A Low-Cost Semi-Active Orthosis and Neural Network Based Prediction System for Alleviating Gait Degradation

Tom Slucock

Thesis Submitted to the University of Kent for the Degree of Doctor of Philosophy in Electronic Engineering

University of Kent

School of Engineering

19 August 2024

Acknowledgements

I would like to give thanks to Kostas, Gareth, and Sanaul for acting as supervisors during these somewhat chaotic three and a half years of Global Pandemics, Cyberattacks, Funding Issues, Career Changes, and occasional struggles with Motivation, as well as to the entirety of Jennison and The Shed's Support Staff for putting up with my occasional silly questions and helping me with technical issues and exoskeleton construction. I'd additionally like to thank Mohammahdi Sarachi, Rania Kolghassi, Nathan Brabon, and all of my Exoskeleton Test Subjects for their contributions to the research project over its lifetime.

Throughout my almost 8 years at the University of Kent as both an Undergraduate Student and Postgraduate Researcher I am happy to have benefitted from the resources and knowledge I have been provided, and the aid offered to me. Even when opposed by unfortunate external circumstances such as global pandemics and cyberattacks. Hopefully I've helped impart some of that knowledge to others in the Demonstrator work I've been able to perform and the people I met whilst doing so.

And, finally, to my parents, for the lifelong support they have provided me and the opportunities I have had as a result. I can only attempt to live up to such good fortune to by attempting to surpass you.

Included Academic Work

This Study and Thesis has also resulted in the following additional Journal Article publications, some chapters within this thesis include work from these papers. The Author of this research is in all instances the Primary Author of these papers:

- **Slucock T**. A Systematic Review of Low-Cost Actuator Implementations for Lower-Limb Exoskeletons: A Technical and Financial Perspective. Journal of Intelligent and Robotic Systems. 2022;106(1):3
- **Slucock T**, et al. Development of a Microcontroller-Based Recurrent Neural Network Predictive System for Lower Limb Exoskeletons. Journal of Intelligent and Robotic Systems. 2025;111:32

Other Declarations

Ethics Approval: Consent was given for the collection of Gait data in order to create predictive models. Ethics approval gained for collecting training data from participants. [CREAG012-11-22].

Contents

List of Figures

List of Tables

List of Equations

List of Acronyms

Acronym	Description
API	Application Programming Interface
CNN	Convolutional Neural Network
CoM	Centre of Mass
CoR	Centre of Rotation
CP	Cerebral Palsy
DoF	Degree of Freedom
EEG	Electroencephalography
EMG	Electromyography
FSR	Force Sensitive Resistor
GRNN	General Regression Neural Network
GRU	Gated Recurrent Unit
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
LiPo	Lithium Polymer
LSTM	Long Short-Term Memory
MAE	Mean Average Error
MEMS	Microelectromechanical Systems
MLP	Multi-Layered Perceptron
NN	Neural Network
PD	Proportional Derivative
PETG	Polyethylene Terephthalate Glycol
PI	Proportional Integral
PID	Proportional Integral Derivative
PLA	Polylactic Acid
PRISMA	Preferred Reporting Items for Systematic reviews and Meta-Analyses.
RNN	Recurrent Neural Network
RTOS	Real-Time Operating System
SCI	Spinal Cord Injury
SEA	Series Elastic Actuator
TBI	Traumatic Brain Injury
TNN	Transformer Neural Network

Abstract

Exoskeletons are one method by which a patient's ability to walk can be maintained or even rehabilitated in the face of pathologic gait disorders, however in many cases more advanced, active exoskeletons suffer from being prohibitively expensive for the average user. The muscular imbalances caused by Cerebral Palsy, Strokes, and such conditions can cause excessive strain upon musculature and bone over time, and considerably increase the risk of osteoporosis, pathological gait issues, and stunted growth. As a result, sufferers of such conditions may begin expressing pathologic gait disorders or lose the ability to walk at all should these issues continue. The cost of exoskeleton-based treatment however imposes great strain on a medical system to aid in funding these devices for affected individuals or limits the time such individuals have to access these devices, such as within set "rehabilitation sessions". As neither of these cases are ideal, there is an opportunity to explore methodologies that seek to reduce the financial impact of these devices. This could be achieved by creating a Low-Cost Orthosis that may aid in reducing gait degradation in between such rehabilitation sessions whilst remaining as inexpensive as is feasible without seriously reducing the device's effectiveness.

To this end, this Thesis seeks to propose and test the effectiveness of a Low-Cost Exoskeleton Methodology. It will in so doing examine the structure, actuation methods, and control systems of other Exoskeleton Implementations for potentially innovative solutions to maintain effectiveness even within considerable limitations. This Low-Cost Methodology consists of analysing the cost of exoskeleton components from a perspective of not only Financial cost, but also of Power Consumption, Weight, and the Effectiveness of key exoskeleton components such as the actuator, sensor, and control system.

This study then puts this methodology into practice by outlining the creation of a simple exoskeleton design that follows the principles of the Low-Cost Methodology. This exoskeleton makes use of a rotary electric motor driven series elastic actuator and is controlled by an embedded microcontroller implemented Neural Network capable of real-time predictions of the wearer's knee angle into the future with a level of accuracy equivalent to that of models on far more powerful hardware. The exoskeleton itself will be capable of reliably following and pre-empting user motion that the model had been trained to recognise. The impact of different Neural Network types, such as Recurrent and Convolutional Neural Networks, and factors such as input size, model size, training data quality and quantity, and depth of prediction were considered and reviewed in the creation of the prediction system.

