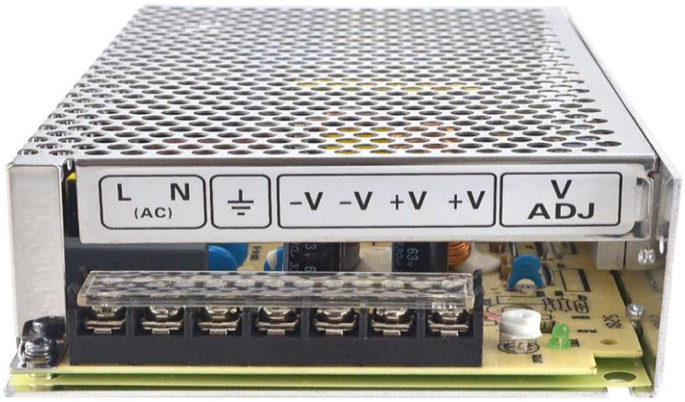


Figure A6 Power Supply

This power supply comes equipped with potent features such as PWM control, which ensures superior efficiency and dependability. Its professional design further bolsters its reliability and sturdiness. Moreover, the low cost of the Professional 150W 48V 3.1A Switching CNC Power Supply renders it an appealing choice for general use. Some of the key features of the power supply are highlighted in Table A6.

Table A6 Power Supply Features

| | |
|---|---|
| 48V DC 3.1 A output | High efficiency low cost |
| AC input voltage range:92~132V/180~264VAC | Free air-cooling convection |
| 115V/230V AC selected by switch | Over current, over voltage, short circuit and overheat protections |