Cape Peninsula
University of Technology

Optimisation of the stability of an assistive lower limb exoskeleton

By

Robert Anthony

Thesis submitted in fulfilment of the requirements for the degree

Master of Engineering: Mechanical Engineering

in the Department of Mechanical and Mechatronic Engineering

Faculty of Engineering and the Built Environment

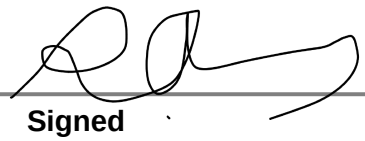
at the Cape Peninsula University of Technology

Supervisor: Michael Petersen
Co-supervisor: Professor Graeme John Oliver

Bellville
Date submitted December 2023

DECLARATION

I, Robert Anthony, declare that the contents of this dissertation/thesis represent my own unaided work, and that the dissertation/thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.



29 February 2024

Signed **Date**

ABSTRACT

Currently, most patients who have suffered from a stroke are treated through-one-on rehabilitation training administered by a physiotherapist. Powered Lower Limb Exoskeletons (PLLEs) have been designed to bring about advantages over traditional treatment. Although PLLEs have been invented overseas it has done little help in South Africa (SA), apart from sparking interest and showing what could be possible if research and development are put into this technology. It is hoped that this research will help solve gaps in the existing body of literature, from a South African perspective.

This thesis focused on the optimisation of stability of a PLLE, aiming to enhance the walking gait and overall performance of the device. The research explored two main approaches: the Linear Inverted Pendulum Model (LIPM) and an optimisation technique utilising a cost function and genetic algorithm.

The study involved the use of software in a simulation environment using MATLAB, which allowed for the prediction of hardware requirements and optimisation for stability. The requirements and desirable traits of a stable PLLE were identified through a literature review.

The design considerations encompassed a realistic size and weight based on existing exoskeletons. Using existing motors and motor controllers, a test platform was designed to closely represent the simulation model. Although the test platform did not achieve successful walking functionality, its development process yielded invaluable insights and knowledge, contributing to a deeper understanding of the challenges and complexities involved in attaining the desired performance.

The simulated results were analysed based on predetermined criteria, including the distance walked within 10 seconds, maximum torque of ankle, knee, and hip joints, torso height for upright walking, and human-likeness of the gait. The findings highlighted the achievement of a stable gait through the LIPM approach while the optimisation technique showcased progressive improvements throughout iterations, leading to a refined walking pattern.

This thesis contributes to the field of PLLEs, providing valuable insights for the development of more stable and human-like walking gaits. It is hoped that this outcome will help to form the groundwork for a rehabilitation system which would improve mobility for SCI patients and support healthcare professionals such as occupational therapists.

ACKNOWLEDGEMENTS

I wish to thank:

- Gina, my wife, who I owe everything to.
- My mom, Nicolette, who has always supported my learning and financed my studies.
- My dad, Johan, who was a great example of a lifelong learner.
- My grandma, Pam, who has encouraged me along the way.
- Mr Petersen, my supervisor, for his invaluable guidance, mentorship, and expertise.

DEDICATION

For Gina

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ABBREVIATIONS

BLDC	Brushless Direct Current
CAD	Computer Aided Design
CAN	Controller Area Network
CoM	Centre of Mass
CPUT	Cape Peninsula University of Technology
DD	Direct Drive
D-H	Denavit-Hartenberg
DOF	Degree of Freedom
EMG	Electromyography
FDA	Food and Drug Administration
GA	Genetic Algorithm
LIPM	Linear Inverted Pendulum Model
LLE	Lower Limb Exoskeleton
MPC	Model Predictive Control
PEA	Parallel Elastic Actuator
PoE	Product of Exponentials
PLA	Polylactic Acid
PLLE	Powered Lower Limb Exoskeleton
QDD	Quasi-Direct Drive
RoMeLa	Robotics and Mechanisms Laboratory
ROS	Robot Operating System
SA	South Africa
SCI	Spinal Cord Injury
SEA	Series Elastic Actuators
TSCI	Traumatic Spinal Cord Injury
URDF	Unified Robot Description Format
XML	Extensible Markup Language
ZMP	Zero Moment Point

CHAPTER 1

INTRODUCTION

1.1 Statement of the Research Problem

The ability to travel from one place to another, known as mobility, is a crucial factor for individuals to actively participate in various aspects of professional, social, and economic life. Being able to travel is essential for individuals to engage in activities such as employment, education, healthcare, and social interactions. However, if personal travel capabilities are limited, individuals may face challenges in participating fully in these activities, leading to inadequate integration into the labour market, loss of independence, and reduced social interactions, ultimately impacting their quality of life (Neven & Ectors, 2023).

In South Africa, the overburdened health system poses challenges for individuals to access existing rehabilitation methods and assistive devices, resulting in reduced mobility and thus a reduced quality of life for those who have suffered a stroke or spinal cord injury (Magaqa, Ariana & Polack, 2021). The limited availability and accessibility of rehabilitation services, including assistive devices, contribute to unmet needs among individuals with disabilities (Magaqa, Ariana & Polack, 2021). This lack of access to necessary care and resources can lead to inadequate integration into the labour market, loss of independence, and a lack of social interactions, ultimately impacting the overall well-being of individuals (Magaqa, Ariana & Polack, 2021).

PLLEs are said to have become a viable tool in helping people who have suffered a Spinal Cord Injury (SCI) restore mobility (Schrade et al., 2019). In a survey by van Silfhout et al. (2023), it is stated that, 'Individuals with SCI considered exoskeletons as a positive and desirable innovation. But based on the findings from the surveys, major points of improvement are necessary for exoskeletons to replace wheelchairs in the future.' In the same study 74.7% of the participants expressed favourable views towards this innovative technology. However, some individuals who held negative opinions about exoskeletons criticised their impracticality due to the requirement of using crutches, perceived slowness, and robot-like appearance. Among the respondents, 24 individuals (25.3%) anticipated that exoskeletons could potentially replace wheelchairs in the future. Nevertheless, concerns were raised regarding the necessity of crutches for ambulation in exoskeletons, which could pose challenges in carrying objects. Notably, respondents emphasised the need for future improvements in exoskeleton design to accommodate individuals with higher spinal cord injuries and enable ambulation without the reliance on crutches, thereby facilitating the carrying of items. Additional areas for enhancement, as identified by the participants, included independent walking in exoskeletons,

traversing uneven surfaces, reduced weight, and simplifying putting the exoskeleton on (van Silfhout et al., 2023).

1.2 Background to the Research Problem

For over a decade, there has been an increase in research on bipedal robots as well as PLLEs. One of the reasons for this increase is that technology has advanced to the point where PLLEs have become a viable tool in helping people who have suffered a spinal cord injury restore mobility (Schrade et al., 2019). Apart from SCI, surviving stroke patients are also able to benefit from robotic-assisted rehabilitation training (Wu & Wu, 2018). There are many health benefits associated with standing up and walking after SCI, both physically and psychologically. The physical advantages encompass enhanced bone density, improved cardiovascular and respiratory function, better gastrointestinal function, improved sitting balance, and reduced pain and spasticity (Geigle & Kallins, 2017). Psychological confidence is promoted in patients who have an accelerated recovery of their motor function due to robot-assisted training (Wu & Wu, 2018).

In South Africa, we have an overburdened health system, which means many people do not have access to existing methods of rehabilitation or assistive devices. This leaves many people with a reduced quality of life after suffering a stroke or SCI where there may have been a possibility for them to recover. Statistics South Africa (2011) states that “population group variations show that white persons had a higher proportion of persons who have access to assistive devices, while black Africans had the lowest proportions for all types of assistive devices” and that “urban dwellers had higher proportions of persons using assistive devices for all types of assistive devices compared to rural areas”. This reiterates the issue that many South Africans do not have access to the necessary care needed. There is little epidemiological data on SCI in South Africa, but there are population estimates on the number of traumatic spinal cord injury (TSCI) incidences. The reported figures indicate an incidence rate of 75.6 cases per million individuals. The primary cause was identified as assault, accounting for 60% of all incidents, followed by transport-related causes (26%), while falls constituted the remaining 12% (Joseph et al., 2017).

In the same paper by Joseph et al. (2017), it was made clear that both the primary prevention of SCI and the prevention of scenarios which hinder recovery are hugely important. If SCI affected rural South Africans are to lead a better life with more independence, they need improved access to treatment and for their environment to cater for them to travel freely so that they can gain employment.

It is clear that globally, PLLEs and bipedal robots deserve to be researched and developed to help improve the lives of many. In SA, the research of this topic should aim to make this technology not only effective, but affordable for it to be accessible to the greater population.

South African universities have made progress in sectors such as medical and engineering services (Ndabeni, Rogerson & Booyens, 2016). The government has also recognised the importance of science, technology, and innovation (Ndabeni, Rogerson & Booyens, 2016). By continuing to prioritise innovation and knowledge integration, South African universities can contribute to regional and local development, strengthen local systems of innovation, and drive the transformation of the economy to a knowledge-based one (Ndabeni, Rogerson & Booyens, 2016). Although PLLEs have been invented overseas it has done little to help us here but to spark an interest and show us what could be possible if we put our minds together and give our time towards researching and developing this technology; specifically in a way that our country will be able to use it. It is hoped that this research will help solve gaps in the existing body of literature, from a South African perspective.

1.3 Research questions

The optimisation of stability of a PLLE is a critical area of research that aims to enhance the naturalness and human-like quality of the walking gait. This section presents a set of research questions that will guide the investigation into achieving optimal stability in a PLLE. These questions encompass various aspects, including the hardware requirements for untethered walking, the utilisation of simulation environments for stability optimisation, objective assessment criteria for evaluating walking gait, and the comparison of different optimisation techniques.

- How can stability be optimised in a PLLE to achieve a more natural and human-like walking gait?
- What are the hardware requirements necessary for the untethered walking (without crutches) of a PLLE, and how can these be predicted through simulation?
- Can a simulation environment using MATLAB be utilised as a tool to optimise stability and inform design decisions for a PLLE?
- What are the criteria for objectively assessing the stability and human-likeness of the walking gait in a PLLE, and how do different optimisation techniques compare in meeting these criteria?

By addressing these research questions, this study aims to contribute to the advancement of knowledge and understanding in the field of PLLEs, ultimately leading to the development of more stable and human-like walking assistance devices.

1.4 Research Objectives

The main objective of this research is to use mathematical modelling to predict the hardware requirements for untethered walking of a PLLE with a human payload, and to use this simulation to optimise for stability in a simulated environment using MATLAB. As well as to design a scaled test platform to gain insight into the process of going from a simulated environment to hardware. The points below illustrate how the researcher has split the overarching objective to accomplish the goals of the research.

- Design a simple to-scale model of a PLLE.
- Implement robust controllers or a gait pattern generator which will enable the model to walk within a virtual environment.
- Simulate the system and analyse the results using MATLAB.
- Optimise the system for stability.
- Build a test platform.
- Test optimisation.

1.5 Budget

The initial design and development phase of this project will not need a budget as they will be carried out in software. CPUT has a license for MATLAB which is the primary software that will be used. However, a scaled test platform will be developed to validate the system. The researcher will use parts available to them, free of charge, at their place of employment, and return these parts after the research is completed.

1.6 Delineation of the research

- This research will focus on an assistive PLLE. Augmentation and rehabilitation type robots will not be considered.
- Only lower limbs will be considered, the design will be of a powered lower limb locomotion assistive robot.
- This research will be limited by the amount of time available for simulation as well as the software available for use.
- The research will focus on a PLLE that does not have feedback control. It will not respond to varying levels of input effort.
- Specific actuator parameters will be neglected but joint controllers will be used and tuned.

1.7 Significance of the research

If this research proves successful, it will establish the foundation for an assistive PLLE. The assistive PLLE would be aimed at improving the daily life of individuals who have suffered a SCI by improving mobility and aiding treatment by occupational therapists.

The significance of enabling untethered walking holds immense implications for individuals grappling with Spinal Cord Injuries or complete immobility in their lower limbs. Untethered walking, particularly for individuals with SCI or those with limited mobility in their lower limbs, holds significant importance in the field of assistive technologies. The ability to walk without the need for external support, such as crutches, can greatly enhance the mobility and independence of individuals with SCI. Research has shown that PLLEs have the potential to improve walking mechanics, energetics, and overall walking performance in individuals with SCI (Chang et al., 2017; Orekhov et al., 2021). Studies have also explored the physiological aspects of using powered exoskeletons which investigated the heart rate and oxygen demand of powered exoskeleton-assisted walking in individuals with paraplegia (Asselin et al., 2015). The study found that persons with paraplegia were able to ambulate efficiently using a powered exoskeleton, potentially leading to functional gains and improved fitness (Asselin et al., 2015).

The development of optimized stability in PLLEs can contribute to the restoration of walking function, allowing individuals to engage in daily activities, improve their quality of life, and potentially reduce the risk of secondary health complications associated with immobility (Orekhov et al., 2021; Naka et al., 2022). Understanding the hardware requirements, simulation methodologies, and objective assessment criteria for optimising stability in exoskeletons is crucial for the development of effective and user-friendly devices (Orekhov et al., 2021).

By addressing the research questions related to stability optimisation in PLLEs, this research aims to advance the field of assistive technologies and contribute to the improvement of mobility and independence for individuals with SCI or limited mobility in their lower limbs.

1.8 Expected outcomes, results, and contributions of the research

It is expected that the outcome of this research will produce a working simulated model and a scaled test platform of a PLLE that can perform a dynamic walk on flat terrain.

This simulation will aid in design choices for actuators, materials, linkage setups, etc. by providing valuable information as to how the design may act if it were built. In this way, many

iterations can be tested through software before any hardware has to be purchased, which in turn saves cost.

This simulation, along with the test platform, should provide the groundwork and form a solid platform for CPUT to further develop this technology and potentially solve global research gaps in the future.

1.9 Chapter outline

This chapter outlines the structure and content of the thesis, providing an overview of the chapters and their respective focus areas.

Chapter 1: Introduction
This chapter has introduced the research problem, presented the background and context of the study, and outlined the research questions and objectives. It also provided information about the budget allocated for the research and delineated the scope and significance of the study. The expected outcomes, results, and contributions of the research were discussed in this chapter.

Chapter 2: Literature Review
This chapter presents a comprehensive review of the existing literature, relevant to PLLEs. It explores the classification of PLLEs, surveys existing designs, and discusses the concepts of anthropomorphism and ergonomics. The chapter also covers the main hardware components, LLE model design, gait and trajectory considerations, control strategies, controllability, stability, optimisation, and simulation methodologies.

Chapter 3: Simulation Methodology
The methodology chapter outlines the research methodology employed in this study. It covers design considerations, CAD model development, and the MATLAB and Simulink workflow. The chapter details the application of the Linear Inverted Pendulum Model (LIPM) and an optimisation technique using a genetic algorithm.

Chapter 4: Simulation Results
This chapter presents the results obtained from the LIPM simulations and the optimisation process. It provides an analysis of the LIPM results, the best optimisation outcomes, and a comparison between the different results.

Chapter 5: Attempted Physical Validation of Aspects of the Numerical Modelling

This chapter focuses on the design considerations, requirements, and constraints of the test platform. It discusses the simulation, hardware and electronics, mechanical design, and performance testing of the demonstrator.

Chapter 6: Test Platform Experimental Methodology

This chapter presents the experimental methodology employed to evaluate the performance and functionality of the test platform. This chapter outlines the steps taken to conduct the experimental trials, including the setup, data collection procedures, and measurement techniques. It also discusses the specific objectives and hypotheses tested during the experiments.

Chapter 7: Test Platform Results

This chapter presents the results and outcomes of the test platform's performance. It discusses the first attempt at a ground contact walk, and the observations made during the testing phase.

Chapter 8: Discussion and Conclusion

The discussion and conclusion chapter summarises the findings of the research and their implications. It provides a concise overview of the contributions made by the study and offers insights into the future directions and potential applications of PLLEs.

Chapter 9: Recommendations

This chapter offers recommendations based on the research findings and presents suggestions for improving the design and control of PLLEs.

CHAPTER 2

LITERATURE REVIEW

This chapter will examine existing legged platforms to gain insight into their respective strengths and/or compromises. The information collated here will help to identify gaps in current research and provide the foundational knowledge needed to design and simulate a PLLE to optimise for stability and lead to a solution for untethered walking.

2.1 Introduction

LLEs are robotic structures worn by humans to augment the ability of the user's legs. They are designed for multiple use cases such as to help people in need of gait rehabilitation due to injury, disease or a neurological condition (Anam & Al-Jumaily, 2012). Those with complete paralysis of their lower limbs are also able to benefit from LLEs as a locomotion device. Another application is strength augmentation which increases the natural ability of the user enabling them to carry loads greater than they are physically capable of.

In a paper by Pons (2010), it was stated that the field of exoskeleton research began as early as the 1960s but has only recently been applied to rehabilitation and locomotion assistive use cases. As this technology evolves to focus more on human-robot interaction there is an emphasis on the need for a platform which is safe, robust and user friendly.

2.2 Classification of PLLEs

Chen et al. (2016) describe three broad categories of LLEs. The first is for gait rehabilitation which is designed to aid people with walking disabilities restore mobility by supplementing the strength of their muscles during rehabilitation training.

The second category are LLEs designed for human strength augmentation of nondisabled people such as military personnel which enable the user to carry heavy loads.

The third category is the assistive type which provides locomotion assistance to people who have complete paralysis of their lower limbs, allowing them to stand up and walk. This is a replacement of wheelchairs and as previously mentioned has health advantages as well as greater access to the environment. This category is the focus of this research.

To clarify, gait rehabilitation and locomotion assistive powered exoskeletons are two distinct approaches in the field of assistive technologies. Gait rehabilitation exoskeletons are primarily designed for therapeutic purposes, aiming to assist individuals with impaired gait patterns in relearning and improving their walking abilities (Rodríguez-Fernández, Lobo-Prat & Font-

Llagunes, 2021; Zhu et al., 2021). These exoskeletons often incorporate active joints at the hip and knee. The ankle joint is typically passive, as its main functions during walking are body weight support and propulsion (Chen et al., 2016). Gait rehabilitation exoskeletons are commonly used in clinical settings, where patients wear the exoskeletons for gait retraining or rehabilitation, often in combination with stationary facilities such as treadmills (Hsieh et al., 2020).

On the other hand, locomotion assistive powered exoskeletons are designed to aid the walking abilities of individuals with mobility impairments, such as those with paraplegia or lower limb weakness (Esquenazi et al., 2012; Chen et al., 2016). These exoskeletons aim to enable individuals to walk independently and perform activities of daily living with reduced reliance on external support, such as crutches (Esquenazi et al., 2012). Locomotion assistive exoskeletons often incorporate active joints at the hip, knee, and ankle, providing powered assistance to facilitate walking (Chen et al., 2016). The goal is to restore ambulatory function and enhance mobility, allowing individuals to engage in community ambulation and improve their overall quality of life (Esquenazi et al., 2012). While gait rehabilitation exoskeletons primarily focus on therapeutic interventions and retraining of gait patterns, locomotion assistive powered exoskeletons aim to provide functional assistance and augment the walking abilities of individuals with mobility impairments.

2.3 Survey of Existing Locomotion Assistive LLEs

This chapter is a short review of existing designs of powered locomotion assistive LLEs and their respective capabilities.

Assistive PLLEs, such as MINDWALKER, ReWalk, Vanderbilt, REX, and Wandercraft's Atalante, have emerged as promising technologies in the field of rehabilitation and mobility assistance (Esquenazi et al., 2012; Chen et al., 2016; Huynh et al., 2021).

MINDWALKER (Figure 2.1) is a PLLE made to assist paraplegics in being able to walk again. It has five DOF per leg, a number are powered by series elastic actuators (SEA) while the rest are passively sustained. Ground-level walking experiments were conducted using the MINDWALKER. Ten participants were involved – six health individuals and four SCI individuals. SCI patients had to hold a handrail to keep balance. Upper limb muscle measurements were shown through electromyography (EMG) patterns. The analysis of EMG patterns revealed interesting findings regarding the involvement of muscles during stepping with the exoskeleton. It was observed that the upper limb muscles exhibited increased activation during the stepping process, indicating augmentation provided by the exoskeleton. In contrast, the activation of leg muscles was relatively low or even absent. For healthy

participants, the EMG activities of leg muscles were comparable or slightly higher when walking with the exoskeleton compared to free-level walking. However, these activities were still lower compared to walking without the assistance of the exoskeleton (Yan et al., 2015).

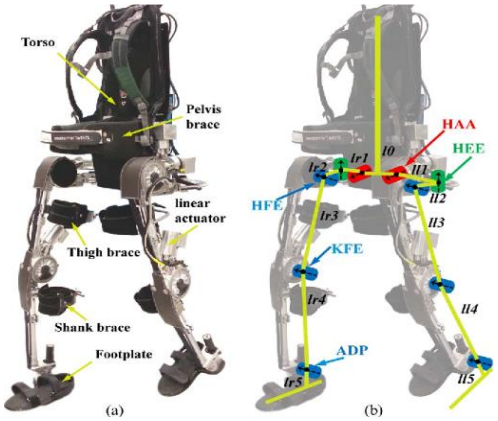


Figure 2.1: MINDWALKER

(Wang et al., 2015)

The ReWalk exoskeleton has shown potential in restoring ambulatory function to individuals with thoracic-level motor-complete spinal cord injuries (Esquenazi et al., 2012). There was a large amount of variability within the participants of the study but most were able to gain a level of walking that would allow for limited community ambulation (Esquenazi et al., 2012). ReWalk, as shown in Figure 2.2, incorporates hip and knee movements, providing support and mobility to users. Notably, in 2014, the ReWalk became the pioneering exoskeleton suit to receive approval from the U.S. Food and Drug Administration (FDA) as a personal device. This marked a significant milestone as it became the first commercially available walking exoskeleton, manufactured by Argo Medical Technologies (Baldovino & Jamisola, 2017).

Figure 2.2: ReWalk

(ReWalk Robotics, 2020)

The control system of the ReWalk exoskeleton relies on onboard computers and motion sensors to enable individuals with paralysis to regain self-initiated walking. By detecting the forward tilt of the upper body, the system mimics the natural gait pattern observed in able-bodied individuals. Clinical studies conducted with the ReWalk exoskeleton have demonstrated its effectiveness in helping paralysed patients stand upright and walk with increased independence, significantly improving their overall quality of life. Moreover, the outcomes of these studies have indicated a reduction in secondary complications associated with wheelchair-bound lifestyles, including a decrease in instances of depression among the patients (Chen et al., 2016).

The Vanderbilt exoskeleton, depicted in Figure 2.3 and developed by Goldfarb et al., is a LLE that offers paralysed patients the ability to perform essential movements such as walking, sitting, standing, and navigating stairs. Its modular-based design allows patients themselves to easily assemble and disassemble the exoskeleton. The hip and knee joints are actuated by two brushless direct current motors integrated within each thigh segment. Remarkably, the exoskeleton weighs a mere 12 kg, making it relatively lightweight compared to similar devices. A patient with sensory complete injury was outfitted with the Vanderbilt exoskeleton, resulting in a repeatable gait pattern characterised by hip and knee joint amplitudes similar to those observed during non-spinal cord injury walking. With the assistance of the Vanderbilt exoskeleton, the patient has regained the ability to stand up, sit down, walk, turn, and ascend or descend stairs, significantly enhancing their mobility and independence (Chen et al., 2016).

Figure 2.3: Vanderbilt

(Vanderbilt University, 2020)

The REX exoskeleton (Figure 2.4) is a PLLE assistive device designed to aid in the restoration of ambulatory function to people who suffer from thoracic-level motor-complete spinal cord injuries (Esquenazi et al., 2012). It offers the potential for safe ambulation and limited

community ambulation, which can allow users to walk with a level of proficiency that nears what is needed for limited community ambulation (Esquenazi et al., 2012). The REX exoskeleton is unique in that it enables untethered walking without the need for crutches, providing greater autonomy and freedom of movement (Chua & Kuah, 2017). This feature sets it apart from other exoskeletons that require auxiliary equipment or are fixed to a treadmill (Penzlin et al., 2021). The REX exoskeleton has shown promise in improving gait speed, functional exercise capacity, and overall quality of life for individuals with spinal cord injuries (Birch et al., 2017; Spratt, Spratt & Marquez, 2021). However, challenges remain, such as the slow gait speed and limited range of movement, which may impact its usability in certain situations (Riener, 2016). Further research and development are needed to optimise the REX exoskeleton and address these limitations, ultimately improving its effectiveness and expanding its potential applications in rehabilitation and mobility assistance. The REX system weighs 38 kg (Luo et al., 2021).

Figure 2.4: REX

(Birch et al., 2017)

Wandercraft's exoskeleton, Atalante (Figure 2.5), is a crutchless LLE designed to provide autonomy and freedom of movement for individuals with mobility impairments (Gurriet et al., 2020; Huynh et al., 2021). Atalante has demonstrated stable bipedal walking and dynamic motions, offering potential benefits for users with mobility limitations. The exoskeleton is designed to allow hands-free walking, enabling users to perform daily tasks without the need for additional stabilization tools such as crutches. Atalante has been developed with advanced mechatronic design and active cooperation between the user and the device, resulting in stable dynamic movements, more natural stand-up and turning motions, and improved physical condition for the user (Huynh et al., 2021). The Atalante PLLE weighs approximately 60 kg and can walk while carrying up to a 90 kg passenger (Luo et al., 2021).

Figure 2.5: Atalante by Wandercraft

(Gurriet et al., 2018)

Overall, the literature on assistive PLLEs, such as REX, ReWalk, and Wandercraft's Atalante, demonstrates their potential in restoring ambulatory function, improving mobility, and enhancing independence for individuals with mobility impairments.

2.4 Main Hardware Components

This section focuses on the main hardware components of legged robots, specifically actuators and sensors. Actuators are responsible for generating the necessary forces and torques to actuate the joints and enable legged locomotion. Sensors, on the other hand, provide feedback to the control system, allowing the robot to perceive its environment and adjust its movements accordingly. The selection and understanding of these hardware components are crucial for the successful design and operation of legged robots. In this section, we will discuss different types of actuators commonly used in PLLEs and legged robots, as well as various sensors employed for position, force, and torque sensing.

2.4.1 Actuators

It is pointed out by Katz (2018), who has developed a “low cost, high performance, modular actuator, intended for use in legged robots and other machines which dynamically interact with the world, make and break contact and experience collisions” that selecting or developing an actuator for legged robots is no easy task. There are different types of actuators which have been implemented successfully on PLLEs all having advantages and disadvantages. The ideal actuator would have specific torque, high actuation speed, high acceleration speed, low reflected inertia, impact robustness, a high force control bandwidth and high efficiency (Katz, 2018). A higher specific torque indicates that the actuator can generate more torque for its size or weight, which is desirable for applications where space and weight constraints are important

factors. Achieving high specific torque is crucial for designing efficient and compact legged robots and powered prostheses. Commonly used legged actuator types will be discussed below.

2.4.1.1 Direct Drive

Direct drive (DD) applications do not use reduction gearboxes, belts, or pulleys. The motor is used to drive the joint it is actuating directly, which means no torque amplification is applied.

The main advantage of this setup is the inherent transparency gained by mitigating negative characteristics of transmissions, such as backlash, reflected inertia and energy wasted due to friction. One can obtain actuation with high mechanical stiffness that benefits from minimal dynamics and a high level of efficiency as there are no losses from a transmission. Doing away with a gearbox also increases robustness as there are no gear teeth that can suffer damage from continuous impact stresses. Lastly, DD benefits from high-bandwidth signal flow; compliance can be implemented at kHz timescales, allowing the joint to react to different terrains and maintain balance (Kenneally, De & Koditschek, 2016).

One of the primary drawbacks of direct drive systems is the limited torque density compared to using a gearbox. This means that direct drive motors may not be able to deliver the same level of torque output as a motor coupled with a gearbox, which can be a disadvantage in applications requiring high torque. Additionally, direct drive systems can suffer from a loss of efficiency when operating in a high torque but low-speed scenario. The motor's efficiency decreases in such conditions, leading to reduced overall system efficiency. It's important to note that peak power and peak efficiency typically occur at or near the motor's no-load speed, further highlighting the challenges associated with operating direct drive systems in high torque situations (Kenneally, De & Koditschek, 2016).

Because of the power density and force density of direct drive being low, it may not be a viable option for an exoskeleton which has to support a human load (Pratt & Williamson, 1995). The deduction made in 1995 regarding the low power density and force density of direct drive actuators appears to still be relevant today. Since then, advancements in technology and the development of high torque density motors have led to the emergence of new actuation paradigms, such as quasi-direct drive, that offer improved performance for exoskeletons and legged robots. In more recent research papers (Aguilar-Sierra et al., 2015; Belogusev & Egorov, 2019) highlight the use of high torque density motors and low ratio transmission mechanisms to achieve compact and efficient actuation in wearable robots, however, there is no mention of PLLEs which support humans that employ direct drive actuation.

Although not PLLE's, Kenneally et al. (2016) described the effective application of DD on three legged platforms, namely; Delta Hopper, a monoped with three active DOF per leg; Minitaur, a quadruped with two active DOF per leg; and Penn Jerboa, a tailed biped with one DOF per leg. Refer to Figure 2.6 which shows the legged robots in order from left to right.

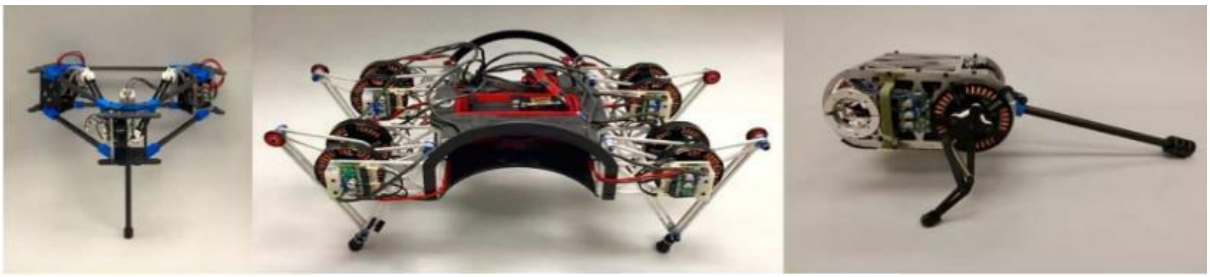


Figure 2.6: Delta Hopper, Minitaur and Penn Jerboa

(Kenneally, De & Koditschek, 2016)

2.4.1.2 Quasi-Direct Drive

Proprioceptive Electric actuators or Quasi-Direct Drive actuators (QDD) are a compelling choice for dynamic legged robots due to their high torque density, high speed capabilities, robustness to external impacts, and the potential for high bandwidth torque control when designed appropriately. These features make them well-suited for achieving agile, stable, and efficient locomotion in legged robotic systems (Katz, 2018; Zhang, 2019).

QDD's are built by coupling a high torque BLDC motor with a low gear ratio transmission such as a planetary gearbox. This combination achieves high actuation transparency and accurate and reliable force control through current sensing, eliminating the need for external force/torque sensors. Part of their attractiveness, especially in a research setting is their relatively low cost and ease of availability of components. This is due to the rapid development of the drone industry which mass produces high torque BLDC motors in many shapes and sizes (Zhang, 2019).

A notable application of a QDD actuator is in the MIT Cheetah robots. As a testament to their robustness, these quadrupeds can handle the impact forces of running at speeds up to 13.5 m/s (Zhang, 2019).

The Robotics and Mechanisms Laboratory (RoMeLa) have also been developing QDD actuators for use on their robots and legged platforms. NABi-2, a bipedal robot and ALPHRED-2, a quadruped, are two examples of applications where they have made use of their QDD actuator (Zhang, 2019).

2.4.1.3 Series Elastic Actuators

Series Elastic Actuators (SEAs) are a type of actuator commonly used in robotic systems that involve physical interaction with the environment. They are designed with a configuration that includes a highly geared electric motor connected in series with a compliant element, typically a spring, and a displacement sensor positioned at the output.

The use of a highly geared electric motor enables SEAs to achieve high gear ratios, which in turn provide advantages in terms of torque amplification and precise control. However, the inclusion of the compliant element, such as a spring, is the key distinguishing feature of SEAs. This compliant element acts as a mechanical buffer between the motor and the load, absorbing and releasing energy during physical interactions or impacts.

The presence of the compliant element offers several benefits in robotic applications. Firstly, it helps to protect the internal components, including the gears, from damage that may occur when interacting with a rigid environment or when subjected to sudden forces. The compliance of the actuator absorbs the impact energy, reducing the risk of gear or motor failure (Katz, 2018).

In this configuration, output force or torque is monitored and controlled by measuring and adjusting the spring deflection. SEAs are used in two humanoid robot platforms developed by RoMeLa, namely, THOR and SAFFiR. ATRIAS, a bipedal robot developed by the Oregon State University Dynamic Robotics Laboratory is another example where SEAs are used. Interestingly, all three of these legged platforms make use of carbon fibre plates as the spring portion of their SEA modules (Zhang, 2019).

A disadvantage of SEAs in a low-cost but high-performance application is both cost and complexity (Katz, 2018).

2.4.1.4 Parallel Elastic Actuators

Parallel elastic actuators (PEA) consist of a highly geared electric motor coupled to a spring in a parallel configuration and a spring displacement sensor at the output. A key feature is the ability of the spring element to share the load with the actuator. This is advantageous as the torque required of the actuator can be minimised, which means a smaller actuator can be used and less energy is required (Mazumdar et al., 2017).

There are numerous examples of bipedal walking robots which make use of a spring-like position vs. force configuration during walking. Two examples where parallel knee springs

have been utilized to reduce motor torque and power reduction are on the ERNIE and Phides robots. As powered robotic walking continues to approach human performance, Mazumdar et al. (2017) are confident that these walking robots will continue to benefit from parallel springs.

2.4.1.5 Hydraulic/Pneumatic Actuators

Hydraulic actuators are widely used in various applications due to their high force density and robustness, making them particularly suitable for tasks that require strong and reliable actuation. However, there are some disadvantages associated with hydraulic actuators that need to be considered.

One significant disadvantage is the relatively low efficiency of hydraulic systems. This inefficiency arises from several factors, including viscous losses in the moving fluid, internal leakage in the servo valves, and pressure drops throughout the system. These losses result in energy dissipation and reduced overall efficiency of the actuator (Katz, 2018).

Hydraulic systems as well as pneumatic systems need pumps and cylinders to operate with supply and return lines running to each cylinder. Pneumatics have the benefit of not needing return lines as they can exhaust into the atmosphere.

2.4.2 Sensors

This chapter will briefly discuss commonly used sensors on legged robots. Sensors are used to supply information to the control system of a legged robot so that it can react and adjust accordingly when it strays from its given trajectory or instruction. Essentially, they “sense” the environment or configuration the robot is in.

Perhaps the most common group of sensors used are position type sensors. For a legged robot to move its limbs in a precise way a sensor is generally required to output the position of the joint. Siciliano et al (2010) states that, “The aim of position transducers is to provide an electric signal proportional to the linear or angular displacement of a mechanical apparatus with respect to a given reference position.” Angular displacement transducers, such as encoders, are usually used on robots where electric actuators are used on prismatic joints. Encoders and resolvers, both which are used as angular position sensors, are the most common transducers due to their precision, robustness, and reliability. Linear displacement transducers can also be used to give feedback of the position of a prismatic joint (Siciliano et al., 2010).

An example of the use of encoders on a PLLE are on the powered joints of the BLEEX robot. Encoders are used for position sensing on each powered joint and the encoder is enclosed to protect it from damage. Refer to Figure 2.7.

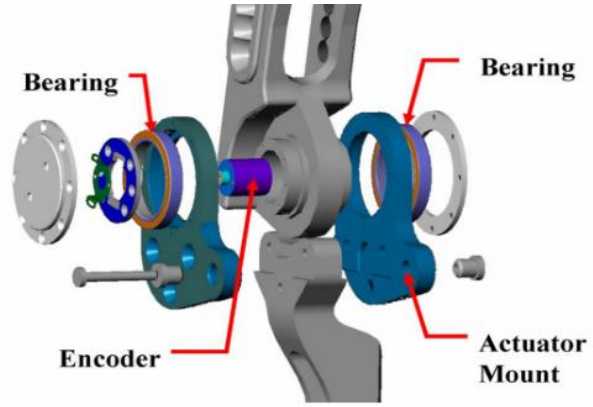


Figure 2.7: BLEEX Powered Joint

(Zoss, Kazerooni & Chu, 2005)

Force or torque sensors in legged robots use strain gauges, which measure the change in electrical resistance of a wire when subjected to strain. These sensors provide feedback on the forces and torques experienced during robot locomotion and interaction with the environment. By measuring the resistance changes, strain gauge sensors enable accurate detection and quantification of applied forces and torques. They are widely used due to their reliability, accuracy, and cost-effectiveness. These sensors enhance control, stability, and performance of legged robots by providing crucial information for motion planning and adaptation (Siciliano et al., 2010).

BLEEX is an example of a PLLE where a load or force distribution sensor has been implemented. The sensor is in the form of a pressure tube that is placed between the users' foot and the exoskeletons foot. The sensor is then used to measure the forces exerted by the users' feet. This data is fed to the control algorithm which then detects how much weight the user places on their left foot versus their right (Zoss, Kazerooni & Chu, 2005).

In a paper by Wu et al. (2019) force sensors were designed to be used on small legged robots. The authors point out that the terrain a legged robot walks on greatly varies the ground contact interactions it experiences. Different gaits also affect performance on different types of terrain, the same gait used on a hard slippery surface would not perform as well on a surface such as sand. By applying capacitive tactile sensors to the feet of a hexapod, they were able to use the feedback from the sensors to adjust the gait to suit different surfaces. The researchers observed a 17.1% increase in body speed and a 13.2% improvement in efficiency during a

surface transitioning test. This highlights the effectiveness of gait adjustment in enhancing the robot's mobility and energy efficiency.

Accelerometers, which measure acceleration forces on an object, are also a form of feedback device that has been used on legged robots to adjust gait, especially on uneven terrain. They have also been used to record actual human gait with which the data can be applied to trajectory generation for bipedal locomotion (Varma, Jolly & Suresh, 2018).

2.5 LLE Model Design

To design a multi-link robot model, many parameters must be considered. Depending on the control strategy implemented, the masses of each segment of the structure along with their varying lengths and DOF can all affect how the robot will move in space and what torques are needed to manipulate it. In the field of mechanics there have been multiple techniques and standard notations developed to aid one in recording the geometry of robots as well as calculating joint angles and joint torques to accomplish different tasks, such as forward and inverse kinematics and forward and the inverse dynamics.

2.5.1 Forward and Inverse Kinematics

Kinematics, a branch of mechanics, focuses on the motion and relationships between objects or points within a system, without considering forces (Housner & Hudson, 1983). It plays a crucial role in the design and analysis of robot geometry and motion in space. Forward kinematics involves using kinematic equations to determine the position of a robot's end-effector based on given joint parameter values. On the other hand, inverse kinematics involves calculating the joint parameters necessary to achieve a desired position of the end-effector. In the context of a LLE, the foot or end of a limb can be considered as the end-effector.

2.5.1.1 Forward Kinematics

The Denavit-Hartenberg (D-H) method is commonly used for forward and inverse kinematics in multi-link robots (Amiri et al., 2020). Another widely used approach for forward kinematics in open chains is the product of exponentials (PoE) method. In the D-H method, for an n-joint multi-link robot, $3n$ parameters are used to define the link structure, and n parameters are used to represent the joint values. In contrast, the PoE method requires $6n$ parameters to describe the n screw axes and n parameters for the joint values. One advantage of the PoE method is that it eliminates the need for link frames (Lynch & Park, 2017).

The D-H convention provides a standardised approach for selecting reference frames in a multi-link robot. Each transformation between frames is represented as a composition of four basic transformations: a rotation around the z-axis, a translation along the z-axis, a translation

along the x-axis, and a rotation around the x-axis. These transformations define the relative position and orientation of a link frame with respect to the previous link frame. A representation of D-H frame assignment can be seen in Figure 2.8.

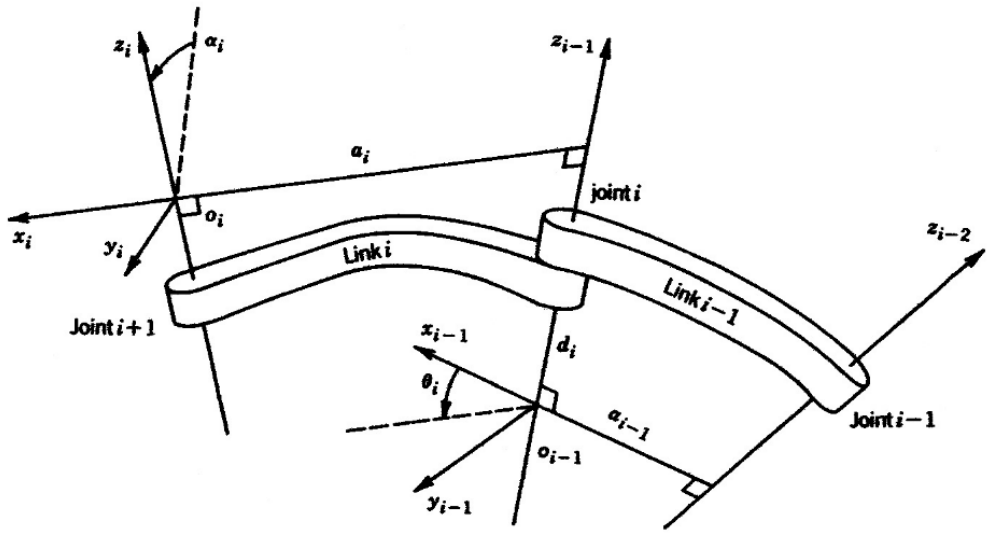


Figure 2.8: D-H Frame Assignment

(Spong, Hutchinson & Vidyasagar, 2020)

The transformation can be expressed using equation (2.1) as shown below:

$$A_i = R_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} R_{x,\alpha_i}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

$$= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The four parameters of link i and joint i given as a_i, α_i, d_i and θ_i refer to link length, link twist, link offset, and joint angle (Spong, Hutchinson & Vidyasagar, 2020). To derive the forward kinematics of a multi-link robot, one can calculate the transformation matrix for each link and multiply them together. This process begins with the first joint and proceeds up to the end effector, resulting in the position of the end effector relative to the base frame of reference. By combining the transformation matrices for each link, the overall transformation from the base frame to the end effector can be obtained, providing the forward kinematics of the robot.

Apart from the D-H method, the product of exponentials is another method which makes use of a formula to describe the forward kinematics of open chains. Lynch and Park (2017) states that, “The PoE formula is directly derived from the exponential coordinate representation for rigid-body motions. Aside from providing an intuitive and easily visualisable interpretation of the exponential coordinates as the twists of the joint axes, the PoE formula offers other advantages, like eliminating the need for link frames (only the base frame and end-effector frame are required, and these can be chosen arbitrarily).” The POE method is a mathematical framework used in robotics and kinematics to describe the motion of rigid bodies and the relationship between their joint variables. It decomposes the overall transformation between two coordinate frames into a sequence of elementary transformations represented as exponential twists or screw motions. The screw axis, a fundamental concept within the POE framework, represents the instantaneous axis of rotation or translation associated with a joint. It provides information about the type and direction of joint motion, enabling the analysis and synthesis of complex robotic motions (Lynch & Park, 2017).

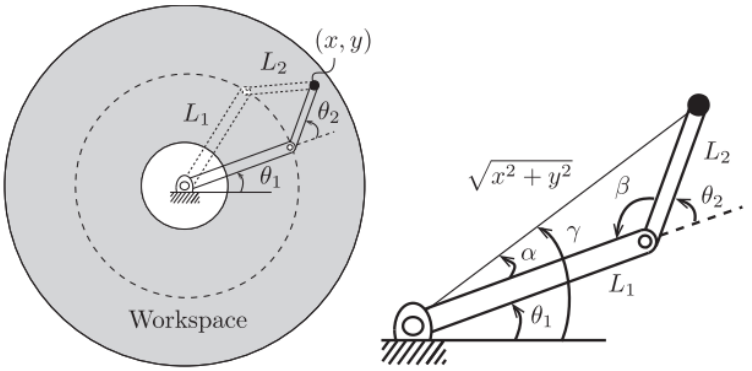
2.5.1.2 Inverse Kinematics

Inverse kinematics is the process of determining the joint configurations or motions needed to position an end effector in a desired location or pose. It allows for precise control of a robot's movements based on specified end effector positions. The calculation is not as straightforward as forward kinematics for various reasons. The first reason being that the equations to be solved are mostly nonlinear which can make closed-form solutions not possible. There may also be multiple solutions to the problem; the end effector could be positioned in a specified position using a combination of multiple sets of joint angles. In some cases, infinite solutions may exist where there is redundancy in the kinematic chain. Lastly the structure of the multi-link joints may not support a solution for a desired position of an end effector (Siciliano et al., 2010).

The Modern Robotics textbook by Lynch (2017) provides a good example of a 2-link planar open chain robot where there are two solutions to find the end-effector position, the angle at the elbow or second joint can be positive or negative. Lynch (2017) states that, ‘These two solutions are sometimes called “lefty” and “righty” solutions, or “elbow-up” and “elbow-down” solutions.’ The forward kinematics to find the end-effector position of such a robot can be seen in equation (2.2) below:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 cos\theta + L_2 \cos(\theta_1 + \theta_2) \\ L_1 sin\theta + L_2 \sin(\theta_1 + \theta_2) \end{bmatrix} \quad (2.2)$$

In Figure 2.9 the lefty and righty positions can be seen within the workspace displayed. The linkages are displayed as dotted or solid lines to indicate the two positions which end in the same end-effector position.



(a) A workspace, and lefty and righty configurations.

(b) Geometric solution.

Figure 2.9: Inverse kinematics of a 2R planar open chain

(Lynch & Park, 2017)

Siciliano et al states that, “Computation of closed-form solutions requires either algebraic intuition to find those significant equations containing the unknowns or geometric intuition to find those significant points on the structure with respect to which it is convenient to express position and/or orientation as a function of a reduced number of unknowns.”

As it can be difficult to find closed-form solutions using intuition, or verify that there is no solution, an alternative is to make use of numerical solution methods. The inverse kinematics approach has the advantage of being applicable to any kinematic structure, but it may not provide all possible solutions for the end effector position. Algorithms using the manipulator Jacobian can be employed to solve inverse kinematics and find feasible solutions (Siciliano et al., 2010).

The Newton-Raphson method is an iterative numerical technique used to solve the inverse kinematics problem in robotics. It is employed to find the joint coordinates that correspond to a desired end-effector position by iteratively updating the joint coordinates until the error between the desired end-effector position and the actual end-effector position become sufficiently small. The method utilises the differential kinematics equations based on the product-of-exponential (POE) formulas (Chen et al., 1999). Although it is frequently utilised, a closed-form solution is preferable due to its computational efficiency, which holds significance in simulations as well as in anticipated future implementations (Seven, 2007).

In the general form (Equation (2.3)), the Newton-Raphson method formula is written as follows:

$$x_n = x_{n-1} - f(x_{n-1})/f'(x_{n-1}) \quad (2.3)$$

In this context:
xn-1 represents the estimated root of the function at step *(n-1)*, *f(xn-1)* denotes the function's value at the $(n\text{-}1)^{th}$ estimated root, and *f'(xn-1)* signifies the value of the first-order derivative of the function at *xn-1*.

2.5.2 Forward and Inverse Dynamics

Dynamics is a division of mechanics which deals with the motion of bodies under the action of forces. When related to LLEs it is the calculation of the relation between the forces applied to a linkage and the acceleration produced by that force (Housner & Hudson, 1983).