The Resulting prediction system that was developed was capable of running on inexpensive, Low-Cost, Low-Impact Microcontrollers. This system was capable of predicting the wearer's knee angle movements 150ms into the future, with a Mean Average Error (MAE) from Reality of 2.5%. It had a maximum potential prediction depth of one second into the future.

Chapter 1: Introduction

Exoskeletons provide support to the human body, reducing strain when performing otherwise intensive tasks by aiding in movement. These tasks may be naturally intensive, such as industrial labour or construction work, in which the exoskeleton would, for example, allow the user to more easily lift and transport heavy objects whilst supporting the back, legs, and arms to provide additional strength to the user. Alternatively, it may be that the tasks are intensive as a result of a disability or injury that reduces the user's natural abilities and are where medical exoskeleton technologies can prove useful for aiding a wearer in performing these tasks. For example, a user suffering from injuries or disabilities such as a Spinal Cord Injury (SCI), who may be Quadriplegic (some to all loss of muscle use in 4 limbs and torso) or Paraplegic (some to all loss of muscle use from the lower limbs), may be unable to walk and be consigned to a wheelchair. A Lower body exoskeleton is a method to improve the user's independence and ability to perform basic tasks unassisted, allowing such individuals to perform limited ambulation without the use of a wheelchair, it has also been cited as improving psychological conditions according to the Beck Depression Inventory scale [1], as well as self-perception and body uneasiness.

Exoskeletons can be aimed to assist in either the upper body, lower body, or both, with lower body being the focus of the thesis. Lower-Limb Exoskeletons within the medical field may aid in reducing the energetic costs of walking, providing rehabilitation for pathologic walking gaits, and providing a general independence for a disabled user by allowing them to walk and interact with less human assistance.

The most common exoskeleton subcategories seen within literature are Active, Passive, or a Mixture of both depending on their actuation method. Passive Exoskeletons are designed to reduce strain and load upon the legs without the use of any powered actuation, instead more often using tension or elastic. Active Exoskeletons use powered actuation which can be provided by electric motors, pneumatics, levers, hydraulics, cables, etc. Naturally, exoskeletons may also include both, such as via a Series Elastic Actuator [2]. Similarly exoskeletons will be considered either "Hard" or "Soft" depending on their construction method, with many exoskeletons making use of a solid frame to support actuators and drive legs and so being a "Hard" Exoskeleton, whereas a "Soft" exoskeleton makes use of cabling and elastic material to wrap around a user's limbs, allowing for more free range of movement although with the cost of not being as supportive for those with limited to no muscle strength.

In General, disabilities that would benefit from an exoskeleton often become more common in elderly patients. Due to the UK's (and many other developed countries) rising proportion of people over the age of 65, such disabilities will continue to become a growing burden on medical and human resources. Exoskeletons allow for a way to reduce the human resource burden by providing semi-automated, continuous rehabilitative efforts similar or superior to those of a human therapist without such being present. The Exoskeleton here fulfils the assistive role, by making activities easier for its users to perform and reducing the workload of already existing human assistants.

However, for lifelong conditions such as Cerebral Palsy or curable effects of conditions such as brain damage, the Exoskeleton fulfils a more rehabilitative role, and if used early can help to correct a faulty gait pattern that would lead to muscle strain and permanent damage later in life [3]. Exoskeletons have the potential to be more useful in this field than regular therapy due to being able to provide consistent assistance with or without constant professional therapist supervision. As such, this thesis seeks to document the methodology behind developing an inexpensive exoskeleton implementation designed for such consistent, semi-automated assistance.

1.1 Overview

While the idea of Humanoid Robotics has existed in one form or another for centuries [4], the original concept of an exoskeleton was first developed in 1890 by Nicholas Yagn [5]. This consisted of a simple passive spring connecting along the leg, designed to store energy during flexion and release it during extension. It would however take until the late 1960's before actual powered devices would be built, the first examples of these seen in 1969 with the Mihajlo Pupin Institute in Belgrade, Serbia, developing the "Kinematic Walker" [6] designed for use with disabled wearers. This consisted of a 2 Degree of Freedom (DoF's) active exoskeleton with actuated hip joint and passive ankle joint, with the knee being kept locked, and was further developed on in later years, with a 3 DOF system known as the "Partial Exoskeleton". In 1971 General Electrics in the USA would complete the "Hardiman I" [7], a full body exoskeleton designed to enhance the user's strength. This prototype had a considerable weight in excess of 650kg, but could lift an equal weight in turn, although it had a limited walking capacity.

The early 2000's saw several more advanced exoskeleton designs arising in the USA as a result of DARPA's Exoskeleton program. One of the most well-known exoskeletons that was developed during this time was the "Berkeley Lower Extremity Exoskeleton" (BLEEX) in 2004 [8] [9], which made use of linear hydraulic actuators enhancing the user's strength and providing 7 Degrees of Freedom throughout the leg. Another example developed in 2002 was HAL [10], developed by the Japanese company Cyberdyne, which made use of EMG (Electromyography) sensors to detect the nerve signals of the wearer as was designed to allow paralysed patients to walk.

Within the modern day the exoskeleton industry has grown to encompass a far wider variety of potential implementations, from augmentative designs seen within the military and industry to rehabilitative or assistive designs seen within healthcare.

1.1.1 Current Limitations

As of 2022, the worldwide exoskeleton market size has a total value of roughly $334.5 Million, with the majority of this market held by Mobile, Powered Lower-Limb/Body Exoskeleton technologies within the field of Healthcare [11]. Despite this expansion, exoskeletons remain outside the realm of feasibility for much of the average populace, for example, medical spending in the UK was estimated in 2019 to be ~£2,347 per person [12] and up to an estimated £4,192 per person in 2022 [13], an increase likely as a result of COVID-19. The costs of well-known exoskeletons meanwhile such as the EksoNR are upwards of £126,000 [14], which is infeasible for many without external funding. Even for the developed country with the highest medical spending per capita, the United States, with ~£10,356 ($12,914) [15] per person the average exoskeleton cost far exceeds the average spending. For developing countries such as India with considerably lower medical spending and individual wealth, acquiring such equipment is effectively impossible for the average person without external funding. Beyond this initial investment cost comes potential future maintenance costs which would vary depending on the reliability of the exoskeleton's components, the intensity and frequency of its use, and the environment the exoskeleton is used within. These factors are considered of similar importance to initial cost [16], and whilst difficult to calculate, some estimates of more generic "maintenance contracts" [17] for exoskeleton usage are as much as ~£7,400 a year ($10,000) with a 5 year lifespan for a device of a £110,000 ($150,000) initial cost, in this instance, assuming that after the 5-year lifespan the exoskeleton would need replacement, maintenance would have represented ¼ of the total cost. It is however likely that initial costs would decrease at a more rapid rate than maintenance costs with the development of the exoskeleton industry, as much of the initial cost can be linked to a necessity to recoup the investment of research into the device as opposed to simply material and construction costs.

There are several limitations seen within various exoskeleton examples which can limit their effectiveness. For example, whilst a pre-set gait, which is where the exoskeleton itself drives the walking motion of the user, is useful for disabled patients who have no muscle use whatsoever within their legs it can be restrictive for those with some muscle use and such a system can result in the pre-set gait conflicting with the natural gait of the user, leading to increased muscle strain. An exoskeleton designed for a person with limited muscle usage would instead need to work with the user rather than for them.

Exoskeletons may also have issues with increasing undesired muscle usage if they do not properly allow for movement in line with the capabilities of the average human leg. For example, exoskeletons incapable of significant sideways motion may cause muscle strain on users whose gait naturally involves sideways motion. Additionally, such limitations may make movement as a whole more difficult. This is commonly seen in exoskeletons which may have limited Degrees of Freedom about their joints, either with some Degrees of Freedom not actuated, or not able to move at all, often due to the increased technical complexities that would come from such implementations.

Separate from the exoskeleton itself, many studies involving them are tested in controlled laboratory environments, often making use of specialised rehabilitation equipment such as the Stationary Lokomat Rehabilitative Exoskeleton [18]. Such tests suffer from two significant flaws:

- The Majority of testing occurs in the form of occasional timed sessions, as opposed to continuous study, thereby giving time for the wearer's gait to deteriorate in between sessions.
- The cost and often delicate nature of much of the equipment, the necessity of observers to be present to collect data and make sure there are no incidents, or the equipment's complete immobility, can make it impractical to have any form of continuous rehabilitation or testing outside of these sessions for extended periods of time, rendering the exoskeleton unable to be tested or used in conjunction with an every-day life-like environment [19].

The solution would therefore be the development of an exoskeleton capable of being used within a home environment. This would allow for gait rehabilitation to occur continuously, as opposed to during discrete sessions. Such an exoskeleton would need to meet several key criteria in order to be practical, those being:

- Low Cost (Total expense of components should be kept as low as is practical)
- Resilient and Simple in design (Easy to maintain, resistant to sudden forces such as falling over).
- Comfortable and Light (Able to be worn for long periods of time)
- Power efficient (if requiring power at all, batteries should be kept minimal to keep system light)
- Easy to use and take on/off (able to be used by the average person without supervision)
- One size fits most (exoskeleton should be wearable by a wide variety of individuals)
- Safe (the device presents no significant risk of harming the wearer or those around them)

A Similar set of design criteria are seen in some other papers. [20] for example, uses more granular specifications and additionally considers concepts such as integration with protective equipment and potential ethical considerations such as protecting any data collected by an exoskeleton for use in aiding a patient. [21] and [22] also bring up the reasonable psychological consideration of exoskeleton appearance. More aesthetically pleasing devices capable of being worn under clothing and that do not make the wearer look "robotic" were claimed to be preferred by some patients and may contribute to improved motivation. [23] and [24] also employ similar concepts.

Regardless of the type of exoskeleton, their designs must all consider their interactions with human kinematics (described more in 2.2 and 3.3), whether it be that of a healthy patient or of one that may suffer from gait deficiencies. For this thesis, the exoskeleton that shall be developed will possess the designated purpose of either assistive or rehabilitative work for wearers with limited gait capacity, such as those with Mild Cerebral Palsy or other similar conditions that affect Gait. The exact requirements of which are detailed within the next section.