Otten (2003) states that, "connected multi-body systems exhibit notoriously complex behaviour when driven by external and internal forces and torques. The problem of reconstructing the internal forces and/or torques from the movements and known external forces is called the 'inverse dynamics problem', whereas calculating motion from known internal forces and/or torques and resulting reaction forces is called the 'forward dynamics problem'."

Multiple techniques exist for analysing the dynamics of multi-link robots, including Lagrange, Newton-Euler, Kane, and Hamilton equations. Lagrange and Newton-Euler equations are frequently employed for solving forward dynamics, with the former relying on energy principles and the latter on the equilibrium of forces and torques. These methods will be further explored in the following discussion (Minchala et al., 2017; Amiri et al., 2020).

The Lagrange method is said to be best suited to simpler robots with fewer degrees of freedom as calculations can become tedious quickly as DOF go up. The Newton-Euler method is known for its ability to generate efficient recursive algorithms for both inverse and forward dynamics, allowing the derivation of closed-form analytic expressions (Lynch & Park, 2017).

The Lagrange method equations are given in equation (2.4) and equation (2.5):

$$L = K - P \quad (2.4)$$

$$\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \quad (2.5)$$

In this formula, i represents the DOF, q_i is the generalised coordinates (in this case representing the joint articular coordinates θ_i). τ is the force and torque vector applied to joint i. L is the Lagrangian function, K is the kinetic energy of the system, and P is the potential energy of the system (Minchala et al., 2017).

The Newton-Euler method, unlike the Lagrangian method, is particularly suitable for calculating the dynamics of multi-degree-of-freedom robots without the need for differentiation. It offers computational efficiency and successful implementation on a computer. It is worth noting that despite its advantages, the equations of motion derived from the Newton-Euler method are equivalent to those obtained from the energy-based Lagrangian method (Lynch & Park, 2017).

The Newton Euler method combines Newtons second law, equation (2.6):

$$F = ma \tag{2.6}$$

where F is force, m is mass, and a is acceleration, with Euler's equation (2.7):

$$J\dot{\omega} + \omega\, x\, (J\omega) = \tau \tag{2.7}$$

where J is the rotational inertia, $\dot{\omega}$ is the angular acceleration, ω is the angular rate and τ is the torque applied to the body. Newtons law is used to describe the translational motion while Euler's equation is used to describe the rotational motion (Corke, 2020). In addition to its use in inverse dynamics, the Newton-Euler method can also be extended to handle forward dynamics by incorporating constraint equations (such as joint constraints) into the analysis. This allows for the determination of the system's accelerations, forces, and torques based on the applied inputs and the specified constraints.

When simplified, factorised, and expressed in matrix form the rigid-body equations of motion are expressed in equation (2.8):

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) \tag{2.8}$$

Corke (2020) states that, ''τ is a vector of the applied joint torques, q represents the joint coordinates (generally the joint angles of a serial link manipulator), $\dot{q}$ represents the joint velocities – the rates of change of the joint coordinates, $\ddot{q}$ is the joint acceleration, g is a term representing the torque due to gravity acting on the robot manipulator and is a function of only the joint coordinates q, C is the Coriolis and centripetal term which represents the gyroscopic forces that act on the robot joints due to the rotation of the other joints, $M(q)$ is the inertia matrix and is a function only of the joint coordinates." $M(q)\ddot{q}$ gives us the torque that is required to accelerate our serial link manipulator by multiplying $M(q)$ with $\ddot{q}$ (Corke, 2020).

Equation (2.9) gives us the inverse dynamics as it maps the motion to torque:

$$(q, \dot{q}, \ddot{q}) \rightarrow \tau \tag{2.9}$$

The equations of motion can be rearranged to map the torque to motion to give us the forward dynamics, which will result in the following equations (Corke, 2020): (2.10), (2.11) and (2.12).

$$\tau \rightarrow (q, \dot{q}, \ddot{q}) \tag{2.10}$$

This equation represents the input torque vector (τ) as a function of the system's joint positions (q), velocities ($\dot{q}$), and accelerations ($\ddot{q}$). It indicates that the torque applied to the system is dependent on the current state of the system, including its position, velocity, and acceleration.

$$\ddot{q} = M^{-1}(q)\{\tau - C(q, \dot{q})\dot{q} - q(g)\} \tag{2.11}$$

This equation describes the acceleration ($\ddot{q}$) of the system's joints. It is determined by the inverse of the mass matrix (M^{-1}), which relates joint accelerations to applied torques. The equation also includes the Coriolis and centrifugal terms ($C(q, \dot{q})\dot{q}$) and the gravitational term ($q(g)$), which account for the effects of velocity and gravity on the system's motion.

$$\dot{q} = \int \ddot{q}\ dt \quad ; \quad q = \int \dot{q}\ dt \tag{2.12}$$

These equations represent the relationship between joint velocities ($\dot{q}$) and joint positions (q) over time. The first equation states that joint velocity is the integral of joint acceleration with respect to time, while the second equation states that joint position is the integral of joint velocity with respect to time. These equations describe how the motion of the system's joints evolves over time based on their respective accelerations and velocities. This is useful when simulating the robot in a real-world setting.

Equations of motion play a crucial role in both forward and inverse dynamics of mechanical systems. In forward dynamics, the equations relate motion variables to forces and are used to determine the system's motion given the applied forces. In inverse dynamics, the equations establish the relationship between forces and motion and are used to calculate the forces required to achieve a desired motion. Methods such as the Newton-Euler method with constraint equations and the Lagrange method set up these equations as mutually dependent linear equations, ensuring comprehensive consideration of all mechanical couplings, thus enabling accurate simulations of both forward and inverse dynamics (Otten, 2003).

2.6 Gait and Trajectory

To model human locomotion, bipedal robots can be developed which can mimic the motion of walking. To understand how a human walking gait can be mimicked, the gait can be broken up into different stages. In Figure 2.10 is an outline of the different stages of the human gait.

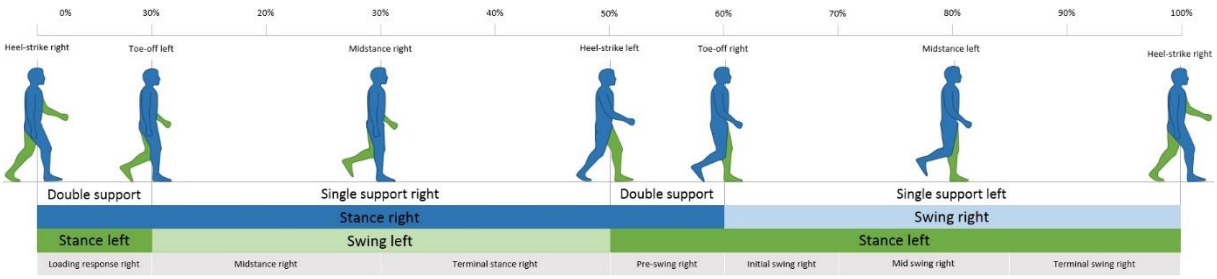


Figure 2.10: Human Gait Cycle

("Gait_cycle.jpg (1860×412)", n.d.)

The human gait comprises two main phases: the stance phase and the swing phase, which can be split into eight sub-phases. These sub-phases are made up of a combination of open and closed-chain movements. The stance phase makes up 60% of the total gait cycle, involves the foot touching the ground and consists of five sub-phases: initial contact (heel strikes the ground), loading response (foot is flat on the ground), mid-stance, terminal stance (heel leaves the ground), and pre-swing (toe leaves the ground) ("The Gait Cycle - Physiopedia", n.d.).

On the other hand, the swing phase, occupying 40% of the total gait cycle, occurs when the foot is not touching the ground, and the body weight is supported by the opposite leg. The swing phase is split into another three sub-phases: initial swing, mid-swing, and late swing ("The Gait Cycle - Physiopedia", n.d.).

During a complete two-step cycle, both feet touch the floor at the same time for 20% of the total gait cycle. 10% happens at the start of the stance phase and 10% at the end of the stance phase, which are referred to as "double-support periods." The remaining time is spent in single support, where only one foot is touching the ground ("The Gait Cycle - Physiopedia", n.d.). By examining these aspects, we can objectively assess the degree of human-likeness in the robot's gait.

Human locomotion is always associated with a system of non-linear equations. As it is difficult to simulate the movement of a walk using the exact bipedal robot model, the model is often simplified and linearized to design a control algorithm (Varma, Jolly & Suresh, 2018).

In the early stages of developing bipedal walkers, researchers in the fields of robotics and biomechanics often adopted a common approach. They created simplified models that focused solely on kinematics, specifically studying the motion in a single plane, typically the sagittal plane, and on flat ground (Kajita et al., n.d.; Collins, Wisse & Ruina, 2001). This approach allowed them to analyse and understand the fundamental principles of bipedal locomotion

without delving into the complexities of dynamic forces, multi-plane motion, or uneven terrains. By studying simplified models, researchers gained insights into the basic mechanics and principles of bipedal walking, paving the way for further advancements in the field (Pratt & Tedrake, 2006).

Passive dynamic walkers are bipedal robots capable of walking without any actuation or control. They are used to aid in the understanding of the dynamic behaviour of human locomotion and for developing solid mathematical governing models along with analytical optimisation methods related to human-like bipedal walking. As opposed to passive walkers, active walking incorporates actuation with feedback control. This aids in the handling of trajectory generation and tracking on uneven terrain (Varma, Jolly & Suresh, 2018).

To achieve desired trajectory or gait patterns in bipedal robots, various mathematical equations and control algorithms are employed. These equations and algorithms enable the synthesis of trajectories or gaits that mimic human locomotion. One common approach is to simplify and linearise the bipedal robot model to design a control algorithm (Haupt & Haupt, 2003). This simplification allows for the development of mathematical equations that describe the desired motion.

The synthesis of trajectories or gaits often involves the generation of joint angle profiles over time. These profiles can be obtained through trajectory optimisation techniques that consider factors such as smoothness, energy efficiency, and task-specific requirements (Haupt & Haupt, 2003). Optimisation algorithms, such as genetic algorithms, can be employed to find optimal joint trajectories by minimizing a cost function associated with performance criteria (Ahn, 2006).

In addition to trajectory optimisation, control algorithms are utilised to generate and track desired gaits. Model Predictive Control (MPC) is an optimisation-based control strategy that uses a dynamic model to optimise control signals while considering constraints on states, inputs, and objectives (Takahashi & Nonaka, 2012). Adaptive control methods continuously adjust control parameters based on system identification and estimation, ensuring robust control performance in the presence of uncertainties and variations (Takahashi & Nonaka, 2012).

The latest trend in biped gait generation and trajectory control is to incorporate the abilities of superfast computing with self-evolving algorithms which can learn from real-time behaviours of the robot and generate responses to adapt to the surroundings (Varma, Jolly & Suresh, 2018).

Common mathematical equations relevant to the synthesis of trajectory or gait in robotic systems include optimisation and control equations.

Optimisation equations are used in trajectory optimisation or gait synthesis to minimise a cost function associated with performance criteria (Singh, Chaudhary & Singh, 2018; Suryadi et al., 2023). Optimisation algorithms, such as genetic algorithms, are employed to find optimal control parameters or joint trajectories.

Control equations involve the control algorithms used to generate and track desired trajectories or gaits (Lin & Brandt, 1998). They can include feedback control laws, such as PID controllers, or model predictive control strategies.

2.7 Control Strategies

In a review conducted by Anam and Al-Jumaily (2012) the different control systems commonly implemented on powered exoskeletons were examined. It is said that modern exoskeletons do not implement all controllers in one system but it is common for them to combine several controllers according to the use case (Anam & Al-Jumaily, 2012). A brief description of each system is outlined below:

2.7.1 Model-based Control Systems

Model-based control in the context of exoskeletons can be classified into two categories: dynamic model-based control and muscle model-based control. In dynamic model-based control, the exoskeleton and human body are modelled as rigid links connected by joints to represent the skeletal structure. This model incorporates inertial, gravitational, Coriolis, and centrifugal effects. The dynamic model can be obtained through mathematical modelling, system identification, or artificial intelligence methods, providing a foundation for controlling the exoskeleton's behaviour (Anam & Al-Jumaily, 2012).

The control system on the BLEEX PLLE is designed to allow the system to move in an as synchronized manner as possible with the movements of the user. For the BLEEX robot, the control algorithm takes data from the forces acting on the exoskeleton and not those acting on the user. By using different dynamic models of the PLLE, the control system can identify the intent of the user as the robot is in different states and produce the required control signals. During locomotion, the control system is designed to determine the current state of the PLLE, distinguishing between single support, double support, and double support with redundancy. Single support refers to when only one foot is in contact with the ground, while double support indicates that both feet are in contact with the ground. In the case of double support with

redundancy, both feet are on the ground, but only one foot is fully flat, while the other may have partial contact. This state identification allows the control system to adapt and respond accordingly for optimal stability and motion control. A weakness of this control strategy is that for the algorithm to be effective, an accurate dynamic model is required. This is because of high parameter sensitivity as a result of high force sensitivity (Jimenez-Fabian & Verlinden, 2012).

The muscle model makes use of Electromyography (EMG) signals and joint kinematics to predict muscle forces in the human limb joint, providing an estimation of the relationship between muscle neural activities and joint movements (Anam & Al-Jumaily, 2012).

Jimenez-Fabian and Verlinden (2012) describe the control system of an active thigh–shank–foot orthosis. The orthoses uses sensory data as the input to a biomechanical model. The biomechanical model is made up of a torso with two legs and two feet. As described above of the muscle model, the control system of the orthoses uses EMG signals that are plugged into the biomechanical model as muscle forces. The model is then able to calculate the knee torque as well as knee acceleration.

2.7.2 Hierarchy Based Control Systems

From a hierarchical perspective, the control system of an exoskeleton can be categorised into three levels: task-level, high-level, and low-level controllers. The task-level controller, being the highest level, operates based on the designed task. The high-level controller follows, regulating the force of interaction between the human and exoskeleton using information from the task-level controller. Finally, the low-level controller, functioning at the lowest level, directly controls the position or force of the exoskeleton joints, establishing a direct connection with the exoskeleton itself (Anam & Al-Jumaily, 2012).

2.7.3 Physical Parameters-Based Control Systems

The control system of an exoskeleton can be categorized into position, torque/force, and force interaction controllers based on its physical parameters. The position controller, typically implemented as a low-level controller, ensures specific angles of the exoskeleton joints. The torque/force controller, also applied at the low-level, manages torque or force in the joints. At the high-level, the impedance controller governs the interaction force between the human and exoskeleton, aiming to minimise the force of human-exoskeleton interaction during task execution (Anam & Al-Jumaily, 2012).

2.7.4 Usage Based Control Systems

Exoskeleton control systems can be categorised based on their applications, such as virtual reality controllers, teleoperation controllers, and gait controllers. These different types of controllers are designed to serve specific purposes in their respective contexts (Anam & Al-Jumaily, 2012).

2.8 Stability

In a paper by Pratt and Tedrake (2006) it is pointed out that one must first define what stability is. As it is in their paper, for this research, stability was defined as whether the PLLE falls down or not. This was further clarified by a fall being when any part of the PLLE touches the ground plane other than the feet. The careful selection of stability margins for bipedal walking is crucial, as they must serve as sufficient conditions for stability, facilitate comparisons between different walking algorithms, be quantifiable and calculable, and possess meaningful significance.

There are many complex mathematical models and numerical optimisation techniques that have been designed and developed which can be coded into sophisticated controllers for experiments and validation of bipedal robots. Among them, the zero-moment point (ZMP) was a milestone in the era of humanoid robot development. ZMP can be thought of as being the Centre of Pressure (CoP) of the forces distributed on the sole of the foot which keeps the biped stable (Harib et al., 2018). The simplified linear inverted pendulum model is another widely used modelling strategy. The linearised inverted pendulum model can be thought of as what links the ZMP to the centre of gravity (CoG) and as a system is important in the dynamic balancing and stability of LLE's (Menga & Ghirardi, 2019).

Sardain and Bessonnet (2004) assert that although ZMP has been around for decades, it is by no means outdated and is here to stay. This statement is still relevant as ZMP continues to be a valuable tool for understanding and controlling the dynamics of bipedal locomotion (Bagheri et al., 2006; Suzumura & Fujimoto, 2012; Joe & Oh, 2019; Sugihara & Morisawa, 2020). In the paper by Sardain and Bessonnet (2004), it is clarified that the CoP refers to the point related to contact forces whereas the ZMP is the point related to gravity and inertia forces.

Poincaré return maps are a valuable tool for analysing the stability and dynamics of systems, including walking robots. The concept of Poincaré return maps is widely used in various fields, such as physics, mathematics, and control theory. These maps provide a way to study the behaviour of periodic orbits and limit cycles by reducing the dimensionality of the system's dynamics.

The stability analysis of limit cycles often involves examining the fixed points of the Poincaré return map. Stability properties of the periodic orbits can be assessed by analysing the eigenvalues of the linearized Poincaré map. If all the eigenvalues have negative real parts, the limit cycle is stable, indicating that small perturbations will diminish over time. Conversely, if any eigenvalue has a positive real part, the limit cycle is unstable, implying that deviations will amplify, leading to unpredictable behaviour.

The Poincaré return map approach has been applied in various studies related to stability analysis and control of walking robots (Hamed et al., 2010; Hamed & Grizzle, 2013; Yan et al., 2014). It has been used to assess the stability of periodic orbits, design feedback control laws, and analyse the existence and characteristics of limit cycles (Hamed et al., 2010; Hamed & Grizzle, 2013; Yan et al., 2014). By utilising the Poincaré return map, researchers can gain insights into the stability and behaviour of walking robots, facilitating the development of robust control strategies and understanding the dynamics of bipedal locomotion.

In a thesis by Seven (2007) the linear inverted pendulum model (LIPM) is said to be a powerful tool due to its simplicity. By not using a complex model basic walking control is easier to achieve while being more intuitive. The LIPM looks at the bipedal robot as a single point mass located at its CoM. The CoM is connected to the ground by an assumed to be massless leg, which makes up the pendulum as the CoM pivots on the point on the ground made by the massless leg. To make the system linear, the CoM is assumed to be at a fixed height. The equations of motion regarding the CoM in the x and y plane are given below:

$$\ddot{c}_x = \frac{g}{z_c} x + \frac{1}{m z_c} u_p \tag{2.13}$$

$$\ddot{c}_y = \frac{g}{z_c} y + \frac{1}{m z_c} u_r \tag{2.14}$$

In equations (2.13) and (2.14) m is the mass of the body, z_c is the height of the point and g is the gravity constant. u_p and u_r are control torques. u_p represents the pitch about the y-axis and u_r represents the roll about the x-axis (Seven, 2007).

To ensure effective joint control and stability in robotic systems, communication plays a critical role. The CAN (Controller Area Network) bus is a commonly used communication technology that offers several advantages for robotic applications. CAN bus provides electrical robustness and daisy-chaining capability, making it suitable for interconnecting multiple actuators and higher-level controllers.

CAN bus operates at varying speeds, with a maximum data rate typically up to 1 Mbps. While it may not be considered a high-speed interface compared to some other communication

protocols, the actual required speed for stable robot control depends on the specific application and the dynamics of the robot. In general, the communication speed should be sufficient to transmit commands and receive feedback from the actuators at a rate that allows for accurate and timely control of the robot's joints.

The required communication speed for stable robot control can vary depending on factors such as the complexity of the robot's motion, the control algorithms employed, and the desired precision and responsiveness of the system. In some cases, a lower communication speed within the capabilities of the CAN bus may be sufficient for achieving stability, while in other cases, higher speeds may be necessary, especially for applications that involve fast and precise movements.

It is crucial to carefully assess the requirements of the robotic system, considering factors such as control loop frequency, sensor update rates, and the desired level of coordination and synchronization between the actuators. By selecting the communication speed to meet the specific needs of the robot, effective joint control and stability can be achieved.

Overall, the CAN bus serves as a reliable and widely adopted communication technology for joint control in robotic systems, offering flexibility in terms of data rates and robustness. The appropriate communication speed should be determined based on the specific requirements of the robot and its control algorithms to ensure stable and efficient robotic motion (Katz, 2018).

2.8.1 Stability Analysis

Research findings indicate that the commonly used methods for analysing the stability of bipedal walking, namely the ZMP criterion and Poincaré return map method, lack a general qualification for assessing stability. While satisfying the Zero Moment Point criterion is sufficient for stability, it applies only to a subset of the robot's state-space. The Poincaré return map method, while useful for periodic gaits, has limitations and cannot assess stability for non-periodic gaits. Neither criterion offers a comprehensive stability qualification that covers all possible gaits of a bipedal system. These limitations highlight the need for further research to develop a comprehensive stability criterion that overcomes these challenges and provides a universal qualification for bipedal walking stability (Zutven, Kostić & Nijmeijer, 2010).

Fixed points and limit cycles are concepts used to analyse the stability and dynamics of a walking robot's gait. In the context of bipedal locomotion, a fixed point refers to a specific configuration or pose of the robot where the motion repeats exactly in a periodic manner (Vukobratović & Borovac, 2004).This means that the robot's joint angles, positions, and

velocities return to the same values after completing a full gait cycle. In other words, the robot's motion converges to a stable equilibrium at these fixed points (Vukobratović & Borovac, 2004).

On the other hand, a limit cycle represents a set of continuous and repeating motions that the walking robot exhibits (Goswami, 1999). Unlike fixed points, a limit cycle does not necessarily return the robot's state variables to the exact same values. Instead, it creates a pattern of motion that repeats over time while maintaining certain characteristics, such as step length, step frequency, or joint coordination (Goswami, 1999). The robot's motion evolves in a cyclic manner, but with slight variations from one cycle to the next (Goswami, 1999).

To analyse the stability of a gait on a walking robot, the concept of fixed points and limit cycles can be utilised. Stability refers to the ability of the robot to maintain its desired gait pattern and resist deviations or disturbances. By studying the behaviour of the system around fixed points and limit cycles, it is possible to assess the stability properties of the robot's gait (Vukobratović & Borovac, 2004).

In practice, stability analysis often involves linearizing the equations of motion around the fixed points or limit cycles. This linearization process approximates the dynamics of the robot near these points by considering small perturbations. The stability of the gait is then determined by examining the eigenvalues of the linearized system. If all the eigenvalues have negative real parts, the gait is stable, indicating that small deviations from the desired motion will diminish over time and the robot will continue to walk in a controlled manner. Conversely, if any eigenvalue has a positive real part, the gait is unstable, implying that deviations will amplify, and the robot's motion will become unpredictable or divergent (Vukobratović & Borovac, 2004).

By analysing fixed points and limit cycles, researchers can gain insights into the stability and dynamics of walking robots. This knowledge helps in designing and controlling robotic systems to achieve stable and robust gaits, contributing to the development of effective locomotion strategies for walking robots (Garcia et al., 1998; Awad et al., 2017).

2.9 Optimisation

Optimisation techniques play a significant role in enhancing the performance of robotic joint control systems. Several approaches have been developed to improve various aspects of joint control. Some commonly used optimisation techniques in robotic joint control include:

Trajectory Optimisation:

Trajectory optimisation methods aim to generate optimal joint trajectories considering factors such as smoothness, energy efficiency, and task-specific requirements. These techniques help in minimising tracking errors and improving accuracy (Spong, Hutchinson & Vidyasagar, 2005).

Feedback Control Optimisation:

Feedback control techniques involve optimising control signals based on sensor feedback. Control algorithms can be designed, or control parameters can be tuned to achieve stability, robustness, and accurate tracking (Åström & Murray, 2021).

Model Predictive Control (MPC):

MPC is an optimisation-based control strategy that uses a dynamic model to optimise control signals while considering constraints on states, inputs, and objectives. This technique enables effective handling of complex tasks and adaptation to changing conditions (Garcia, Prett & Morari, 1989).

Adaptive Control:

Adaptive control methods optimize control strategies by continuously adjusting control parameters based on system identification and estimation. These techniques ensure robust control performance in the presence of uncertainties and variations (Slotine & Li, 1991).

Energy Optimisation:

Energy optimisation techniques focus on minimising the energy consumption of robotic joints while maintaining desired performance. Optimal control strategies, motor efficiency, actuator selection, and motion planning algorithms are considered to achieve energy-efficient joint control (Roussel, Canudas-De-Wit & Goswami, 1998).

Multi-objective Optimisation:

Joint control optimization often involves multiple conflicting objectives, such as accuracy, speed, energy consumption, and stability. Multi-objective optimisation algorithms provide a set of optimal solutions along the trade-off curve, allowing for the selection of control strategies that best balance the objectives (Deb et al., 2002).

Optimisation using Genetic Algorithms:

Genetic algorithms (GAs) have proven to be effective in optimising robotic joint control systems. GAs are a type of evolutionary algorithm inspired by natural toe and genetics. They involve iteratively evolving a population of potential solutions to a problem by applying genetic operators such as selection, crossover, and mutation (Lamont & Veldhuizen, 2007).

Genetic algorithms can be utilised to optimise joint control by reducing a cost function associated with performance criteria. The cost function represents a measure of the system's objective, such as tracking error, energy consumption, or a combination of multiple objectives (Lamont & Veldhuizen, 2007). By formulating the joint control problem as an optimisation task, the genetic algorithm searches for the set of control parameters that minimises the cost function (Ramadan, 2018).

The process begins with the initialisation of a population of potential control parameter sets, often referred to as individuals or chromosomes. These individuals undergo selection based on their fitness, which is determined by evaluating their performance using the cost function (Rigatos & Abbaszadeh, 2021). The fitter individuals are more likely to be selected for reproduction, where genetic operators such as crossover and mutation are applied to create new offspring (Berninger & Rixen, 2019). This iterative process continues until a termination criterion is met, such as reaching a maximum number of generations or achieving a desired level of optimisation (Rigatos & Abbaszadeh, 2021).

Genetic algorithms offer several advantages for optimising joint control. They can handle complex, nonlinear, and multi-objective optimisation problems (Jianwei & Zhou, 2023). The population-based nature of GAs allows for exploration of the solution space and the potential to find global optima (Kurennov, Barakhov & Vambol, 2022). Additionally, GAs can handle constraints on control parameters or system dynamics, ensuring that the optimised solutions are feasible (Azizi, 2020).

The application of genetic algorithms in joint control optimisation has been demonstrated in various studies. For example, researchers have used GAs to optimise the control parameters of robotic arms, improve posture control, and enhance the efficiency of joint movements (Singh, Jadhav & Krishna, 2014; Zhang, He & Wang, 2017; Azizi, 2020). By employing genetic algorithms, researchers can effectively optimise joint control systems by reducing a cost function associated with performance criteria, leading to improved performance, stability, and efficiency.

These optimisation techniques, along with others, contribute to improving the performance, stability, and efficiency of robotic joint control. They provide valuable tools for achieving precise control, adaptability to dynamic environments, and energy efficiency.

2.10 Simulation

A PLLE can be simulated in software using different techniques and mathematical models to test its behaviour. There are several platforms and programming languages which can aid one in simulating a model, an overview of the Robot Operating System (ROS), MATLAB and Simulink will be discussed below.

The ROS middleware has gained significant popularity in recent years as a platform for robotics, enabling seamless integration of hardware and software components. ROS has become widely adopted, particularly in the research community. In the context of simulating a LLE, the kinematic and dynamic properties of the robot can be defined using the Unified Robot Description Format (URDF) within ROS. URDF, an XML-based specification, is commonly used in academia and industry for modelling multibody systems. The LLE can then be simulated in Gazebo, a powerful 3D open-source dynamic simulator plugin for ROS. Gazebo provides accurate and efficient simulation of various robot types in different environments, offering high-fidelity physics simulation. By leveraging ROS-control packages and the Gazebo plugin adapter, one can effectively simulate a robot's controllers (Amiri et al., 2020). Amiri et al (2020) used the ROS platform with Gazebo to simulate their own 6 DOF biped and implemented a walking algorithm in the form of a Python script which ran in a loop and specified joint angles.

MATLAB and Simulink are a powerful set of software which work together and have also been used in research (Amiri et al., 2020). These systems offer seamless compatibility with hardware platforms like Arduino microcontrollers and Raspberry Pi single-board computers, enabling an external mode for interactive tuning of parameters and real-time signal monitoring during algorithm execution on the device. You can generate C code directly from the model created in Simulink and deploy it to hardware such as an Arduino and then monitor it running in real-time. To simulate a LLE a model can be defined using Unified Robot Description Format (URDF) or the robotics toolkit can deal with D-H parameters. Alternatively, a LLE CAD model can be imported into Simulink. Simulink makes use of a proprietary programming language of function blocks to allow you to model and simulate your entire system before moving to hardware. An advantage of MATLAB is that it contains a ROS Toolbox which enables you to connect MATLAB and Simulink to ROS. By establishing a connection to an existing ROS network, it becomes possible to communicate with simulators, hardware devices, or software nodes on the network, allowing for the transmission of commands and reception of data. This

seamless integration facilitates interaction with various components within the ROS ecosystem.

It may be noted that CAD software packages such as SolidWorks can export models in the URDF format, which can then be imported into Gazebo or Simulink.

2.11 Summary

This chapter reviewed current LLE's and their different designs and applications. Difficulties relating to ergonomics were reviewed and it was seen that a focus should be put on user-cenred design as well self-aligning mechanisms to accommodate different users. The different actuator types commonly used on legged robots were reviewed along with their strengths and weaknesses. Common sensors used on legged robots were also briefly discussed.

A review of existing controller types and typical control of bipeds was done. Methods for model design, simulation as well as gait and trajectory were also researched to gain an insight into modern techniques.

This chapter helped the researcher formulate a methodology by reviewing methods in design, control and simulation.

CHAPTER 3

SIMULATION METHODOLOGY

This chapter discusses the procedures and techniques used to achieve the goals of this research which were to use mathematical modelling that would predict the hardware requirements for untethered walking of a PLLE, and to use this simulation to optimise for stability in a simulated environment. Accomplishing this involved the use of several software tools and programming languages, including Onshape, MATLAB and Simulink. The simulation began firstly by making use of Onshape, a CAD software, which was used to generate a 3D model of the exoskeleton with a human-sized test dummy attached. The model was then imported into Simulink, a multi-body dynamics simulation software, which was used to simulate the movement of the exoskeleton during walking. Finally, MATLAB was used to optimise the simulation for stability, using a range of stability metrics to guide the optimisation process. The resulting simulation provided insight into the hardware requirements for untethered walking of a PLLE, as well as providing a tool to optimise for stability in a simulated environment.

3.1 Design Considerations

In developing the simulation, several design considerations were taken into account to ensure that the model accurately represented a PLLE. One of the main considerations was the size and weight of the exoskeleton, which needed to be based on realistic measurements and specifications. Inertia and moment of inertia of the system were also considered and incorporated into the simulation using CAD files and the parameters associated to them. Once the 3D model had been created, the data for the mass, centre of mass, moments of inertia, and products of inertia could be output for each subsection of the system using Onshape. These values were then entered and set up in the simulation.

To achieve this, a thorough literature review was conducted to identify existing PLLEs and their specifications. These specifications were then used as a basis for developing the 3D CAD model of the exoskeleton in Onshape. The weight of the model was also considered, with efforts made to ensure that the model's weight was as close as possible to the weight of an existing PLLE. This consideration was important to ensure that the simulation accurately reflected the physical properties of a real-world exoskeleton, which is crucial for accurate prediction of hardware requirements and stability. Components of the model, including the human test dummy and the frame components of the PLLE, were assigned uniform densities to control their masses. The densities were adjusted to match a desired total system mass, which was based on literature. The test dummy density was manipulated to have a mass of what could be considered the upper range of a user of certain existing PLLEs. The frame parts densities were overridden to achieve the target system mass. The inertial properties of the

system were subsequently derived based on these masses. This will be expanded on in the CAD section of this chapter.

3.2 Requirements

The goals of this research, as mentioned in Chapter 1, should include a realistic 3D CAD model of the PLLE assembly with respective masses and inertial properties. The simulation should accurately predict the hardware requirements for untethered walking of the PLLE and provide insight into how to optimise for stability in a simulated environment. Additionally, the simulation should ideally be user-friendly and accessible to researchers, to facilitate the adoption of PLLE technology and its research within CPUT. The requirements relevant to the simulation portion of this research are as follows:

- The simulation should accurately portray the movement of the PLLE with realistic impact forces, inertial properties, and the effects of gravity.
- The simulation should provide insights into optimising stability in a simulated environment, allowing for the comparison of different walking algorithms.
- The model of the PLLE system should be full-scale, representing a realistic size and proportions of a full-sized walker capable of carrying human-like payloads.
- The simulation should be user-friendly and accessible to researchers, facilitating the adoption of PLLE technology in research at CPUT.
- The stability margins employed in the simulation must serve as sufficient conditions for stability, providing meaningful and quantifiable results.
- The simulation should account for the trade-off between complexity and computational resources, ensuring a reasonable computational time for repetitive simulations.
- Assumptions may be made to simplify the model and leverage available resources without sacrificing the goal of accurately predicting hardware requirements for untethered walking.
- Specific actuator modelling and production-ready design of the PLLE are not necessary for the simulation, as long as the simplified model is compatible with human body movements.

It should be noted that there is a limit to the accuracy of simulations, as increasing complexity and accuracy requires significantly more computational power and there is a trade-off between accuracy and the computational resources available. In addition, when running repetitive simulations, the time it takes for the simulation to run becomes an important factor. As the complexity and accuracy of the simulation increase, the computational time required also increases, and this can limit the number of simulations that can be run within a reasonable

time frame. Therefore, the researcher needed to leverage the resources available and make assumptions as to what can be simplified and what was necessary in order to achieve the goal of accurately predicting hardware requirements for untethered walking, such as actuators. It was assumed that it was not necessary to model specific actuators, or to design a production ready PLLE. A simplified model, so long as it is compatible with the movements of the human body, should suffice as a tool to optimise for stability and predict hardware requirements. The simulation should however accurately portray the movement of the PLLE with impact forces, inertial properties, and the effects of gravity.

These design choices will be expanded on in the subsections to follow.

3.3 CAD Model

To simulate a walking robot, a CAD model was needed to import into the MATLAB Simulink software. For the CAD design of the PLLE platform, Onshape was chosen. Onshape is a cloud-based computer-aided design software that provides a comprehensive 3D modelling environment accessible through a web browser or mobile app. Onshape enables users to design and collaborate on projects. This meant that the researcher could easily make the model available to future researchers within CPUT.

In order to create an accurate CAD design of the PLLE, it was crucial to ensure that the combined weight of the structure, along with a human test dummy attached, was realistic. By considering the weight of the exoskeleton and its attachment, the simulation could more accurately reflect real-world conditions and provide meaningful insights. Additionally, it is important to note that for simulation purposes, the structure of the actual PLLE was not of primary importance. Instead, a generalised model was sufficient as long as it exhibited anthropomorphic behaviour and resembled the expected movements and interactions with the human body. This allowed for simplifications in the design, focusing on capturing the essential dynamics and functionality rather than replicating every intricate detail of the physical structure.

The weight of the system was based on the specifications of the Atalante exoskeleton developed by Wandercraft (Huynh et al., 2021; Luo et al., 2021). This PLLE was chosen because it is focussed on untethered walking without crutches, as opposed to most other exoskeletons which make use of crutches or operate more like motorised braces. The Atalante PLLE weighs approximately 60 kg and can walk while carrying up to a 90 kg passenger. This meant that the researcher needed to design the model with a combined weight of less than 150 kg.

An existing model of a human test dummy was taken from the public library repository within Onshape. As can be seen in Figure 3.1, there are several models which use the same generic test dummy model. The researcher was able to make a copy of one of these models as a base to design around. With a height of approximately 185 cm, the dummy falls between the Hybrid III 50th Male, with a height of 175 cm, and the Hybrid III 95th Male with a height of 188 cm (“Hybrid III - Wikipedia”, n.d.).

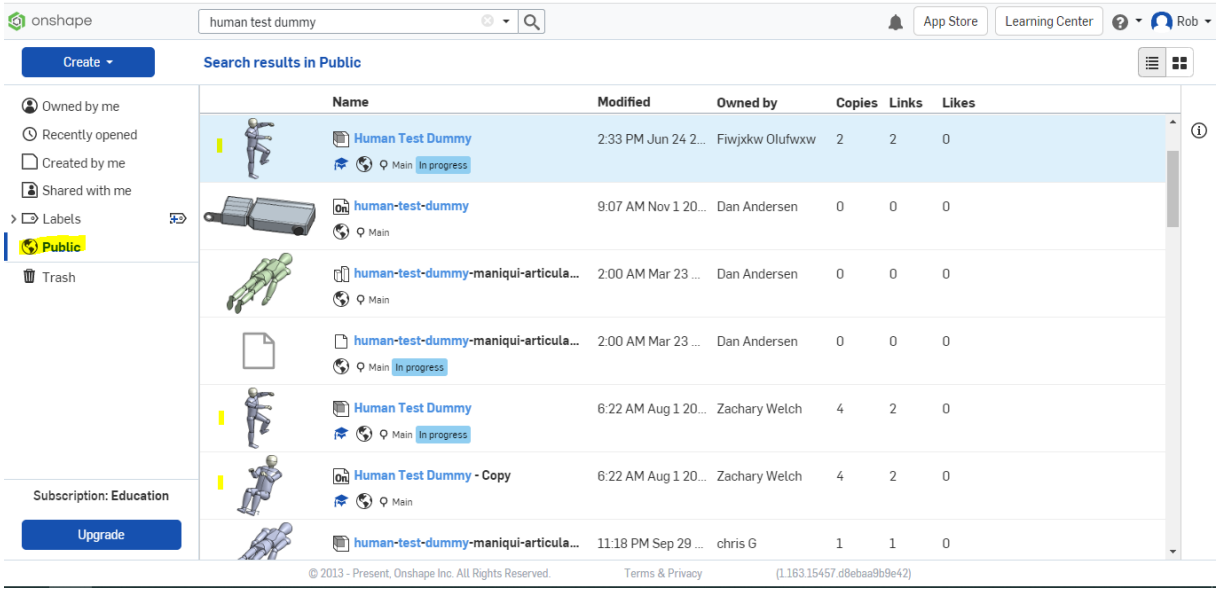


Figure 3.1: Onshape Public Models

After the test dummy model was obtained, a simple exoskeleton frame was designed around it. The joints were designed to be colinear with the joints of the test dummy in the Y frame.

As will be seen in the following chapter, the existing Simulink model which the researcher was using split the PLLE into the following sections, namely a torso, an upper left leg, an upper right leg, a lower left leg, a lower right leg, a left foot and lastly a right foot. This meant the CAD model needed to be split in the same manner to be able to be imported into Simulink. This arrangement can be seen in the final CAD model in Figure 3.2. The final CAD assembly is a simple structure designed by the researcher, mated to the test dummy model that was obtained. It was used to primarily represent how weight could be distributed through the system and to specify joint positions. The joints and frame parts are simplified bodies which would, in a real PLLE, need motors, a power source, control hardware, wiring and various other hardware to function.

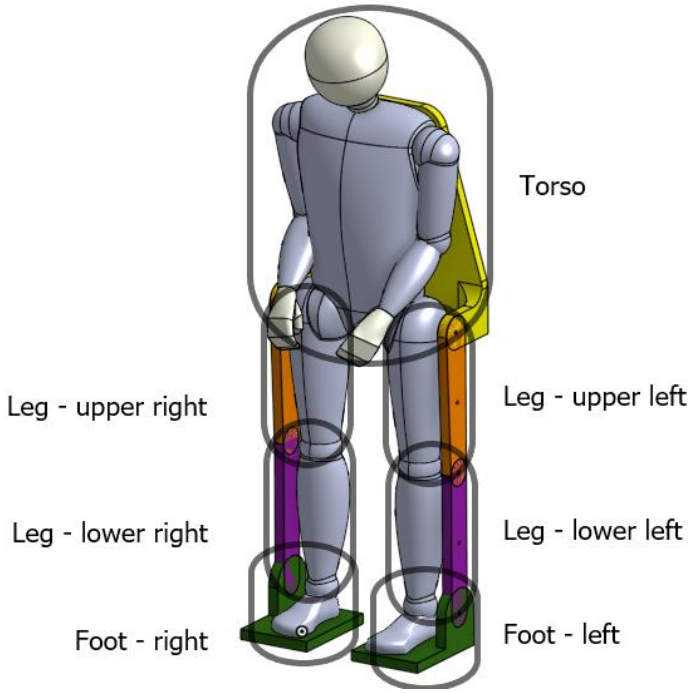


Figure 3.2: Final CAD assembly

To assemble the CAD model in this manner, composite parts were designed by the researcher. A composite part in Onshape is the combining of two different parts with different materials into one part. By using composite parts, the researcher was able to combine the torso of the test dummy with the upper structure of the PLLE, the upper leg with the corresponding upper leg linkage, the lower leg with the lower leg linkage and the feet with the foot platform of the PLLE. In the Figures 3.3, 3.4 and 3.5, an example is shown of how the lower leg and lower leg linkage were combined into a composite part. Separate materials were applied with different densities to each individual part and when combined, one is able to see the combined centre of mass, as well as mass moments of inertia. Specific materials were deemed to not be important with a simplified model, therefore materials were chosen which had a desired density rather than structural properties. These values were later inserted into Simulink when importing the model.

As was previously stated, the approximations for the masses assigned to the components were based on the Atalante PLLE having a combined mass of 150 kg, 60 kg for the structure and up to a 90 kg user. The mass of the test dummy was made to be 92.7 kg. This was accomplished by assigning a custom density to the test dummy components of 1 g/cm3. The PLLE frame components were assigned a density of 2.7 g/cm3 (Aluminium material in Onshape). This resulted in the PLLE structure having a mass of 53.3 kg.

Figure 3.3 shows the right bottom leg component of the test dummy alongside the bracket that was designed by the researcher. The right bottom leg is highlighted to display the custom

material properties that were assigned to it. In this case, the density was assigned a value of 1 g/cm3 with the alias of ‘Human’.

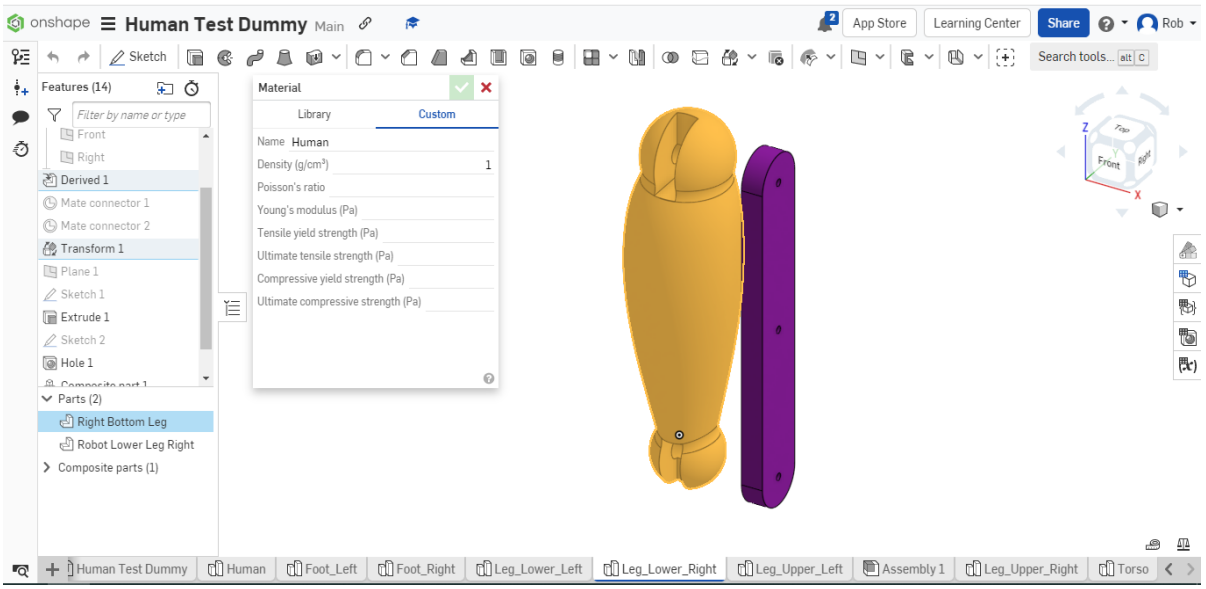


Figure 3.3: Right lower leg material properties

Figure 3.4 displays the right lower leg bracket that is highlighted. The material properties show a density of 2.7 g/cm3, and a material of ‘Aluminium’. The material in this case could also have been a custom material and assigned an appropriate density, if a suitable material was not found to achieve a desired approximate mass.

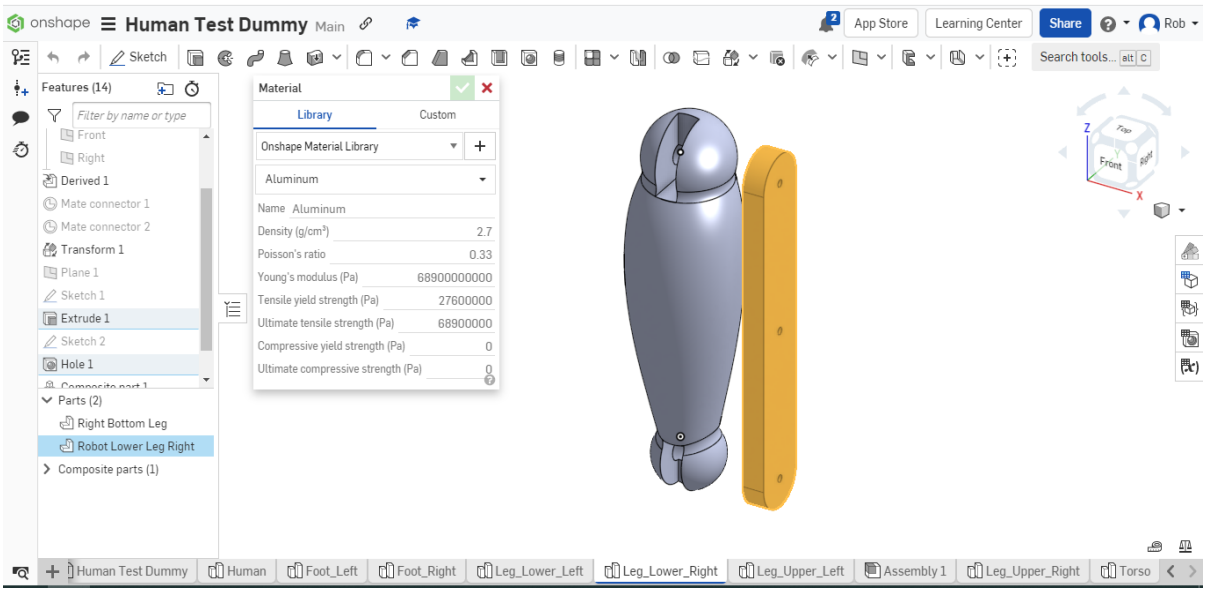


Figure 3.4: Right lower leg linkage material properties

It can now be seen that if ‘Composite part 1’ is selected in the bottom left of Figure 3.6, the mass properties of the combination of the two parts can be viewed. Along with the total combined mass, the centre of mass and mass moments of inertia are also displayed. These values will later be used in the simulation section of this research.

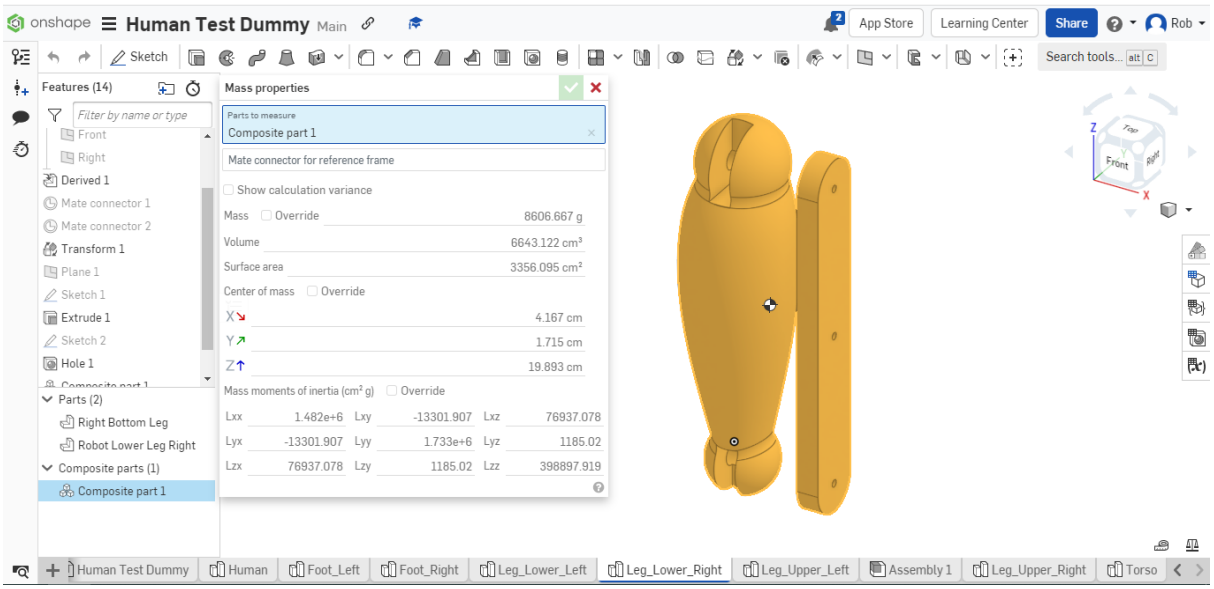


Figure 3.5: Right lower leg composite part

In Figure 3.6 it can be seen that the mass of the combined PLLE and test dummy assembly is approximately 146 kg, which was within specification of being less than but near the 150 kg maximum of Atalante.

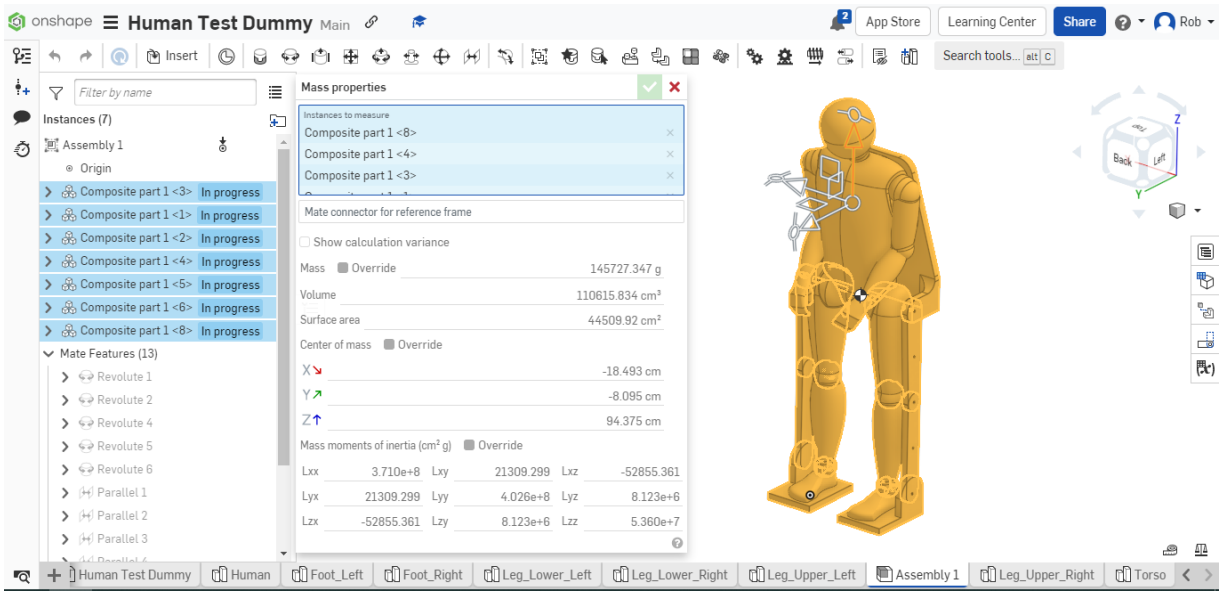


Figure 3.6: Exoskeleton frame with test dummy

3.4 Methods of Obtaining Walking Gaits

The following sections will document the workflow used to obtain stable walking gaits using two different methods. As mentioned in the literature review, there was existing software available to students written by staff at MathWorks containing a bipedal walker, (MathWorks Student Competitions Team, 2023). The following subsections will explain how the software was understood and manipulated to get results. A more detailed explanation of the software can be found in Appendix A. Two different methods were used to obtain a walking gait, firstly

the LIPM to produce a walking pattern, followed by another technique which makes use of motion planning and a genetic algorithm to produce a more dynamic gait which will later be optimised.

3.4.1 Linear Inverted Pendulum Model

The Linear Inverted Pendulum Model (LIPM) is a widely employed low fidelity technique for modelling bipedal walking robots, known for its simplicity, and frequently used in the design and analysis of legged locomotion (Castro, 2019). Using the tools created by the MathWorks team, the researcher was able to import the CAD model that was prepared and manipulate variables to obtain a walking gait. The resulting body trajectory assumes that the walking pattern is symmetric and that there are no disturbances.

To obtain a walking gait, the researcher started by employing the Linear Inverted Pendulum Model method, reaping several notable advantages and benefits.

The LIPM's simplicity offered a foundational comprehension of bipedal walking dynamics. It served as a fundamental platform for understanding the underlying principles of bipedal locomotion, priming the researcher before delving into more intricate simulations.

LIPM's analytical approach facilitated a deep exploration of walking dynamics. Its simplified equations allowed for analytical solutions, aiding in the derivation of crucial correlations between variables like step length, step time, and centre of mass movement. This analytical insight subsequently guided the researcher's subsequent research directions.

With LIPM being such a well-known solution with working solutions that could readily be used, experimentation was expedited. By simplifying the walking dynamics, the researcher could swiftly implement and assess diverse scenarios. This experimental agility supported iterative enhancements and the exploration of various walking configurations.

LIPM's adoption as a baseline allowed for meaningful comparisons. As the researcher advanced towards advanced simulation methods or optimisation techniques, they could contrast the outcomes with the results based on LIPM. This comparative evaluation facilitated the assessment of the efficacy of subsequent research undertakings.

Moreover, utilising LIPM aligned the researcher's work with established educational paradigms in robotics and biomechanics. This alignment broadened the accessibility and relatability of their work to a wider audience.

Transitioning to a genetic algorithm for optimisation, the researcher aimed to leverage insights from LIPM simulations to inform optimisation objectives and constraints. The results from LIPM simulations were employed as initial approximations or boundaries for optimisation parameters, potentially enhancing convergence and efficiency.

Although it is possible to optimise LIPM walking gaits, this research chose not to. The creation and optimisation of walking gaits with a genetic algorithm and cost function was done starting from zero motion initial conditions. This approach was chosen to avoid biasing the genetic algorithm towards a predetermined path. By starting from zero motion initial conditions, the researcher aimed to isolate the effects of the cost function and genetic algorithm on the results. This would allow for a more accurate evaluation of the performance of the cost function and genetic algorithm in optimising the walking gaits.

3.4.1.1 Walking Pattern Generation

The LIPM folder in the workflow contains all the tools needed to firstly create a walking pattern, use the pattern to transform the reference frame from the CoM to the foot positions, use inverse kinematics to calculate the joint trajectories to achieve the foot positions, and finally simulate the walking robot using the open loop trajectory that has been calculated by the software.
To obtain a walking pattern, the researcher made use of a parametric set of initial conditions. By ticking the box useParametricICs to set it to true, the findInitialConditions function found within the animateLIPM.mlx Live Script is used (see Appendix B, Code Snippet 1). The findInitialConditions function uses the theory of orbital energy to find the initial conditions for the LIPM system, which are then assigned to the variables dx0, y0, dy0, and tSingleSupport. These variables represent the initial X velocity, initial Y position, initial Y velocity, and time for single support phase, respectively. Initial conditions are used to create a symmetric trajectory and move the pendulum in a positive Y direction, starting with a positive initial X position x0.

The animateLIPM.mlx Live Script uses a number of functions based on maths found in the paper titled, ‘’The 3D Linear Inverted Pendulum Mode: A Simple modeling for a biped walking pattern generation’’ by Kajita et al. (2001). The resulting pattern after having run the script can be seen in Figure 3.7 . It can be seen that the pendulum starts off by lowering to the specified walking height, does a half step to position itself as defined by the initial conditions and then continues in a symmetrical walking pattern. The circle at the top represents the CoM and has a virtual leg which leads to the floor to indicate the foot placement.

There is only one virtual leg as it is assumed that there will only ever be one foot on the floor at once. The virtual leg flips from left to right to indicate the foot placement of both the left and right foot. A single virtual leg is a limitation of the LIPM being a simplified representation of a bipedal walking system. It assumes the CoM of the walker to be modelled as an inverted

pendulum, which in turn helps in understanding the overall dynamics and stability of bipedal locomotion. However, the LIPM considers only one foot on the ground at any given time, which means it represents the single support phase of walking. During single support, the model assumes that the CoM of the walker moves along a straight line in a 2D plane or along a simple trajectory in a 3D plane. In reality, bipedal walking involves both single support and double support phases. During the double support phase, both feet are on the ground, and the CoM is supported by two points. In this case having such a simplified model is not an issue, as the stability will later be verified within the Simulink simulation. If the model proves to be unstable, adjustments can be made.

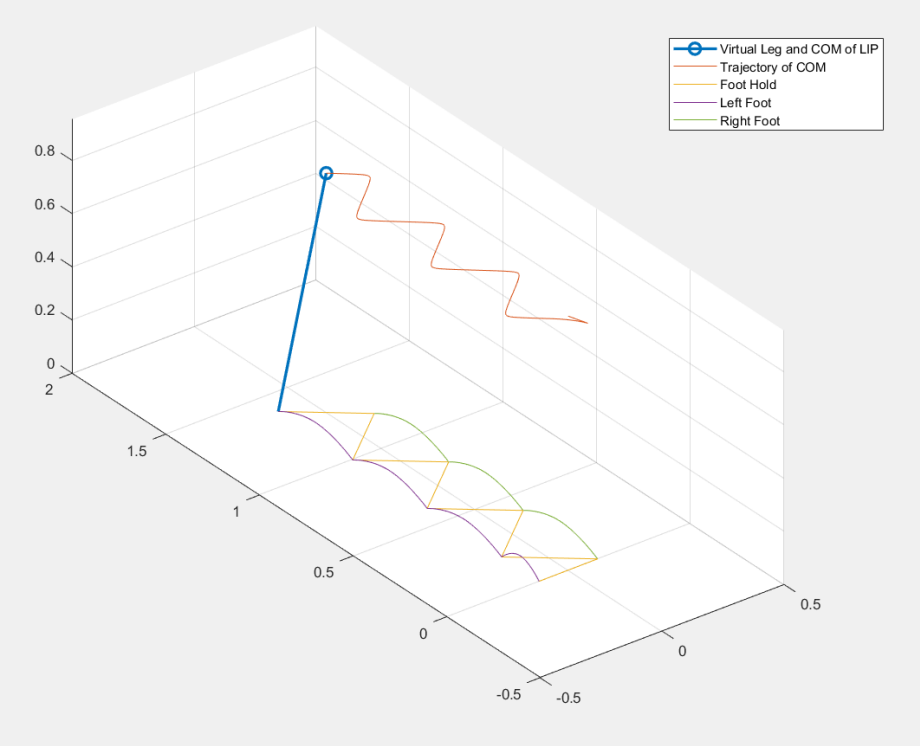


Figure 3.7: Pendulum Walking Pattern

3.4.1.2 Frame Transformations

After the walking pattern in Figure 3.7 was obtained, the reference frame needed to be transformed from the CoM with respect to the world to the foot positions with respect to the robot body. The preceding script (animateLIPM.mlx) accomplished two tasks. Firstly, it generated the centre of mass trajectory for the robot and determined the trajectories of the swinging feet. These trajectories are defined in relation to the world frame, providing global position information. However, our objective is to obtain position information relative to the robot's local body frame since our focus is on moving the robot's feet for walking. To achieve this, the MathWorks team's script undertakes a transformation of reference frames, converting the body and swing trajectories from global origin-based coordinates to foot trajectories relative

to the body's centre of mass. In the provided code, the function being used to get different positions with respect to the world frame or the local frame is simple vector addition. The fundamental concept is straightforward and can be explained using the following equation:

$$\overrightarrow{ac} = \overrightarrow{ab} + \overrightarrow{bc} \tag{3.1}$$

Assume:

ab (the global position of the body) is [2, 3, 1] in meters.
bc (the relative position of the swing foot with respect to the body) is [1, -1, 0.5] in meters.
Now, let's find ac (the global position of the swing foot):

ac (global frame) = ab (global frame) + bc (local frame)
ac (global frame) = [2, 3, 1] + [1, -1, 0.5]

Performing the vector addition elementwise:

ac_x = 2 + 1 = 3 meters
ac_y = 3 - 1 = 2 meters
ac_z = 1 + 0.5 = 1.5 meters

So, ac is [3, 2, 1.5] meters.

This means that, based on the given values, the swing foot is located at [3, 2, 1.5] meters in the global frame. With any two elements of equation 3.1, the third element can be calculated (MathWorks Student Competitions Team, 2023).

The walking pattern being in the perspective of the foot frame in relation to the body frame would later be important when simulating the robot within the Simulink environment, as the robot body would be unconstrained with six degrees of freedom.

The script opens an animation when run which visualizes the trajectories of the feet, this can be seen in Figure 3.8.

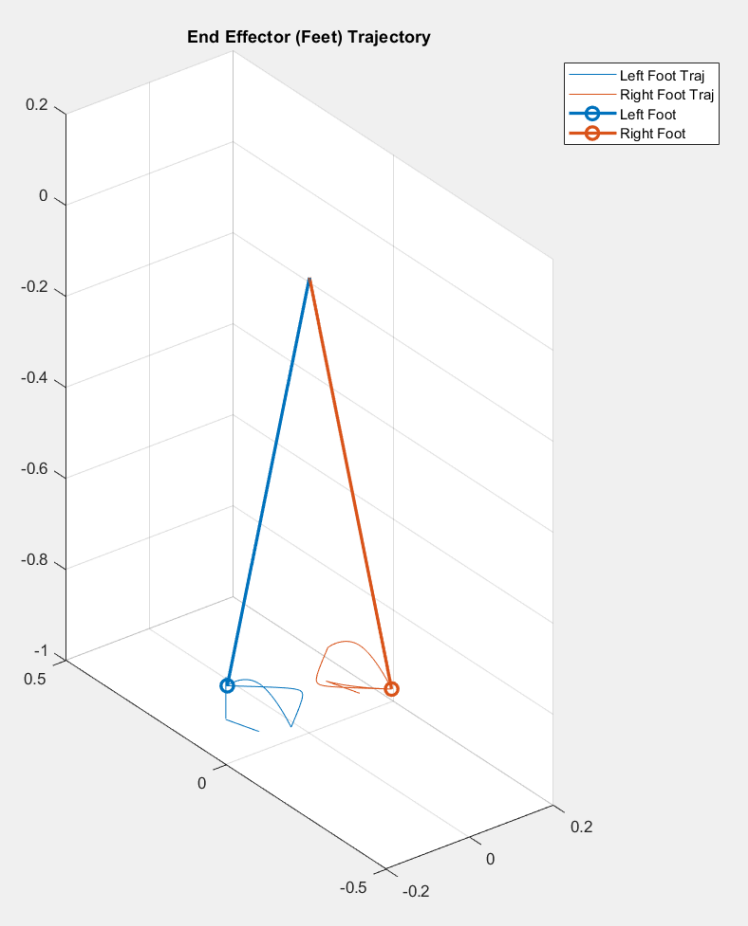


Figure 3.8: Feet Trajectories

3.4.1.3 Inverse Kinematics

The next step in the process was to convert the foot trajectories from the previous section to joint angles which can be used to control a Simulink model. To solve this problem, inverse kinematics would need to be employed on a robot structure containing the desired joints and degrees of freedom. The previous section contained only foot trajectories with a reference point, the ‘legs’ were simply for visualisation purposes.

This was accomplished by using the Live script animateInverseKinematics.mlx which sets up a rigid body tree using DH parameters to set up the kinematic chain. In this case the staff at MathWorks chose to use DH parameters which represent the legs of the HUBO humanoid robot, which are said to be common throughout many bipedal walking robots (Castro & Kim, 2020).

As described in Chapter 2.6, Denavit-Hartenberg parameters are a set of four parameters that describe the kinematic relationship between two adjacent links in a serial robotic manipulator. They are widely used in robotics to model the transformations between coordinate frames of consecutive robot joints.

The four Denavit-Hartenberg parameters are:

1. **Link Length (a_i):** Which represents the distance along the common normal between the *zi-1* and *zi* axes. It is the distance from the origin of frame *i-1* to the intersection of *zi-1* with the *xi* axis.
2. **Link Twist (α_i):** Which represents the angle between the *zi-1* and *zi* axes, measured about the *xi* axis. It is the angle from the *zi-1* axis to the *zi* axis, measured along the *xi* axis.
3. **Link Offset (d_i):** Which is the signed distance from the origin of frame *i-1* to the intersection of the *xi* and *zi* axes. It is the distance between the origins of frame *i-1* and frame *i* along the *zi-1* axis.
4. **Joint Angle (θ_i):** Which represents the rotation angle between the *xi-1* and *xi* axes, measured about the *zi-1* axis. It is the angle from the *xi-1* axis to the *xi* axis, measured along the *zi-1* axis.

The DH parameters can be seen for both the right and left leg in Table 3.1 and Table 3.2 respectively.

Table 3.1: Right Leg DH Parameters

Joint i	a_i	α_i	d_i	θ_i	Description
-	*L1*	*0*	*-L2*	*0*	Base to hip yaw
1	*0*	*-pi/2*	*0*	*0*	Hip yaw to hip roll
2	*0*	*-pi/2*	*0*	*0*	Hip roll to hip pitch
3	*L3*	*0*	*0*	*0*	Hip pitch to knee pitch
4	*L4*	*0*	*0*	*0*	Knee pitch to ankle pitch
5	*0*	*pi/2*	*0*	*0*	Ankle pitch to ankle roll
6	*0*	*0*	*L5*	*0*	Ankle roll to end effector (foot)

Table 3.2: Left Leg DH Parameters

Joint i	a_i	α_i	d_i	θ_i	Description
-	*-L1*	*0*	*-L2*	*0*	Base to hip yaw
1	*0*	*-pi/2*	*0*	*0*	Hip yaw to hip roll
2	*0*	*-pi/2*	*0*	*0*	Hip roll to hip pitch
3	*L3*	*0*	*0*	*0*	Hip pitch to knee pitch
4	*L4*	*0*	*0*	*0*	Knee pitch to ankle pitch
5	*0*	*pi/2*	*0*	*0*	Ankle pitch to ankle roll
6	*0*	*0*	*L5*	*0*	Ankle roll to end effector (foot)

Going on the previous description of the four DH parameters, the right leg parameters will be explained below.

1. Base to Hip Yaw (Joint 1):
    - Link Length ($a1$): $L1$
    - Link Twist ($\alpha 1$): 0 (No twist)
    - Link Offset ($d1$): $-L2$ (Negative $L2$, meaning the origin of frame 1 is below frame 0 along the z-axis)
    - Joint Angle ($\theta 1$): 0 (No rotation)

2. Hip Yaw to Hip Roll (Joint 2):
    - Link Length ($a2$): 0 (No length)
    - Link Twist ($\alpha 2$): $-\pi/2$ (90-degree twist along x-axis)
    - Link Offset ($d2$): 0 (No offset)
    - Joint Angle ($\theta 2$): 0 (No rotation)

3. Hip Roll to Hip Pitch (Joint 3):
    - Link Length ($a3$): 0 (No length)
    - Link Twist ($\alpha 3$): $-\pi/2$ (90-degree twist along x-axis)
    - Link Offset ($d3$): 0 (No offset)
    - Joint Angle ($\theta 3$): 0 (No rotation)

4. Hip Pitch to Knee Pitch (Joint 4):
    - Link Length ($a4$): $L3$
    - Link Twist ($\alpha 4$): 0 (No twist)
    - Link Offset ($d4$): 0 (No offset)
    - Joint Angle ($\theta 4$): 0 (No rotation)

5. Knee Pitch to Ankle Pitch (Joint 5):
    - Link Length ($a5$): $L4$
    - Link Twist ($\alpha 5$): 0 (No twist)
    - Link Offset ($d5$): 0 (No offset)
    - Joint Angle ($\theta 5$): 0 (No rotation)

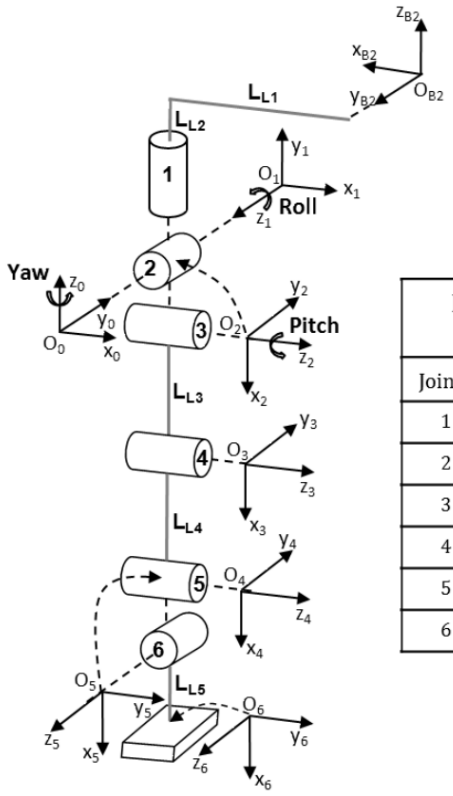
6. Ankle Pitch to Ankle Roll (Joint 6):
    - Link Length ($a6$): 0 (No length)
    - Link Twist ($\alpha 6$): $\pi/2$ (90-degree twist along x-axis)
    - Link Offset ($d6$): 0 (No offset)
    - Joint Angle ($\theta 6$): 0 (No rotation)

7. Ankle Roll to End Effector (Foot) (Joint 7):
    - Link Length ($a7$): 0 (No length)
    - Link Twist ($\alpha 7$): 0 (No twist)
    - Link Offset ($d7$): $L5$
    - Joint Angle ($\theta 7$): 0 (No rotation)

As can be seen in Table 3.1 and Table 3.2, the parameters are the same, except for $L1$. $L1$ is positive for the right leg, and negative for the left, indicating that they are positioned in opposite directions. The rigid body tree represents a robot with 6 DOF per leg, defined as ankle roll and pitch, knee pitch and finally hip roll, pitch, and yaw.

Having a rigid body structure, the foot trajectories could now be mapped onto that structure. To do this, the animateInverseKinematics.mlx script calls a function named invKinBody2Foot.m, written by the MathWorks staff team. The function makes use of an analytical solution to solve the inverse kinematics, based on the solution found in the paper titled, ''Closed-Form Inverse Kinematic Joint Solution for Humanoid Robots'' by Ali, Andy Park and Lee (2010). See Appendix B, Code Snippet 2 for variables changed by the researcher to suite their PLEE design.

In the paper by Ali, Andy Park and Lee (2010), it is said that Inverse kinematics in humanoid robotics often relies on iterative methods using the Jacobian matrix. However, closed-form joint solutions offer advantages over iterative approaches. Their study introduces a novel reverse decoupling mechanism method that analyses the kinematic chain of a limb in reverse order, decoupling it into positioning and orientation mechanisms. By applying the inverse transform technique, a consistent joint solution for humanoid robots can be derived. Their method provides a simple and efficient procedure for obtaining joint solutions for various humanoid robots. Computer simulations on a Hubo KHR-4 humanoid robot validate the effectiveness of the approach, which is said to be easily adapted for use with other humanoid robots with minor modifications. The final equations used in their solution will be outlined below for the right leg in the order given in the paper. Figure 3.9 shows the link coordinate frames of the right leg of a Hubo KHR-4 Robot and its D-H parameters.



Right Leg link coordinate parameters				
Joint i	θ_i	α_i	a_i	d_i
1	0	90°	0	0
2	−90°	−90°	0	0
3	0	0	L_{L3}	0
4	0	0	L_{L4}	0
5	0	90°	0	0
6	0	0	L_{L5}	0

Figure 3.9: Link Coordinate Frames of the Right Leg of a Hubo KHR-4 Robot and its D-H Parameters

(Ali, Andy Park & Lee, 2010)

To determine the position and orientation of the end-effector of a limb, the spatial displacement of the 6th coordinate frame relative to the base/reference coordinate frame can be obtained by multiplying the 6 link-transformation matrices together. This can be achieved by using the established link coordinate frames and the D-H parameters seen in Figure 3.9 (Ali, Andy Park & Lee, 2010).

$$
\begin{aligned}
{}^{0}\boldsymbol{T}_{6} = \prod_{i=1}^{6} {}^{i-1}\boldsymbol{A}_{i} = {}^{0}\boldsymbol{A}_{1}\,{}^{1}\boldsymbol{A}_{2}\,{}^{2}\boldsymbol{A}_{3}\,{}^{3}\boldsymbol{A}_{4}\,{}^{4}\boldsymbol{A}_{5}\,{}^{5}\boldsymbol{A}_{6} \\
= \begin{bmatrix} \boldsymbol{x}_6 & \boldsymbol{y}_6 & \boldsymbol{z}_6 & \boldsymbol{p}_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.2}
$$

The unit vectors along the principal axes of the coordinate frame i are denoted as x_i, y_i, and z_i. The general link transformation matrix, $i{-}1A_i$, relates the i^{th} coordinate frame to the $(i-1)th$ coordinate frame. Additionally, n, s, a, p represents the normal vector, sliding vector, approach vector, and position vector of the hand, respectively. This forward kinematic equation can be used to derive the closed-form joint solution (Ali, Andy Park & Lee, 2010).

$$
\begin{aligned}
\boldsymbol{T}' = \begin{bmatrix} \boldsymbol{n} & \boldsymbol{s} & \boldsymbol{a} & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{n}' & \boldsymbol{s}' & \boldsymbol{a}' & \boldsymbol{p}' \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
= {}^{6}\boldsymbol{A}_{5}\,{}^{5}\boldsymbol{A}_{4}\,{}^{4}\boldsymbol{A}_{3}\,{}^{3}\boldsymbol{A}_{2}\,{}^{2}\boldsymbol{A}_{1}\,{}^{1}\boldsymbol{A}_{0} = {}^{6}\boldsymbol{A}_{0}
\end{aligned}
\tag{3.3}
$$

To approach the inverse kinematics problem the joint angles are considered in reverse order. In this approach, the position and orientation of the base coordinate frame with respect to the end-effector coordinate frame can be represented by a new position vector, denoted as p'. This new position vector, p', is solely dependent on the three joint angles θ_4, θ_5, and θ_6. By decoupling the limb into positioning and orientation subsystems, the IK-problem can be solved in this reverse manner. To achieve this, we take the inverse of both sides of Equation (3.2), resulting in the new composite link transformation matrix T' being referenced to the end-effector coordinate frame (Ali, Andy Park & Lee, 2010).

By using the transform in equation (3.3), the joint solutions of the right leg of the Hubo robot can be derived (Ali, Andy Park & Lee, 2010). The first transformation goes from the base coordinate frame B2, which is attached to the body, to the first coordinate frame of the right leg (Equation (3.4)).

$$ {}^{B2}A_0 = \begin{bmatrix} -1 & 0 & 0 & L_{L1} \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -L_{L2} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4} $$

Where l_{Li} are geometric link parameters, as seen in Figure 3.9.

Equations (3.5) to (3.10) show the final equations from the paper for θ_1 to θ_6. They are given in the order presented by Ali, Andy Park and Lee (2010).

$$ \theta_4 = \text{atan2}(\pm\sqrt{1 - C_4^2}, \text{C}_4) \tag{3.5} $$

Where $\text{C}_i \equiv cos\,\theta_i$.

$$ \theta_5 = \text{atan2}\left(-p'_z, \pm\sqrt{(p'_x + L_{L5})^2 + {p'_y}^2}\right) - \psi \tag{3.6} $$

$$ \theta_6 = \text{atan2}(p'_y, -p'_x - L_{L5}) \tag{3.7} $$

$$ \theta_2 = \text{atan2}(\pm\sqrt{1 - (S_6 a'_x + C_6 a'_y)^2}, S_6 a'_x + C_6 a'_y) \tag{3.8} $$

Where $\text{S}_i \equiv sin\,\theta_i$.

$$ \theta_1 = \text{atan2}(-S_6 s'_x - C_6 s'_y, -S_6 n'_x + C_6 n'_y) \tag{3.9} $$

$$\theta_3 = \theta_{345} - \theta_4 - \theta_5 \quad (3.10)$$

When the script is run the joint trajectories are displayed through an animation to verify if they are moving as expected, this can be seen in Figure 3.10.

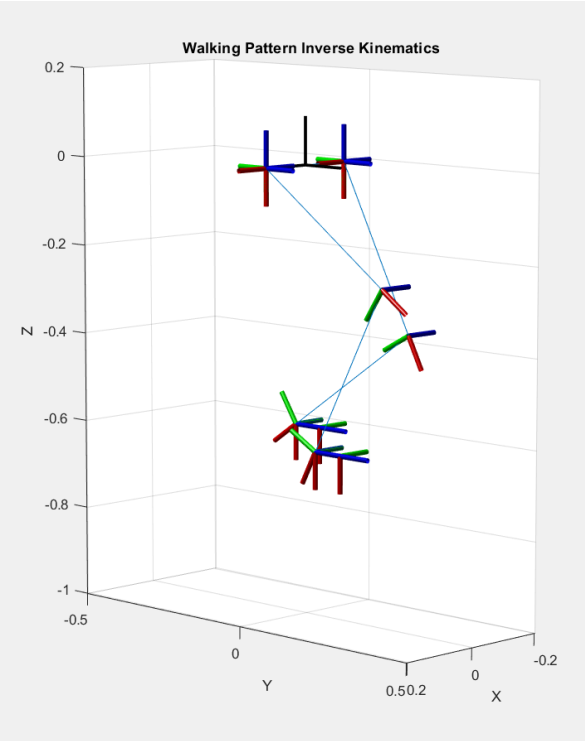


Figure 3.10: Inverse Kinematics Animation

The walking pattern inverse kinematics was verified by comparing the feet trajectories in the Inverse Kinematics Animation in Figure 3.10, to the desired Feet Trajectories that were seen in Figure 3.8. A visual comparison was made between the step lengths and feet placement positions. A check was also done to see that the joints were moving within their defined ranges of motion and that the knees were bending in the correct orientations. Having verified the walking pattern inverse kinematics pattern, the researcher could move on to using the Simulink model to verify if the robot was able to maintain balance and perform a walk. The resulting transform matrices for the feet that contains position and orientation information will later be used as an input to the simulated robot model. The Simulink model verification will be discussed further in the next section.

3.4.1.4 Verification using the Simulink Model

Where the inverse kinematic model was used to verify the joint trajectories, the Simulink model is used to model the physical 3D mechanics of the PLLE to see the walking pattern on a flat terrain. The same joint trajectories of the walking pattern inverse kinematics animation, which can follow the given trajectory exactly, will be applied to the Simulink model.

By opening the Simulink model walkingRobotLIPM.slx, one can see how the flow of the software operates. Refer to Figure 3.11. The simulation inputs highlighted on the left of Figure 3.11, siminR, and siminL, contain the necessary transforms calculated using the previous scripts for the foot trajectories. The inverse kinematics block once again uses the invKinBody2Foot.m function to calculate the joint trajectories. In the centre of the image is a Motion block, this defines which type of robot joint controller will be used when running the simulation.

The options for ‘’Set actuator type:’’ above the Motion block allow the user to select which type of actuator to use. ‘’Motion’’ refers to idealised motion of the joints, meaning that the joints are given a reference trajectory which they can then execute without any limitation of torque or actuation speed.

‘’Torque + Servo’’ refers to a joint controller which has a torque limit, apart from the torque limit, the torque is controlled by a closed loop torque controller using PID control.

‘’Motor + Servo’’ allows one to model and specify the properties of specific actuators, it combines a closed loop torque controller with a motor model. For this research which seeks to discover the necessary specifications for the hardware necessary to complete untethered walking of a PLLE, the ‘’Motor + Servo’’ option was not used.

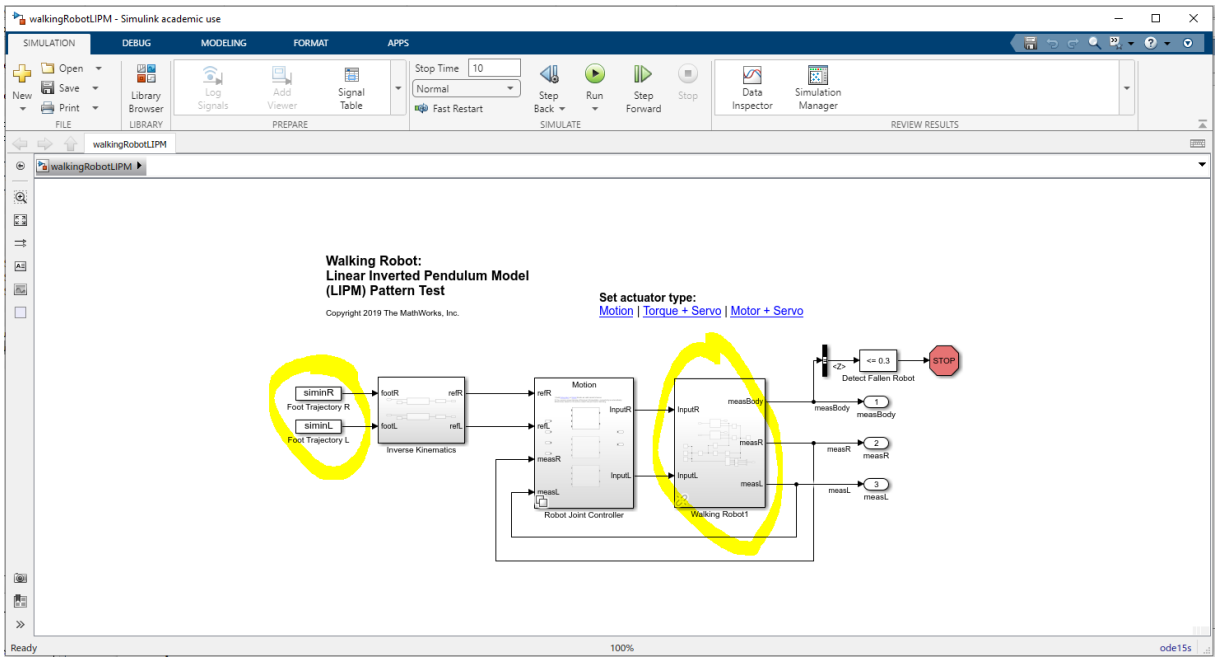


Figure 3.11: Walking Robot LIPM Simulink Model

3.4.1.5 Simulink Model CAD Referencing

In order for the Simulink Model to simulate the walking trajectory on the CAD model that was completed, it was necessary to reference the CAD files within the model. To import a model into Matlab, Simscape Multibody 'File Solid' blocks were used. These blocks allowed the linking of a CAD file and setting up of frames, inertia as well as mass properties. By keeping the same layout of the workflow, the Brick and Cylindrical Solids could be replaced directly without impacting the Simulink document.

By entering the ''Walking Robot1'' block, which is highlighted in Figure 3.11, a block for ''Torso'' can be seen. In Figure 3.12 the Torso File Solid can be seen. The researcher deleted the previous brick solid which was simply a rectangular body with parameterised variables to modify its size (parameters found in robotParameters.m file). As can be seen in the Figure, the values for the mass and mass moments of inertia have been taken from Onshape and entered here. The reference frames were also modified to match the frames previously depicted in the brick solid.

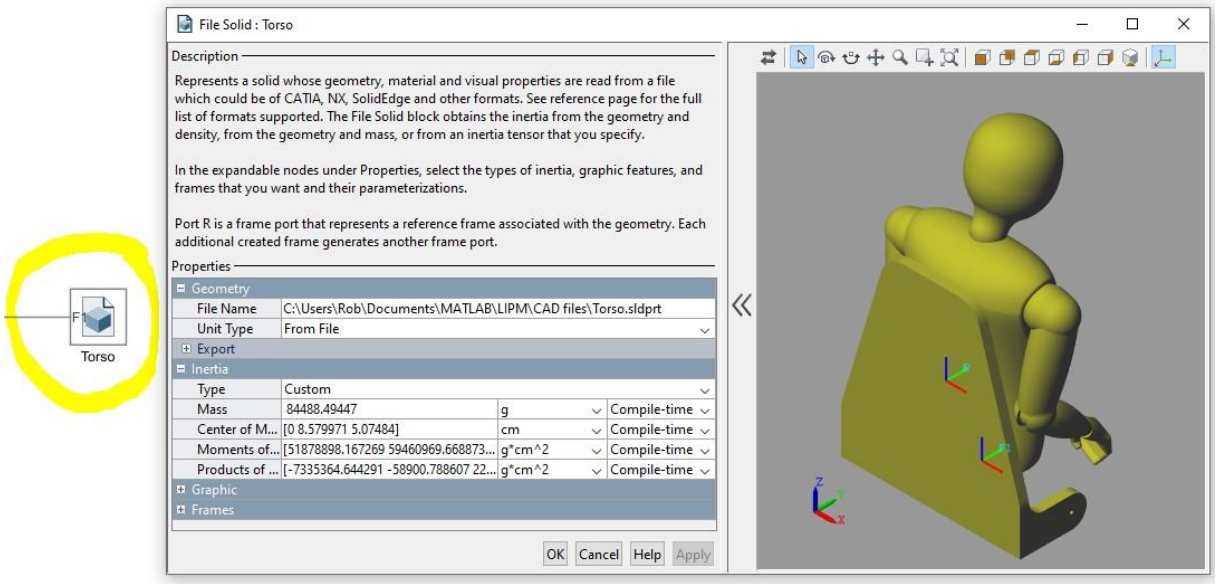


Figure 3.12: Torso File Solid

The same procedure was followed for the Foot, the Lower Leg, and the Upper Leg for both the left and the right leg.

Figure 3.13 illustrates a good example of frames created on the right foot to match the previous frames that existed. On the block named ''Foot_Right'' five frames were defined. Frames F1, F2, F3 and F4 are connected to spheres which model the contact force of each corner of the foot, effectively creating a contact plane as the foot is flat.

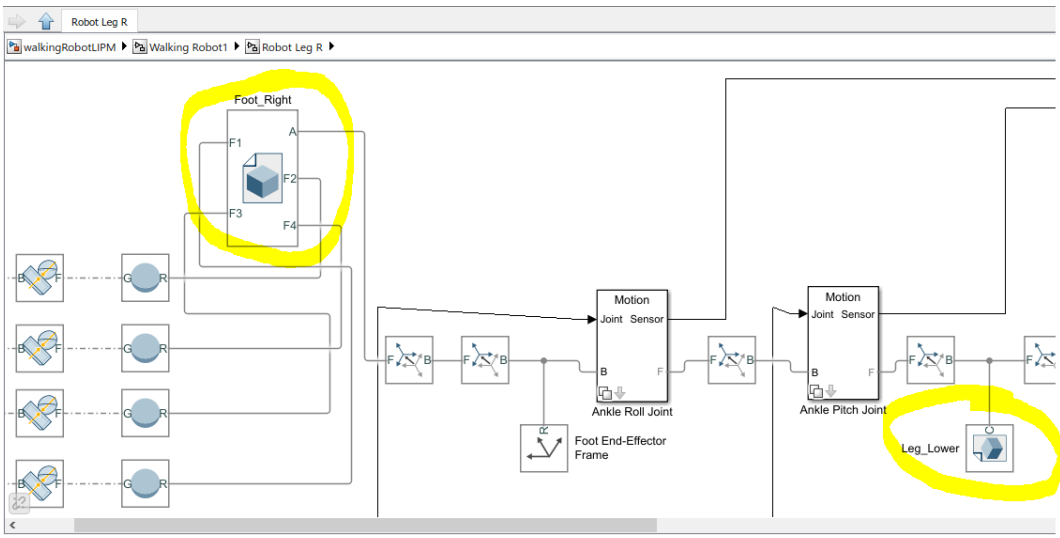


Figure 3.13: Simulink right leg model

3.4.1.6 Running Simulink Simulation

Once the CAD model was set up, the walking pattern could be tested. By running the Simulink model, the Mechanics Explorer is opened which displays an animation of the PLLE walking on a surface (Figure 3.14). The mechanical system is simulated with Simulink's solvers, which take into account the mechanical systems contact forces, as well as the physical properties of the solid bodies accelerating within the world environment. The researcher observed whether the PLLE fell down or not and modified the input variables in the previous steps to alter the walking pattern until the PLLE was able to perform a walk without falling during 10 seconds of simulation time. At this stage of the research the ''Motion'' profile was used first, as it was easier to obtain a stable walking gait by first not imposing limits on the motion of the joints. After the stable walking was obtained, the joints were actuated using the ''Torque'' profile to try to obtain results which better represent what a motor with a closed loop controller could achieve. The results will later be discussed in the next chapter.

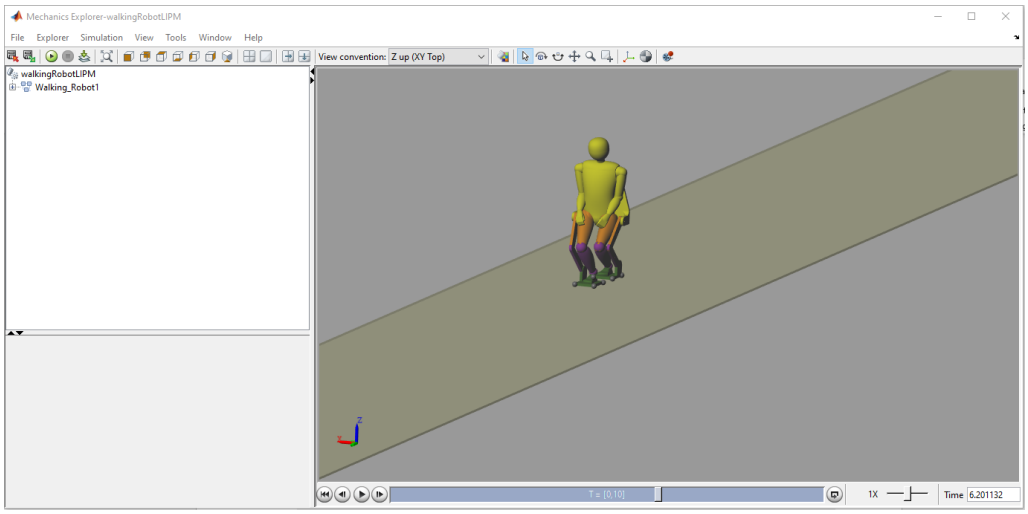


Figure 3.14: Mechanics Explorer

3.4.2 Optimisation using a Genetic Algorithm

This section will discuss another method of obtaining a walking pattern and how this walking trajectory was optimised. The results from this section will later be compared to the results from the previous LIPM walking trajectory in the previous section. For this method, Version 2.0.0.0 of the Walking Robot software released by the MathWorks Student Competitions Team was used (MathWorks Student Competitions Team, 2023).

To first gain an initial walking pattern, the researcher ran the optimizeRobot.m script with zero motion initial conditions as to not guide the algorithm to a certain predefined gait pattern. The simulation was also initially set to run with idealised motion of the joints, this would allow the algorithm to converge to a walking pattern that is able to walk more quickly.

The model used in this section was modified to have larger feet platforms, as the reduced degrees of freedom per leg (3 DOF as opposed to 6 DOF in the LIPM setup) left the robot unable to converge to a stable walking pattern when run with the same feet platforms as those used in the LIPM simulation.

Once a walking pattern was achieved, the simulation was run using the walking pattern as its initial condition. The researcher also moved to using the torque actuated joints to achieve a result which would more closely resemble a result that actuators could achieve.

Initially a large unrealistic torque value was used to aid the researcher to select PID values for the joints which would still allow the model to maintain balance. Subsequently the best simulations were chosen as initial conditions and the torque was gradually lowered as soon as a lower value achieved a satisfactory walking gait.

The following sections will elaborate on how this was achieved.

3.4.2.1 Model Import and Setup

To begin using the optimisation workflow, it was first necessary for the researcher to import the CAD model as before. The same procedure was followed as before, by replacing the Brick or Cylindrical Solids to File Solid blocks, the CAD model can be referenced. For this Simulink model, the robot's degrees of freedom have been reduced to 3 DOF per leg. This can be seen in Figure 3.15 where there are only motion blocks for Ankle Joint, Knee Joint and Hip Joint for the right leg.

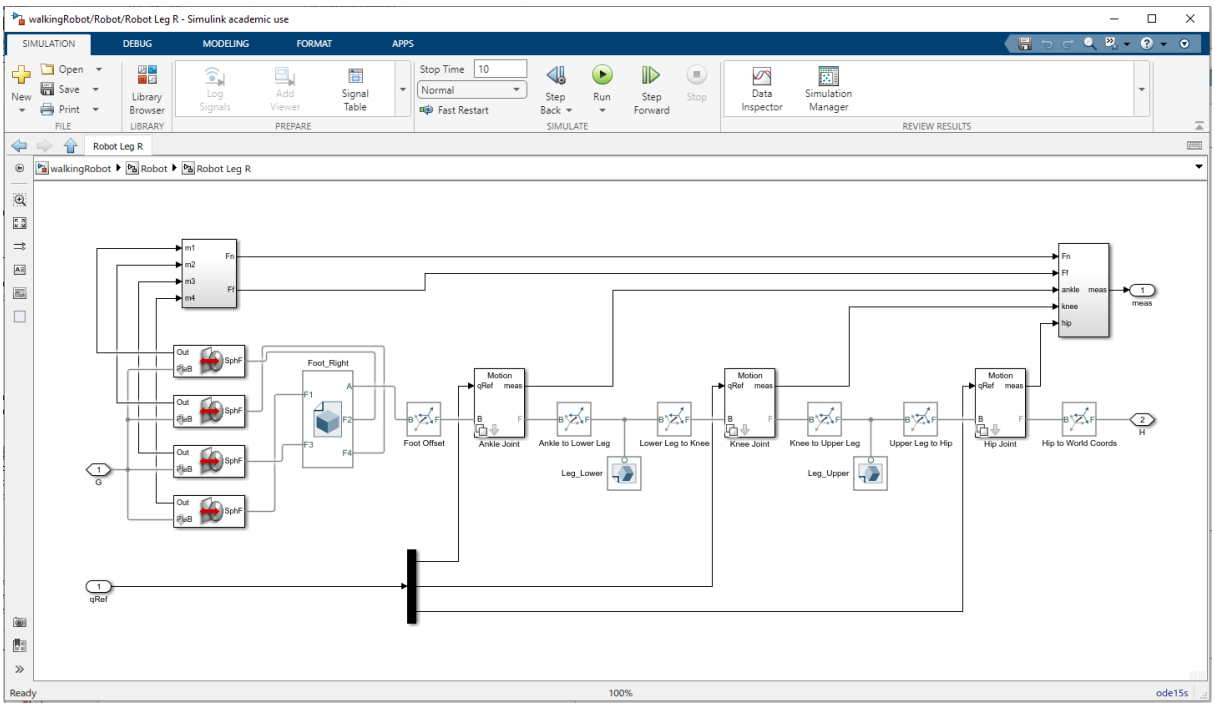


Figure 3.15: Right Leg Simulink Model

3.4.2.2 Optimisation Variable Entry

After having imported the CAD model into the Simulink model, the researcher could move on to entering the necessary variables in the robotParameters.m script. The values for these variables were mostly copied from the robotParameters.m script in the previous workflow of the LIPM. The values which differ or were not used in the LIPM workflow are listed in Appendix B, Code Snippet 3.