1.2 Research and Project Scope

Based on the identified limitations of existing exoskeleton designs, this thesis seeks to investigate the feasibility of constructing a functional knee actuator exoskeleton that minimises *Financial*, *Weight*, and *Energy* costs without excessive sacrifice to *Effectiveness*. This exoskeleton could therefore perform a role of reducing gait degradation in between rehabilitation sessions with more advanced exoskeleton designs. This will involve researching how best to define a Low-Cost Methodology, which exoskeleton components from *Actuators*, *Sensors*, and *Control Systems* fit this methodology, and then designing, constructing, and testing an exoskeleton that fits best this Low-Cost Methodology to see whether or not it is capable of meeting both the Aims and Objectives outlined below.

1.2.1 Aims and Objectives

The primary aim of this research project is to address the issue that many rehabilitative/assistive exoskeletons are either prohibitively expensive to be used by the average person, or are designed to be used within laboratory/testing environments as a part of discrete rehabilitation sessions, making them impractical for continual, ongoing rehabilitation within a home environment due to the downtimes between these rehabilitation sessions. This downtime potentially leads to gait degradation as the wearer will be limited to less effective portable solutions such as Knee-Ankle Orthoses. Therefore, this project will therefore see the creation of a relatively inexpensive exoskeleton and governing control system that could function in tandem with more intense rehabilitation sessions which make use of a more advanced design. It will formulate a methodology by which this exoskeleton will be built by that will seek to minimise cost where possible and ascertain whether the resultant exoskeleton is feasibly able to aid a wearer.

Therefore, based upon this aim, a set of objectives can be constructed. Both in the development of the exoskeleton, and the necessary surrounding methodology and research require to do so. Such a device could theoretically support the knee passively to reduce muscle strain whilst also then being able to actively adjust as the wearer moves. The Device itself would be designed to work in conditions found in a home environment and for long periods of time, therefore necessitating low power usage and simplistic design.

To Summarise the Objectives and corresponding research questions of the research project:

- Define a methodology to minimise costs of an Exoskeleton Implementation. *What would be a Low-Cost Methodology for Exoskeletons?*
- Consider how the current State of the Art of Exoskeleton technology and its components within a comprehensive Literature Review, and how they exist in relation to this Methodology. *Which Components are most used within the wider literature? Which would be most appropriate for a Low-Cost Methodology? How does a Low-Cost Methodology fit within the wider Industry?*

- Based off of the State of the Art, design a Knee Exoskeleton, choosing components that fit the Low-Cost methodology and previously mentioned Aims. For example, limiting to off-the-shelf, inexpensive components. *What components are most appropriate for a Low-Cost Implementation?*
- Create a Control System that allows for the function of this Exoskeleton that minimises cost, such as through running entirely off of an inexpensive consumer-grade microcontroller. *What methods exist to minimise loss in effectiveness of control system accuracy due to limited processing power?*
- Test the Effectiveness of the exoskeleton and whether it is capable of reducing wearer strain. *Can the Low-Cost Methodology Successfully create an exoskeleton that does not sacrifice Effectiveness? Does this Low-Cost Implementation provide suitable support to a wearer? Where are its limitations?*
- Based on these results, ascertain the legitimacy of the Low-Cost Methodology either in whole or part. *Is the Low-Cost Methodology Appropriate?*

This thesis will conclude with an overview of these research questions, determining whether the Low-Cost Methodology, and the exoskeleton created under its guidance are an effective exoskeleton implementation.

1.3 Contributions

This research primarily focuses on the development of an inexpensive implementation designed to aid in alleviating gait degradation, the outcomes as a result of this development are summarised below:

- Proposed a Methodology for minimising cost across Financial, Weight, and Power usage metrics.
- Developed a Novel Knee Actuator implementation designed to minimise cost, based off of a comprehensive literature review into the State of the Art of Exoskeleton Components to find the optimal Low-Cost Actuator type. As well as general mathematical theory and kinematic simulations to optimise its placement within an exoskeleton to maximise applied force across a given knee angle range.
- Developed a Microcontroller-based Long Short-Term Memory Neural Network Prediction System implementation capable of predicting knee angles up to 1 second into the future with an error of ~2.5% achieved at 150ms prediction. Results of a similar accuracy to more powerful Laptop Implementations on far inferior and less expensive hardware, far slower sample rates than seen in similar implementations.
- Explored the effectiveness of such models in predicting knee angles at different depths into the future, across a variety of test participants, and at different speeds and styles of walking and other movement states. Observed the effects of deploying such a prediction system to control a physical exoskeleton.

1.4 Thesis Structure

The Structure of the rest of the Thesis is as follows:

- **Chapter 2: Literature Review,** will detail the Low-Cost Methodology and review core exoskeleton components within literature to understand the state of the art of exoskeleton technology. Actuators, Sensors, Control Systems, Power Sources, and Materials will be considered.
- **Chapter 3: Theoretical Development** will cover common concepts within human kinematics, the initial developments of exoskeleton technology including mathematical and computer simulations that would guide the creation of the final exoskeleton design. It will also review the first basic hardware implementation.
- **Chapter 4: Exoskeleton Development,** will document the development process of the Exoskeleton hardware and structural design from basic prototyping to full implementation.
- **Chapter 5: Prediction System Development**, will document the development process of the exoskeleton Neural Network Control System and its implementation into the hardware.
- **Chapter 6: Discussion and Conclusion,** will finalise and review what has been learned from the development of the thesis, and summarise its contributions, flaws, and potential future developments. It also provides links to further reading and addendums.