With the Robot Simulink model set up and the robotParameters.m script values modified; the researcher could move on to using the optimisation script. Within the ‘’Optim’’ folder, there is a script named optimizeRobotMotion.m. The script is used to optimise the walking trajectory of the robot by running a Genetic Optimisation algorithm which repeatedly runs the Simulink model towards optimising a cost function (Castro, 2017). It can be seen that in a section of the code within the optimizeRobotMotion.m script in Appendix B, Code Snippet 4, the cost function being used is a function named **costFcn** which calls a script named simulateWalkingRobot.m.

Within the simulateWalkingRobot.m script, there is a section of code which is used to run the Simulink model and unpack the logged data from the simulation. The function calculates a penalty value based on the logged data. The penalty is then returned to the **costFcn** function, and the optimisation process is programmed to minimise this cost function (see Appendix B, Code Snippet 5).

The code has two main components to calculate the penalty:

1. Positive Reward:

- It calculates a positive reward based on the distance travelled without falling (xEnd), the MATLAB sign function of xEnd, and the duration of the simulation (tEnd).
- The reward is determined by multiplying the square of xEnd by tEnd, with the sign of xEnd taken into account.

In the form of a mathematical expression, the positive reward could be expressed as in Equation 3.11.

$$positiveReward = sign(xEnd) \times (xEnd)^2 \times tEnd \tag{3.11}$$

Here:

- $xEnd$ is the distance the PLLE has covered before the simulation terminates.
- $sign(xEnd)$ represents the MATLAB sign function, which returns -1 for negative values of $xEnd$, 1 for positive values, and 0 for zero.
- $tEnd$ is the time taken to before the simulation terminates.

2. Negative Reward:

- It calculates a negative reward based on two factors: angular velocity (wAvg), which amounts to deviation from a straight line, and trajectory aggressiveness.
- Trajectory aggressiveness is determined by analysing the derivatives of the hip_motion, knee_motion, and ankle_motion arrays.
- It counts the number of times the derivative changes sign (flips from positive to negative or vice versa) and increments the aggressiveness value accordingly.
- The aggressiveness value is multiplied by the maximum of abs(wAvg) and 0.01, with a minimum value of 1.

The pseudocode broken down into steps, to calculate the $negativeReward$ found in Appendix B, Code Snippet 5 is outlined below. Variables and functions are displayed in bold.

1. Initialize **aggressiveness** to 0.
2. Compute the differences and concatenate them into **diffs** for **hip_motion**, **knee_motion**, and **ankle_motion**. The diff function calculates the differences between

consecutive elements in an array. In simpler terms, it computes the "difference" between each element and the one that comes immediately before it in the array. This helps one find the increments or changes between adjacent elements in a sequence.

3. Iterate over the elements of **diffs** from 1 to **numel(diffs) - 1**. The numel function is short for "number of elements." It returns the total number of elements in an array.
   a. Check if the sign of the ratio of **diffs(idx)** and **diffs(idx + 1)** is negative.
   b. Also, check if the index **idx** is not a multiple of **N**.
   c. If both conditions are true, increment **aggressiveness** by 1.
4. Calculate **negativeReward** as the product of the maximum of the absolute value of **wAvg** and 0.01, and the maximum of **aggressiveness** and 1.

Finally, the penalty value is computed as the negative ratio of the positive reward to the negative reward. This penalty is used in an optimisation process where the goal is to minimise the cost function. In the form of a mathematical expression, the penalty could be expressed as in Equation 3.12.

$$penalty = -\frac{positiveReward}{negativeReward} \tag{3.12}$$

The options and functionality for the optimizeRobotMotion.m script will be discussed below, each relevant subsection of code will be linked in the appendix, and its function subsequently explained. The first section of code can be found in Appendix B, Code Snippet 6.

The next section of code is where the user can input what type of actuation strategy the simulation should be run with, with the three actuation types that were previously discussed as options (see Appendix B, Code Snippet 7).

The scalingFactor function is used to scale from degrees to integers. The variable called numPoints is used to set how many waypoints each joint should have. In the context of trajectory planning for joints, waypoints represent specific key positions or configurations that a joint must pass through during its motion. These waypoints serve as essential reference points along the trajectory, helping to define the desired path that the joint should follow. In this case the number of waypoints has been set to 6. With 6 waypoints and 3 joints, being the ankle knee and hip joint, a total of 18 optimisation parameters are set (6 waypoints x 3 joints). Because the left leg follows the same trajectory as the right leg but lags behind it by half a period, the optimisation parameters needed are reduced by a half (Castro, 2017). In summary, the chromosome is coded as an 18 integer vector. Therefore there are 6 unknown parameters for each joint, the ankle, knee, and hip joint respectively.

As is stated in the comment, the gaitPeriod variable sets the period of the walking gait (see Appendix B, Code Snippet 8).

Within the optimizeRobotMotion.m script, initial conditions of the walking gait can be set to generate from zero motion initial conditions, or from a previous gait by loading the file of previously recorded data.

The user is also able to define the number of generations, as well as the population size of the genetic algorithm. The script contains upper and lower limits for the joint movements' settings, for example the lower bound of the knee being set to zero makes sure the knee joint never bends forwards (see Appendix B, Code Snippet 9).

3.4.2.3 GA Function Outline

The MATLAB Help Centre documentation ("How the Genetic Algorithm Works - MATLAB & Simulink - MathWorks United Kingdom", n.d.) states that the GA function works as follows,
"

1. The algorithm begins by creating a random initial population.
2. The algorithm then creates a sequence of new populations. At each step, the algorithm uses the individuals in the current generation to create the next population. To create the new population, the algorithm performs the following steps:
    a. Scores each member of the current population by computing its fitness value. These values are called the raw fitness scores.
    b. Scales the raw fitness scores to convert them into a more usable range of values. These scaled values are called expectation values.
    c. Selects members, called parents, based on their expectation.
    d. Some of the individuals in the current population that have lower fitness are chosen as *elite*. These elite individuals are passed to the next population.
    e. Produces children from the parents. Children are produced either by making random changes to a single parent—*mutation*—or by combining the vector entries of a pair of parents—*crossover*.
    f. Replaces the current population with the children to form the next generation.
3. The algorithm stops when one of the stopping criteria is met.
4. The algorithm takes modified steps for linear and integer constraints.

"

For the first optimisation, as in step 1, the initial population for this research was set to values of 0 and progressed from there. The population size was set to 20. For further optimisations the initial population could be set to add copies of the initial gait of a previous optimisation. The cost or fitness function as discussed in step 2 was detailed in the previous section. As the

options for the ‘ga’ functions children creation has been left to default for this research, the genetic algorithm creates both crossover and mutation children for the next generation. The simulation is set to terminate when the maximum generations have been met, which in this case was 50. Integer constraints were also set to place restrictions on joint movements. The mutation rate was not specified in the options and would therefore be the default of 1/n, where 'n' is the number of genes (parameters) in the chromosome. Although the options for the ‘ga’ function were discussed here, they can also be seen in Appendix B, Code Snippet 9.

The MATLAB Help Centre documentation (“How the Genetic Algorithm Works - MATLAB & Simulink - MathWorks United Kingdom”, n.d.) defines the method for creating crossover and mutation children as follows:

The algorithm produces crossover offspring by merging pairs of parents in the existing population. For each coordinate in the offspring vector, the default crossover procedure randomly selects a gene from one of the two parents at the corresponding coordinate and assigns it to the offspring. In scenarios involving linear constraints, the default crossover function generates the offspring as a random weighted average of the parental vectors.

The algorithm generates mutation offspring by introducing random alterations to the genes of individual parents. By default, for problems without constraints, the algorithm incorporates a random vector from a Gaussian distribution into the parent's genetic makeup. In situations where constraints exist, either in the form of bounds or linear constraints, the resulting offspring remains within the feasible solution space.

3.4.2.4 GA Function Live Plot

In Figure 3.16 is an example of what the MATLAB live plot of the Genetic Algorithm looks like.

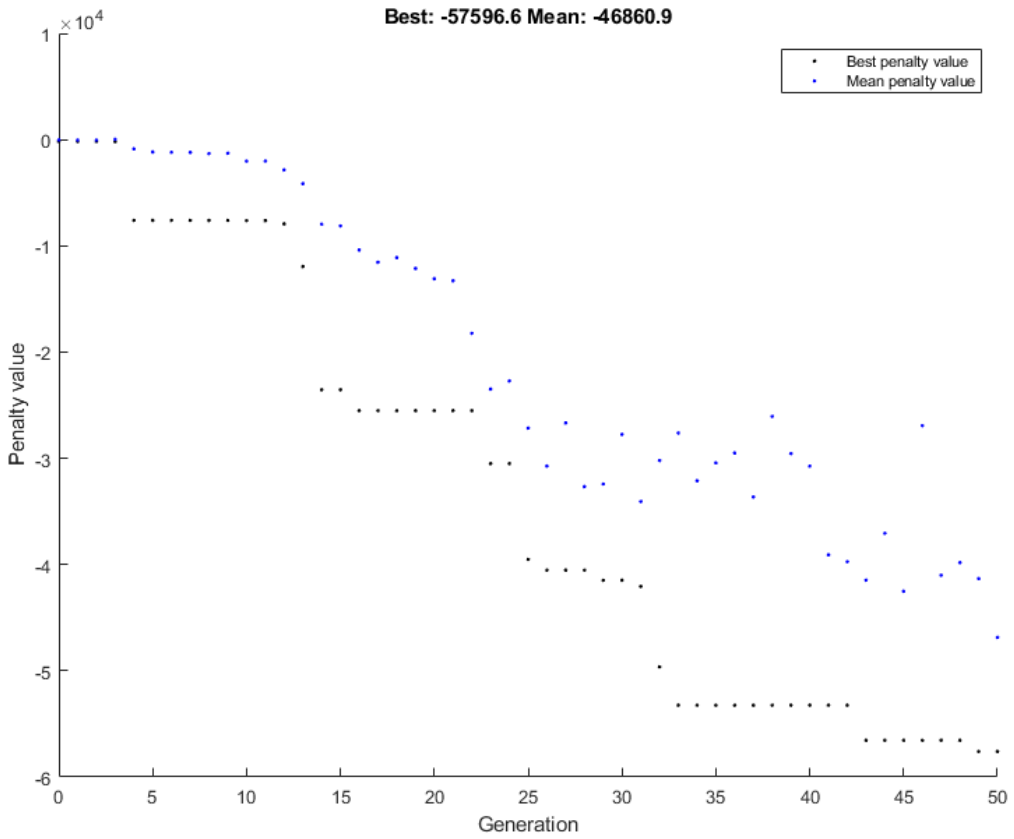


Figure 3.16: Genetic Algorithm Plot

A further explanation of Figure 3.16 is given below:

X-Axis: Generation

The X-axis represents the generations or iterations of the genetic algorithm. Each generation corresponds to a cycle of the algorithm's operations, including selection, crossover (recombination), mutation, and evaluation of individuals (potential solutions).

Y-Axis: Penalty Value

The Y-axis represents the penalty value associated with the solutions generated by the genetic algorithm. The penalty value is a metric used to quantify how well a particular solution satisfies certain constraints or objectives. It's a way to measure the quality of a solution.

In optimisation problems, the goal is usually to minimise the penalty value. A lower penalty value indicates a better solution that is closer to meeting the desired objectives or constraints.

The dots on the graph represent the penalty values for different individuals (potential solutions) in each generation. Each dot corresponds to a solution generated by the genetic algorithm.

The vertical position of a dot indicates the penalty value associated with the corresponding solution. The plot includes values to indicate the best penalty value achieved across all generations as it progresses. These values represent the lowest penalty value found during the optimisation process. It signifies the best solution found so far. Similarly, the plot shows markers for the mean (average) penalty value of all solutions in each generation. These dots provide an indication of the overall progress and convergence of the genetic algorithm. As the algorithm converges toward an optimal solution, the mean penalty value is expected to decrease over generations (“An Introduction to Genetic Algorithms”, 1996).

Ideally, as the generations progress, one would observe a trend of decreasing penalty values. This signifies that the genetic algorithm is improving the quality of solutions over time. Fluctuations in the penalty values, especially in the early generations, are common. They might represent exploration of the solution space and attempts to find promising regions. A steady decrease in the mean penalty value and convergence of the best penalty value indicate that the genetic algorithm is making progress toward finding a good solution (“An Introduction to Genetic Algorithms”, 1996).

In summary, the "Genetic Algorithm Plot" provides a visual representation of how the penalty values of different solutions change over generations. It helps one track the performance of the genetic algorithm optimisation process and assess whether it is successfully converging towards optimal or near-optimal solutions (“An Introduction to Genetic Algorithms”, 1996).

As in the LIPM simulation, the compareActuatorTypes.m script was used to obtain the data documented in the following chapter.

CHAPTER 4

SIMULATION RESULTS

This chapter presents the results of the two methods employed to achieve a stable walking gait, as discussed in the preceding chapter. The first method involves the conventional LIPM, while the second method utilises an optimisation technique incorporating a cost function and genetic algorithm. The results will be analysed based on the criteria specified in Chapter 3's requirements section. These criteria include the distance covered by the robot within a 10-second simulation time, the maximum torque observed at the ankle, knee, and hip joints during the simulation, the torso height to measure the robot's upright walking posture, and an objective assessment of the human-likeness of the gait. The results will be presented in graph format, generated using MATLAB software. After many iterations, both the LIPM simulation and the optimisation method simulation successfully achieved the objective of walking for the full 10 seconds of simulation time without experiencing any instances of falling.

4.1 LIPM Results

This section presents the final results obtained from the idealised motion simulation of the LIPM approach. As mentioned in the methodology, achieving this gait required an iterative process of guess-and-check, refining the parameters until the robot demonstrated a stable walking pattern. The following section will explore the optimisation technique, highlighting the iterative nature and the progressive improvements made throughout the iterations. Subsequently, a comparison will be drawn between the final result of the optimisation technique and the outcome presented in this section.

In Figure 4.1 a screen shot can be seen of the Mechanics Explorer within MATLAB which is playing the animation of the final LIPM walking pattern result. The video displaying the simulation can also be found by following this link. The robot can be seen to walk along a flat surface runway. The direction of travel is in the y-direction, the x-direction is to the side of the robot and the z-direction is the vertical position.

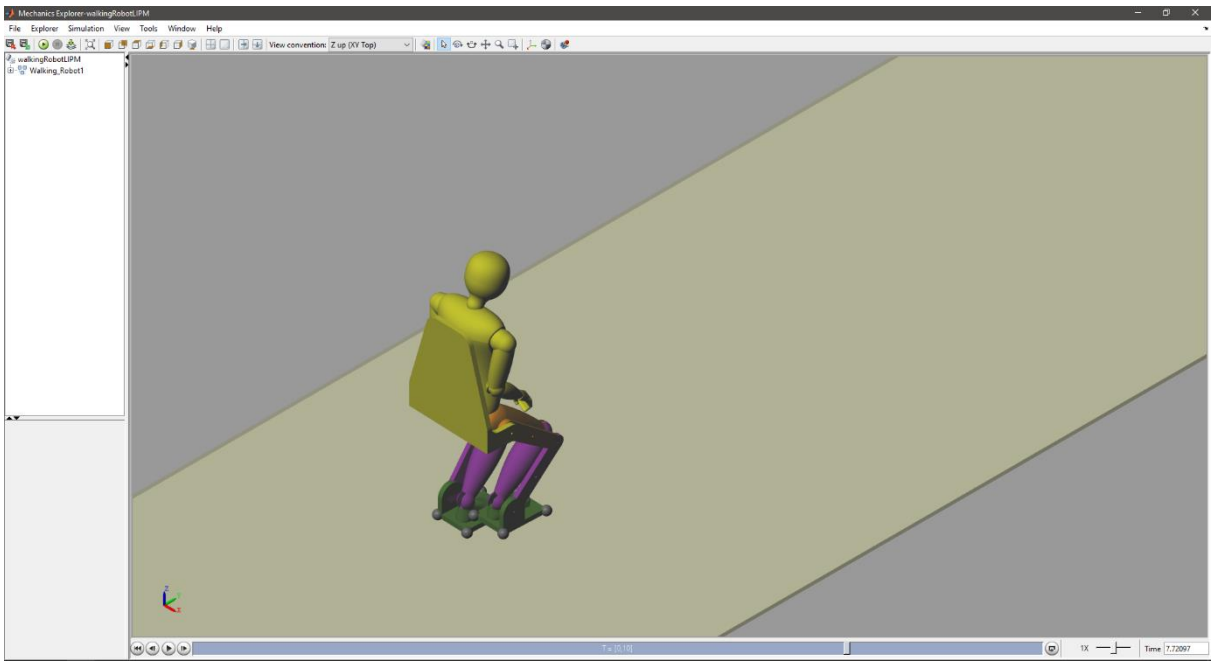


Figure 4.1: LIPM Mechanics Explorer Animation

The first step in the analysis of the LIPM simulation was to evaluate the human-likeness of the robot's gait. To accomplish this, it is necessary to compare the components of the gait to that of the human gait discussed in the literature review.

In Figure 4.2, an overlay of the different stages of the gait from the simulation onto the gait cycle that was previously shown can be seen.

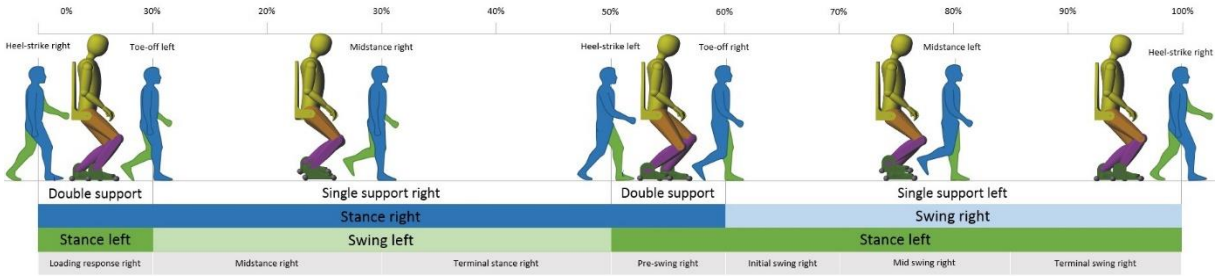


Figure 4.2: Robot Gait Cycle Overlay

(Adapted from “Gait_cycle.jpg (1860×412)”, n.d.)

Figure 4.2 shows that the robot does have discrepancies to a normal human walking gait. Firstly, it can be seen that the robot walks with knees bent, in a squat-like posture. Robots that employ the Linear Inverted Pendulum Model often exhibit a walking pattern with bent knees or a slightly squat-like posture as a result of fixing their CoM to a specific height (Brasseur et al., 2015). This is because the LIPM is a simplified model that approximates human walking dynamics using an inverted pendulum analogy (Kajita et al., n.d.). In the LIPM, the robot's

centre of mass is treated as a point mass located at a fixed height above the ground, and its motion is governed by principles of pendulum dynamics (Li et al., 2021).

When walking, humans tend to have a more upright posture with extended legs during the stance phase, as well as a smoother transition between the stance and swing phases (Hong & Lee, 2016). However, the LIPM does not account for the complexities of human musculoskeletal systems and the intricate control mechanisms involved in maintaining balance and executing a natural gait (Yuan & Liu, 2018).

Another area where this simulation differs from the natural walking gait cycle is during the double support phases. There is no heel-strike or toe-off phase because the model is restricted to having the feet always be normal to the ground. Another simplification is that the moment one foot strikes the floor, the other foot lifts, making it that there is essentially no double support phase (Castro & Kim, 2020). This could contribute to higher torques being seen during the gait cycle. Brasseur et al. (2015) state that, ‘’Imposing the CoM to move on a horizontal plane typically leads to unnatural walking motions, with low trunk and bent knees, that require greatly increased torques and excessive speeds in knee joints, both impacting negatively the efficiency of the resulting motion in terms of amplitude, speed and energy consumption.’’ This could lead to actuator requirements that are hard to achieve if the actuator needs to comply to being light weight, high torque, as well as being able to achieve high speeds.

Lastly it can be seen that the robot is taking very small steps. Hong and Lee (2016) assert that maintaining a constant centre of mass height restricts a humanoid robot's ability by enforcing a limited step length and results in an unnatural walking pattern. This may help keep the robot stable but would mean that the robot does not cover as much distance as if it were to take larger steps at the same speed. Of course, there is a balance between speed and stability but in this case, it would be desirable to take a stride that resembled a human stride. It also does not look natural to take steps where the swing foot does not advance much further than the stance foot during a step.

As depicted in Figure 4.3, the graphs display the total distance covered within the given timeframe as well as the torso height during this time. The top of the Figure reveals 1.1 meters walked by the LLE during the 10-second period. The Torso Height graph illustrates the variation in torso height throughout the walking pattern. Starting at a height of 0.95 meters, the torso drops to 73.5 centimetres within the first second and maintains this height for the remainder of the 10-second duration. This drop of height is consistent with what was seen in the previous section, where the robot had a squat-like posture and walked with bent knees

throughout the simulation, remaining at approximately the same height throughout once it has lowered.

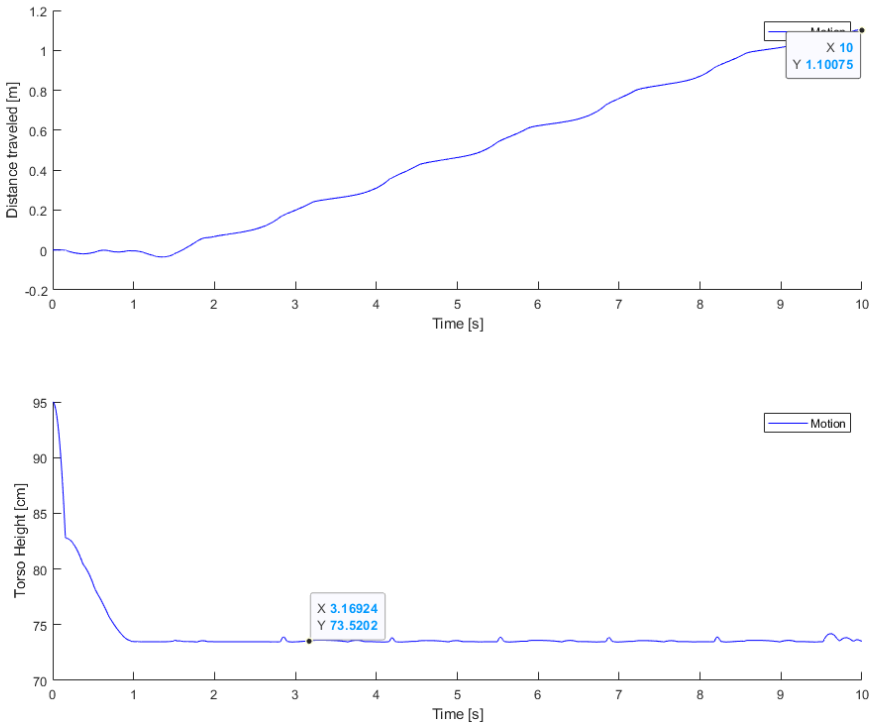


Figure 4.3: LIPM Global Data

Figure 4.4 shows the joint angles and joint torques over time of the ankle roll. Joint torque spikes can be seen to reach 1064.87 Nm during the repeating pattern and a very high spike of 2676.89 Nm towards the end. This could be an area where a double support phase could help to reduce torque.

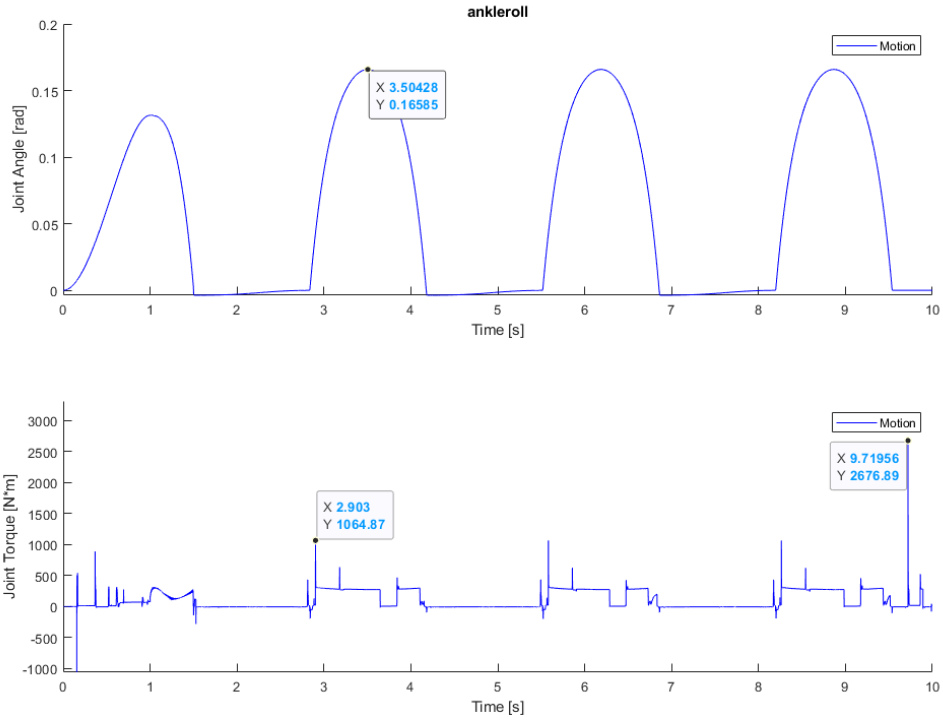


Figure 4.4: LIPM Ankle Roll

In Figure 4.5, the joint angles and joint torques are shown for the ankle pitch. Joint spikes of 472.40 Nm can be seen in the positive direction and 983.74 Nm in the negative direction.

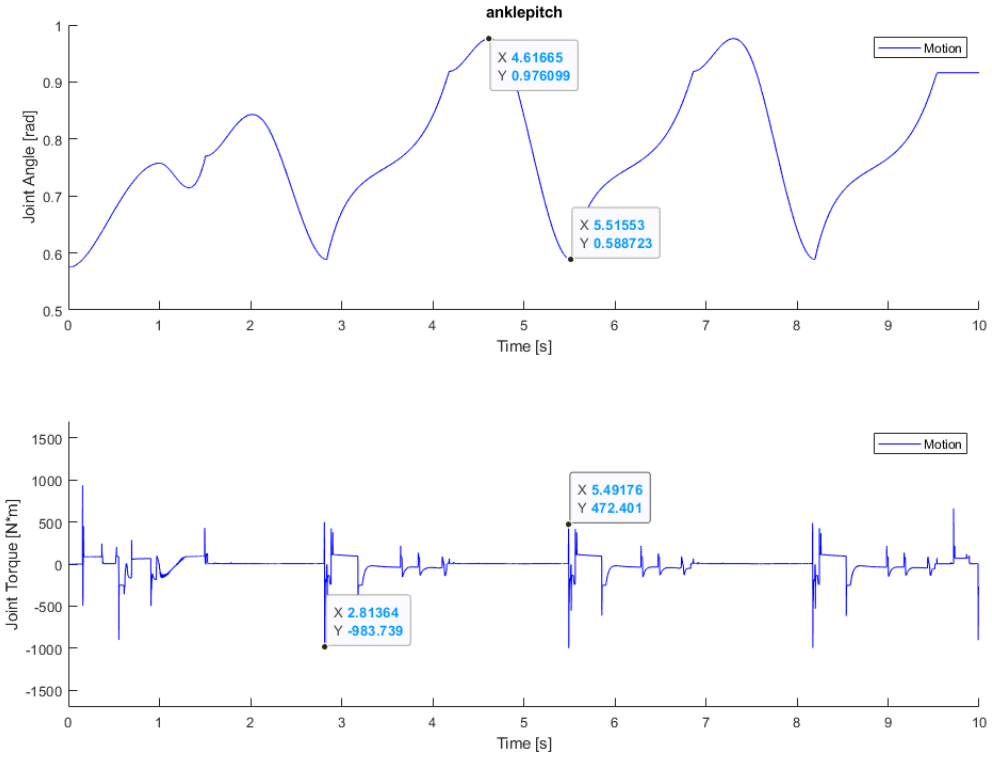


Figure 4.5: LIPM Ankle Pitch

The joint angles and joint torques' for the knee joint are shown in Figure 4.6. Joint spikes of 1894.47 Nm can be seen in the positive direction, whereas spikes of 440.48 Nm can be seen.

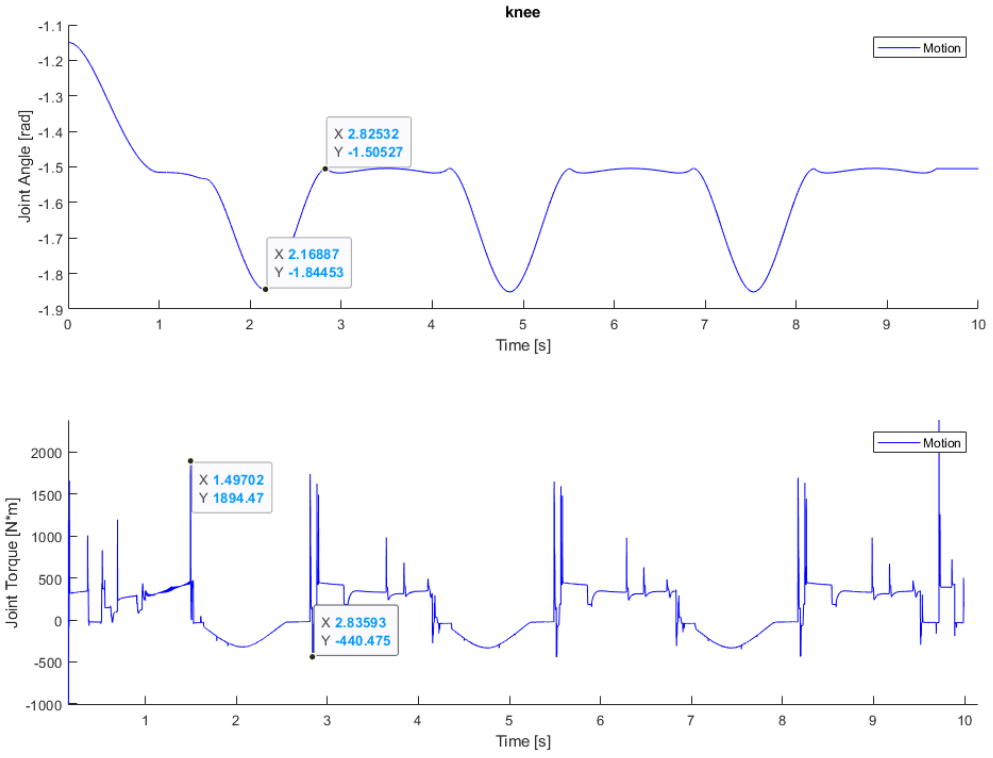


Figure 4.6: LIPM Knee Pitch

Figure 4.7 displays the data for the hip roll of the hip joint. Joint torque spikes of 2432.44 Nm can be seen in the positive direction and spikes of 1444.08 Nm can be seen.

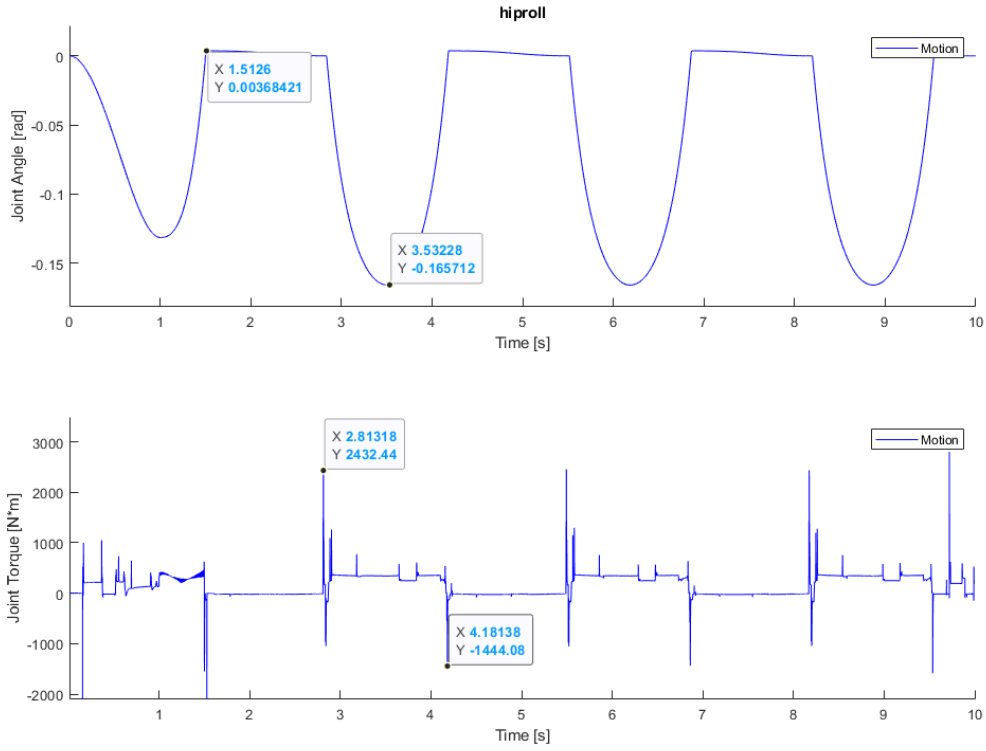


Figure 4.7: LIPM Hip Roll

Hip pitch data is displayed in Figure 4.8. Joint torques range from 1187.66 Nm to 1251.52 Nm.

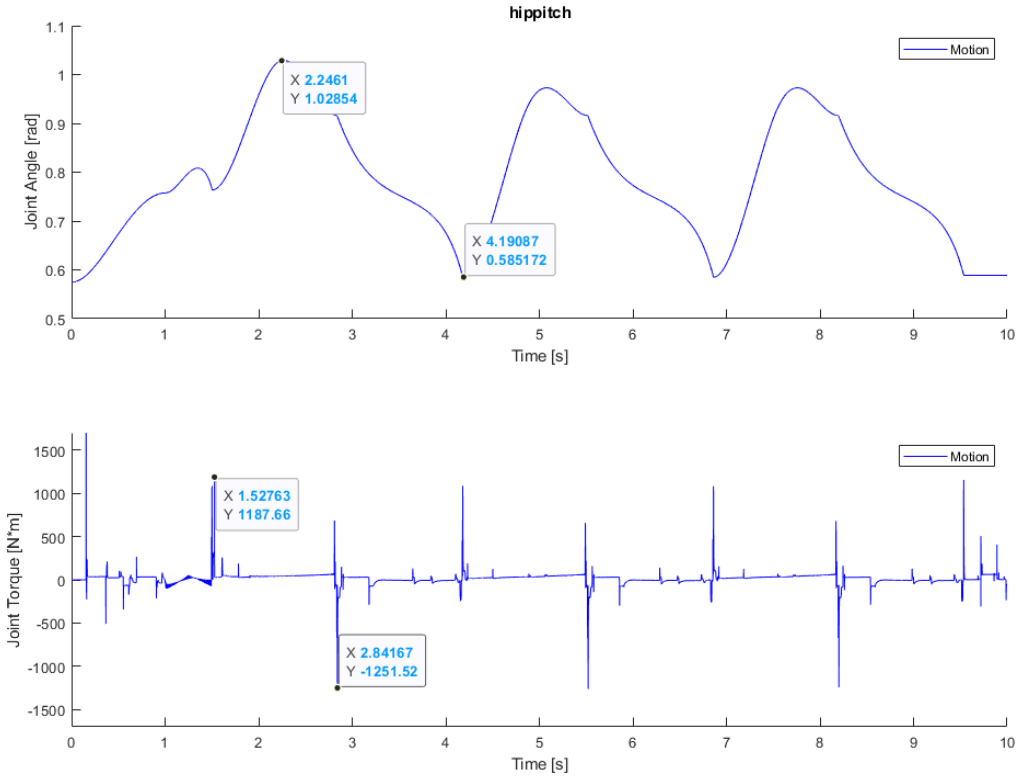


Figure 4.8: LIPM Hip Pitch

Figure 4.9 shows joint torque spikes of 948.805 Nm and 474.22 Nm for hip yaw.

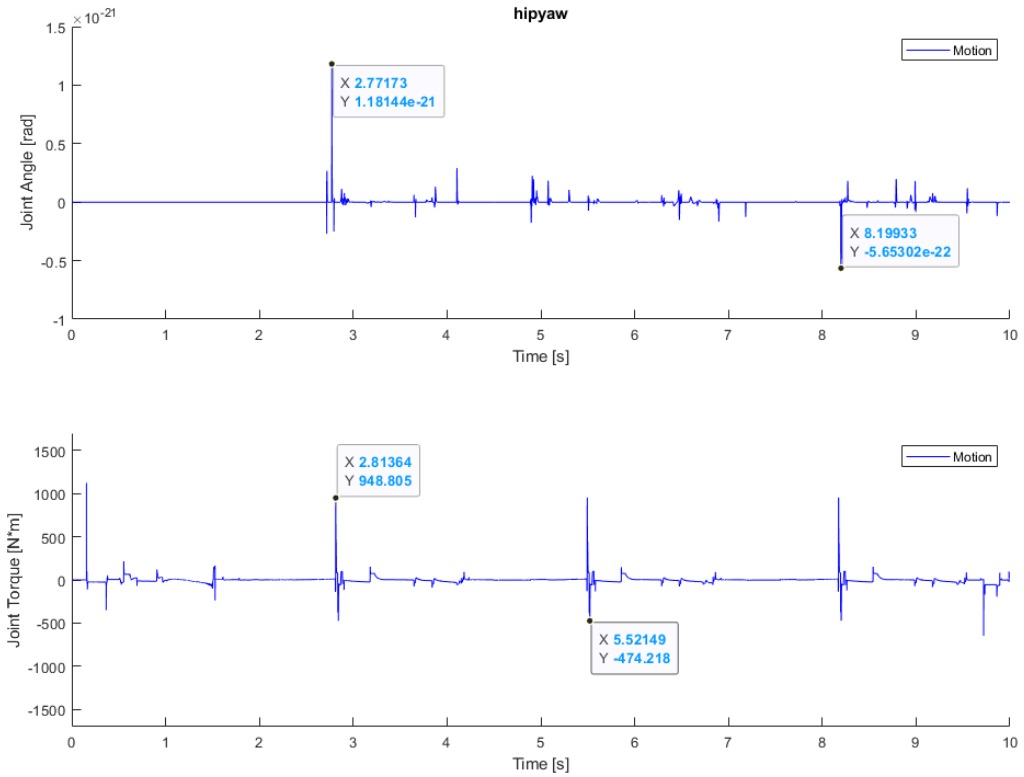


Figure 4.9: LIPM Hip Yaw

4.2 Optimisation Results

This section focuses on the optimisation technique employed to enhance the stability and performance of the PLLE. This section highlights the iterative nature of the optimisation process and the progressive improvements achieved throughout the iterations. By utilising a combination of a cost function and a genetic algorithm, the system parameters were refined to optimise the exoskeleton's walking gait. The subsequent analysis delves into the details of three iterations, showcasing the advancements made in terms of gait stability, torque distribution, and anthropomorphic characteristics. Additionally, a comparison will be drawn between the best optimised result and the reference gait achieved through the LIPM approach in the next section, providing insights into the strengths and weaknesses of both methods.

4.2.1 Best Optimisation Results

In Figure 4.10 a screen shot can be seen of the animation of the final optimisation technique walking pattern result. As was discussed in Chapter 3, to reach this final ‘best’ optimisation, the researcher started by using idealised motion. Idealised motion did not give a satisfactory result, as the robot used very high torque values in the range of 2000 Nm and sprinted across the screen. Although not useful as a final optimised gait, using this gait as initial conditions and

imposing torque limits on the joints by using the torque profile allowed the simulations to converge to stable walking patterns more quickly. The researcher reduced the torque limit as much as possible to see the threshold where the simulations would stop converging to a stable walking pattern. The torque limit ended with a stable simulation that was restricted to 700 Nm. Intermediate results can be found in Appendix C which further illustrates the progression of the walking gait.

The video displaying the simulation can also be found by following this link. The robot can be seen to walk along a flat surface runway. The direction of travel is in the x-direction in this case, the y-direction is to the side of the robot and the z-direction is the vertical position. As was mentioned in Chapter 3, it can be seen that the feet platforms needed to be enlarged in order for the optimisation algorithm to converge to a stable walking pattern.

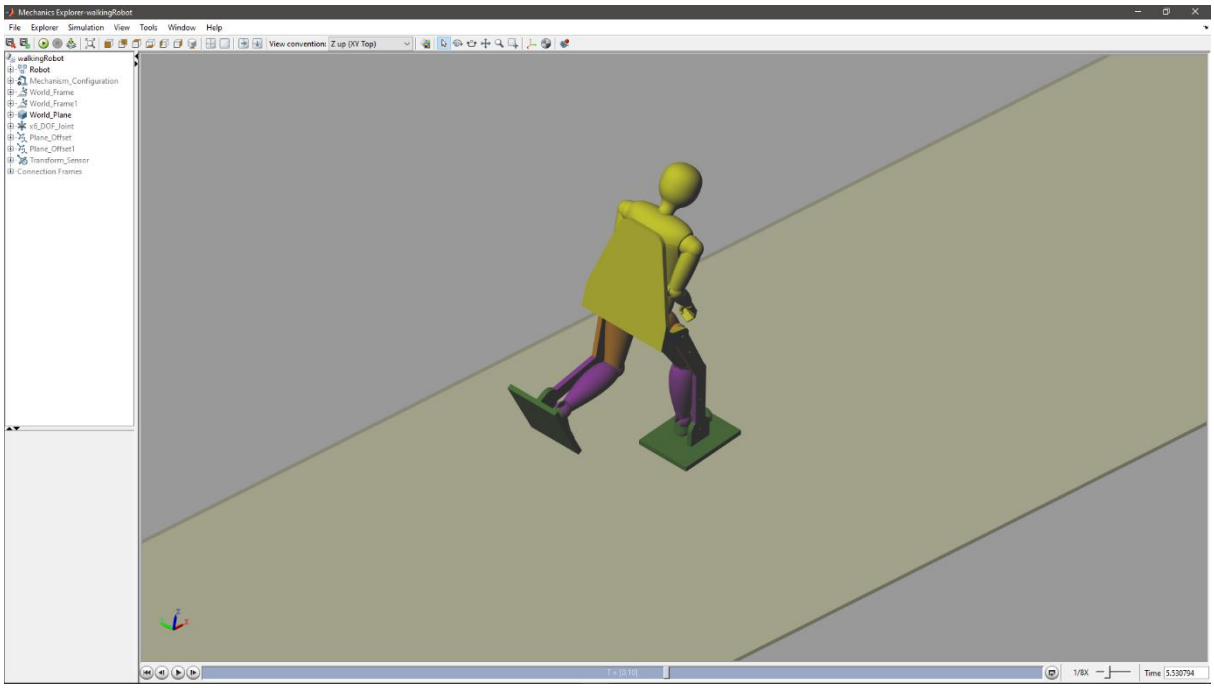


Figure 4.10: Optimisation Mechanics Explorer Animation

As was done for the LIPM simulation, the first step was to evaluate the human-likeness of the robot's gait. In Figure 4.11, an overlay of the different stages of the gait from the simulation onto the gait cycle that was previously shown can be seen.

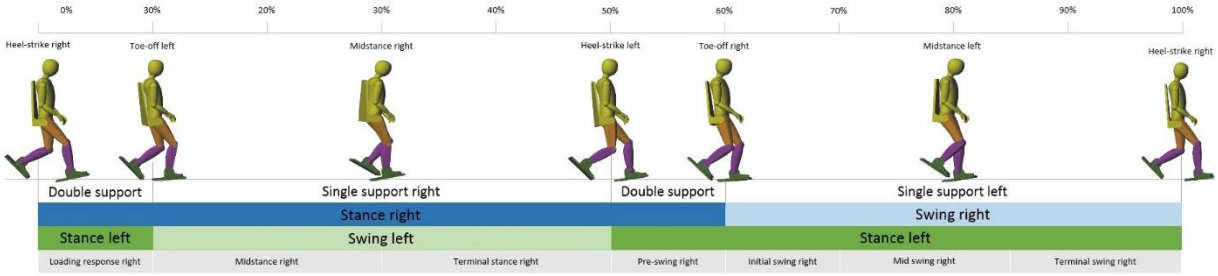


Figure 4.11: Optimisation Gait Cycle

(Adapted from “Gait_cycle.jpg (1860×412)”, n.d.)

Figure 4.11 shows an upright posture through the walking gait. The step length appears a lot larger than that of the LIPM simulation. Another advantage with this simulation in terms of human-likeness is the inclusion of the heel-strike and toe-off phases within the double support phases of the gait.

Although the gait appears more human-like than the LIPM simulation, the robot appears to be skipping rather than performing a smooth steady-paced walk. This may be due to the limited degree of freedom of the joints. In order to maintain stability, the time during single-support phases is kept to a minimum and the robot flicks from double-support to double-support in quick succession. It may also be due to how the combination of the cost function and genetic algorithm are set up, as they function to optimise for a greater distance covered while moving the joints smoothly and maintaining stability. The result does not show an adequate walking gait for a user in an assistive PLLE. Although the LIPM covered far less ground, the walking pattern it achieved appeared much more suitable for carrying a human safely.

The results in this section were achieved by using the torque profile for the simulation in which a torque limit can be set for the joint controller. The torque was set to be limited to a maximum of 700 Nm.

In Figure 4.12 there is a displacement and height graph. In the displacement graph it can be seen that the PLLE was able to walk 5.98 meters within the 10 seconds of simulation time. The line of the graph is more or less straight, indicating a steady progression forward. For the height graph, there are a series are peaks and troughs as the PLLE goes through the different gait stages that were depicted in Figure 4.11. These peaks and troughs will be further explained in the next section, with the help of Figure 4.13.

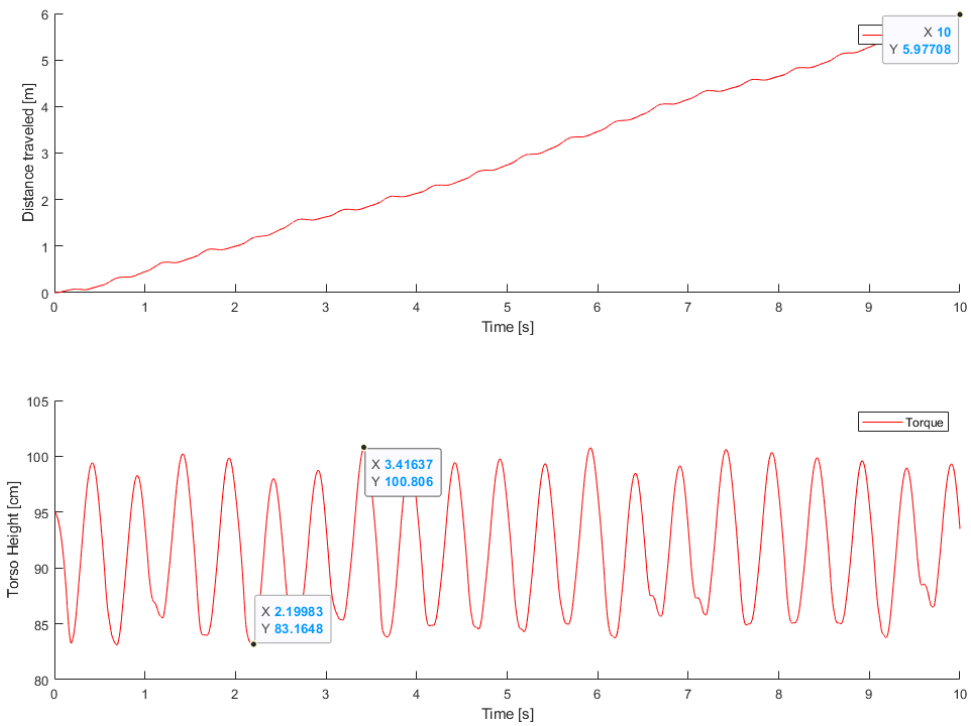


Figure 4.12: Optimised Global Data

Figure 4.13 shows an enlarged section of the Torso Height graph in Figure 4.12. It can be seen that the trough with the value of 83 cm shown to the right of the 2 second mark corresponds to the toe-off right stage. The next peak corresponds to the midstance left stage. The trough which follows corresponds to the heel-strike right phase, and finally the peak which follows corresponds to the toe-off left phase.