Chapter 2: Literature Review

2.1 Introduction

Exoskeleton technology requires the understanding of the human that moves the exoskeleton (or is moved by the exoskeleton). It is essential that the innate constraints and natural gait of a human is understood as to prevent an exoskeleton from moving in ways a human cannot, this leading to excess strain and joint damage. Additionally, a proper walking gait is required for effective rehabilitation, to prevent a circumstance where the user is working against the exoskeleton.

The structure of an exoskeleton should ideally be light as is feasible to reduce inertia but also sufficiently sturdy as to support the weight of the wearer, the exoskeleton itself, and the applied torque of its actuators. Similarly, the Actuators should ideally be able to provide sufficient force to aid in the wearer's movement at a low latency as to best align with the wearer's own movements rather than lag behind, but also not consume excessive amounts of power as to limit usage time. In both of these a balance must be found to find effective solutions.

The final aspects of the exoskeleton would be the control system and the sensors which gather the information for it. The control system must consider both the low-level requirements of moving the exoskeleton in such a way that it is ideally capable of balancing itself, even in response to external forces, as well as moving in ways that work with the users' gait. In the long term it may also possess some ability to adjust itself to its particular user in ways that would allow for improved rehabilitation, or to better align a user's current trajectory with a more desired one. This requires the ability to not only take in information, but to store it and make judgements on this information in order to determine future movements from those in the past and present. The sensors should be capable of providing a comprehensive set of data consisting of the movement of each relevant body part, as well as the potential forces applied to them and the surrounding environment.

Each of these points will be further expanded upon by detailing those methods utilised by others over the last few decades of exoskeleton development as part of a detailed literature review of each of these "key components" of the exoskeleton whole that have been mentioned.

These can be summarised as the following:

- Exoskeleton Type and Purpose.
- Material Consideration and Structural Design.
- Actuation Technologies.
- Sensor Technologies.
- Control System Implementation.

A Low-Cost Methodology can then be created that applies to these key components to determine which would be most effective for a Low-Cost Exoskeleton.

2.2 Human Kinematics

A Human leg consists of 3 segments, the foot, shank and thigh, with there being a total of 6 Degrees of Freedom (DoF) split between them, or in other words axes in which these joints can rotate, with 2 in the foot, one in the shank, and 3 in the thigh. Each of these Degrees of Freedom come with their own effective ranges of motion, both comfortable and maximum ranges which may vary depending on the position of the other joints as a result of musculature limitations. While it is not required to reach the maximum range of each of these Degrees of Freedom with an exoskeleton, it should not ever be forcefully exceeded as to avoid serious risk of bodily damage.

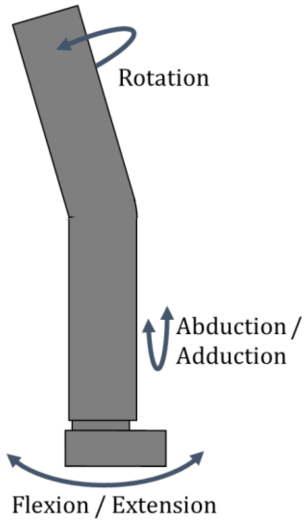


Figure 1 - Simple Anatomical Movement Diagram

In addition to body segments, Anatomical descriptors are often used throughout many papers due to the inherent links between the human body and exoskeletons [25] and describe orientations to view the body. The Anatomic planes and their associated movements (Figure 1, Figure 2 [26]) are:

- Frontal/Coronal Plane, which splits the body between the front and back. Movement in this plane is referred to as Abduction (outward movement of limbs away from the body), and Adduction (inwards movement of limbs towards the body).
- Transversal Plane, which splits the body between upper and lower halves through the waist.
- Sagittal/Lateral Plane, which splits the body into left and right parts. Movement of joints within this plane are referred to as Flexion (Reducing angle between the bones either side of a joint) and Extension (Increasing angle between bones either side of a joint)

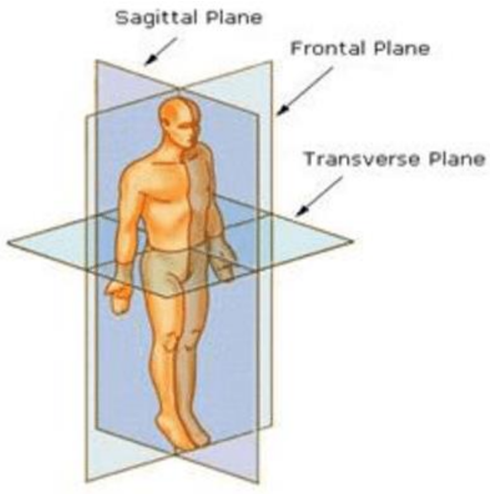


Figure 2 - Anatomical Planes [26]

2.3 Disabilities that Benefit from Assistive Exoskeletons

Cerebral Palsy

Cerebral Palsy (CP) is a series of conditions usually caused due to complications at birth which can result in mental and physical developmental issues [27] [28]. These issues can manifest as difficulties with eating, muscular overstressing and imbalance, speech and language disorders, spasms, and so on. Of particular relevance is the muscular imbalance which can result in excess strain upon the bones, stunted growth, and increased likelihood of further disorders such as osteoporosis [29] [30].