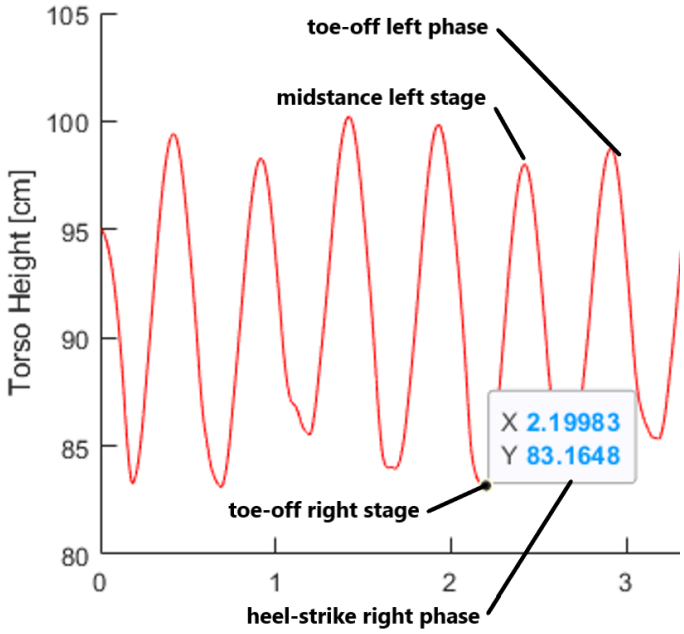


Figure 4.13: Enlarged section of Figure 4.12

Figure 4.14 shows the joint angles over time of the three degrees of freedom per leg of the walker; the ankle, knee and hip joints. It is perhaps the most intuitive to look at the Right Leg Hip Angle graph to get an idea of what the curves are representing. For each step, the leg extends forward, steps through and then returns at the hip. The next section will elaborate on this, with the help of Figure 4.15.

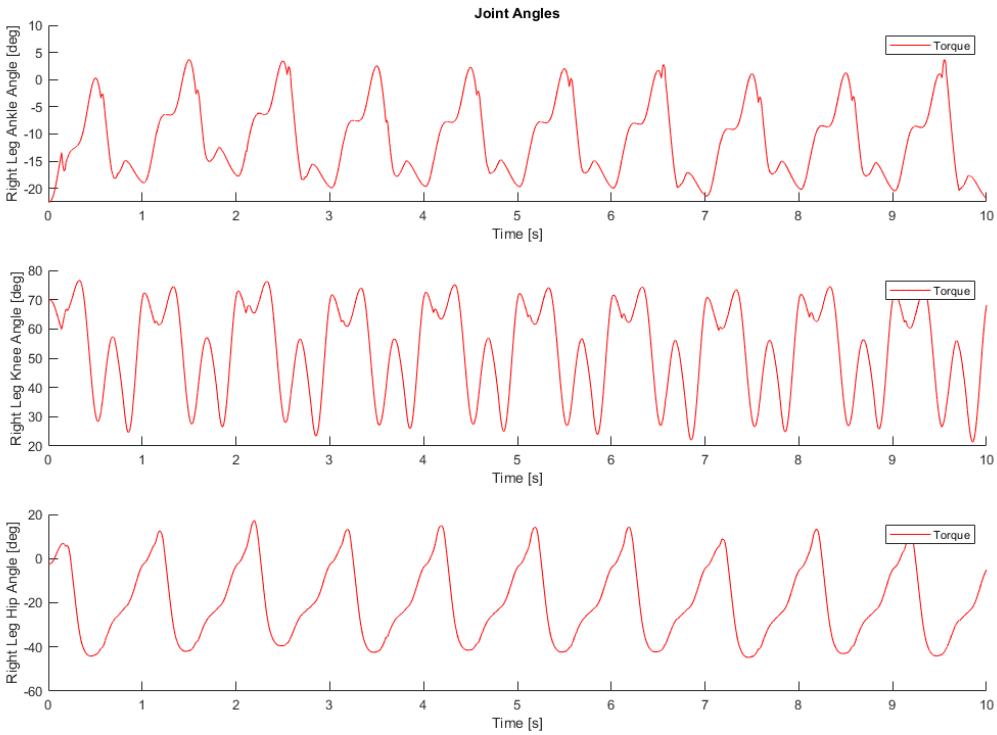


Figure 4.14: Optimised Joint Angles

Figure 4.15 shows an enlarged section of the Right Leg Hip Angle graph in Figure 4.14. It can be seen that the slope is steeper as it advances towards the troughs than it is when it advances towards the peaks, this indicates that the leg is extending forwards faster than it is retracting backwards while stepping through. The negative angle can be thought of as the leg rotating at the hip in an anti-clockwise direction. The gait period of this simulation was set to 1 second, this too can be seen in the hip angle graph which is extending and retracting in that timeframe.

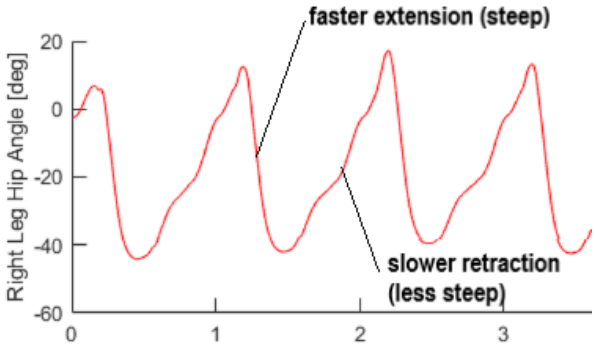


Figure 4.15: Enlarged section of Figure 4.14

Figure 4.16 shows the torque data for the joints over time. It can be seen that the upper limit of 700 Nm is reached on all three joints at different times during the gait cycle. Figure 4.17 will expand on points of interest on the graph.

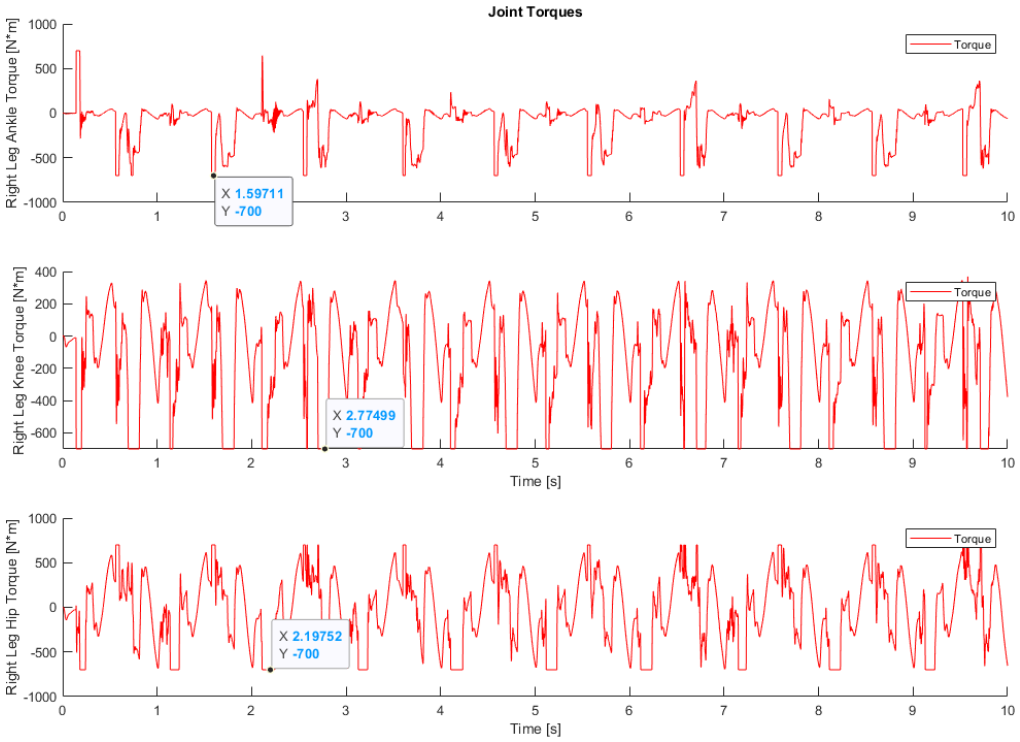


Figure 4.16: Optimised Joint Torques

Figure 4.17 contains an enlarged section of Figure 4.16. In the Ankle Torque graph, it can be seen that there are two distinct troughs in the second half of each period. The first trough with the flattened section where the torque limit is reached corresponds to the heel-strike right stage of the walking gait. The second trough which follows corresponds to the toe-off left stage of the walking gait.

In the Knee Torque graph, the troughs with the square bottom section that are reaching the torque limit also correspond to the toe-off left stage of the walking gait.

In the Hip Torque graph, the troughs with the square bottom section that are reaching the torque limit correspond to the toe-off right stage of the walking gait. This is where the right leg is not in contact with the ground and is therefore being supported entirely by the hip.

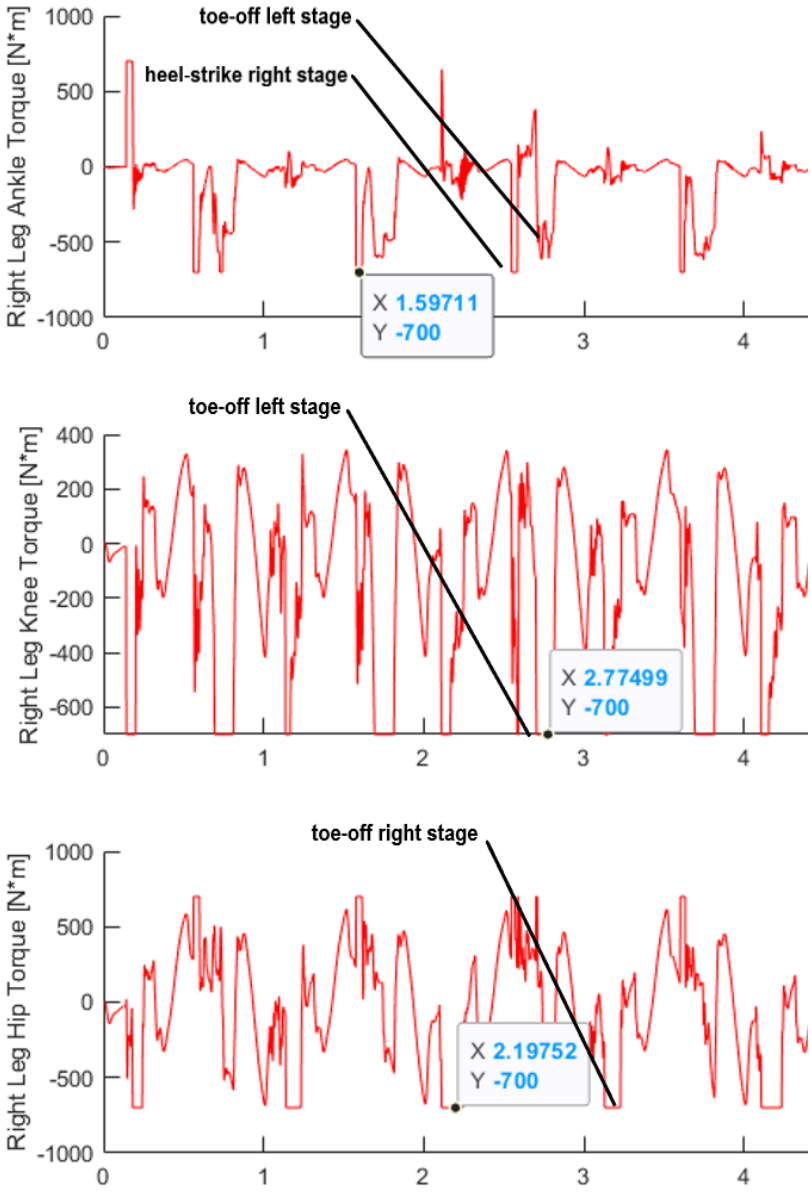


Figure 4.17: Enlarged section of Figure 4.16

4.3 Results Comparison

This section will show a direct comparison between the final results shown for the LIPM simulation and the final optimisation technique results.

In Table 4.1 a summary of the comparison of the results is presented. The distance walked in the Optimisation simulation as opposed to the LIPM simulation was approximately 6 meters to 1 meter. The torque values for the ankle, knee and hip pitch are all lower in the Optimisation simulation. Finally, it can also be seen that the Optimisation simulation walks with a higher torso height. The values in Table 4.1 were taken from Figures 4.15 to 4.19.

Table 4.1: Results Comparison

Criteria	LIPM	Optimisation
Distance walked in 10s (m)	1	6
Minimum Z torso height (m)	0.73	0.83
Ankle pitch torque maximum (Nm)	984	700
Knee pitch torque maximum (Nm)	1894	700
Hip pitch torque maximum (Nm)	1188	700

Figure 4.15 shows a simple displacement graph. Both the blue line representing the LIPM simulation, as well as the red line representing the Optimisation simulation are more or less straight, indicating that in both simulations the PLLE walked forwards without stumbling backwards. The Optimisation model was just able to progress a lot further in the given time.

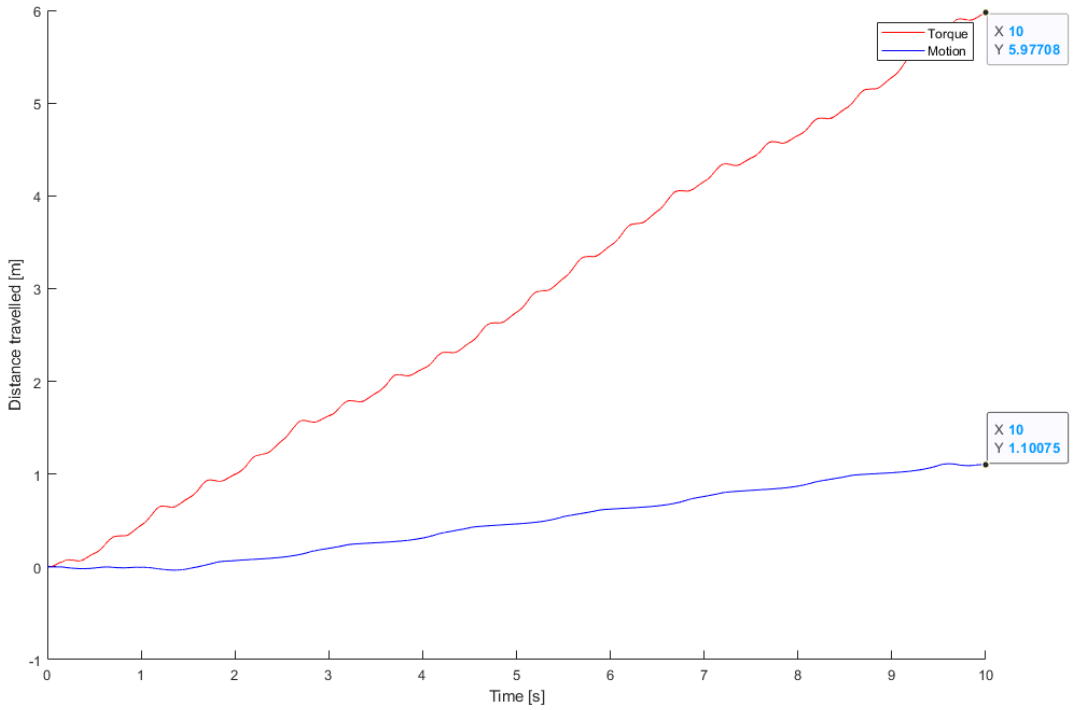


Figure 4.15: Distance Travelled Comparison

In Figure 4.16 the difference between the LIPM simulation and the Optimisation simulation in terms of torso height during a walking gait is made clear. Once in a walking position, the torso height of the LIPM simulation is virtually fixed, whereas in the Optimisation simulation, it varies as the gait progresses. The Optimisation simulation also walks with a higher torso height, even though it varies.

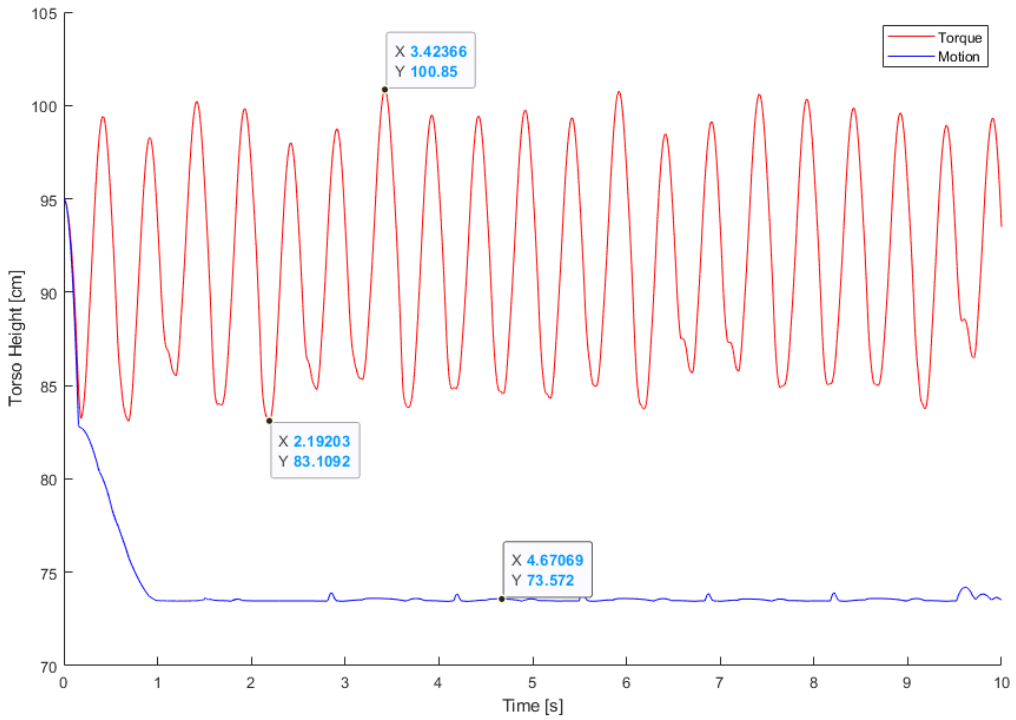


Figure 4.16: Torso Height Comparison

In Figure 4.17 it can be seen that the Optimisation simulation results had lower peak torque values for the ankle torque as opposed to the LIPM simulation. Flat troughs and peaks indicate areas where the Optimisation simulation reached the torque limit that was set at 700 Nm.

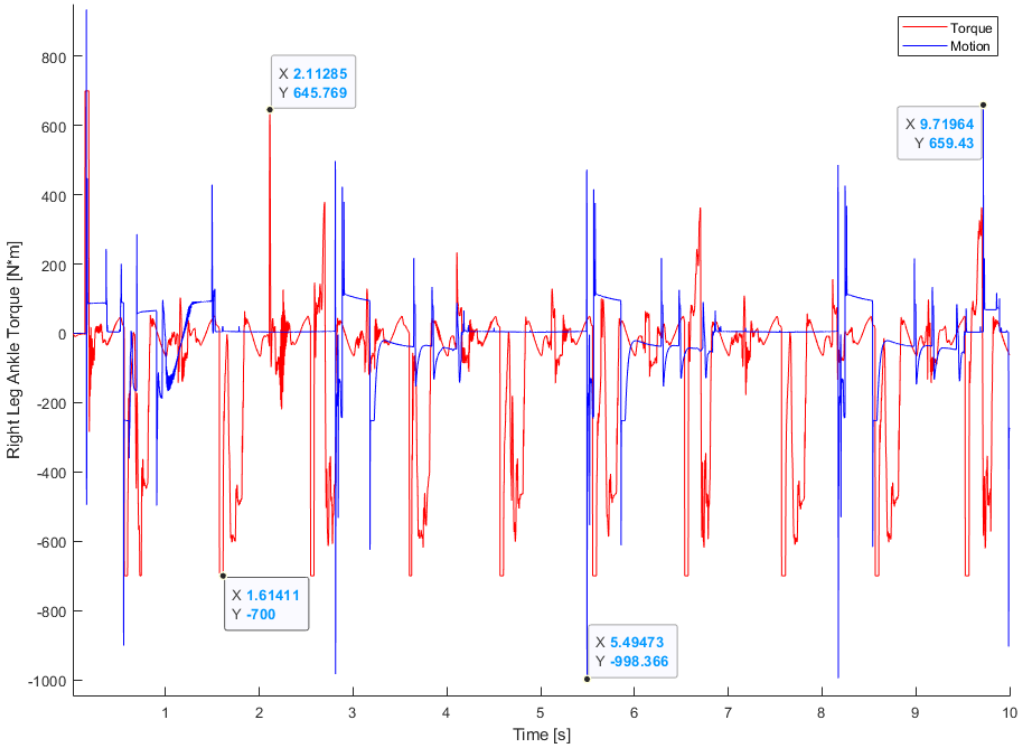


Figure 4.17: Ankle Torque Comparison

In Figure 4.18 it can be seen that the Optimisation simulation results had lower peak torque values for the knee torque as opposed to the LIPM simulation. Flat troughs and peaks indicate areas where the Optimisation simulation reached the torque limit that was set at 700 Nm.

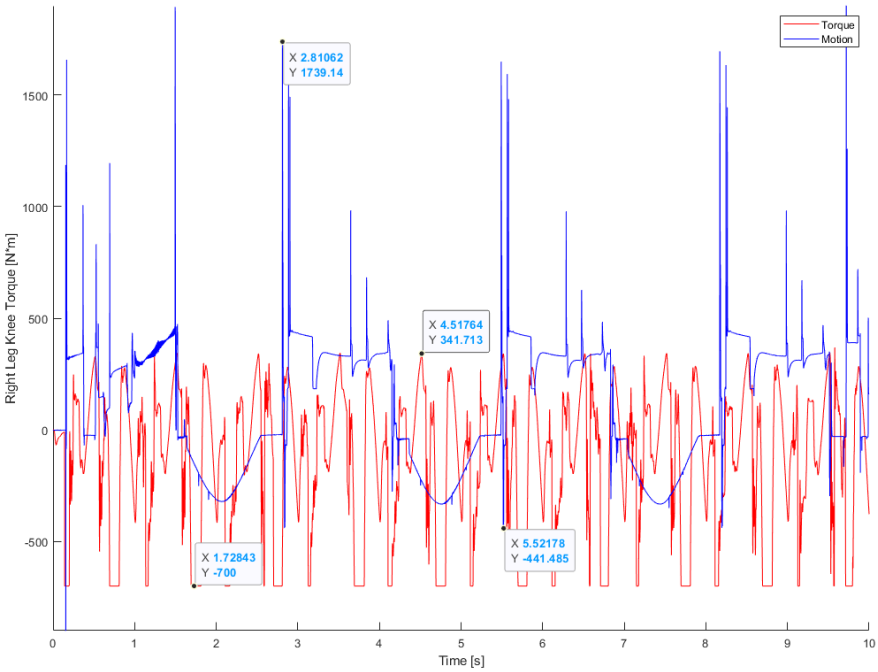


Figure 4.18: Knee Torque Comparison

In Figure 4.19 it can be seen that the Optimisation simulation results had lower peak torque values for the hip torque as opposed to the LIPM simulation. Flat troughs and peaks indicate areas where the Optimisation simulation reached the torque limit that was set at 700 Nm.

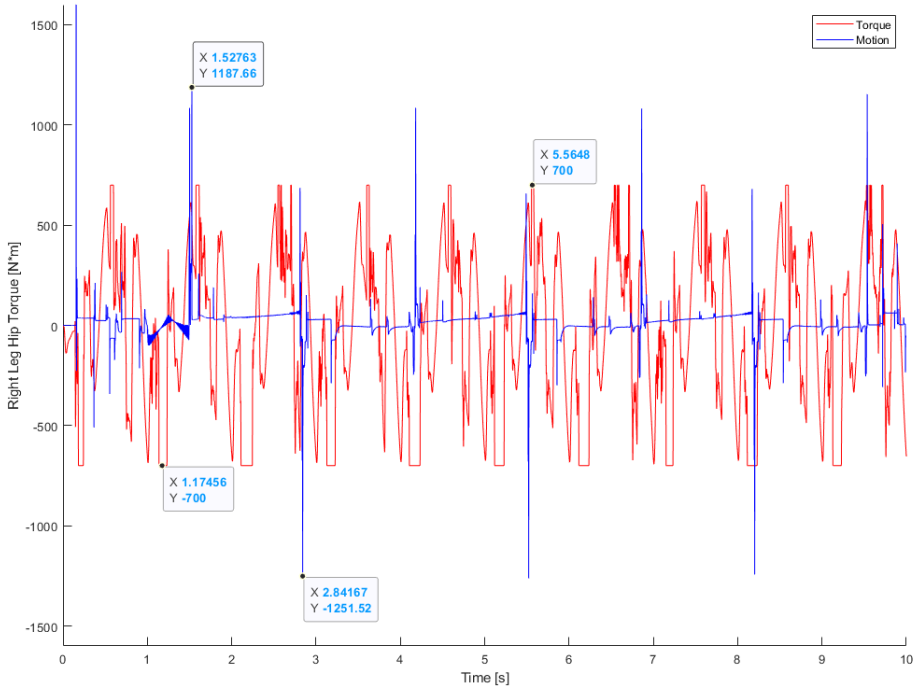


Figure 4.19: Hip Torque Comparison

Although the data from the graphs in Figures 4.15 to 4.19 appear to favour the Optimisation simulation in terms of having lower torque values and a further distance walked with a more upright posture, it was noted in Chapter 4.2 that the Optimisation result does not show an adequate walking gait for a user in an assistive PLLE. Although the LIPM covered far less ground with higher torque values, the walking pattern it achieved appeared much more suitable for carrying a human safely.

CHAPTER 5

ATTEMPTED PHYSICAL VALIDATION OF ASPECTS OF THE NUMERICAL MODELLING

This chapter covers the design, the experimental methodology, the testing, as well as the test results of an attempted physical validation of aspects of the numerical modelling.

5.1 Design Methodology

This section will deal with the design of the experiment. The purpose was twofold: to anticipate potential challenges when transitioning from simulation to physical hardware, and to explore the disparities between simulated and real-world results of the PLLE's locomotion. Furthermore, this endeavour aimed to delve into the complexities of translating the intricacies of a walking gait pattern into code that hardware could faithfully follow. In this chapter, the steps taken to acquire invaluable insights from this test platform will be outlined, breaking them down into easily comprehensible sub-sections. By doing so, it is hoped that light will be shed on the intricacies of bringing the PLLE to life and bridge the gap between theory and practice.

5.1.1 Design Considerations

At this stage of the research the goal was to identify the design envelope the researcher had to work in as this would point to the specifications that were necessary for the concept phase. Apart from availability of components, the design needed to be anthropomorphic as this is most suitable for use with humans. Although no human testing was to be carried out in this study, the aim of the study was to improve PLLE technology to help people. The design also needed to be representative, as far as possible, of the physical structure and degrees of freedom of the PLLE model used in the Optimisation section in Chapter 3, as the purpose of the experiment was to fulfil the same purpose of a scaled model and to test the final walking pattern. This meant having 3 DOF per leg, actuation at the ankle, knee, and hip joint on each leg. The design flowchart can be seen in Figure 5.1.

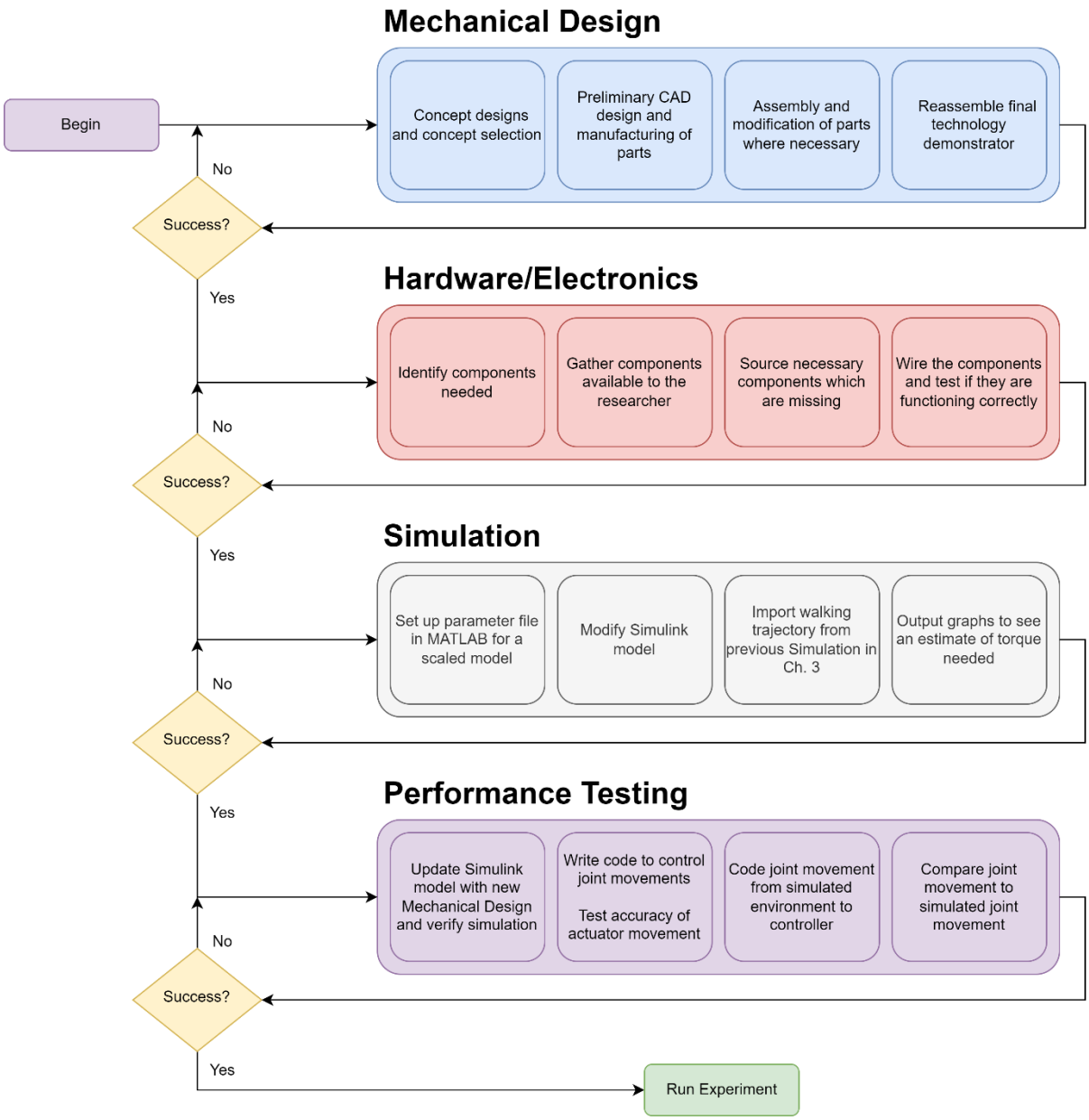


Figure 5.1: Experiment Flowchart

5.1.2 Requirements and Constraints

The purpose of this section is to outline the specific requirements and constraints for the robot developed as part of this research. The test platform aimed to closely represent a scaled model of the main simulation conducted in the study. However, unlike the full-scale simulation, the test platform did not need to carry a payload. Additionally, certain constraints forced the use of specific motors and motor controllers.

The following requirements were identified for the test platform:

5.1.2.1 Scale Representation

The test platform needed to accurately represent the scaled model of the main simulation. The dimensions, proportions, moments of inertia and kinematics of the LLE should be accurate, to allow for a realistic representation of its behaviour and dynamics. This will assist the researcher in comparing the simulation to the real-world result.

5.1.2.2 Motor Compatibility

Given the availability of existing motors and motor controllers, the design of the test platform needed to incorporate these components. The motors and motor controllers should be seamlessly integrated into the demonstrator to facilitate easy control and operation during testing and evaluation. Usually, motor components are selected from the specifications of the system design requirements, but in this case due to constraints, the design was scaled/synthesized around available motors.

The following constraints were considered during the development of the test platform:

5.1.2.3 Payload Exclusion

Unlike the full-scale simulation, the test platform was not required to carry a payload. This assumption allowed for a simplified design and reduced the complexity associated with accounting for additional weight and balance considerations.

5.1.2.4 Motor and Controller Utilisation

Given the researcher's existing motors and motor controllers, the design of the test platform needed to accommodate and utilise these components. This constraint ensured cost-effectiveness and leveraged the availability of suitable hardware.

5.1.2.5 Summary

This section highlighted the requirements and constraints that guided the development of the test platform for this research. The test platform aimed to closely represent a scaled model of the main simulation, while excluding the payload requirement. It also incorporated existing motors and motor controllers to maximize resource utilisation. These requirements and constraints were fundamental in ensuring the successful development of the test platform and aligning it with the research objectives.

5.1.3 Mechanical Design

A mechanical design was needed to manufacture the test platform. The process for obtaining this design is outlined below.

Concept designs were done before a preliminary CAD design was done. The two concepts are explained below.

5.1.3.1 Concept 1

The first concept which was done tried to keep the design as simple as possible. The idea was to connect the actuators using 3D printed linkages. As there is a lot of flexibility in the shape and type of mounting that can be done with 3D printing, it would make the design easier to accomplish. The actuators were also placed at their respective joints they were actuating, so there would be no need to transfer power from another mounting location. The output shafts of the gearboxes would be used as the joints so there would also be no need for shafts and bearings for the linkages to rotate about.

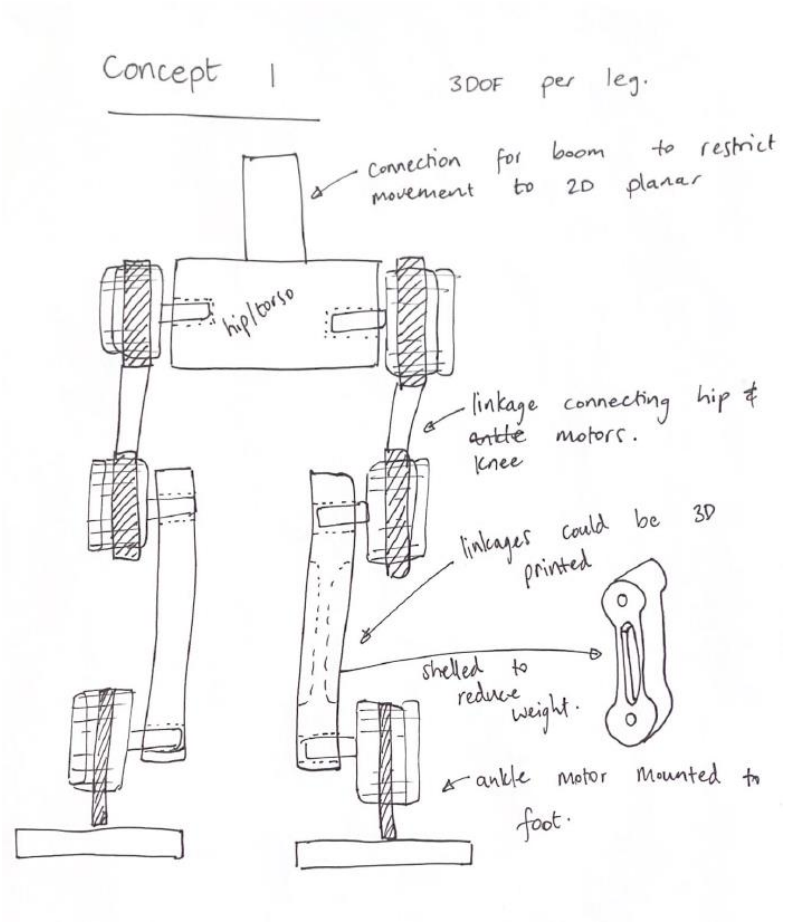


Figure 5.2: Concept 1

5.1.3.2 Concept 2

The second concept involved moving the CoG upwards by mounting both the knee and hip actuators at the top of the structure and then transferring torque to the knee joints via pulleys and a belt drive. As the CoG of the model in the main simulation is high up, this model may represent a scaled model more accurately than Concept 1. For the linkages connecting the motors, it was thought that they could be made of 3D printed parts as in Concept 1, or out of carbon fibre rods with 3D printed connections at their ends.

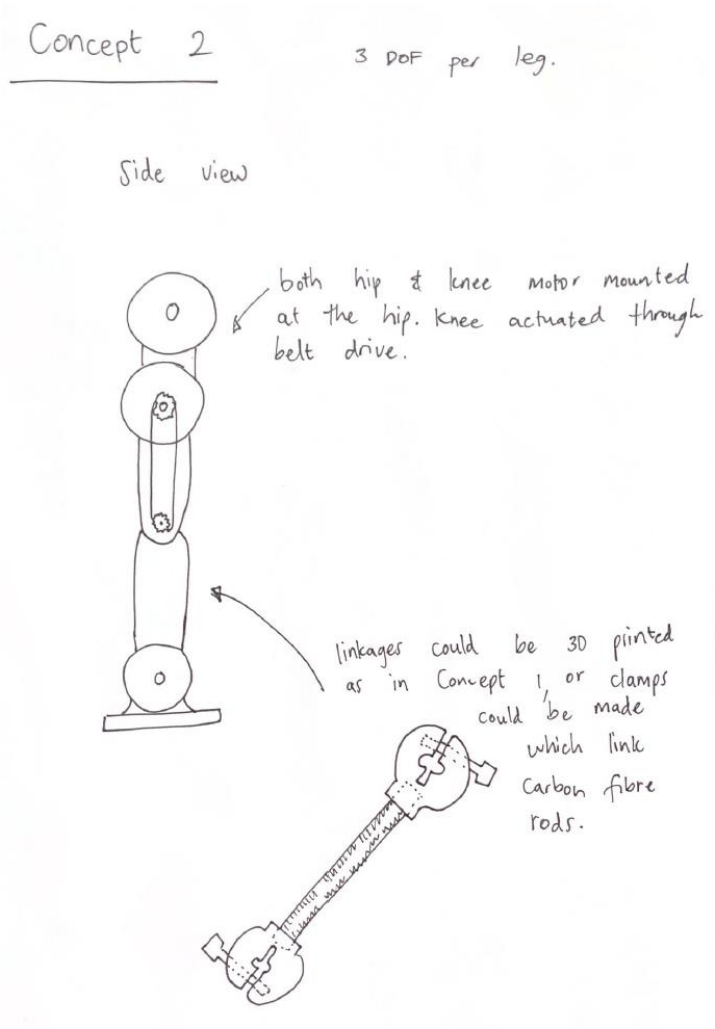


Figure 5.3: Concept 2

A decision matrix was done to decide on which concept to choose based on the following criterion: availability of parts, cost, ease of assembly, robustness, and the accuracy of the scaled model representation. A score was assigned to each concept design for each criterion on a scale of 1 to 3, with 3 being the highest. The decision matrix can be seen in Table 5.1.

Table 5.1: Decision Matrix

Criteria	Concept Design 1	Concept Design 2
Availability of Parts	3	2
Cost	3	2
Ease of Assembly	3	3
Robustness	3	2
Model accuracy	2	3
Total	**14**	**12**

Equal weights for each criterion were assumed since their relative importance did not vary greatly.

Based on the total scores, Concept Design 1 has a higher score of 14 compared to Concept Design 2 with a score of 12. Therefore, considering the given criteria and scores, Concept Design 1 was the preferred choice.

5.1.3.3 Computer Aided Design (CAD) Modelling

Having selected Concept Design 1 as the preferred option to produce, a CAD model was drafted. As stated in the design, the design was centred around being as simple as possible. The researcher started by downloading a STEP file of the motor and gearbox from the Maxon website. The STEP file was imported, and a mounting method was designed for the actuator. Before the linkages were designed, the mounting method was 3D printed and the fitment was tested. This can be seen in Figure 5.4.

Figure 5.4: Actuator Mounting Test

Once the fitment was satisfactory, the mounting method was essentially modified into a foot, lower leg and upper leg. The fitment of the linkages on the output shafts of the actuators was also tested beforehand. The linkages connecting the joints were designed as simple rectangles that have been shelled to reduce weight.

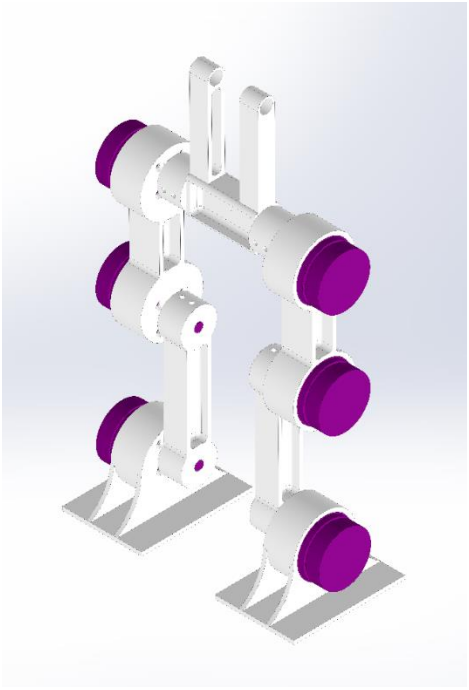


Figure 5.5: Test Platform CAD Model

After the design was complete the parts were 3D printed from white PLA with an infill of 70%. The parts before assembly can be seen in Figure 5.6.

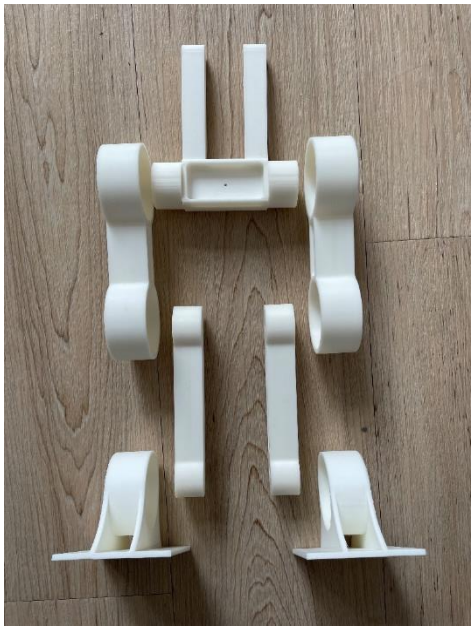


Figure 5.6: 3D Printed Parts

5.1.4 Hardware and Electronics

This section will outline the main hardware components needed to design the test platform and how the selection of these components was decided, as well as what hardware the researcher selected based on availability and constraints.

5.1.4.1 Actuators

Motors were needed to actuate the joints of the test platform. The motors were one of the components mentioned as a constraint. The researcher had 6 Maxon 310194 combination drives available to them. The combination drive coupled a BLDC motor with hall effect sensors for position control with a 26:1 spur gear output. The combined weight of the motor and gearbox came to 308 grams. The data sheet (Appendix C) states a peak torque output of 263 mNm which when multiplied by 26 due to the gear ratio, resulted in the value of 6.838 Nm of peak torque.

The concept designs in the mechanical design section of this chapter needed to consider the use of these actuators.

Figure 5.7: Maxon Actuator

5.1.4.2 Motor Controller Board

A motor controller was needed to control the actuators on the test platform. In this case the motor controller was limited to one which the researcher had access to using at their place of employment. Six Nanotec N5-1-2 CANopen 10A RMS were used, one to power each actuator. The motor controller was capable of running the selected BLDC motor and was able to accept inputs from a computer so that a microcontroller was not necessary. To control all six motors at once, the motor controllers can be daisy chained together while still only requiring one input to control them using CANopen commands. Further information can be found in the datasheet in Appendix C.

Figure 5.8: Nanotec N5-1-2 Controller

5.1.4.3 Power Supply

A power supply was necessary to power the motor controllers. The power supply needed to meet the voltage and current supply needs specified on the Nanotec controller boards. As the Nanotec controllers could accept a wide range of DC voltage inputs, from 12 VDC to 72VDC, choosing a power supply was made easier. A 19VDC 6.32A power supply was used for each motor controller.

Figure 5.9: Power Supply

5.1.4.4 Hardware and Electronics Verification

To test that the hardware was working as it should, a preliminary test was done with two motor controllers and two actuators with no load attached. The first motor controller was assigned to be node 1 and the second motor controller was assigned to be node 2. A short test program was written to command node 1 to move from 0 degrees to 360 degrees, then return to 0 degrees. As node 1 returns to 0 degrees, node 2 should repeat the same movement. This behaviour was then set to loop to see if the actuators could hold their positional accuracy.

In a 3-phase motor with 3 Hall sensors, the electrical commutation sequence typically follows a pattern called a "6-step" or "trapezoidal" commutation. In this commutation scheme used by

the Nanotec controllers, the motor's magnetic field is generated by sequentially energizing two of the three motor phases while the third phase remains off.

With 3 Hall sensors, the motor control system can determine the rotor position within one electrical revolution. This means that there are six electrical sectors or positions within each electrical revolution. Each sector represents a unique combination of the energized motor phases and the state of the Hall sensors.

Therefore, in a 3-phase motor with 3 Hall sensors, there are typically 6 electrical positions within one electrical revolution. From the data sheet for the actuators, it was seen that the Maxon motor has 8 pole pairs. By multiplying the 6 electrical positions with the 8 pole pairs, there are 48 electrical positions. Multiplying the 48 positions by the gear ratio of 26 gives 1248 positions per 360 degree rotation. The test setup can be seen in Figure 5.10 with tape on the output shafts to more easily observe their movement.

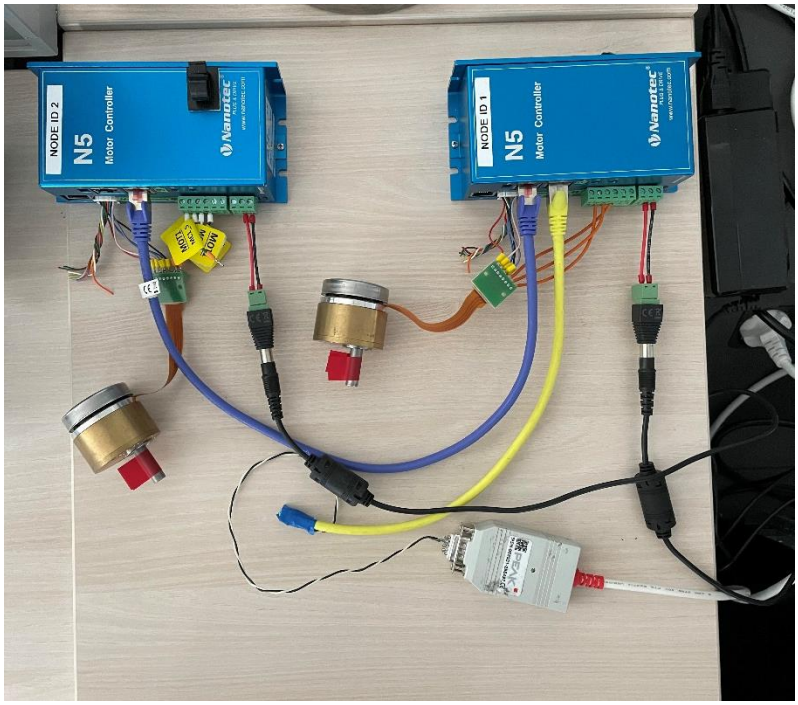


Figure 5.10: Actuator Testing

After verifying that the wiring of the motors was functioning correctly, wiring harnesses could be made for each motor. A harness of 1.5 meters was prepared and connected to each motor controller. An example of one of the harnesses can be seen in Figure 5.11.