Muscle Tone is the ability to maintain partial muscle contraction and is important for a person's reflexes and posture. Cerebral palsy can cause this to be abnormal, leading to the previously mentioned spasms, imbalances, and stresses. The nature of this abnormality can vary and is classed as different types of CP as defined by SCPE (Surveillance of Cerebral Palsy in Europe) [31], Examples include Spastic CP, Dyskinetic CP, Dystonic CP, etc.

Regardless of the exact type of CP, there are universal factors that many of them share that contribute towards a pathological gait disorder Such as loss of motor control, abnormal muscle tone, and a general weakness and imbalance of muscles which can lead to an unstable stance. Gait features such as a crouch gait could be abated with a more assistive exoskeleton type that aids in supporting the wearer. It would likely be however that a Low-Cost Exoskeleton would only be effective for those individuals with low levels of Cerebral Palsy, such as GMFCS (Gross Motor Function Classification System) [32] Level 1-2 due to higher levels requiring highly specialised care.

Spinal Cord Injury

Spinal Cord Injury (SCI) is a broad cover for some kind of damage to the spinal cord. This in and of itself can result in incomplete damage to the nervous system, with reduction in the effectiveness of sensory and motor functions below the injured area. In particularly bad circumstances this may also result in total loss of feeling and motor function below the damaged area. Those suffering from Spinal Cord Injury may be quadriplegic (loss or reduction of function of all limbs and lower body), or Paraplegic (loss or reduction of function of lower body and legs). Therefore, they will likely be unable to walk without assistance or not at all, in addition to loss of feeling in these areas and a host of other issues such as difficult bowel control, spasms, and intense pain in the spinal region [33], a Low-Cost exoskeleton may be able to benefit those who have not fully lost the use of their limbs.

Traumatic Brain Injury

Traumatic Brain Injury (TBI) is caused as a result of sudden damage to the head or body in such a way that the brain is damaged, this may be from an external object that damages the skull and brain tissue. Such injuries range from temporary damage to permanent bruising and bleeding to the brain tissue, which may in the long-term lead to death. Mild symptoms of this condition include temporary loss of consciousness or states of confusion, speech problems, blurred vision, and concentration issues. Moderate Symptoms include Seizures, loss of coordination, and comas.

This can lead to a difficulty in maintaining balance and a consistent walking pattern. With the most common issues seen to be slower speed, difference in cadence, step length, and stance time on an affected leg, and difficulty with maintaining correct knee flexion [34]. These issues may be overcome to an extent with rehabilitation as TBI does not seem to have much effect on a person's ability to adapt their own movement patterns over time [35].

2.4 The Low-Cost Methodology

A Low-Cost exoskeleton implementation would be beneficial in reducing gait degradation as well as other pathologic issues such as crouch gait. "Low-Cost" will refer to minimising the total financial cost of the exoskeleton through reducing various types of cost of its components.

- *Financial cost* of components: The overall cost of components comes both from individually expensive components but also large numbers of inexpensive components, therefore attempts should be made to limit the design to easily acquiring, off-the-shelf components and limit their numbers to what is minimally necessary. Such as [36], a gait-trainer used in developing countries.
- *Power cost* of components: The amount of power drained by components over time, a high power cost reduces the overall use-time of the exoskeleton unless additional batteries are added, which increases financial and weight costs. Therefore, components should ideally only be active when necessary and drain little power when active. Where possible, assistance may be provided without electrical energy expenditure. For example, [37] made use of an passive design to aid in squats.
- *Weight cost* of components: The overall weight of components, as well as the inertia they cause, a high weight cost requires more powerful actuators to overcome and a stronger frame, increasing financial and power cost. Therefore, components should be as light as is reasonable without such sacrifice to durability that the device is not safe or practical to use. For example, [38] minimised the effect of weight using a series of Bowden cables to allow much of the exoskeleton to be above the waist, and so minimising its effect on the metabolic cost of walking.
- *Effectiveness* of components: While keeping costs low is ideal, if the components themselves are not effective in aiding in the assistance or rehabilitation of the wearer then they are not useful in the exoskeleton. For example, a more expensive component that is considerably more effective than a less expensive one may be a superior choice, a Low-Power actuator incapable of providing required torque would be ineffective, a very light component that is not durable enough for its function would not be desired. As such, a certain amount of Financial, Power, and Weight cost must be considered as to not sacrifice Effectiveness.

Each Component Type will be judged by these four elements, the financial cost of their implementation as well as any peripherals they require, the power cost of their running and how power efficient they are, the weight cost on the exoskeleton as well as of any peripherals they require.

On top of this, the effectiveness of the components will be considered, with both benefits and disadvantages of each component type outside of Low-cost considerations. The Literature Review sections will then conclude with a summary of which implementations are considered optimal for a low-cost implementation, these ideal components will then be used to design an implementation that attempts to minimise financial cost, power cost, and weight cost without sacrificing too much effectiveness within the next section of the Thesis. For General usage, the Significance of these Low-Cost Factors of Financial, Power, Weight, and effectiveness may vary dependant on the purpose of the exoskeleton, for example, an exoskeleton designed for sit-to-stand movements would prioritise high force output over absolute precision than an exoskeleton designed for correcting a pathologic gait deficiency.

While this Low-Cost Methodology was developed independently of other literature, a similar set of exoskeleton objectives were outlined in [22]. This paper also recognised electric motors as being the most commonly used actuator method, which is in line with the data set found later within this Literature Review.

2.5 Research Methodology

To gain an insight into the current state of the Exoskeleton Literature, papers have been collated from Web of Science, Scopus, and other Grey Literature from 1990 to 2021 in the subject area of Electrical and Electronic Engineering from Conference Papers and Journal Reports. The Search Procedures used were the following:

Title/Abstract/Keywords: (leg OR foot OR hip OR knee OR ankle OR (lower AND (limb OR extremity OR body)))*
AND
Title/Abstract/Keywords: (rehab OR assist* OR treat* OR pathological)*
AND
Title/Abstract/Keywords: (wearable OR ortho OR robot* OR exoskeleton OR actuat* OR powered)*
AND
Language: (English)
AND NOT
Title/Abstract/Keywords: (control OR classif OR recognition OR review OR analysis OR examin* OR comparison OR investig* OR estimation OR effect OR simul* OR assess* OR evaluation)*

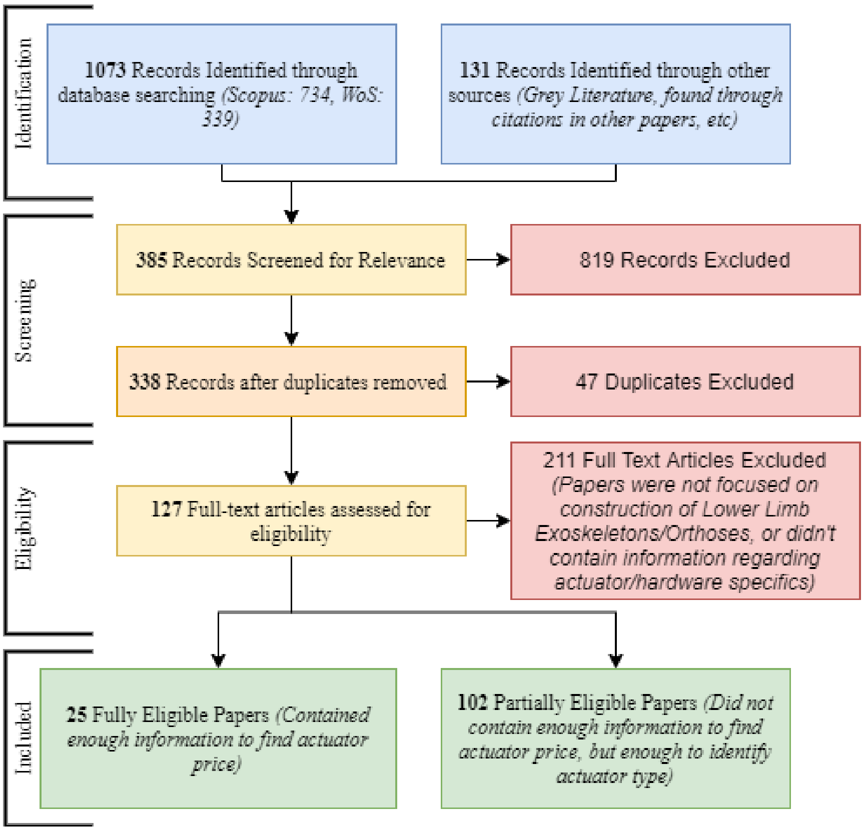

Identification
1073 Records Identified through database searching (Scopus: 734, WoS: 339)
131 Records Identified through other sources (Grey Literature, found through citations in other papers, etc)
Screening
385 Records Screened for Relevance
819 Records Excluded
338 Records after duplicates removed
47 Duplicates Excluded
Eligibility
127 Full-text articles assessed for eligibility
211 Full Text Articles Excluded (Papers were not focused on construction of Lower Limb Exoskeletons/Orthoses, or didn't contain information regarding actuator/hardware specifics)
Included
25 Fully Eligible Papers (Contained enough information to find actuator price)
102 Partially Eligible Papers (Did not contain enough information to find actuator price, but enough to identify actuator type)

Figure 3 - Data Acquisition via PRISMA Guidelines

This Produced 734 results from Scopus, and a similar search used for Web of Science produced a further 339 results, as of June 2021. Finally, an additional 131 results were obtained through miscellaneous collection prior to the development of this thesis primarily through a process of extracting relevant references from other academic papers. Collectively this produced an initial total of 1204 papers, after several rounds of exclusions 127 articles were considered eligible. The Exclusion process is shown in Figure 3, following PRISMA Guidelines.

For a paper to be considered eligible, it was to contain information directly regarding a Lower Limb Exoskeleton, or part of one (such as a leg or ankle orthosis), as well as mention of what actuator technology was used in its construction (expanded on in 2.7). To be eligible for full use the exoskeleton had to specify which actuator specifically was used with enough information where one could then find details about it such as price, power requirements, weight, and so on. While similar, for the purposes of uniformity, protheses and static walking aids such as static rehabilitation systems were not considered eligible as these are subject to different design philosophies. Stationary exoskeletons designed for use on treadmills, designs that made use of external connected hardware, or that only cover part of the leg such as a knee orthosis, were however permitted. In other words, Papers should be for Lower Limb Exoskeletons or orthoses that directly aid locomotion and must contain some information regarding the actuation methods and specific hardware of said exoskeleton or orthosis.