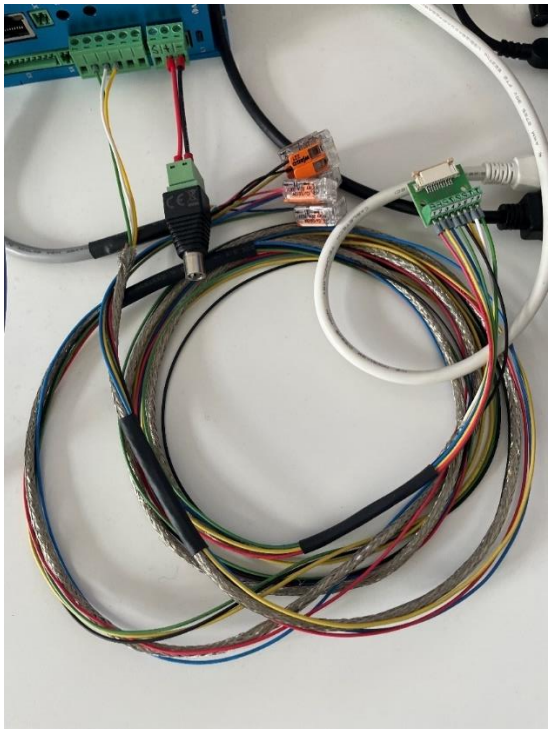


Figure 5.11: Harness

5.1.5 Simulation

To gauge whether the actuators the researcher had access to would have sufficient torque to actuate the joints, an initial simulation was done using the default parameterised model (Figure 5.2) provided in the Walking Robot MATLAB software.

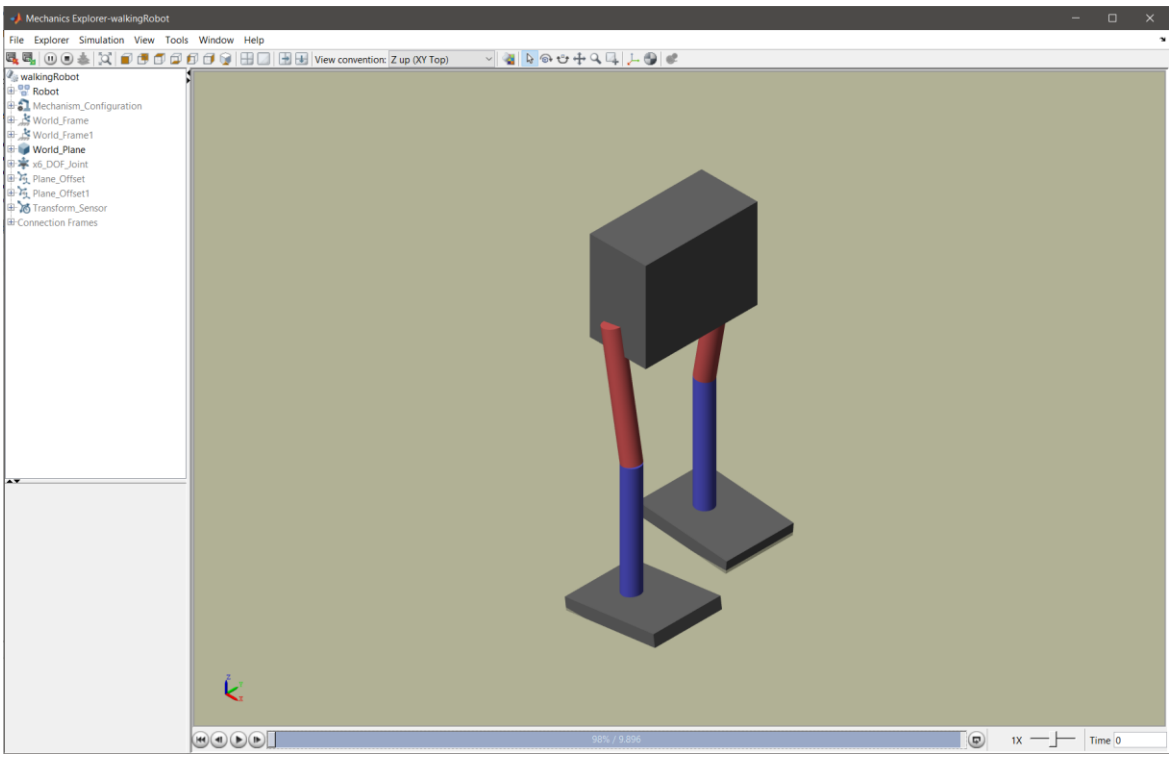


Figure 5.12: Test Platform Parameterised Model

The following parameters were modified in the robotParameters.m file (see Appendix B, Code Snippet 10). The maximum torque value was taken from the data sheet of the motor. The peak torque of 263 mNm was multiplied by 26 due to the gear ratio, which resulted in the value of 6.838 Nm of peak torque.

To modify the simulation to represent the test platform more accurately, the Simulink model was modified. As can be seen in Figure 5.13, the Foot, Lower Leg and Upper Leg Solid blocks' masses were altered. This was done for the left and right leg in the model. Each solid block was set to be 308 grams, this mass came from the weight of the motor and gearbox combination. The weight of the linkages connecting the joints was ignored at this stage, to get an estimate of torque values. The system would later be simulated again with the linkage weights to verify if the system would still be able to achieve the specified torque values.

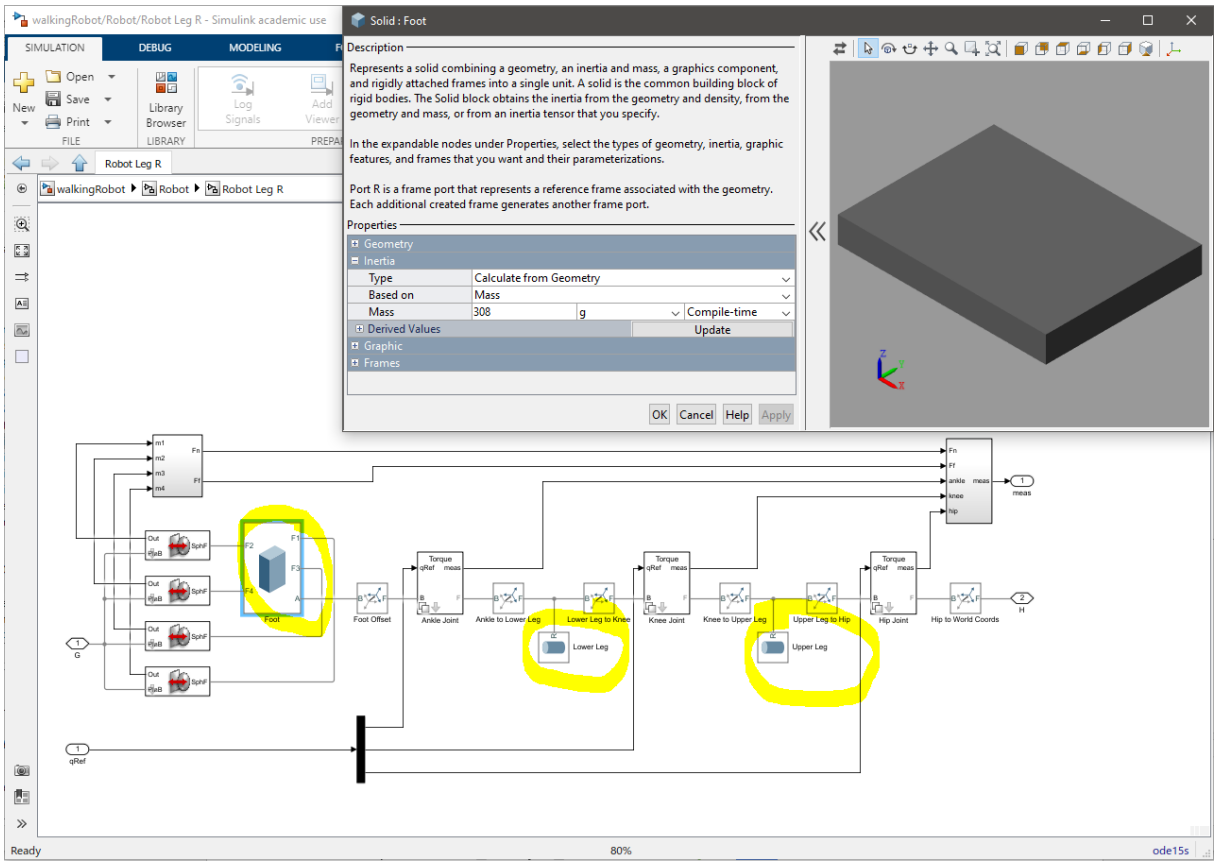


Figure 5.13: Test Platform Simulink Right Leg

After running the script which compares the actuator types, it can be seen that the model is able to follow the walking gait successfully under the torque limit. The output of the joint torques recorded during the simulation can be seen in Figure 5.14.

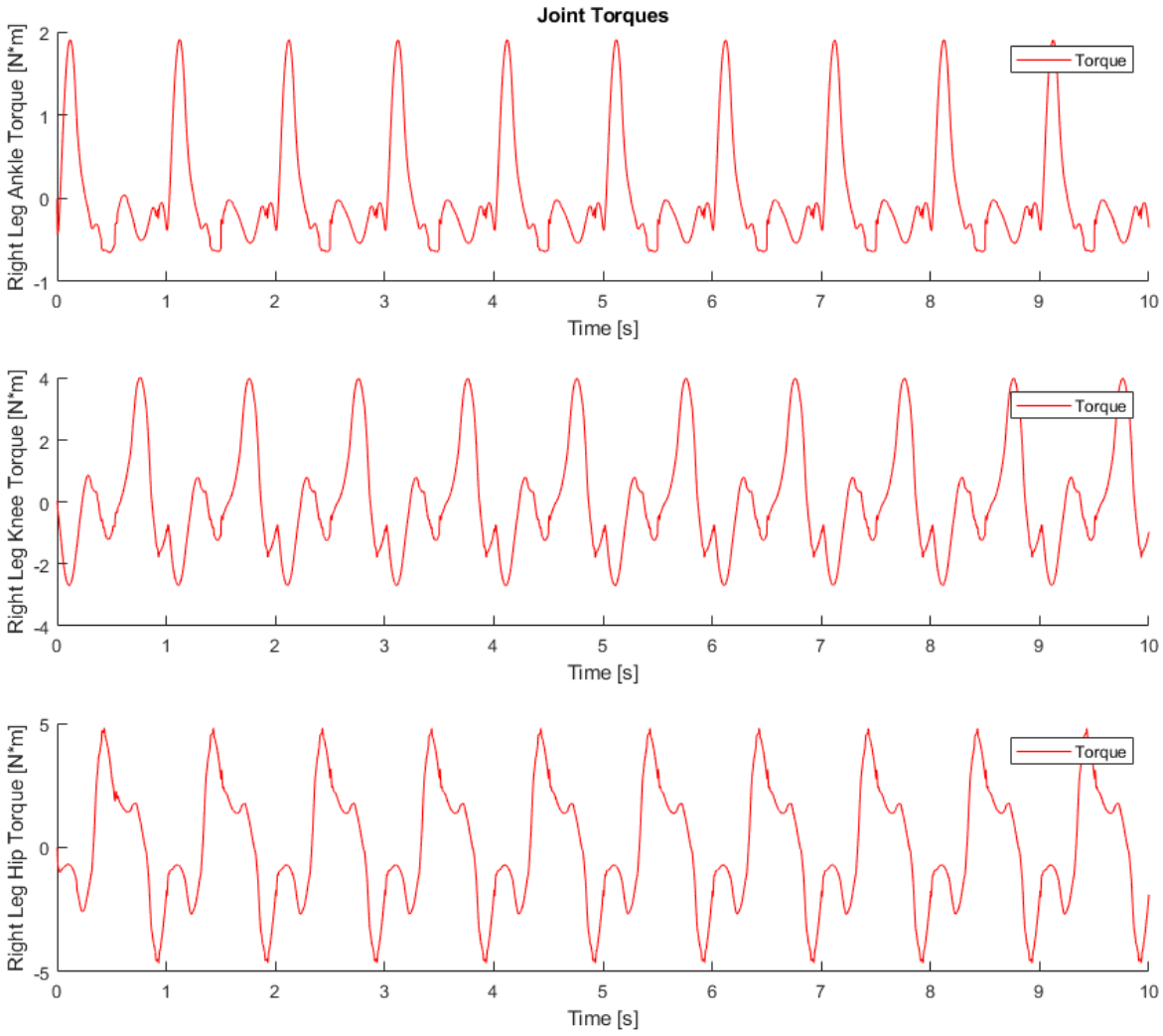


Figure 5.14: Preliminary Simulated Joint Torques

5.1.6 Performance Testing

This chapter documents the testing of the movement of the platform.

5.1.6.1 Model Verification in Simulink

With the mechanical design of the test platform completed, the CAD components could be imported into the Simulink model using the method described in Chapter 3. This step was necessary to verify if the updated geometry could still manage to follow the walking trajectory without falling in the simulated environment. If the robot falls, it would then be necessary to either modify the physical geometry or run the optimisation with the walking gait set as the initial conditions. The 3D Simulink model can be seen in Figure 5.15.

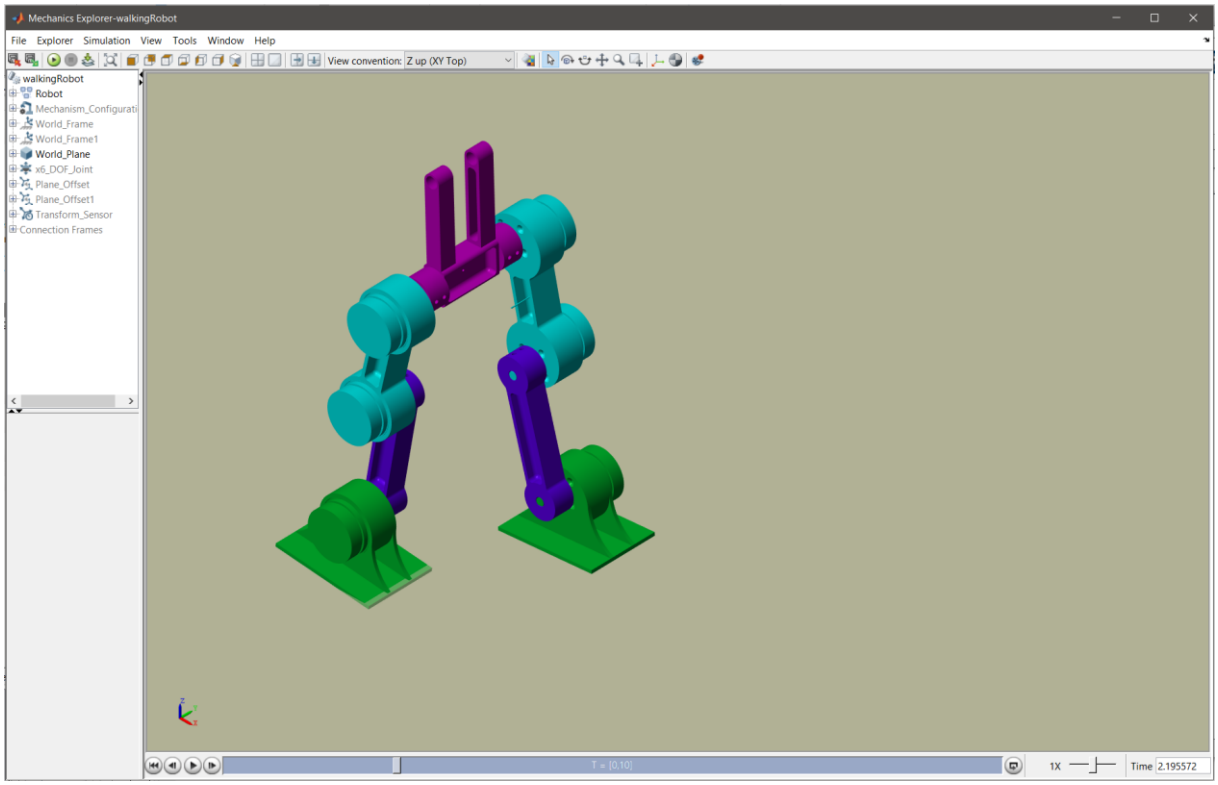


Figure 5.15: 3D Simulink Model

The test platform was able to walk for the full 10 seconds of simulation time using the best gait pattern from section 4.2.1 and the torque constraint of the actuators. The gait pattern of the walk can be seen in Figure 5.16. In Figure 5.17 the joint torques can be seen, which output a similar result to the preliminary simulation (Figure 5.14) when the linkage masses were not yet included as they were with this simulation.

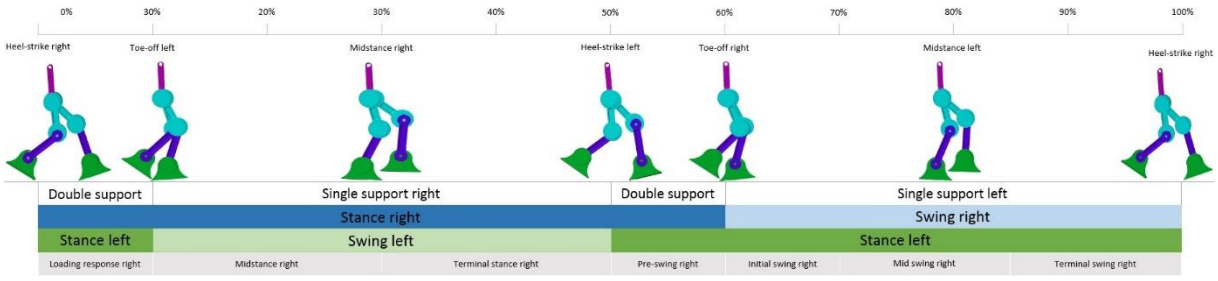


Figure 5.16: Test Platform Gait Cycle

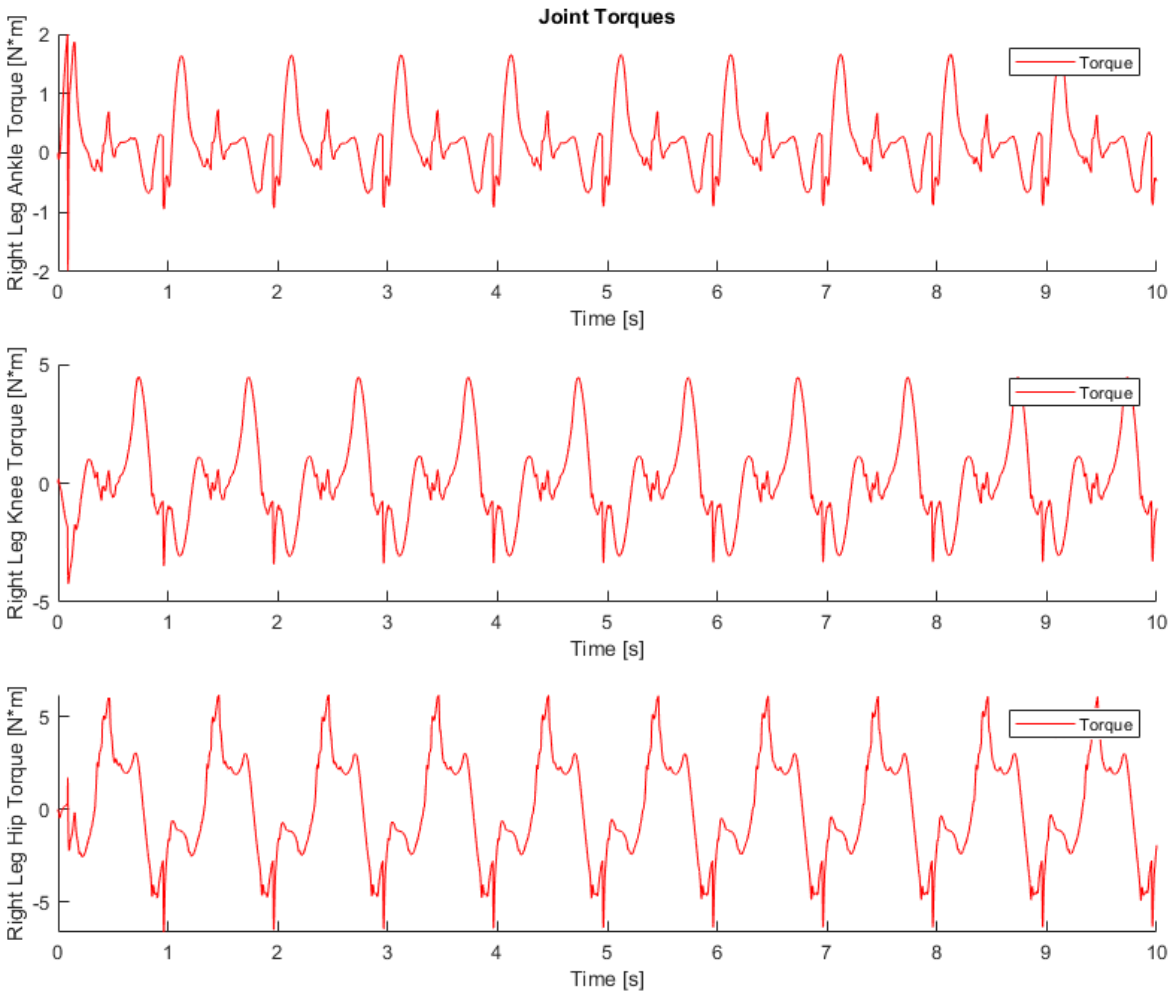


Figure 5.17: Test Platform Simulated Joint Torques

Similar to the test program written in the Hardware and Electronics section of this chapter, test programs were written to test each joints movement separately. This was done to ensure all joints were moving as expected and in the correct direction. An example of one of the programs was to command the left ankle to move 30 degrees and back to its initial 0 position in a loop.

The test platform was hung off a pole to suspend it in the air while testing took place to allow for unobstructed movement of the joints. The setup can be seen in Figure 5.18.

Figure 5.18: Hanging Test Setup

5.1.6.2 Joint Accuracy Test

A setup to test the accuracy of the joint movement and how faithfully it could match the movement of a joint in the simulated environment was also conducted. The test consisted of a simple hip movement from 0 to 45 degrees which repeats over a 2 second period. The hip of the robot as fixed in space. In the figures below the comparison of the simulated results and the real-world results can be seen. 3 sets of 2 second periods are shown. It was observed that in the real world result the leg could achieve the motion of going from 45 degrees to zero within the required 1 second (half a period) but lagged when going from 0 to 45 degrees. This can be seen in seconds 3 to 4 and 6 to 7. In general, the leg had either over-shoot or under-shoot when stopping at the required position and would subsequently correct itself. This would also cause the real-world result to go out of sync with the simulation.

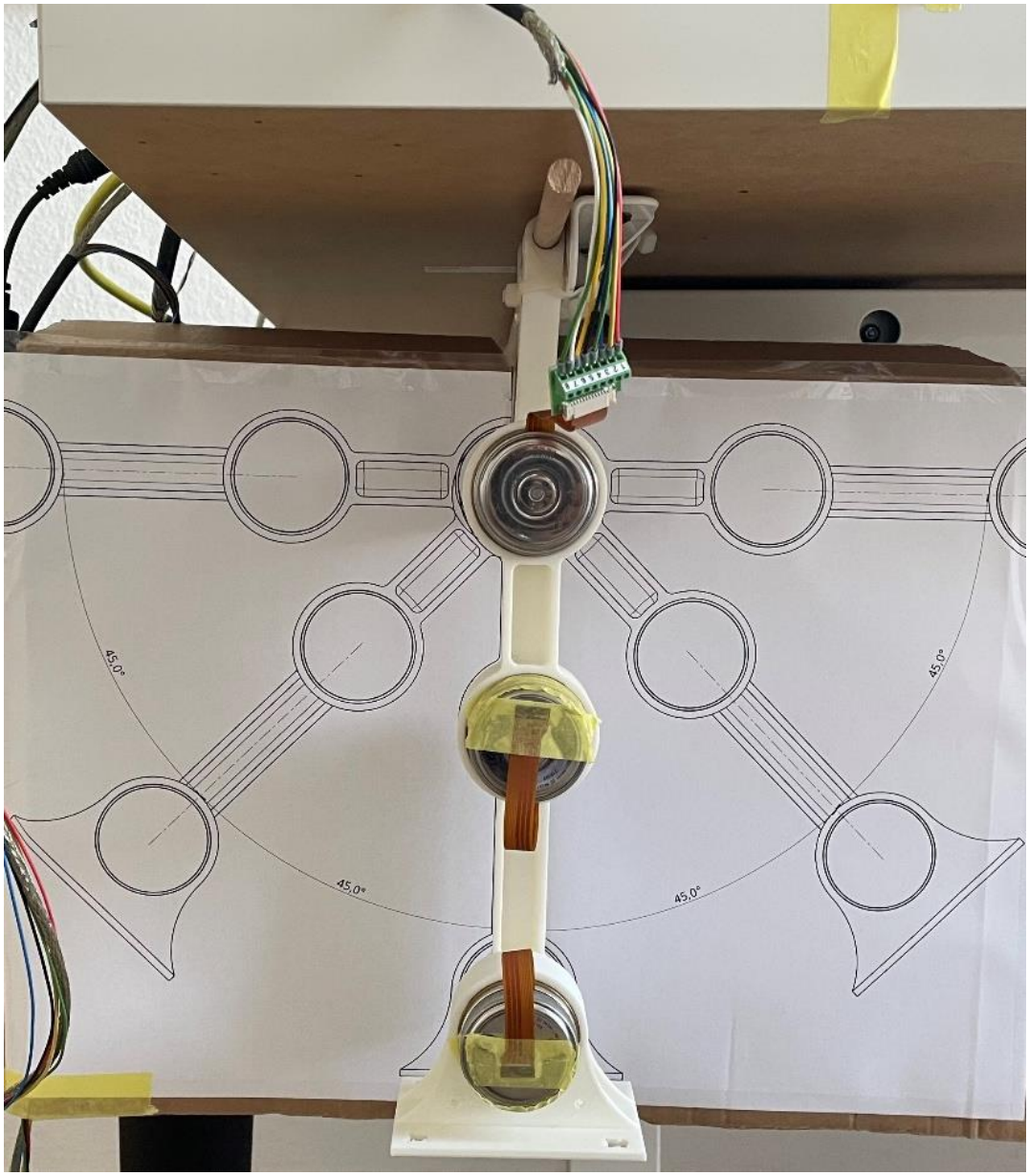


Figure 5.19: Hip Motor Test Setup

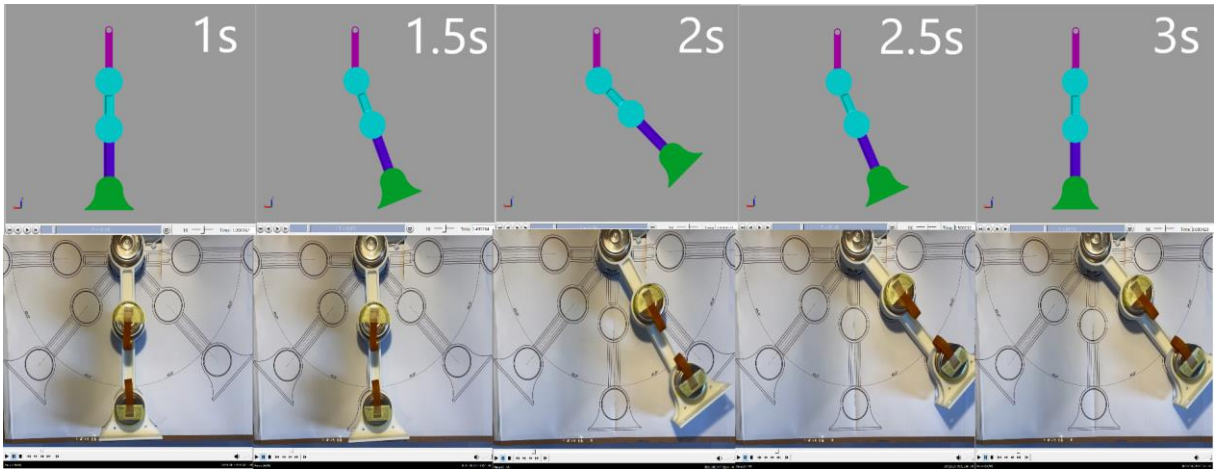


Figure 5.20: Comparison 1s to 3s

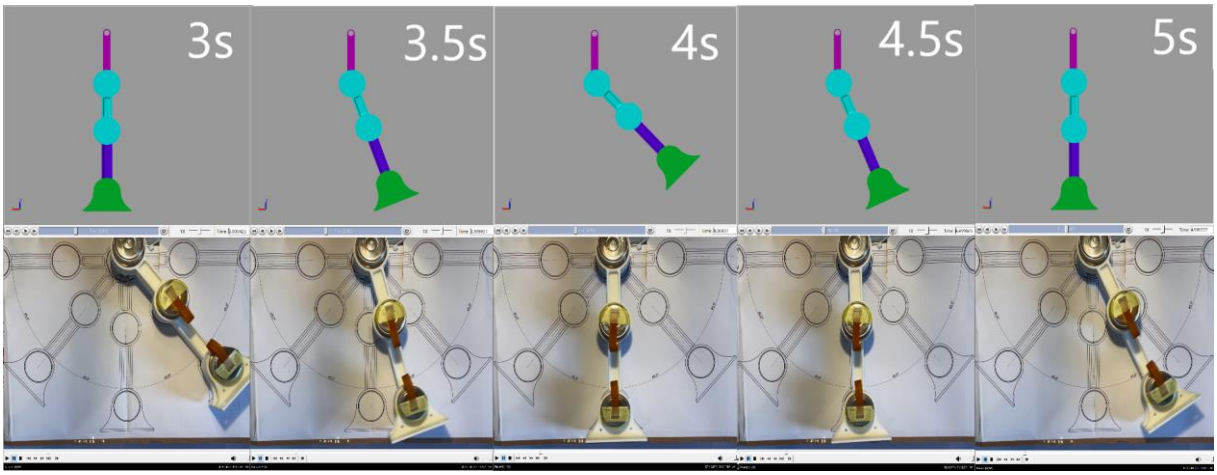


Figure 5.21: Comparison 3s to 5s

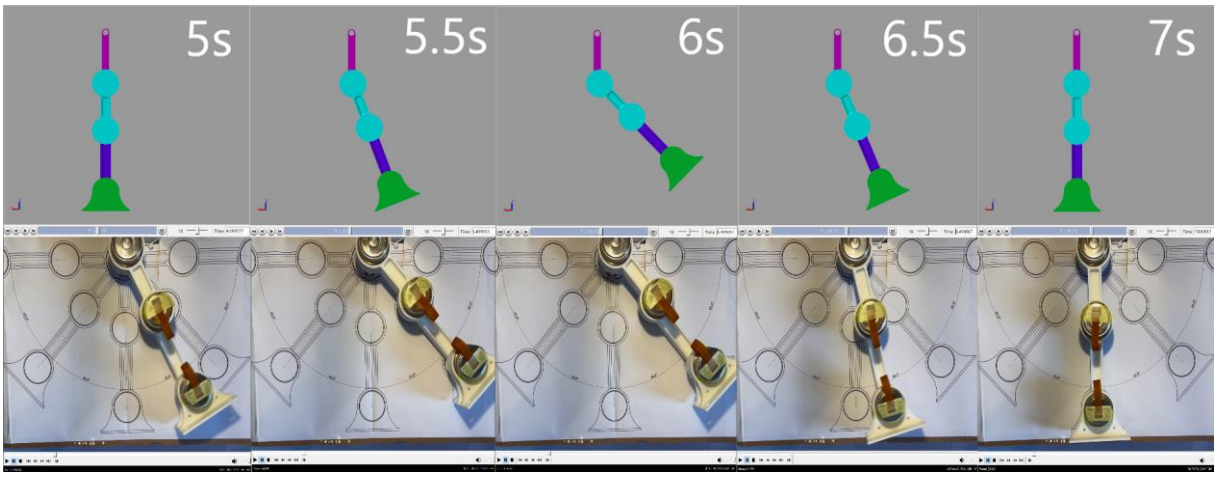


Figure 5.22: Comparison 5s to 7s

5.1.6.3 Suspended Movement Test

Having tested the joints individually, the walking algorithm could be coded and then tested while the robot was still suspended. As can be seen in Figure 5.23, the program to run the motion profile of multiple walking patterns were done and the profiles which were not in use were commented out.

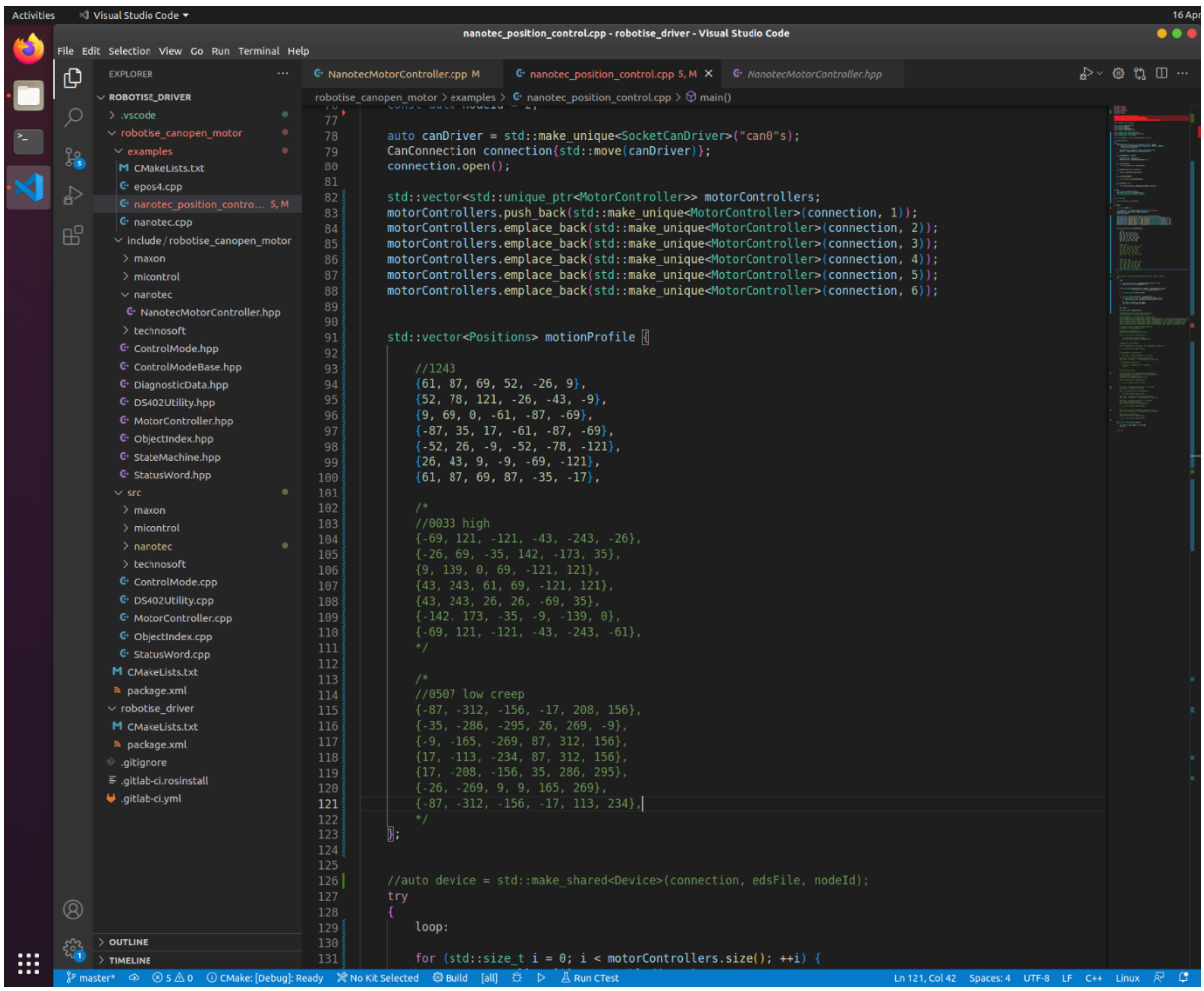


Figure 5.23: Motion Profile Program

To code a motion profile, the 6 waypoints for each joint are defined. Each array contains six elements, a position for each joint. The first element in the array represents the first joint, the second element the second joint and so on. In this case, joints 1, 2 and 3 are the ankle, knee and hip of the right leg. Joints 4, 5 and 6 are the ankle, knee and hip of the left leg. The left leg lags the right leg by half a period, so if the first 3 elements are seen as the base walking pattern, the second 3 elements begin halfway through the first 3. This can be seen in the section of code below. Looking at the first 3 elements which are in blue, the last 3 elements in blue start halfway through. The section of code is set to loop so that the pattern repeats.

{61, 87, 69, 52, -26, 9},
{52, 78, 121, -26, -43, -9},
{9, 69, 0, -61, -87, -69},
{-87, 35, 17, -61, -87, -69},
{-52, 26, -9, -52, -78, -121},
{26, 43, 9, -9, -69, -121},
{61, 87, 69, 87, -35, -17},

After these initial tests, the researcher moved on to conducting a ground walking experiment.

5.2 Experimental Methodology

This chapter presents the experimental methodology employed to validate the performance and functionality of the test platform. The purpose of this section is to provide a detailed description of the experimental setup, data acquisition procedures, and analysis methods used in the evaluation of the test platform. The methodology aims to ensure reliable and repeatable results, allowing for meaningful conclusions to be drawn regarding the performance and feasibility of the developed system.

5.2.1 Experimental Setup

The experimental setup was designed to closely resemble the simulation environment. Every effort was made to replicate the conditions and parameters used in the simulation, ensuring a consistent and comparable evaluation of the test platform.

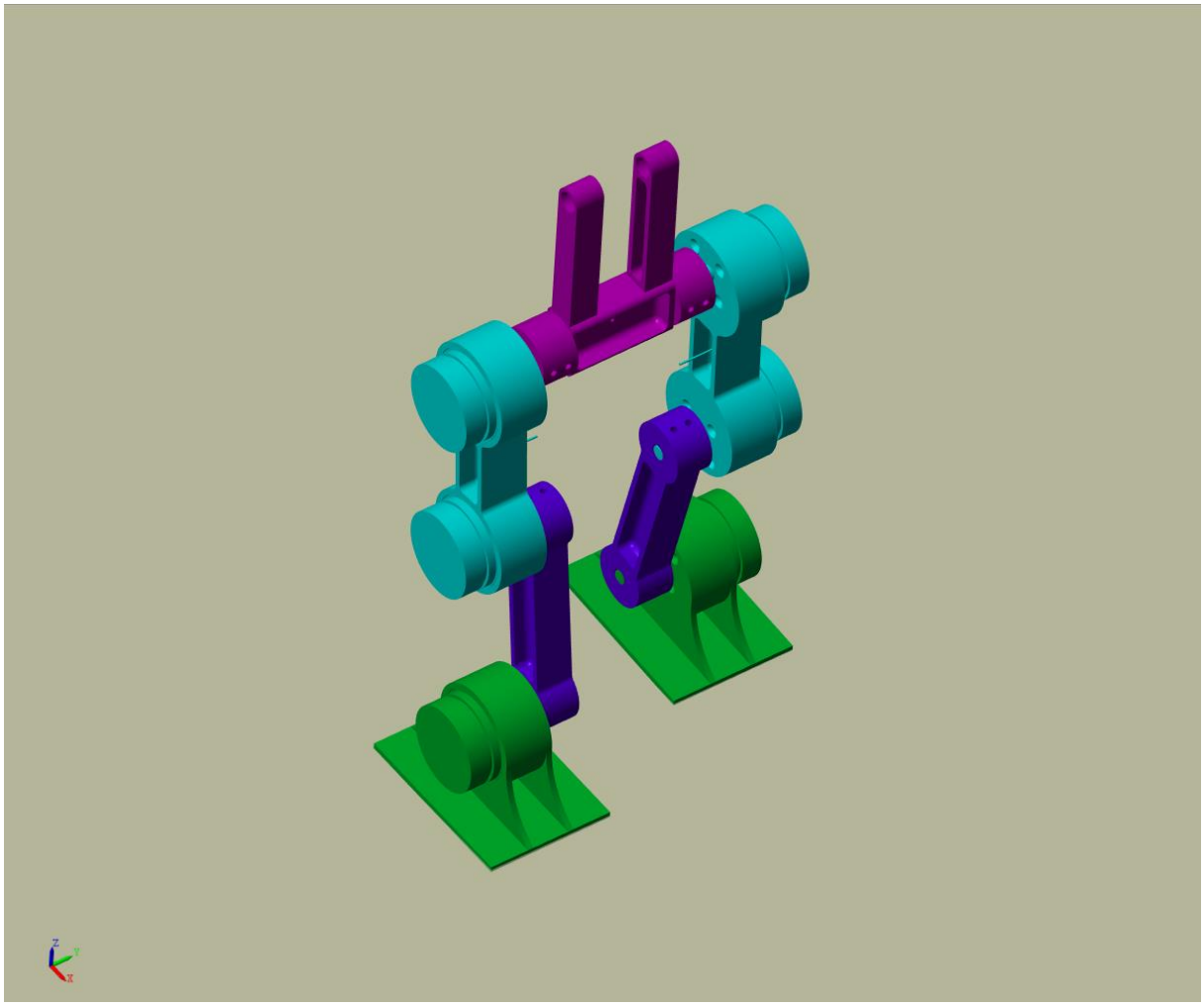


Figure 5.24: Test Platform Simulation Environment

As can be seen in Figure 5.24, the simulation environment is the same as the previous full-scale simulations, where the robot walks down a runway on a flat surface.

What is not seen in the image is that the z-rotation of the robot can either be damped or restricted within the simulated environment. The robot was simulated walking with a restricted z-rotation, and without. It was thus necessary to set up the experiment for both scenarios separately.

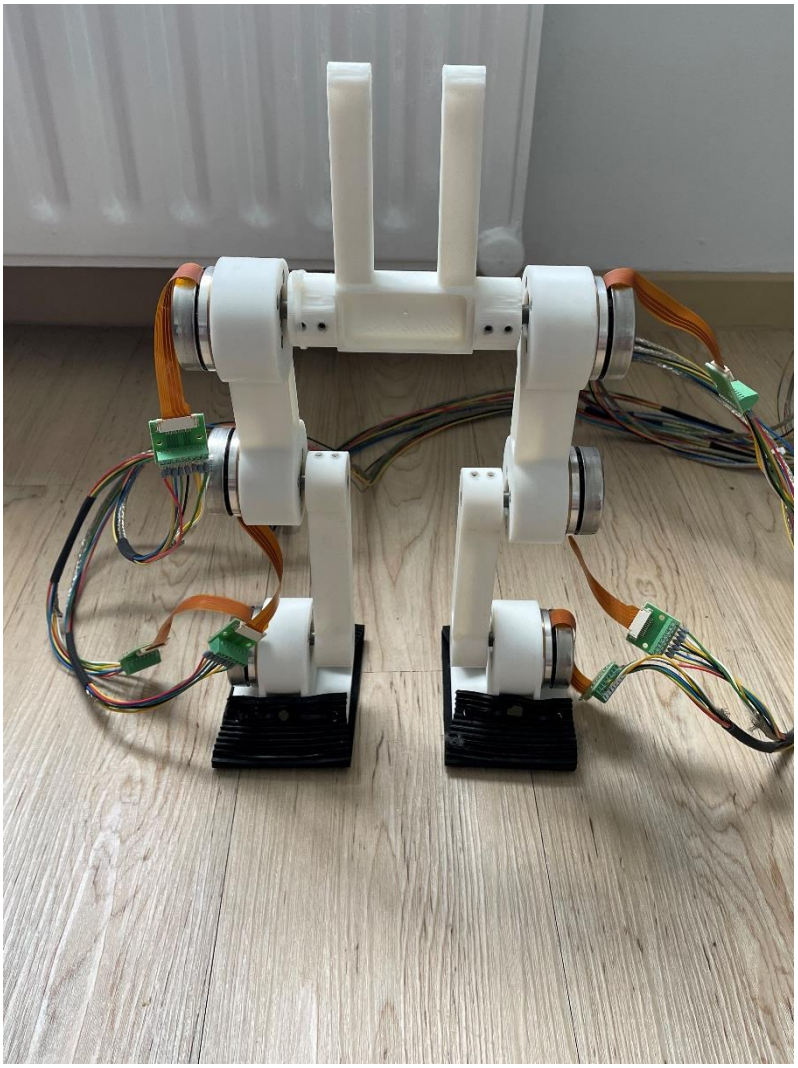


Figure 5.25: Test Platform Unrestricted

Figure 5.25 shows the robot set up in an unrestricted manner. The robot would be free to move on the flat surface.

Figure 5.26will show how a flexible boom arm was set up to restrict side to side movement but allow for movement in the z-axis. The boom arm forces the robot to move around a pivot point, which tests 2D planar stability. The arm also assists in getting the actuator cables out of the way of the moving joints as they can be fastened with cable ties.

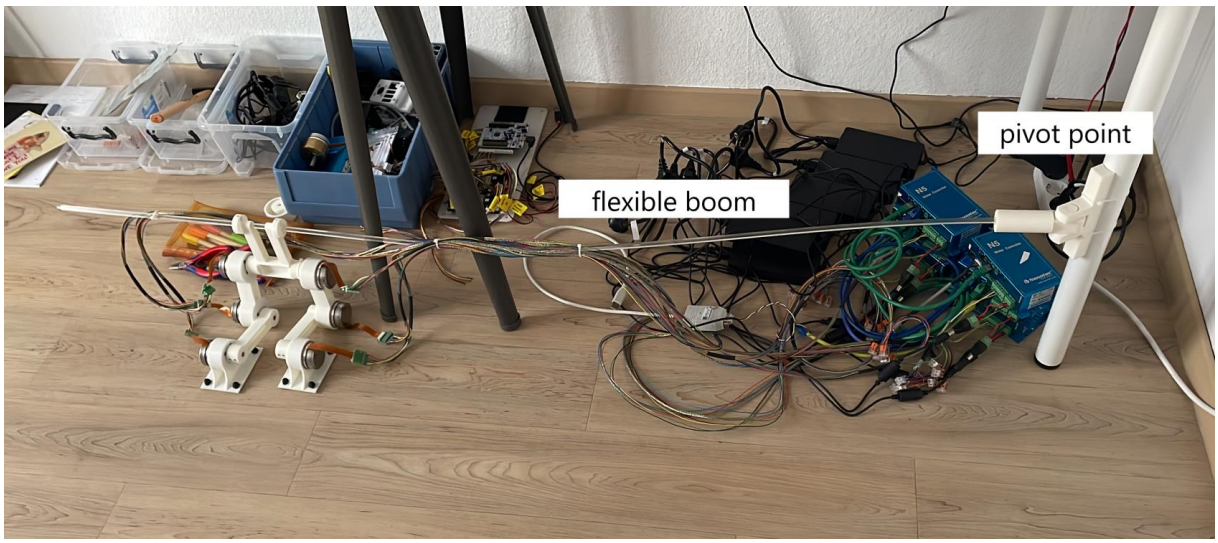


Figure 5.26: Test Platform Restricted

5.2.2 Data Analysis

The data analysis of the test platform is primarily based on qualitative observations. The analysis involved comparing the behaviour of the test platform with the simulated results obtained from the corresponding simulations. Specifically, the actions and movements of the test platform during a specific timeframe were observed and compared to the simulated actions during the same timeframe. This process was repeated multiple times to identify any anomalies or variations between the demonstrator's behaviour and the simulation. The focus was on qualitative observations rather than quantitative measures, aiming to gain insights into the performance, alignment, and discrepancies between the test platform and the simulation.

5.3 Results

In this chapter the results of the test platform experiment will be analysed.

5.3.1 First attempt at a ground contact walk

Having observed the walking algorithm in mid-air, the researcher tested if the test platform could perform a walk on a flat surface. It was observed that when the test platform attempted to take a step, the walking pattern was immediately out of sync. The leg would either lift and overstep, causing the robot to stall in a position with the feet widely spread, or not be able to step at all and rock back and forth with both feet staying on the floor. It appeared that although the peak torque of the motors was enough to allow for a steady and stable walking gait in a simulated environment, it did not work in a real-world environment.

It should be noted that there were limitations in terms of hardware accessibility which constrained further experimentation. However, a future direction for this research should focus on overcoming these limitations by addressing hardware-related issues and refining the experimental setup.

To enhance the experimental setup and bridge the existing disparities between simulation and practical implementation, the adoption of more suitable actuators is paramount. These actuators should possess a high torque-to-weight ratio, enabling them to effectively manage the complex dynamics of bipedal locomotion. Furthermore, incorporating encoders into the actuator design is crucial for enabling precise position control and providing valuable feedback. The availability of accurate position feedback is essential as it empowers the researcher to meticulously analyse the platform's movements and behaviour. In addition to position feedback, it is advisable to integrate force/torque sensors to gain comprehensive insights into the interactions between the platform and its environment. These sensors will facilitate a more profound understanding of the forces at play during walking. While focusing on actuator and feedback enhancements, it is equally important to evaluate the entire experimental setup comprehensively. Potential areas for improvement could include refining mechanical components, optimizing sensor configurations, and implementing advanced control algorithms to ensure the test platform operates seamlessly in a real-world environment.

CHAPTER 6

DISCUSSION AND CONCLUSION

The conclusion chapter provides a comprehensive summary of the research findings and highlights the key contributions and implications of the study. It recaps the research objectives, methodology, and results, emphasising how the research has addressed the research questions or objectives. Additionally, the conclusion chapter discusses the significance of the findings in the broader context of the field and identifies any limitations or areas for further research. Ultimately, the chapter aims to provide a concise and conclusive overview of the study's outcomes, underscoring its relevance and potential impact.

6.1 Summary of Findings

This research aimed to utilise mathematical modelling to predict the hardware requirements for untethered walking of a PLLE and optimise its stability in a simulated environment. The procedures and techniques employed involved the use of Onshape for generating a 3D model of the exoskeleton, Simulink for simulating its movement during walking, and MATLAB for optimising the simulation for stability using various metrics.

The results obtained from the simulations using the Linear Inverted Pendulum Model and the optimisation technique were analysed and compared.

6.1.1 LIPM Simulation Findings Summary

The LIPM simulation served as a foundational exploration of stable walking patterns for a PLLE. The key findings are as follows:

1. Stability Achievement: The LIPM simulation successfully achieved stability in the simulated environment, demonstrating the feasibility of maintaining an upright posture during walking.
2. Limitations in Human-likeness: Despite achieving stability, the simulated walking gait exhibited significant deviations from a natural human gait. Notable discrepancies included bent knees, a lack of a double support phase, and restricted step lengths, highlighting the limitations inherent in the simplified LIPM.
3. Trade-offs of Simplifications: The squat-like posture and limited step length observed in the simulation were identified as trade-offs resulting from the simplified assumptions of the LIPM. The linear motion imposed by the model oversimplified the complexities of musculoskeletal dynamics involved in human walking.

4. Insights into Efficiency and Torque Requirements: The LIPM simulation provided insights into the efficiency of walking patterns but raised concerns about the high torques required for stability. The imposed constraints, while ensuring stability, resulted in unnatural joint movements, emphasising the need for a more realistic representation of human walking dynamics.
5. Foundation for Future Models: While revealing the limitations, the LIPM simulation laid the groundwork for future models, encouraging a nuanced approach that considers the intricate interactions of joints and musculoskeletal systems for more accurate simulations.

The LIPM simulation findings underscore the necessity for advancements in modelling approaches to capture the complexities of human locomotion accurately. The trade-offs between stability and naturalness in gait patterns emphasise the need for more sophisticated models in the pursuit of designing efficient and human-like PLLEs.

6.1.2 Optimisation Simulation Findings Summary

The Optimisation technique, employing a cost function and genetic algorithm, aimed to refine the walking gait of a PLLE. The simulation demonstrated improved performance, with the exoskeleton walking 5.98 meters within 10 seconds and joint torques limited to a maximum of 700 Nm. The key findings are outlined below:

1. Human-like Gait Enhancement: The optimisation technique demonstrated promise in achieving a more human-like gait compared to the Linear Inverted Pendulum Model simulation. The incorporation of a cost function and genetic algorithm allowed for progressive improvements in walking patterns.
2. Increased Distance Coverage: The optimised gait surpassed the LIPM simulation in terms of the distance covered within a specified time frame, showcasing the potential for enhanced performance through iterative optimisation processes.
3. Challenges in Gait Smoothness: Despite achieving greater distance coverage, the optimised gait exhibited a skipping motion, raising questions about the algorithm's ability to generate a smooth, steady-paced walk. This observation highlighted challenges in balancing joint degrees of freedom and optimising for both distance and gait smoothness.
4. Degree of Freedom Limitations: The skipping motion suggested potential limitations in the number of joint degrees of freedom considered in the optimisation process. Understanding and addressing these limitations became crucial for refining the algorithm to achieve a more natural walking pattern.

5. Comparison with LIPM Results: While surpassing the LIPM simulation in certain aspects, the optimisation technique faced challenges in achieving a gait pattern suitable for practical implementation in a PLLE.

The Optimisation technique results underscore the potential for algorithmic enhancements in refining walking patterns. The trade-offs between increased distance coverage and gait smoothness highlight the complexities involved in optimising for human-like locomotion.

6.1.3 Test Platform Experimental Implementation Summary

A test platform experiment was conducted. The experimental implementation of the test platform aimed to bridge the gap between simulated success and real-world functionality. The key findings from the experimental phase are as follows:

1. Real-world Discrepancies: The transition from simulation to practical implementation revealed significant disparities. When attempting to walk on a flat surface, the test platform exhibited walking patterns immediately out of sync, with issues such as overstepping, stalling positions, and an inability to synchronise the walking pattern observed.
2. Hardware Limitations: Challenges in real-world implementation were attributed to hardware limitations, particularly in the performance of actuators. The observed discrepancies emphasized the critical role of hardware, indicating that the peak torque of the motors, while sufficient in a simulated environment, did not translate seamlessly to practical locomotion.
3. Suspended Movement Test Insights: The suspended movement test, while conducted in a controlled environment, provided crucial insights into the platform's behaviour. Dynamic interactions during suspended movement highlighted the complexities of translating theoretical models into physical systems.

The Test Platform Experimental Implementation results shed light on the intricacies of translating simulated success into practical functionality. Hardware limitations emerged as a focal point, highlighting the need for advancements in actuator technology and precise control mechanisms.

6.2 Contributions and Implications

This research makes several contributions to the field of PLLEs. The application of mathematical modelling and simulation techniques, combined with optimisation methods, provides valuable insights into the hardware requirements for untethered walking and the importance of stability in achieving a successful walking gait. The findings from the LIPM simulation and the optimisation technique demonstrate the significance of joint torque limitations and the impact on overall performance.

The results obtained can inform the design and development of future PLEEs, enabling researchers and engineers to optimise hardware requirements and improve stability. The limitations identified during the test platform experiment highlight the challenges of translating simulated results to real-world performance, emphasising the need for further investigation and refinement of control algorithms and hardware implementation.

In conclusion, this research has made progress in utilising mathematical modelling, simulation, and optimisation techniques to predict hardware requirements and optimise stability for untethered walking of a PLLE. The findings contribute to the understanding of the complex dynamics involved in exoskeleton locomotion and provide valuable insights for future advancements in this field. By addressing the challenges identified and exploring further research directions, the vision of enabling enhanced mobility and quality of life through PLLEs can be realised.

CHAPTER 7

RECOMMENDATIONS

These recommendations aim to provide actionable steps for addressing the identified shortcomings in each results section.

7.1 Recommendations for the Linear Inverted Pendulum Model Simulation

Improvements to the LIPM simulation are outlined below:

1. Model Refinement for Human-likeness: Address the limitations in human-likeness observed in the LIPM simulation by refining the model to capture more intricate dynamics of human walking. Consider incorporating additional joint degrees of freedom and biomechanical principles to achieve a gait pattern that more closely resembles natural human locomotion.
2. Integration of Musculoskeletal Complexity: Explore models that integrate the complexities of musculoskeletal systems, allowing for a more realistic representation of joint movements and interactions. Consider leveraging biomechanical data to inform the simulation, providing a foundation for a more accurate and natural walking gait.
3. Trade-off Analysis: Conduct a comprehensive analysis of the trade-offs associated with the simplified assumptions in the LIPM model. Evaluate the impact of these trade-offs on stability and efficiency and seek a balanced approach that considers both stability and human-like gait characteristics.

7.2 Recommendations for the Optimisation Simulation

Improvements to the Optimisation simulation are outlined below:

1. Gait Smoothness Optimisation: Focus on enhancing the optimisation technique to achieve a smoother, more natural walking gait. Investigate parameters within the cost function and genetic algorithm that contribute to the observed skipping motion. Iteratively refine the optimisation process to balance distance coverage with gait smoothness.
2. Joint Degrees of Freedom Analysis: Conduct an in-depth analysis of the joint degrees of freedom considered in the optimisation process. Explore the impact of limiting or expanding the degrees of freedom on the resulting gait pattern. Strive for a more nuanced optimisation strategy that addresses the intricacies of joint coordination.
3. Comparative Analysis with Human Gait Data: Include a comparative analysis with human gait data to validate and improve the optimisation results. Use biomechanical

data to guide the optimisation process, ensuring that the resulting gait closely aligns with natural human locomotion. This approach will provide a more objective assessment of the algorithm's performance.

7.3 Recommendations for the Test Platform Experimental Implementation

Improvements to the test platform experiment are outlined below:

1. Actuator Technology Advancements: Invest in advancements in actuator technology to address the observed hardware limitations. Focus on acquiring actuators with higher torque-to-weight ratios to enhance the platform's ability to execute stable and synchronised walking patterns in the real world.
2. Precise Position Control Integration: Integrate precise position control mechanisms, such as encoders, into the actuator design. This enhancement will enable more accurate and synchronised joint movements, addressing issues related to overstepping, stalling positions, and overall gait coordination.
3. Comprehensive Sensor Integration: Implement comprehensive sensor integration, including force/torque sensors, to gain insights into the interactions between the platform and its environment. This data will inform control algorithms and contribute to a more profound understanding of the forces at play during walking.
4. Control Algorithm Refinement: Refine control algorithms to address challenges in synchronising actuators and achieving a smooth walking gait. Consider advanced control strategies that optimise joint coordination, minimise discrepancies in joint movements, and enhance the overall stability and efficiency of the test platform.

REFERENCES

Aguilar-Sierra, H., Yu, W., Salazar, S. & Lopez, R.G. 2015. Design and Control of Hybrid Actuation Lower Limb Exoskeleton. *Advances in Mechanical Engineering*. DOI: 10.1177/1687814015590988.

Ahn, C.W. 2006. *Advances in evolutionary algorithms*. Springer.

Ali, M.A., Andy Park, H. & Lee, C.S.G. 2010. Closed-form inverse kinematic joint solution for humanoid robots. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 704–709. DOI: 10.1109/IROS.2010.5649842.

Amiri, M.S., Ramli, R., Tarmizi, M.A.A., Ibrahim, M.F. & Narooei, K.D. 2020. Simulation and Control of a Six Degree of Freedom Lower Limb Exoskeleton. 32(2):197–204.
"An Introduction to Genetic Algorithms". 1996. *Computers & Mathematics With Applications*. DOI: 10.1016/s0898-1221(96)90227-8.

Anam, K. & Al-Jumaily, A.A. 2012. Active exoskeleton control systems: State of the art. *Procedia Engineering*. 41(December):988–994. DOI: 10.1016/j.proeng.2012.07.273.

Asselin, P., Knezevic, S., Kornfeld, S., Cirnigliaro, C.M., Agranova-Breyter, I., Bauman, W.A. & Spungen, A.M. 2015. Heart Rate and Oxygen Demand of Powered Exoskeleton-Assisted Walking in Persons With Paraplegia. *The Journal of Rehabilitation Research and Development*. DOI: 10.1682/jrrd.2014.02.0060.

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

Awad, L.N., Bae, J., O'donnell, K., De Rossi, S.M.M., Hendron, K., Sloot, L.H., Kudzia, P., Allen, S., et al. 2017. A soft robotic exosuit improves walking in patients after stroke. *Science translational medicine*. 9(400):eaai9084.

Azizi, A. 2020. Applications of Artificial Intelligence Techniques to Enhance Sustainability of Industry 4.0: Design of an Artificial Neural Network Model as Dynamic Behavior Optimizer of Robotic Arms. *Complexity*. DOI: 10.1155/2020/8564140.

Bagheri, A., Najafi, F., Farrokhi, R., Moghaddam, R.Y. & Felezi, M.E. 2006. Design, Dynamic Modification, and Adaptive Control of a New Biped Walking Robot. *International Journal of Humanoid Robotics*. DOI: 10.1142/s0219843606000527.

Baldovino, R.G. & Jamisola, R.S. 2017. A Survey in the Different Designs and Control Systems of Powered Exoskeleton for Lower Extremities. *Journal of Mechanical Engineering and Biomechanics*. 1(4):103–115. DOI: 10.24243/jmeb/1.4.192.

Belogusev, V. & Egorov, A. V. 2019. Efficiency Testing of Electric Rotary Actuators With Non-Standard Reduction Units for Lower Limb Exoskeletons. *Istrazivanja I Projektovanja Za Privredu*. DOI: 10.5937/jaes17-20368.

Berninger, T.F.C. & Rixen, D.J. 2019. External Vibration Damping of a Robot Manipulator's TCP Using Acceleration Feedback. *Pamm*. DOI: 10.1002/pamm.201900344.

Birch, N., Graham, J., Priestley, T., Heywood, C., Sakel, M., Gall, A., Nunn, A.J. & Signal, N. 2017. Results of the First Interim Analysis of the RAPPER II Trial in Patients With Spinal Cord Injury: Ambulation and Functional Exercise Programs in the REX Powered Walking Aid. *Journal of Neuroengineering and Rehabilitation*. DOI: 10.1186/s12984-017-0274-6.

Brasseur, C., Sherikov, A., Collette, C., Dimitrov, D. & Wieber, P.-B. 2015. A Robust Linear MPC Approach to Online Generation of 3D Biped Walking Motion. DOI: 10.1109/humanoids.2015.7363423.

Castro, S. 2017. *Walking Robots, Part 3: Trajectory Optimization - MATLAB and Simulink Robotics Arena - YouTube*. Available: https://www.youtube.com/watch?v=-dEX1SZOZEY&ab_channel=MATLAB [2023, June 05].

Castro, S. 2019. *Walking Robot Control: From PID to Reinforcement Learning*. Available: https://blogs.mathworks.com/student-lounge/2019/04/24/walking-robot-control/ [2023, May 30].

Castro, S. & Kim, B. 2020. *Model-Based Control of Humanoid Walking - YouTube*. Available: https://www.youtube.com/watch?v=jnJbXdp2wak&ab_channel=MATLAB [2023, June 02].

Chang, S.R., Nandor, M.J., Li, L., Kobetic, R., Foglyano, K.M., Schnellenberger, J.R., Audu, M.L., Pinault, G., et al. 2017. A Muscle-Driven Approach to Restore Stepping With an Exoskeleton for Individuals With Paraplegia. *Journal of Neuroengineering and Rehabilitation*. DOI: 10.1186/s12984-017-0258-6.

Chen, B., Ma, H., Qin, L.Y., Gao, F., Chan, K.M., Law, S.W., Qin, L. & Liao, W.H. 2016. Recent developments and challenges of lower extremity exoskeletons. *Journal of*

Orthopaedic Translation. 5:26–37. DOI: 10.1016/j.jot.2015.09.007.

Chua, K.S.G. & Kuah, C.W.K. 2017. Innovating With Rehabilitation Technology in the Real World. *American Journal of Physical Medicine & Rehabilitation*. DOI: 10.1097/phm.0000000000000799.

Collins, S.J., Wisse, M. & Ruina, A. 2001. A Three-Dimensional Passive-Dynamic Walking Robot With Two Legs and Knees. *The International Journal of Robotics Research*. DOI: 10.1177/02783640122067561.

Corke, P. 2020. *Rigid Body Dynamics Masterclass*. Available: https://robotacademy.net.au/masterclass/rigid-body-dynamics/?lesson=410 [2020, July 26].

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*. 6(2):182–197.

Esquenazi, A., Talaty, M., Packel, A. & Saulino, M. 2012. The ReWalk Powered Exoskeleton to Restore Ambulatory Function to Individuals With Thoracic-Level Motor-Complete Spinal Cord Injury. *American Journal of Physical Medicine & Rehabilitation*. DOI: 10.1097/phm.0b013e318269d9a3.

Gait_cycle.jpg (1860×412). n.d. Available: https://www.physio-pedia.com/images/1/17/Gait_cycle.jpg [2023, June 30].

Garcia, C.E., Prett, D.M. & Morari, M. 1989. Model predictive control: Theory and practice—A survey. *Automatica*. 25(3):335–348.

Garcia, M., Chatterjee, A., Ruina, A. & Coleman, M. 1998. The simplest walking model: stability, complexity, and scaling.

Geigle, P.R. & Kallins, M. 2017. Exoskeleton-assisted walking for people with spinal cord injury. *Archives of physical medicine and rehabilitation*. 98(7):1493–1495.

Goswami, A. 1999. Postural Stability of Biped Robots and the Foot-Rotation Indicator (FRI) Point. *The International Journal of Robotics Research*. DOI: 10.1177/02783649922066376.

Gurriet, T., Finet, S., Boeris, G., Duburcq, A., Hereid, A., Harib, O., Masselin, M., Grizzle, J., et al. 2018. Towards restoring locomotion for paraplegics: Realizing dynamically stable

walking on exoskeletons. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2804–2811.

Gurriet, T., Tucker, M., Duburcq, A., Boeris, G. & Ames, A.D. 2020. Towards Variable Assistance for Lower Body Exoskeletons. *Ieee Robotics and Automation Letters*. DOI: 10.1109/lra.2019.2955946.

Hamed, K.A. & Grizzle, J.W. 2013. Robust Event-Based Stabilization of Periodic Orbits for Hybrid Systems: Application to an Underactuated 3D Bipedal Robot. DOI: 10.1109/acc.2013.6580811.

Hamed, K.A., Sadati, N., Gruver, W.A. & Dumont, G.A. 2010. Continuous-Time Update Laws With Radial Basis Step Length for Control of Bipedal Locomotion. *Electronics Letters*. DOI: 10.1049/el.2010.2481.

Harib, O., Hereid, A., Agrawal, A., Gurriet, T., Finet, S., Boeris, G., Duburcq, A., Mungai, M.E., et al. 2018. Feedback control of an exoskeleton for paraplegics: Toward robustly stable, hands-free dynamic walking. *IEEE Control Systems Magazine*. 38(6):61–87.

Haupt, R.L. & Haupt, S.E. 2003. Practical Genetic Algorithms. DOI: 10.1002/0471671746.

Hong, Y.-D. & Lee, K. 2016. Stable Walking of Humanoid Robots Using Vertical Center of Mass and Foot Motions by an Evolutionary Optimized Central Pattern Generator. *International Journal of Advanced Robotic Systems*. DOI: 10.5772/62039.

Housner, G.W. & Hudson, D.E. 1983. *Applied mechanics: Dynamics*. V. 8. DOI: 10.1016/0378-3804(83)90056-6.

How the Genetic Algorithm Works - MATLAB & Simulink - MathWorks United Kingdom. n.d. Available: https://uk.mathworks.com/help/gads/how-the-genetic-algorithm-works.html [2024, February 04].

Huynh, V., Burger, G., Dang, Q.V., Pelgé, R., Boéris, G., Grizzle, J.W., Ames, A.D. & Masselin, M. 2021. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.723780. *Hybrid III - Wikipedia*. n.d. Available: https://en.wikipedia.org/wiki/Hybrid_III [2023, July 09].

Jianwei, Z. & Zhou, J. 2023. The Five-Dof Explosion-Removal Manipulator Based on Position and Velocity Feedforward Compensation Control. *Sensors*. DOI: 10.3390/s23094276.

Jimenez-Fabian, R. & Verlinden, O. 2012. Review of control algorithms for robotic ankle systems in lower-limb orthoses, prostheses, and exoskeletons. *Medical engineering & physics*. 34(4):397–408.

Joe, H.-M. & Oh, J.-H. 2019. A Robust Balance-Control Framework for the Terrain-Blind Bipedal Walking of a Humanoid Robot on Unknown and Uneven Terrain. *Sensors*. DOI: 10.3390/s19194194.

Joseph, C., Scriba, E., Wilson, V., Mothabeng, J. & Theron, F. 2017. People with Spinal Cord Injury in Republic of South Africa. *American Journal of Physical Medicine and Rehabilitation*. 96(2):S109–S111. DOI: 10.1097/PHM.0000000000000594.

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K. & Hirukawa, H. 2001. The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*. V. 1. IEEE. 239–246.

Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K. & Hirukawa, H. n.d. Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point. DOI: 10.1109/robot.2003.1241826.

Katz, B.G. 2018. A low cost modular actuator for dynamic robots. *Thesis, Master*. (2016):104. Available: http://hdl.handle.net/1721.1/118671%0Ahttps://dspace.mit.edu/handle/1721.1/118671#files-area.

Kenneally, G., De, A. & Koditschek, D.E. 2016. Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*. 1(2):900–907.

Kurennov, S., Barakhov, K. & Vambol, O. 2022. Topological Optimization of a Symmetrical Adhesive Joint. Island Model of Genetic Algorithm. *Radioelectronic and Computer Systems*. DOI: 10.32620/reks.2022.3.05.

Lamont, G.B. & Veldhuizen, D.A. Van. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. DOI: 10.1007/978-0-387-36797-2.

Li, L., Xie, Z., Luo, X. & Li, J. 2021. Trajectory Planning of Flexible Walking for Biped Robots

Using Linear Inverted Pendulum Model and Linear Pendulum Model. *Sensors*. DOI: 10.3390/s21041082.

Lin, F. & Brandt, R.D. 1998. An Optimal Control Approach to Robust Control of Robot Manipulators. *Ieee Transactions on Robotics and Automation*. DOI: 10.1109/70.660845.

Luo, S., Androwis, G., Adamovich, S., Su, H., Nunez, E. & Zhou, X. 2021. Available: https://www.frontiersin.org/articles/10.3389/frobt.2021.702845.

Lynch, K.M. & Park, F.C. 2017. *Modern Robotics*. Cambridge University Press.

Magaqa, Q., Ariana, P. & Polack, S. 2021. Examining the Availability and Accessibility of Rehabilitation Services in a Rural District of South Africa: A Mixed-Methods Study. *International Journal of Environmental Research and Public Health*. DOI: 10.3390/ijerph18094692.

MathWorks Student Competitions Team. 2023. *MATLAB and Simulink Robotics Arena: Walking Robot*. Available: https://github.com/mathworks-robotics/msra-walking-robot [2023, May 29].

Mazumdar, A., Spencer, S.J., Hobart, C., Salton, J., Quigley, M., Wu, T., Bertrand, S., Pratt, J., et al. 2017. Parallel Elastic Elements Improve Energy Efficiency on the STEPPR Bipedal Walking Robot. *IEEE/ASME Transactions on Mechatronics*. 22(2):898–908. DOI: 10.1109/TMECH.2016.2631170.

Menga, G. & Ghirardi, M. 2019. Estimation and Closed-Loop Control of COG/ZMP in Biped Devices Blending CoP Measures and Kinematic Information. *Robotics*. 8(4):89.

Minchala, L.I., Astudillo-Salinas, F., Palacio-Baus, K. & Vazquez-Rodas, A. 2017. Mechatronic Design of a Lower Limb Exoskeleton. *Design, Control and Applications of Mechatronic Systems in Engineering*. (May). DOI: 10.5772/67460.

Naka, T., Hayashi, T., Sugyo, A., Towatari, F. & Maeda, T. 2022. Effect of Age at Injury on Walking Ability Following Incomplete Cervical Spinal Cord Injury: A Retrospective Cohort Study. *Spine Surgery and Related Research*. DOI: 10.22603/ssrr.2021-0240.

Ndabeni, L.L., Rogerson, C.M. & Booyens, I. 2016. Innovation and Local Economic Development Policy in the Global South: New South African Perspectives. *Local Economy*

the Journal of the Local Economy Policy Unit. DOI: 10.1177/0269094215621865.

Neven, A. & Ectors, W. 2023. “I am dependent on others to get there”: Mobility barriers and solutions for societal participation by persons with disabilities. *Travel Behaviour and Society*. 30:302–311. DOI: 10.1016/J.TBS.2022.10.009.

Orekhov, G., Fang, Y., Cuddeback, C.F. & Lerner, Z.F. 2021. Usability and Performance Validation of an Ultra-Lightweight and Versatile Untethered Robotic Ankle Exoskeleton. *Journal of Neuroengineering and Rehabilitation*. DOI: 10.1186/s12984-021-00954-9.

Otten, E. 2003. Inverse and forward dynamics: Models of multi-body systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 358(1437):1493–1500. DOI: 10.1098/rstb.2003.1354.

Penzlin, B., Bergmann, L., Li, Y., Ji, L., Leonhardt, S. & Ngo, C. 2021. Design and First Operation of an Active Lower Limb Exoskeleton With Parallel Elastic Actuation. *Actuators*. DOI: 10.3390/act10040075.

Pons, J.L. 2010. Rehabilitation Exoskeletal Robotics. *IEEE Engineering in Medicine and Biology Magazine*. 29(3):57–63. DOI: 10.1109/MEMB.2010.936548.

Pratt, G.A. & Williamson, M.M. 1995. Series elastic actuators. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*. V. 1. IEEE. 399–406.

Pratt, J.E. & Tedrake, R. 2006. Velocity-based stability margins for fast bipedal walking. In *Fast motions in biomechanics and robotics*. Springer. 299–324.

Ramadan, S.Z. 2018. Joint <mml:math xmlns:mml="http://www.w3.org/1998/Math/MathML" Id="M1"><mml:mrow><mml:mover Accent="true"><mml:mrow><mml:mi>X</Mml:mi></Mml:mrow><mml:mo>-</Mml:mo></Mml:mover></Mml:mrow></Mml:math> and <mml:math xmlns:mml="http://www.w3.org/1998/Math/Math. *Mathematical Problems in Engineering*. DOI: 10.1155/2018/6516879.

ReWalk Robotics. 2020. *ReWalk*. Available: https://rewalk.com/wp-content/uploads/2015/07/New-ReWalk-Personal-6.01.jpg [2020, August 15].

Riener, R. 2016. The Cybathlon Promotes the Development of Assistive Technology for People With Physical Disabilities. *Journal of Neuroengineering and Rehabilitation*. DOI: 10.1186/s12984-016-0157-2.

Rigatos, G. & Abbaszadeh, M. 2021. Nonlinear Optimal Control for Multi-DOF Robotic Manipulators With Flexible Joints. *Optimal Control Applications and Methods*. DOI: 10.1002/oca.2756.

Rodríguez-Fernández, A., Lobo-Prat, J. & Font-Llagunes, J.M. 2021. Systematic Review on Wearable Lower-Limb Exoskeletons for Gait Training in Neuromuscular Impairments. *Journal of Neuroengineering and Rehabilitation*. DOI: 10.1186/s12984-021-00815-5.

Roussel, L., Canudas-De-Wit, C. & Goswami, A. 1998. Generation of energy optimal complete gait cycles for biped robots. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*. V. 3. IEEE. 2036–2041.

Sardain, P. & Bessonnet, G. 2004. Forces acting on a biped robot. Center of pressure-zero moment point. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*. 34(5):630–637.

Schrade, S.O., Devittori, G., Easthope, C.A., Shirota, C., Lambercy, O. & Gassert, R. 2019. Exoskeleton Knee Compliance Improves Gait Velocity and Stability in a Spinal Cord Injured User: A Case Report. *arXiv preprint arXiv:1911.04316*.

Seven, U. 2007.

Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. 2010. *Robotics: modelling, planning and control*. Springer Science & Business Media.

van Silfhout, L., Hosman, A.J.F., van de Meent, H., Bartels, R.H.M.A. & Edwards, M.J.R. 2023. Design recommendations for exoskeletons: Perspectives of individuals with spinal cord injury. *Journal of Spinal Cord Medicine*. 46(2):256–261. DOI: 10.1080/10790268.2021.1926177/SUPPL_FILE/YSCM_A_1926177_SM6436.DOCX.

Singh, R., Chaudhary, H. & Singh, A. 2018. A Novel Gait-Based Synthesis Procedure for the Design of 4-Bar Exoskeleton With Natural Trajectories. *Journal of Orthopaedic Translation*. DOI: 10.1016/j.jot.2017.09.001.

Singh, S., Jadhav, B.D. & Krishna, K.M. 2014. Posture Control of a Three-Segmented Tracked Robot With Torque Minimization During Step Climbing. DOI: 10.1109/icra.2014.6907470.

Slotine, J.-J.E. & Li, W. 1991. *Applied nonlinear control*. V. 199. Prentice hall Englewood Cliffs, NJ.

Spong, M.W., Hutchinson, S. & Vidyasagar, M. 2005. Robot modeling and control, jon wiley & sons. *Inc, ISBN-100-471-649*.

Spong, M.W., Hutchinson, S. & Vidyasagar, M. 2020. *Robot modeling and control*. John Wiley & Sons.

Spratt, N.J., Spratt, N.J. & Marquez, J. 2021. Physiotherapy Using a Free-Standing Robotic Exoskeleton for Patients With Spinal Cord Injury: A Feasibility Study. *Journal of Neuroengineering and Rehabilitation*. DOI: 10.1186/s12984-021-00967-4.

Statistics South Africa. 2011. Profile of persons with disabilities in South Africa. *Census*. 185. Available: http://www.statssa.gov.za/publications/Report-03-01-59/Report-03-01-592011.pdf.

Sugihara, T. & Morisawa, M. 2020. A Survey: Dynamics of Humanoid Robots. *Advanced Robotics*. DOI: 10.1080/01691864.2020.1778524.

Suryadi, D., Prasetya, A., Daratha, N. & Agustian, I. 2023. Optimum Design of Tuned Mass Damper Parameters to Reduce Seismic Response on Structure Using Genetic Algorithm. *Asean Engineering Journal*. DOI: 10.11113/aej.v13.17890.

Suzumura, A. & Fujimoto, Y. 2012. Control of Dynamic Locomotion for the Hybrid Wheel-Legged Mobile Robot by Using Unstable-Zeros Cancellation. DOI: 10.1109/icra.2012.6224968.

Takahashi, N. & Nonaka, K. 2012. Model Predictive Obstacle Avoidance Control for Leg/Wheel Mobile Robots With Optimized Articulated Leg Configuration. DOI: 10.1109/cdc.2012.6426044.

The Gait Cycle - Physiopedia. n.d. Available: https://www.physio-pedia.com/The_Gait_Cycle [2023, June 28].

Vanderbilt University. 2020. *Vanderbilt*. Available: https://news.vanderbilt.edu/files/Indegouserweb.jpg [2020, August 15].

Varma, N., Jolly, K.G. & Suresh, K.S. 2018. A review on numerical models and controllers for biped locomotion over leveled and uneven terrains. *Advances in Robotics Research*. 2(2):151–159. DOI: 10.12989/arr.2018.2.2.151.

Vukobratović, M. & Borovac, B. 2004. Zero-Moment Point — Thirty Five Years of Its Life. *International Journal of Humanoid Robotics*. DOI: 10.1142/s0219843604000083.

Wang, S., Wang, L., Meijneke, C., Asseldonk, E.V. van, Hoellinger, T., Cheron, G., Ivanenko, Y., Scaleia, V. La, et al. 2015. *MINDWALKER*. Available: https://www.semanticscholar.org/paper/Design-and-Control-of-the-MINDWALKER-Exoskeleton-Wang-Wang/b6e3ab7d4b11f141418183e3bd14f93125d87d31/figure/0 [2020, August 15].

Wu, Q. & Wu, H. 2018. Development, dynamic modeling, and multi-modal control of a therapeutic exoskeleton for upper limb rehabilitation training. *Sensors*. 18(11):3611.

Wu, X.A., Huh, T.M., Sabin, A., Suresh, S.A. & Cutkosky, M.R. 2019. Tactile sensing and terrain-based gait control for small legged robots. *IEEE Transactions on Robotics*. 36(1):15–27.

Yan, G., Tang, C., Lin, Z. & Malloci, I. 2014. Feedback Control for Compass-Like Biped Robot With Underactuated Ankles Using Transverse Coordinate Transformation. *Robotica*. DOI: 10.1017/s0263574714000447.

Yan, T., Cempini, M., Oddo, C.M. & Vitiello, N. 2015. Review of assistive strategies in powered lower-limb orthoses and exoskeletons. *Robotics and Autonomous Systems*. 64:120–136. DOI: 10.1016/j.robot.2014.09.032.

Yuan, K. & Liu, P. 2018. An Improved Formulation for Model Predictive Control of Legged Robots for Gait Planning and Feedback Control. DOI: 10.1109/iros.2018.8594309.

Zhang, X. 2019. Application of Proprioception Quasi-Direct Drive Actuators on Dynamic Robotic Systems. *UCLA Electronic Theses and Dissertations*.

Zhang, L., He, J. & Wang, S. 2017. Inverse Kinematic Solutions of Dual Redundant Camera

Robot Based on Genetic Algorithm. *Mathematical Problems in Engineering*. DOI: 10.1155/2017/7486178.

Zhu, C., Liu, Q., Meng, W., Ai, Q. & Xie, S.Q. 2021. An Attention-Based CNN-LSTM Model With Limb Synergy for Joint Angles Prediction. DOI: 10.1109/aim46487.2021.9517544.

Zoss, A., Kazerooni, H. & Chu, A. 2005. On the mechanical design of the Berkeley Lower Extremity Exoskeleton (BLEEX). In *2005 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. 3465–3472.

Zutven, P. van, Kostić, D. & Nijmeijer, H. 2010. On the stability of bipedal walking. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer. 521–532.

APPENDIX A

MATLAB SOFTWARE OVERVIEW

Upon opening the folder containing the MathWorks staff team's software (Version 4.0), one is greeted with a number of folders on the left of the screen, along with a startup script. Running the startupWalkingRobot.m script does the following:

- Clears everything in the current MATLAB session
- Adds the folders on the left to the path
- Loads the robot parameters by running the robotParameters.m script
- Opens the README file

In Figure A1 it can be seen that the opaque folders on the left have been added to the path, the README has also been opened to display overview information of the software, as well as links where one can learn more about how the software was developed.

Figure A1: Walking Robot Software folders

An extract from the README below outlines the relevant folders to this research and what they contain:

1. `LIPM` -- Shows how to generate a walking pattern using the linear inverted pendulum model (LIPM), which is one of the foundational models for humanoid walking control.

2. `ModelingSimulation` -- Shows how to build the simulation of the walking robot, including contact forces, various actuator models, and importing from CAD.

3. `Optimization` -- Shows how to use genetic algorithms to optimise joint angle trajectories for stability and speed.

In Figure A2, the section of the animateLIPM.mlx Live Script where the researcher modified the variables can be seen.

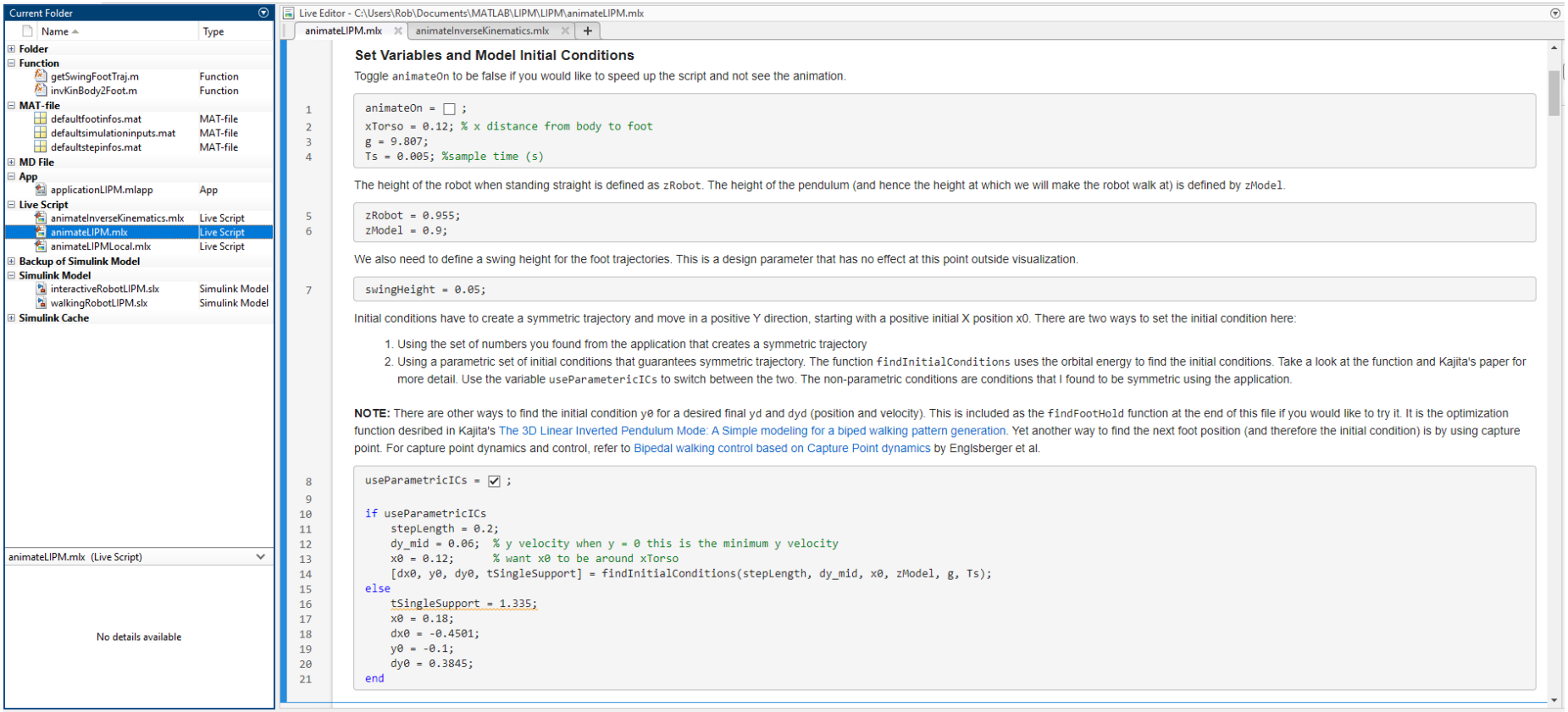


Figure A2: Live Script Animate LIPM

The output files generated by the simulation can be seen in Figure 3.18. In the bottom left corner, the values of the waypoints for each joint can be seen in radians. The value of the final reward output is also stated.

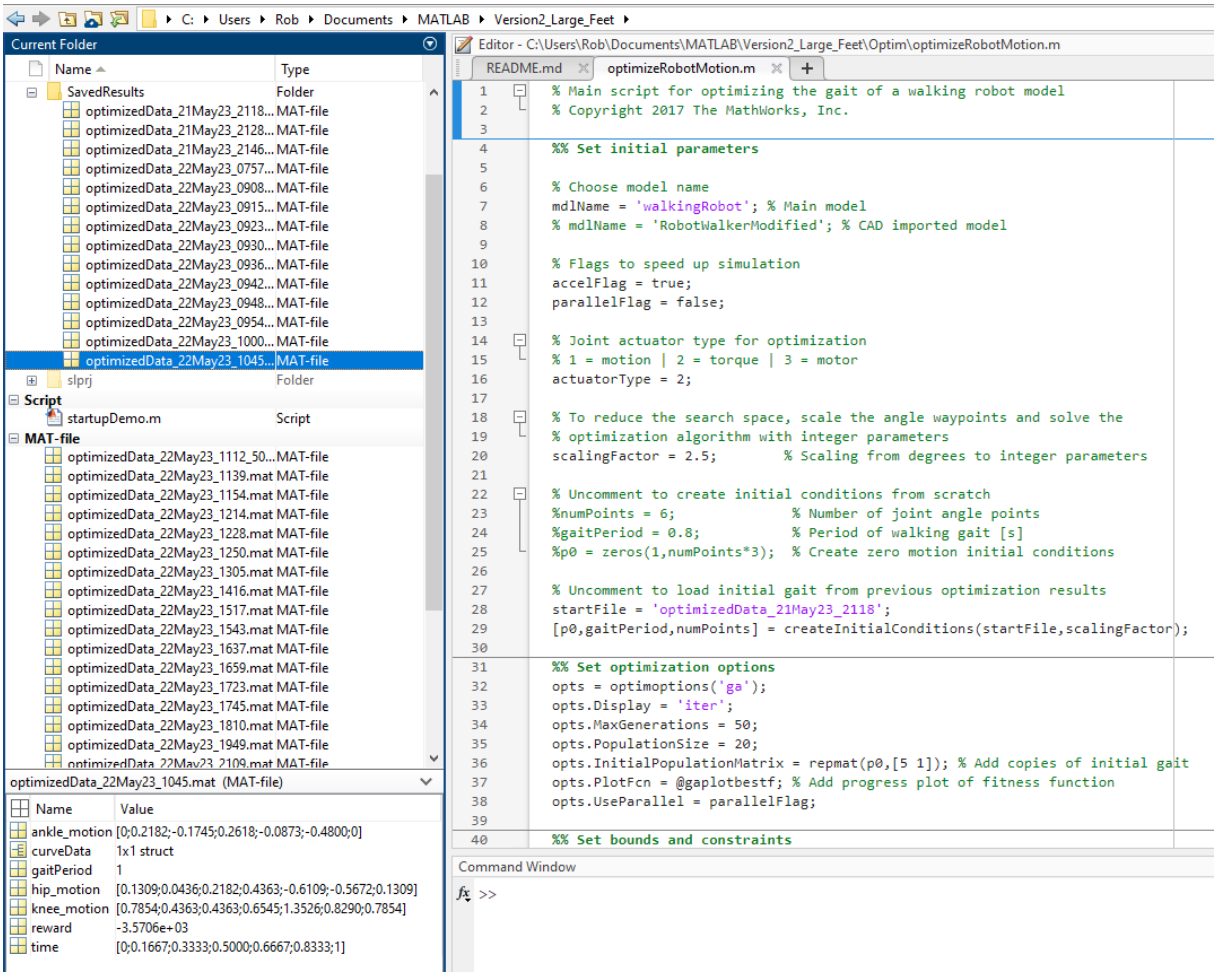


Figure A3: Optimisation Output Files

APPENDIX B

RELEVANT MATLAB CODE SNIPPETS

Code Snippet 1

The following variables within the animateLIPM.mlx Live Script were modified. Comments are in green:

```
xTorso = 0.12;          % x distance from body to foot
zRobot = 0.955;         % height of the robot when standing straight
zModel = 0.9;           % height of the pendulum (and hence the height at which we will make
                        the robot walk at)
swingHeight = 0.05;     % max height of the swing foot
stepLength = 0.2;       % length of a step
dy_mid = 0.06;          % y velocity when y = 0 this is the minimum y velocity
x0 = 0.12;              % want x0 to be around xTorso
```

Code Snippet 2

An extract of the variables which were modified to match the limb lengths of the CAD model can be found below. As was previously noted, comments are in green. The leading comment notes that the values entered in the Live script also needed to be entered within the function so that they matched.

```
% NOTE: make sure parameters match in inverse kinematics function
L1 = 0.12;      % X offset
L2 = 0;         % Y offset
L3 = 0.42;      % Upper leg length
L4 = 0.42;      % Lower leg length
L5 = 0.075;     % Ankle to foot contact offset
```

Code Snippet 3

```
%% General parameters
world_rot_damping = 100;

%% Contact/friction parameters
mu_k = 0.8;
mu_s = 0.99;
```

```
init_height = 95;

%% Robot joint parameters
motion_time_constant = 0.001;

%% Joint controller parameters
hip_servo_kp = 6000;
hip_servo_ki = 1000;
hip_servo_kd = 2000;
knee_servo_kp = 6000;
knee_servo_ki = 500;
knee_servo_kd = 1000;
ankle_servo_kp = 2000;
ankle_servo_ki = 400;
ankle_servo_kd = 800;
deriv_filter_coeff = 100;
max_torque = 200;
```

Code Snippet 4

```
%% Run optimization
costFcn = @(p)simulateWalkingRobot(p,mdlName,scalingFactor,gaitPeriod,actuatorType);
```

Code Snippet 5

```
% Simulate the model
    simout = sim(mdlName,'SrcWorkspace','current','FastRestart','on');

    % Unpack logged data
    wAvg = mean(simout.yout{1}.Values.Data);
    xEnd = simout.yout{2}.Values.Data(end);
    tEnd = simout.tout(end);

    %    Calculate penalty from logged data
    %    Distance traveled without falling
    %    (ending simulation early) increases reward
    positiveReward = sign(xEnd)*xEnd^2 * tEnd;
    %    Angular velocity and trajectory aggressiveness
```

```
%    (number of times the derivative flips signs) decreases reward
%    NOTE: Set lower limits to prevent divisions by zero
aggressiveness = 0;
diffs = [diff(hip_motion) diff(knee_motion) diff(ankle_motion)];
for idx = 1:numel(diffs)-1
    if (sign(diffs(idx)/diffs(idx+1))<0) && mod(idx,N)
        aggressiveness = aggressiveness + 1;
    end
end
negativeReward = max(abs(wAvg),0.01) * max(aggressiveness,1);
%   Negative sign needed since the optimization minimizes cost function
penalty = -positiveReward/negativeReward;
```

Code Snippet 6

```
%% Set initial parameters

% Choose model name
mdlName = 'walkingRobot'; % Main model
% mdlName = 'RobotWalkerModified'; % CAD imported model

% Flags to speed up simulation
accelFlag = true;
parallelFlag = false;
```

As can be seen in the comments of the above code, the first variable to modify is which model to simulate. In this case ‘walkingRobot’ was entered as that is the Simulink model where the researcher has previously imported the CAD model.

The accelFlag and parallelFlag are used to speed up the simulation time and can be set to true or false. The accelFlag when set to true, generates C code from the model and enables the model to simulate repeatedly without compiling. It also prevents the Mechanics Explorer from opening to display the simulation running, which saves time. The parallelFlag, when set to true, allows multiple simulations to run in parallel. This feature relies on CPU’s with multiple cores to spread the workload.

The hardware used for the simulations for this research was a Lenovo ideapad 330 with an Intel(R) Core(TM) i3-8130U CPU and 4 GB of RAM. It was found that the parallelFlag caused

the application to hang and did not provide much benefit in terms of speeding up the simulation when using this dual core CPU. The accelFlag was however very helpful in speeding up the simulation. It should be noted that it can be useful to leave the accelFlag off to be able to see the simulation animation in the Mechanics Explorer, to see how the walking trajectory is progressing.

Code Snippet 7

```
% Joint actuator type for optimization
% 1 = motion | 2 = torque | 3 = motor
actuatorType = 2;
```

Code Snippet 8

```
% To reduce the search space, scale the angle waypoints and solve the
% optimization algorithm with integer parameters
scalingFactor = 2.5;        % Scaling from degrees to integer parameters

% Uncomment to create initial conditions from scratch
%numPoints = 6;              % Number of joint angle points
%gaitPeriod = 0.8;           % Period of walking gait [s]
%p0 = zeros(1,numPoints*3);  % Create zero motion initial conditions

% Uncomment to load initial gait from previous optimization results
startFile = 'optimizedData_21May23_2118';
[p0,gaitPeriod,numPoints] = createInitialConditions(startFile,scalingFactor);
```

Code Snippet 9

```
%% Set optimization options
opts = optimoptions('ga');
opts.Display = 'iter';
opts.MaxGenerations = 50;
opts.PopulationSize = 20;
opts.InitialPopulationMatrix = repmat(p0,[5 1]); % Add copies of initial gait
opts.PlotFcn = @gaplotbestf; % Add progress plot of fitness function
opts.UseParallel = parallelFlag;
```

```
%% Set bounds and constraints
% Upper and lower angle bounds
upperBnd = [30*ones(1,numPoints), ... % Ankle limits
        90*ones(1,numPoints), ... % Knee limits
        45*ones(1,numPoints)] ... % Hip limits
        /scalingFactor;
lowerBnd = [-30*ones(1,numPoints), ... % Ankle limits
          0*ones(1,numPoints), ... % Knee limits
        -45*ones(1,numPoints)] ... % Hip limits
        /scalingFactor;
```

Code Snippet 10

```
%% Leg parameters
lower_leg_length = 10;          % in cm
upper_leg_length = 10;          % in cm

%% Joint controller parameters
max_torque = 6.838;             % in Nm (taken from motor data sheet)
```

APPENDIX C

INTERMEDIATE OPTIMISATION RESULTS

This appendix contains some of the preliminary results of the optimisation simulations to illustrate their progression. The distance walked, and joint torques will be shown.

Result 1: Running gait file optimizedData_21May23_2118.mat on Version2 folder (Robot with small feet)

Comment: Robot falls backwards Immediately.

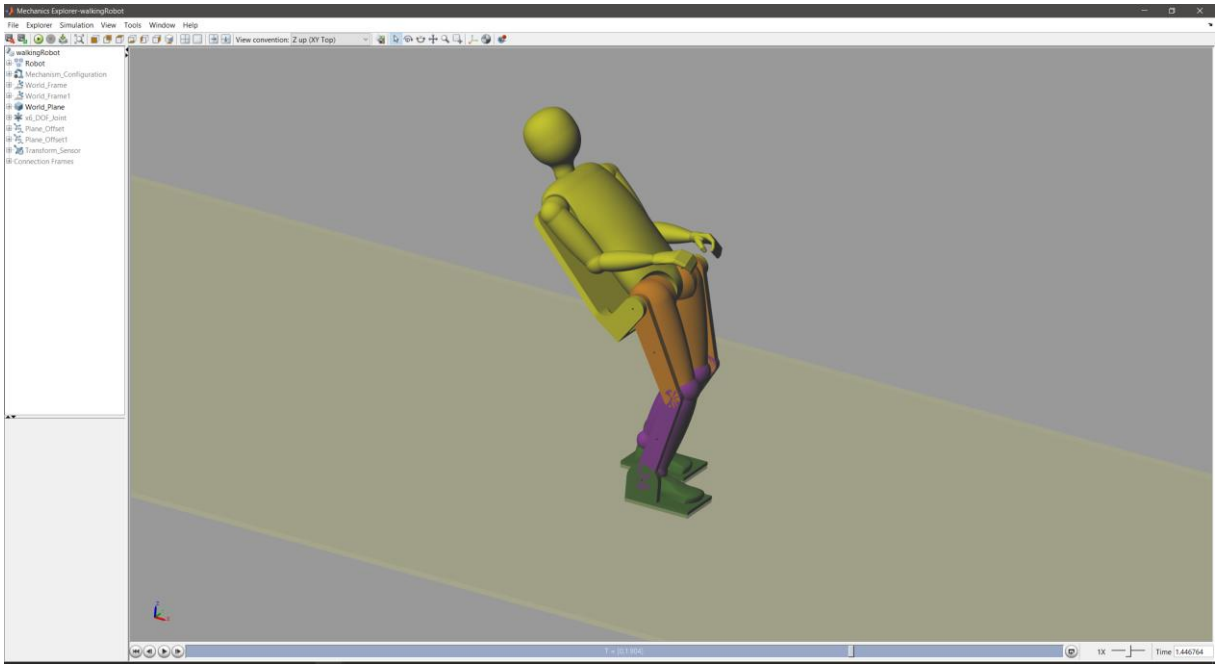


Figure C1: Result 1

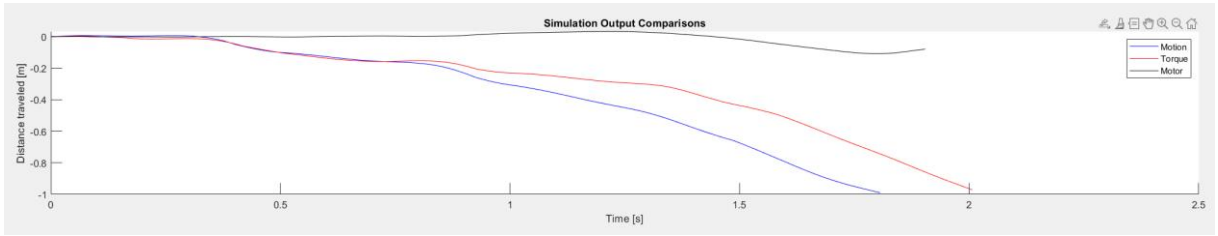


Figure C2: Result 1 Distance Travelled

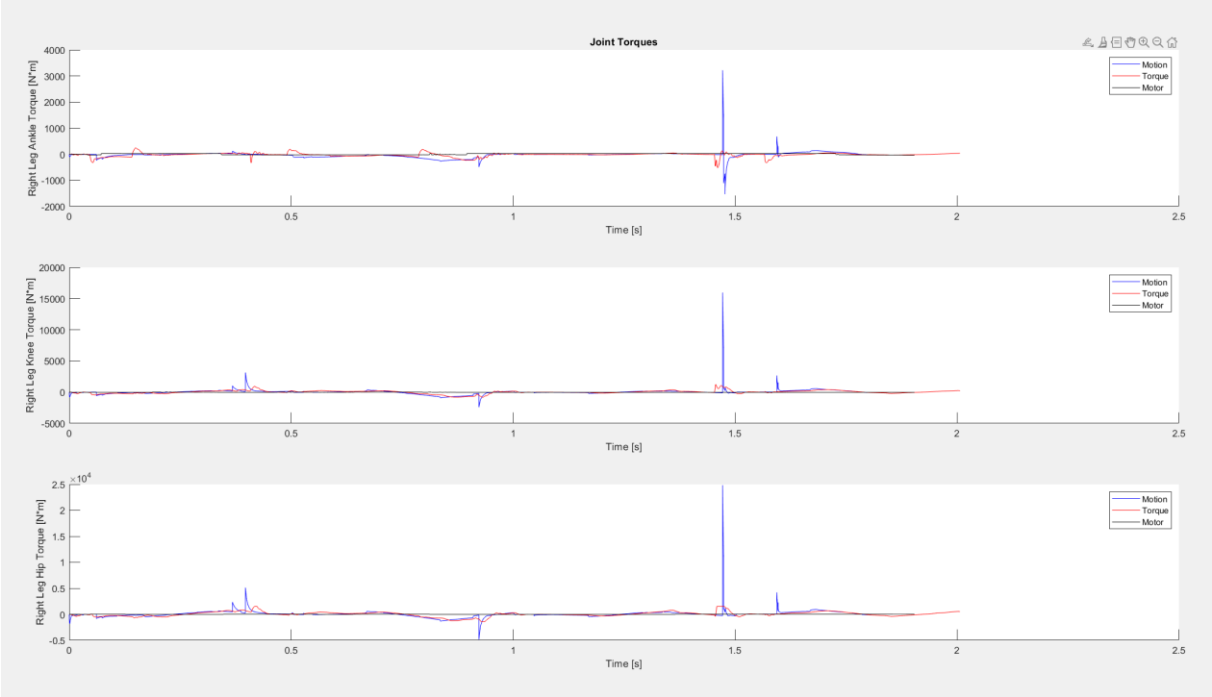


Figure C3: Result 1 Joint Torques

Result 2: Running gait file 1500Nm.mat on Version2 folder (Robot with small feet)

Comment: Robot flicks its legs back and forth and manages not to fall over but does not progress forwards.

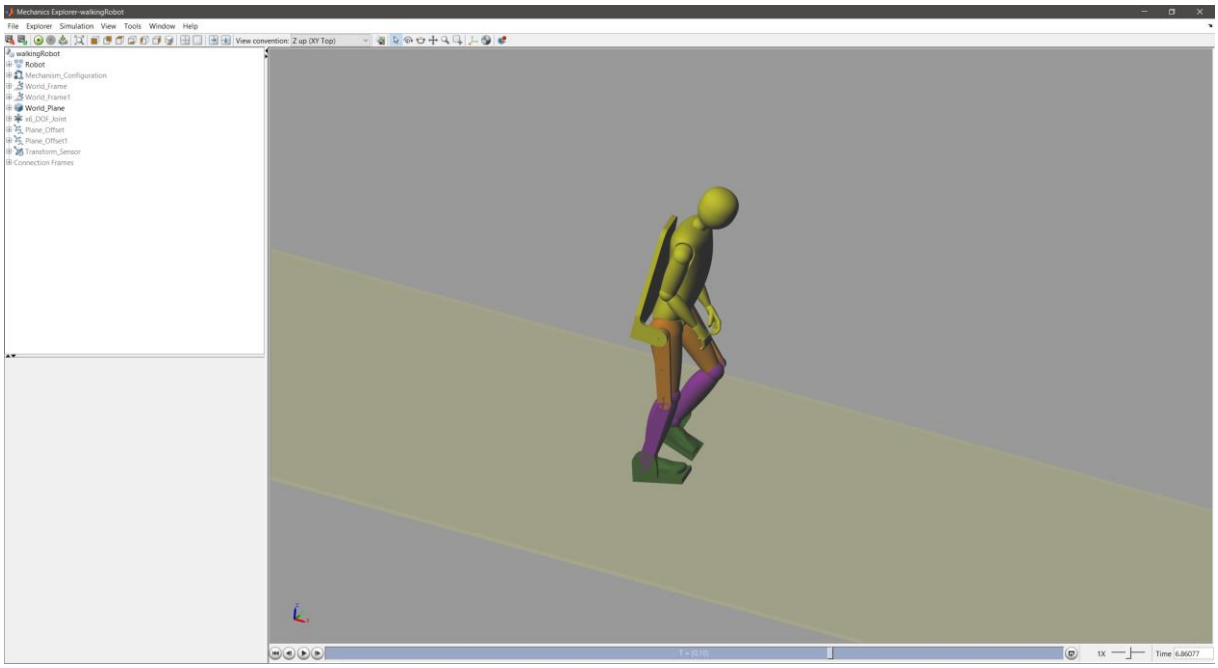


Figure C4: Result 2

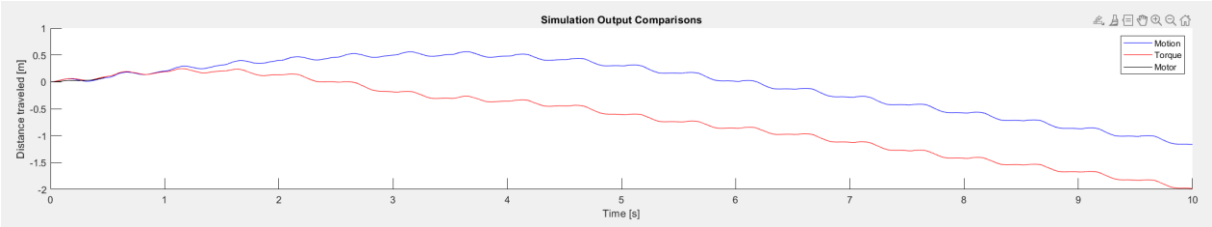


Figure C5: Result 2 Distance Travelled

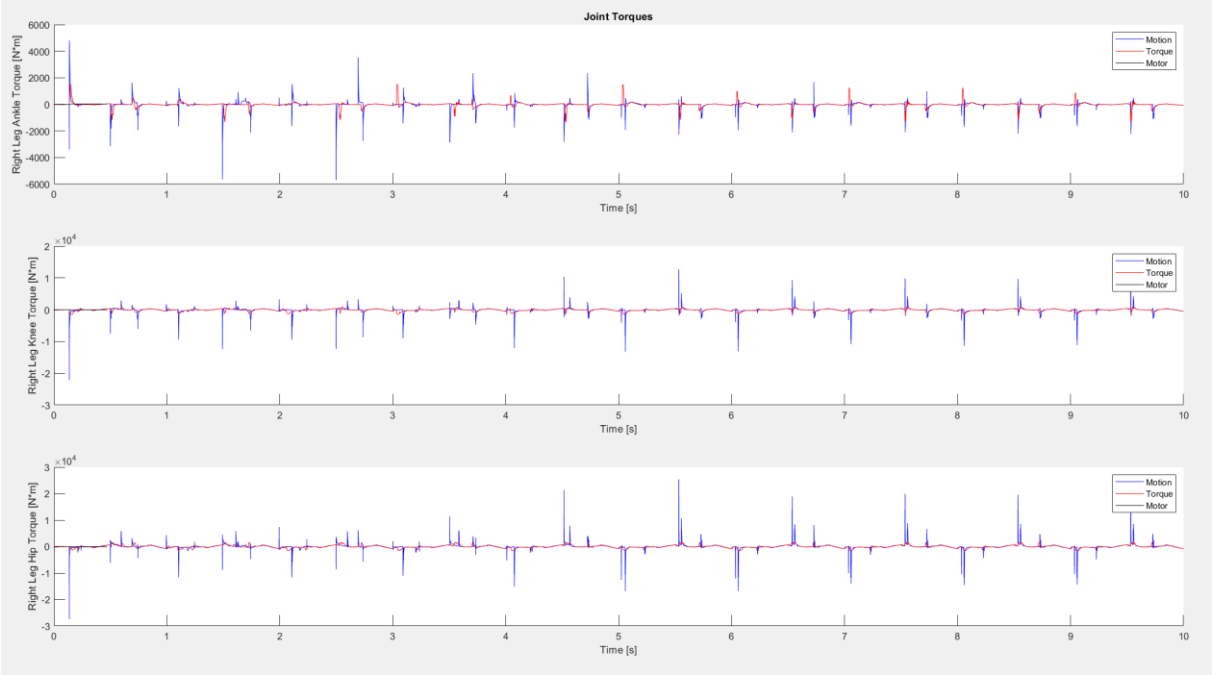


Figure C6: Result 2 Joint Torques

Result 3: Running gait file 1500Nm.mat on Version2_Large_Feet folder (Robot with large feet) with torque limited to 1500Nm.

Comment: Large feet are helping with stability. Robot is able to progress forward without falling and cover a big distance but with very large torques.

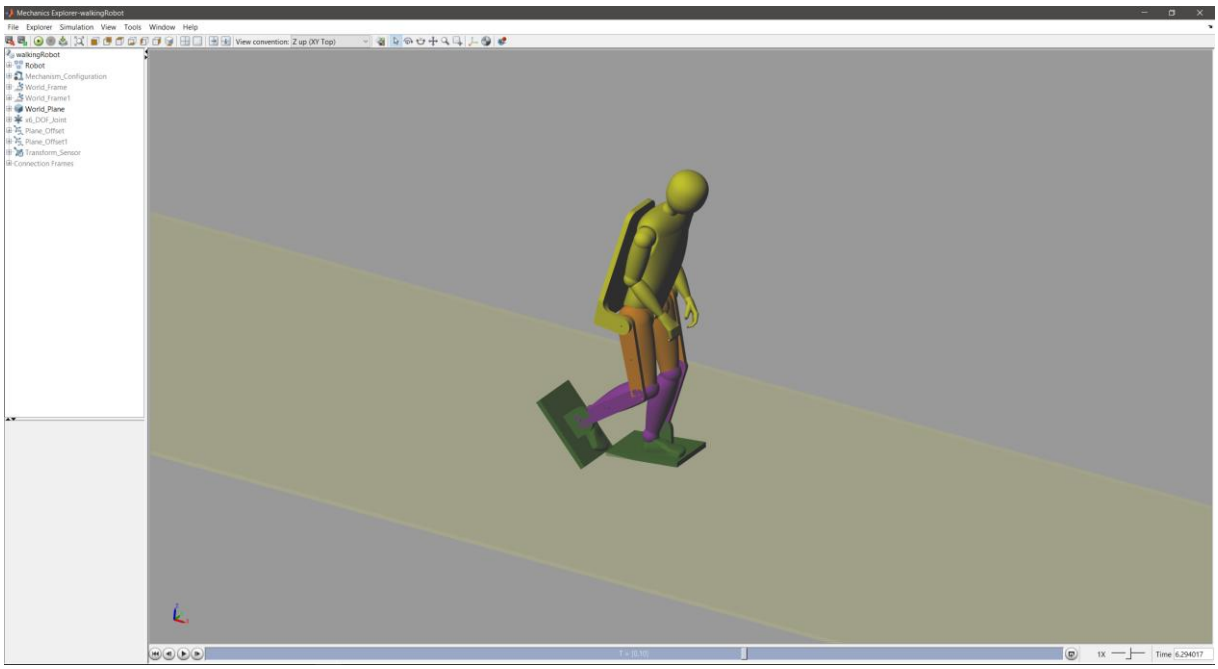


Figure C7: Result 3

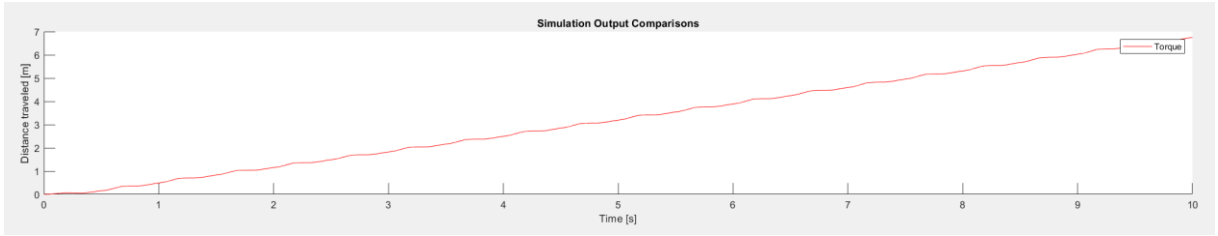


Figure C8: Result 3 Distance Travelled

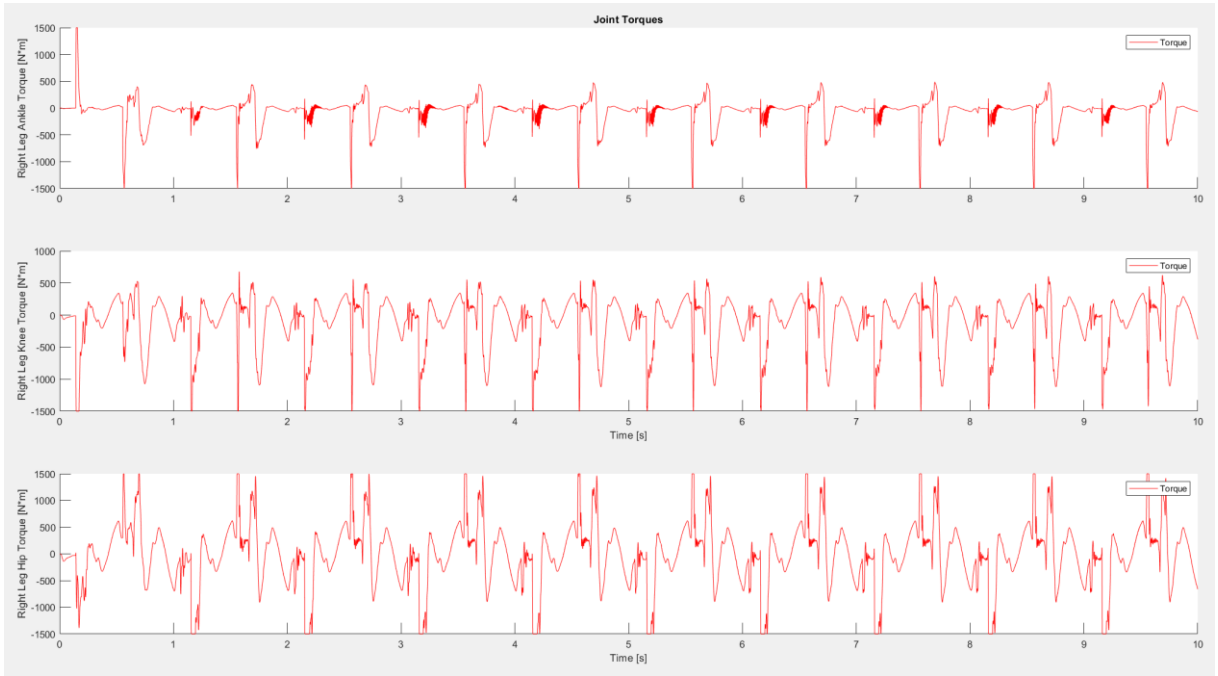


Figure C9: Result 3 Joint Torques

Result 4: Running gait file 1000Nm.mat on Version2_Large_Feet folder (Robot with large feet) with torque limited to 1000Nm.

Comment: Robot is able to progress forward without falling and cover a similar distance to the simulation in Result 3 but with two thirds of the torque available.

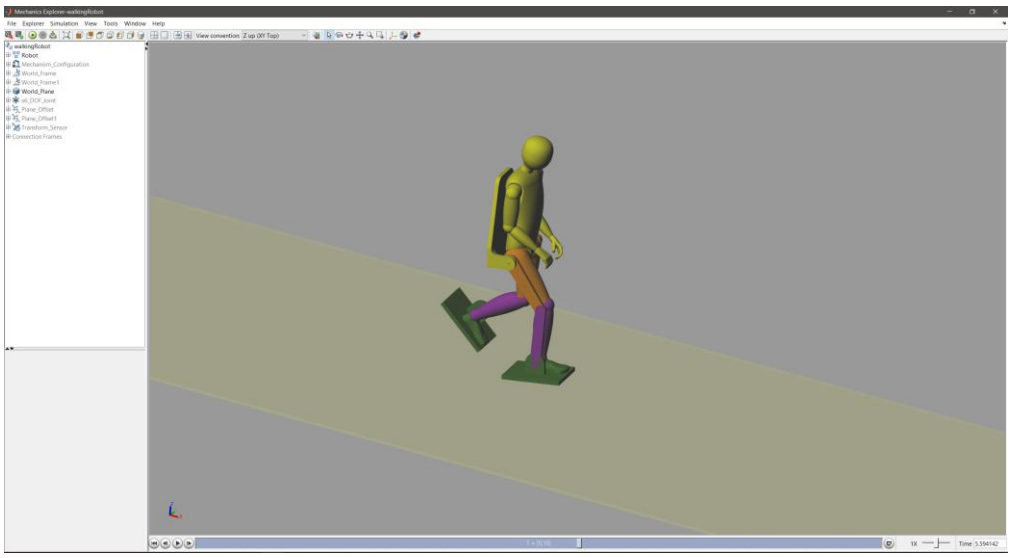


Figure C10: Result 4

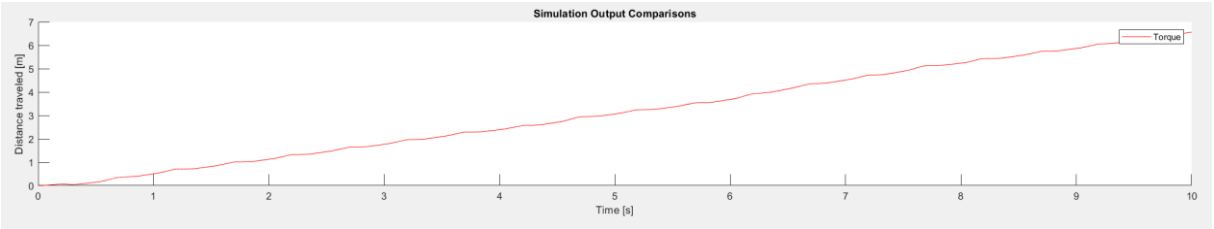


Figure C11: Result 4 Distance Travelled

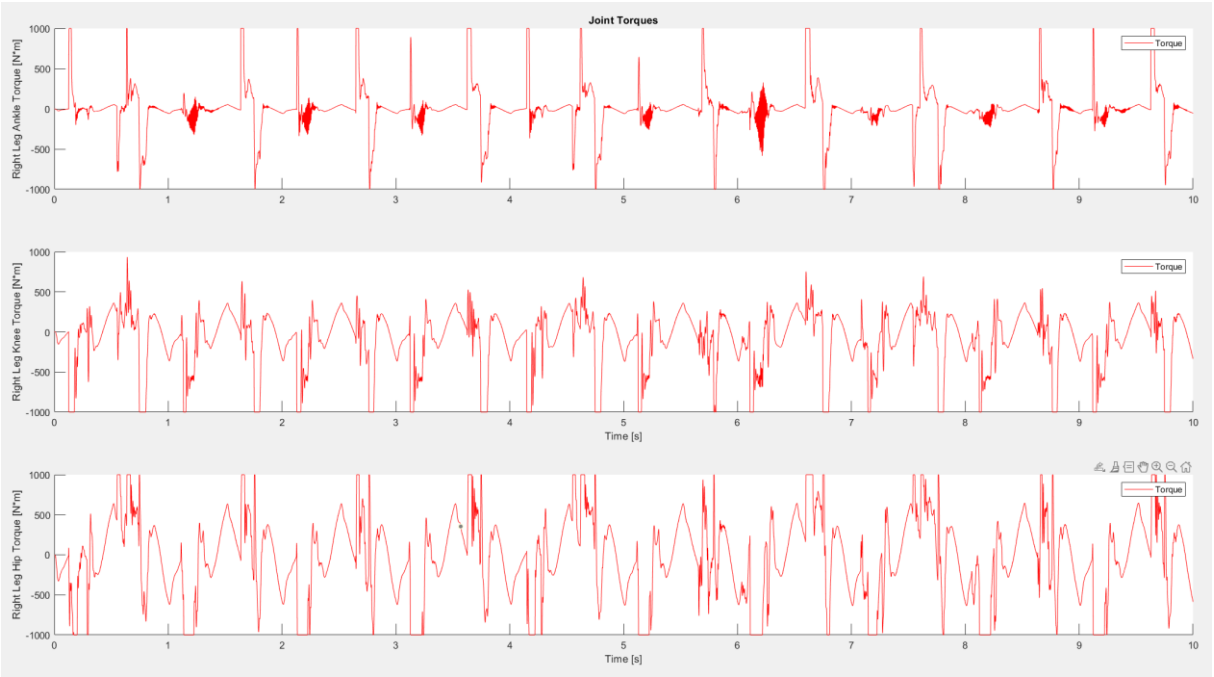


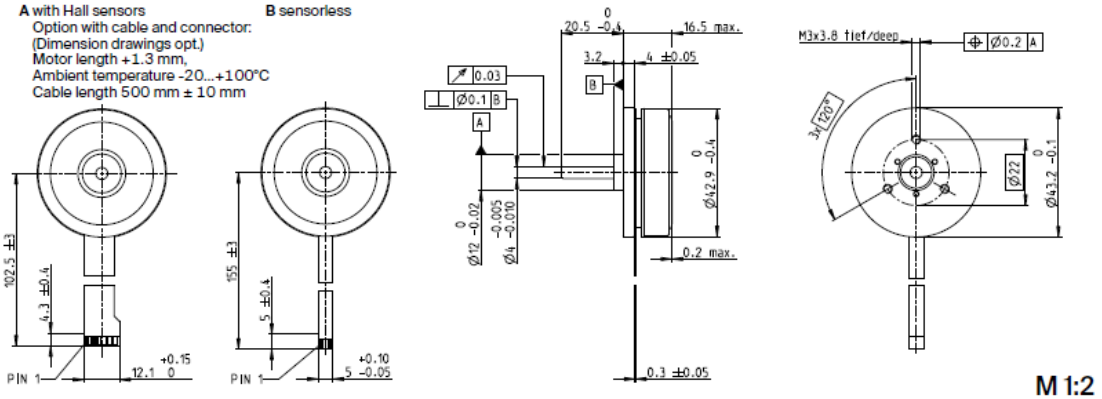
Figure C12: Result 4 Joint Torques

APPENDIX D

HARDWARE DATA SHEETS

EC 45 flat ∅42.9 mm, brushless, 30 watt

A with Hall sensors
B sensorless
Option with cable and connector:
(Dimension drawings opt.)
Motor length +1.3 mm,
Ambient temperature -20...+100°C
Cable length 500 mm ± 10 mm



M 1:2

Stock program
Standard program
Special program (on request)

	Part Numbers							
A with Hall sensors	200142		339281		339282			
Option with Cable and Connector	668555		668556		668557			
B sensorless		200189		339283		339284		
Motor Data								
Values at nominal voltage								
1 Nominal voltage V	12	12	24	24	36	36		
2 No load speed rpm	4370	4350	4360	4380	4750	4760		
3 No load current mA	163	163	81.4	73	61.6	55.3		
4 Nominal speed rpm	2940	2800	2940	2900	3290	3270		
5 Nominal torque (max. continuous torque) mNm	55	54.7	54.8	55.2	66	66.6		
6 Nominal current (max. continuous current) A	2.02	2.02	1.01	1.01	0.847	0.849		
7 Stall torque[1] mNm	255	219	253	243	380	369		
8 Stall current A	10	8.58	4.97	4.77	5.38	5.22		
9 Max. efficiency %	76	75	76	77	80	81		
Characteristics								
10 Terminal resistance phase to phase Ω	1.2	1.4	4.83	5.03	6.69	6.89		
11 Terminal inductance phase to phase mH	0.56	0.56	2.24	2.24	4.29	4.29		
12 Torque constant mNm/A	25.5	25.5	51	51	70.6	70.6		
13 Speed constant rpm/V	374	374	187	187	135	135		
14 Speed/torque gradient rpm/mNm	17.6	20.5	17.7	18.5	12.8	13.2		
15 Mechanical time constant ms	17.1	19.9	17.2	17.9	12.4	12.8		
16 Rotor inertia gcm²	92.5	92.5	92.5	92.5	92.5	92.5		

Specifications

Thermal data

17	Thermal resistance housing-ambient	6.69 K/W
18	Thermal resistance winding-housing	3.92 K/W
19	Thermal time constant winding	11.4 s
20	Thermal time constant motor	295 s
21	Ambient temperature	-40...+100°C
22	Max. winding temperature	+125°C

Mechanical data (preloaded ball bearings)

23	Max. speed	10000 rpm
24	Axial play at axial load < 5.0 N	0 mm
	> 5.0 N	typ. 0.14 mm
25	Radial play	preloaded
26	Max. axial load (dynamic)	4.8 N
27	Max. force for press fits (static) (static, shaft supported)	53 N 1000 N
28	Max. radial load, 5 mm from flange	18 N

Other specifications

29	Number of pole pairs	8
30	Number of phases	3
31	Weight of motor	75 g

Values listed in the table are nominal.

Connection	with Hall sensors	sensorless
Pin 1	V_{Hall} 3.5...24 VDC	Motor winding 1
Pin 2	Hall sensor 3*	Motor winding 2
Pin 3	Hall sensor 1*	Motor winding 3
Pin 4	Hall sensor 2*	⅄ neutral point
Pin 5	GND	
Pin 6	Motor winding 3	
Pin 7	Motor winding 2	
Pin 8	Motor winding 1	

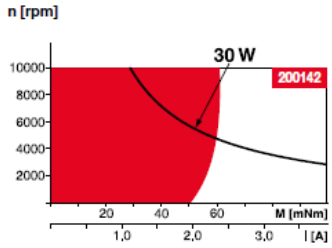
*Internal pull-up (7...13 kΩ) on V_{Hall}
Wiring diagram for Hall sensors see p. 59

Adapter	Part number	Part number
see p. 529	220300	220310
Connector	**Part number**	**Part number**
TE	1-84953-1	84953-4
Molex	52207-1133	52207-0433

Pin for design with Hall sensors:
FPC, 11-pol, Pitch 1.0 mm, top contact style

[1]Calculation does not include saturation effect (p. 71/178)

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

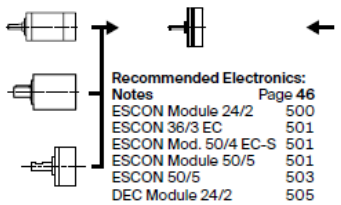
maxon Modular System

Details on catalog page 46

Planetary Gearhead
∅32 mm
0.75 - 6.0 Nm
Page 394/398

Planetary Gearhead
∅42 mm
3.0 - 15.0 Nm
Page 407

Spur Gearhead
∅45 mm
0.5 - 2.0 Nm
Page 409

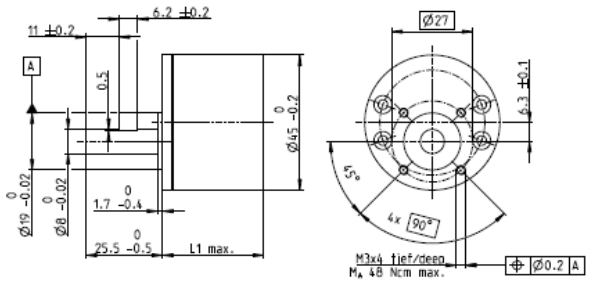


Recommended Electronics:

Notes	Page 46
ESCON Module 24/2	500
ESCON 36/3 EC	501
ESCON Mod. 50/4 EC-S	501
ESCON Module 50/5	501
ESCON 50/5	503
DEC Module 24/2	505
DEC Module 50/5	505
EPOS4 Micro 24/5	509
EPOS4 Mod./Comp. 24/1.5	510
EPOS4 Mod./Comp. 50/5	510
EPOS4 Comp. 24/5 3-axes	511
EPOS4 50/5	515
EPOS4 Disk 60/8	516
EPOS2 P 24/5	520

for motor type A:
Encoder MILE
256 - 2048 CPT,
2 channels
Page 460

Spur Gearhead GS 45 A ∅45 mm, 0.5–2.0 Nm



M 1:2

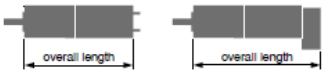
Technical Data	
Spur Gearhead	straight teeth
Output shaft	stainless steel, hardened
Bearing at output	ball bearing
Radial play, 10 mm from flange	max. 0.15 mm
Axial play	0.02–0.2 mm
Max. axial load (dynamic)	60 N
Max. force for press fits	60 N
Max. continuous input speed	6000 rpm
Recommended temperature range	-15...+80°C
Number of stages 2 3 4 5 6	
Max. radial load, 10 mm from flange 120 N 180 N 190 N 190 N 190 N	

gear

Stock program
Standard program
Special program (on request)

Part Numbers	678432	678440	678527	678531	678539
Gearhead Data					
1 Reduction	5:1	18:1	61:1	212:1	732:1
2 Absolute reduction	51/10	459/26	20655/338	125862/595	492790/673
10 Mass inertia gcm²	3.7	1.6	1.0	0.8	0.8
3 Max. motor shaft diameter mm	3	3	3	3	3
Part Numbers	678433	678438	678528	678532	678540
1 Reduction	7:1	26:1	89:1	310:1	1072:1
2 Absolute reduction	209/28	9405/364	66632/745	183281/592	307572/287
10 Mass inertia gcm²	3.1	1.4	1.0	0.8	0.8
3 Max. motor shaft diameter mm	3	3	3	3	3
Part Numbers	678434	678436	678529	678533	678541
1 Reduction	9:1	32:1	111:1	385:1	1334:1
2 Absolute reduction	2295/247	8523/265	334/3	173808/451	198769/149
10 Mass inertia gcm²	2.1	1.4	0.6	0.5	0.4
3 Max. motor shaft diameter mm	3	3	3	3	3
Part Numbers	678435	678437	678530	678536	678542
1 Reduction	14:1	47:1	163:1	564:1	1952:1
2 Absolute reduction	2475/182	6221/132	141157/861	161880/287	1929023/988
10 Mass inertia gcm²	2.2	0.9	0.5	0.5	0.4
3 Max. motor shaft diameter mm	3	3	3	3	3
4 Number of stages	2	3	4	5	6
5 Max. continuous torque Nm	0.5	2.0	2.0	2.0	2.0
6 Max. intermittent torque at gear output Nm	0.75	2.5	2.5	2.5	2.5
12 Direction of rotation, drive to output	=	≠	=	≠	=
7 Max. efficiency %	87	76	66	59	53
8 Weight g	113	113	125	140	149
9 Average backlash no load °	1.6	2.0	2.4	2.8	3.2
11 Gearhead length L1* mm	24.2	24.2	26.9	30.4	33.8

*for EC 45 flat, IE, L1 is max. + 4.0 mm



maxon Modular System				Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts				
+ Motor	Page	+ Sensor/Brake	Page					
EC 45 flat, 30 W	295			40.7	40.7	43.4	46.9	50.3
EC 45 flat, 30 W, cable	295			42.0	42.0	44.7	48.2	51.6
EC 45 flat, 30 W	295	MILE	460	42.9	42.9	45.6	49.1	52.5
EC 45 flat, 50 W	296			46.3	46.3	49.0	52.5	55.9
EC 45 flat, 50 W	296	MILE	460	47.1	47.1	49.8	53.3	56.7
EC 45 flat, 60 W	297			46.3	46.3	49.0	52.5	55.9
EC 45 flat, 60 W	297	MILE	460	47.1	47.1	49.8	53.3	56.7
EC 45 flat, 90 W	298			52.3	52.3	55.0	58.5	61.9
EC 45 flat, 90 W	298	MILE	460	53.1	53.1	55.8	59.3	62.7
EC 45 flat, 70 W	299			51.3	51.3	54.0	57.5	60.9
EC 45 flat, 70 W	299	MILE	460	52.1	52.1	54.8	58.3	61.7
EC 45 flat, 80 W	300			51.3	51.3	54.0	57.5	60.9
EC 45 flat, 80 W	300	MILE	460	52.1	52.1	54.8	58.3	61.7
EC 45 flat, 120 W	301			57.3	57.3	60.0	63.5	66.9
EC 45 flat, 120 W	301	MILE	460	58.1	58.1	60.8	64.3	67.7
EC 45 flat, IE, IP 00	302			59.9	59.9	62.6	66.1	69.5
EC 45 flat, IE, IP 40	302			62.1	62.1	64.8	68.3	71.7
EC 45 flat, IE, IP 00	303			64.9	64.9	67.6	71.1	74.5
EC 45 flat, IE, IP 40	303			67.1	67.1	69.8	73.3	76.7

Short instructions
Original: de

Version 1.0.0

Nanotec Electronic GmbH & Co. KG
Kapellenstraße 6
85622 Feldkirchen, Germany

Phone: +49 (0)89-900 686-0
Fax: +49 (0)89-900 686-50
info@nanotec.de

Introduction

The *N5* is a controller for the *open loop* or *closed loop* operation of stepper motors and the *closed loop* operation of BLDC motors.

This document describes the installation and commissioning of the controller.You can find the detailed documentation for the product on the Nanotec website **us.nanotec.com**. The short instructions do not replace the technical manual oft he product.

Copyright, marking and contact

CE

Intended use

The *N5 controller* is used to control stepper and BLDC motors and is designed for use under the approved **Environmental conditions**.

Any other use is considered unintended use.

Note

Changes or modification to the controller are not permitted.

Warranty and disclaimer

Nanotec produces component parts that are used in a wide range of industrial applications. The selection and use of Nanotec products is the responsibility of the system engineer and end user. Nanotec accepts no responsibility for the integration of the products in the end system.

Under no circumstances may a Nanotec product be integrated as a safety controller in a product or construction. All products containing a component part manufactured by Nanotec must, upon delivery to the end user, be provided with corresponding warning notices and instructions for safe use and safe operation. All warning notices provided by Nanotec must be passed on directly to the end user.

Our general terms and conditions apply: **en.nanotec.com/service/general-terms-and-conditions/**.

Specialist staff

Only specialists may install, program and commission the device:

- Persons who have appropriate training and experience in work with motors and their control.
- Persons who are familiar with and understand the content of this technical manual.
- Persons who know the applicable regulations.

EU directives for product safety

The following EU directives were observed:

- RoHS directive (2011/65/EU, 2015/863/EU)
- EMC directive (2014/30/EU)

Other applicable regulations

In addition to this technical manual, the following regulations are to be observed:

- Accident-prevention regulations
- Local regulations on occupational safety

Safety and warning notices

Note

- Damage to the controller.
- Changing the wiring during operation may damage the controller.
- Only change the wiring in a de-energized state. After switching off, wait until the capacitors have discharged.

Note

- Fault of the controller due to excitation voltage of the motor.
- Voltage peaks during operation may damage the controller.
- Install suitable circuits (e.g., charging capacitor) that reduce voltage peaks.

Note

- There is no polarity reversal protection.
- Polarity reversal results in a short-circuit between supply voltage and GND (earth) via the power diode.
- Install a line protection device (fuse) in the supply line.

Note

- The device contains components that are sensitive to electrostatic discharge.
- Improper handling can damage the device.
- Observe the basic principles of ESD protection when handling the device.

Technical details and pin assignment

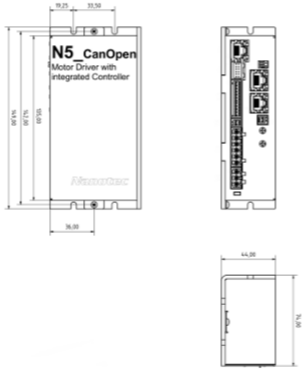
Environmental conditions

Environmental condition	Value
Protection class	IP20
Ambient temperature (operation)	-10 … +40°C
Air humidity (non-condensing)	0 … 95 %
Altitude of site above *sea level* (without drop in performance)	1500 m
Ambient temperature (storage)	-25 … +85°C

Electrical properties and technical data

Property	Description / value
Operating voltage	• 12 V-5% …72 V +4% DC for *low-current version* with designation N5-1-2 • 12 V - 48 V +/-5% DC for the *high-current version* with designation N5-2-2 and up to **hardware version** w007 • 12 V -5% …57.4 V DC for the *high-current version* with designation N5-2-2 and from **hardware version** w007b
Rated current	N5-1-2 (*low current*): 10 A_{rms} N5-2-2 (*high current*): 18 A_{rms}
Peak current	N5-1-2 (*low current*): 10 A_{rms} N5-2-2 (*high current*): 40 A_{rms} for 5 seconds
Commutation	Stepper motor – open loop, stepper motor – closed loop with encoder, BLDC motor – closed loop with Hall sensor, and BLDC motor – closed loop with encoder
Operating modes	*Profile Position Mode, Profile Velocity Mode, Profile Torque Mode, Velocity Mode, Homing Mode, Interpolated Position Mode, Cyclic Sync Position Mode, Cyclic Sync Velocity Mode, Cyclic Synchronous Torque Mode, Clock-Direction Mode*
Set value setting / programming	*CANopen, Ethernet (REST with the NanoIP user interface), clock-direction, analog, NanoJ program*
Interfaces	CANopen, Ethernet
Inputs	• 4 inputs, 5 V/24 V (inputs 1 to 4) individually switchable by means of software, factory setting: 5 V • 2 inputs, wide range 5-24 V (inputs 5 and 6); • 2 analog inputs -10 to +10 V or 0–20 mA (switchable by means of software)
Outputs	2 outputs, (open drain, 0 switching, max. 24 V and 500 mA)
Encoder input	5 V or 24 V signal, differential or single-ended (switchable by means of software), max. resolution 65536 increments per revolution (16-bit)
Protection circuit	Overvoltage and undervoltage protection Overtemperature protection (> 75° Celsius on the power board) Polarity reversal protection: In the event of a polarity reversal, a short-circuit will occur between supply voltage and GND over a power diode; a line protection device (fuse) is therefore necessary in the supply line. The values of the fuse are dependent on the application and must be dimensioned • greater than the maximum current consumption of the controller • less than the maximum current of the voltage supply. If the fuse value is very close to the maximum current consumption of the controller, a medium / slow tripping characteristics should be used.

Dimensioned drawings



Overtemperature protection

Above a temperature of approx. 75°C on the power board (corresponds to 65–72°C outside on the cover), the power part of the controller switches off and the error bit is set . After cooling down and confirming the error , the controller again functions normally.

LED signaling

Power LED

Normal operation

In normal operation, the green power LED L1 flashes briefly once per second.

Case of an error

If an error has occurred, the LED turns red and signals an error number.

The following table shows the meaning of the error numbers.

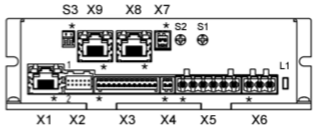
Flash rate	Error
1	General
2	Voltage
3	Temperature
4	Overcurrent
5	Controller
6	Watchdog-Reset

Note

For each error that occurs, a more precise error code is stored in object **1003_h**.

Pin assignment

Pin 1 is marked with an asterisk "*".



Connector	Function	Pin assignment / description
X1	Ethernet	Configuration interface
X2	Encoder and Hall sensor 5 V / 24 V DC signal Max. 1 MHz Switching thresholds: **5 V (factory setting)**: On: >3.8 V; Off: <0.26 V **24 V**: On: >14.42 V; Off: <4.16 V	1. GND 2. Vcc: +5 V (factory setting) /24 V DC output, switchable with object 2059_h 3. A 4. B 5. A\ 6. B\ 7. I 8. I\ 9. Hall 1 10. Hall 2 11. Hall 3 12. Shielding
X3	Inputs and outputs Switching thresholds for digital **inputs 1 - 4**: **5 V (factory setting)**: On: >3.8 V; Off: <0.26 V **24 V**: On: >14.42 V; Off: <4.16 V Switching thresholds for digital **inputs 5 - 6**: On: >3.25 V; Off: <2 V	1. **GND** 2. **Digital input 1**: 5 V / 24 V Signal, switchable with object **3240_h** 3. **Digital input 2**: 5 V / 24 V Signal, switchable with object **3240_h** 4. **Digital input 3**: 5 V / 24 V, switchable with object **3240_h**, max. 1 MHz; *direction input* in clock/direction mode 5. **Digital input 4**: 5 V / 24 V, switchable with object **3240_h**, max. 1 MHz; *clock input* in clock/direction mode 6. **Digital input 5**: 5…24 V signal, not switchable 7. **Digital input 6**: 5…24 V signal, not switchable 8. **Analog input 1**: 10 Bit, 0-10 V oder 0-20 mA, switchable with object **3221_h** 9. **Analog input 1**: 10 Bit, 0-10 V oder 0-20 mA, switchable with object **3221_h** 10. **Digital output 1**: Open drain, max 24 V/500 mA 11. **Digital output 2**: Open drain, max 24 V/500 mA 12. **Shielding**
X4	Brake 24V Brakes have to be connected using an appropriate circuit if **+UB**>24 V!	1. **Brake+**: internally connected to **+UB** 2. **Brake -**: PWM-controlled open-drain output, max 1.5 A