Of each of these papers, the key details of their described exoskeletons were collated, these being the Date of Publishing, the Exoskeleton Coverage, whether it was a Soft or Hard exoskeleton, whether it was Mobile or Stationary, its purpose (Assistive, Rehabilitative, or Augmentative), the Actuator, Sensor, and Control Systems used in its construction, and if available the specific actuator and sensor hardware and its cost. The details will be expanded upon within the later sections of this chapter.

Some trends noticed within the literature include a linear increase in the output of new Literature over the last 2-3 Decades, as shown in Figure 4, this would be expected with increasing technological capacity, as well as the increased demand for exoskeleton use.

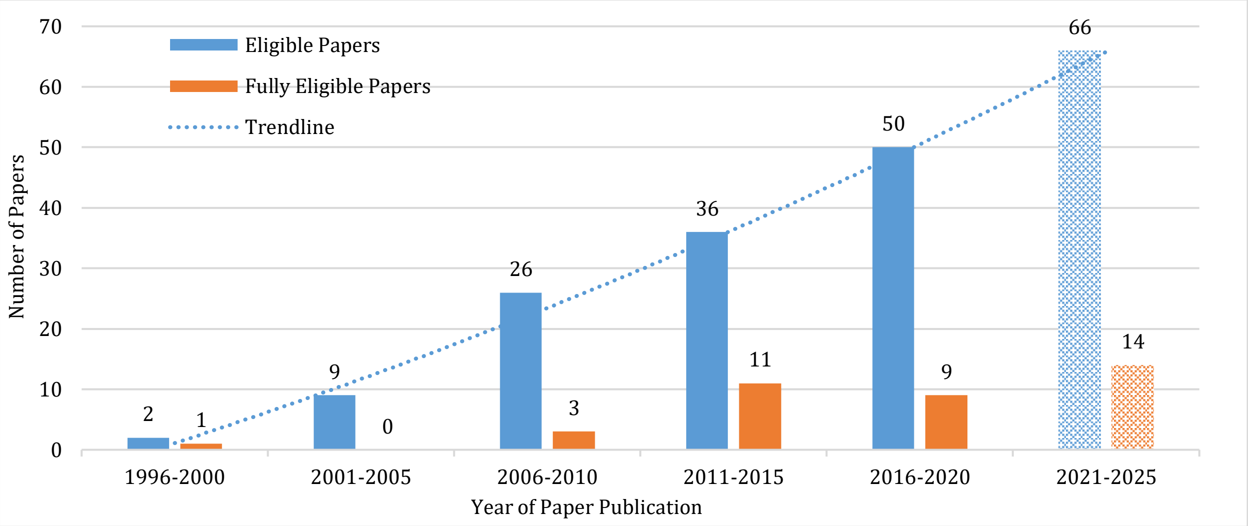


Figure 4 - Graph of Eligible and Fully Eligible Papers, showing gradual upwards trend between 1996 and 2020, theoretical predicted increase in papers for 2021-2025 also graphed based on current trend.

There has also been an uptick within the last decade of collected information when it comes to papers specifically identifying the hardware used within the exoskeletons, with only 4 out of 37 (~10.8%) papers between 1996 to 2011 containing identifiable information compared to the 20 out of 90 (~22.2%) from 2012 to 2021. Across this period of time, the consistent majority of exoskeletons have had a hard structure and were designed for rehabilitative purposes (Figure 5 and Figure 6).

Multiple other Literature Reviews on the current state of Exoskeleton technologies have also been written throughout the prior decades, such as [39], and [40] who came to similar conclusions regarding actuator and sensor groupings. Others, such as [41], [42], and [43] instead grouped exoskeletons by their Control Systems such as “Force Control” and “Position Control”.

2.6 Structural Design

At a basic level, exoskeleton structure makes use of hard materials which are inelastic and somewhat rigid, soft materials which are elastic and pliable, or a mixture of both. Each has their own benefits and drawbacks, at a basic level [44]:

- Soft, compliant materials such as fabrics and elastomers can be used in minimalistic exosuit designs that can match to clothing and work better in tandem with muscles in order to provide assistances. Their natural compliance means that there will be minimal restriction of movement and is therefore ideal in exosuits designed to be supportive or assistive. They are less appropriate for more heavily rehabilitative exoskeletons used to substitute a person's movement ability as a result of physical conditions or injuries for this same reason, their natural compliance limits the amount of torque they can reliably transfer as well as the speeds that they can move at.
- Harder, more rigid materials such as steel and aluminium can deliver reliable, strong forces both quickly and accurately when combined with the proper actuation. They can form a solid structure independent of the user's own strength, making it ideal in cases where the user may be paralysed or otherwise unable to provide sufficient strength themselves. This comes at the cost of reduced manoeuvrability and often increased weight, which itself results in higher power consumption to move the increased weight.

A combination of both hard and soft exoskeletons is not objectively better than either individually, as it would have the benefits and disadvantages of both, with the possible addition of further complexity. Such an implementation however may use softer joints and elastic actuation for freer movement but use rigid hardpoints to allow for more reliable transfer of force. The proportion of Exoskeleton Types within the literature is seen in Figure 5 and Figure 6.

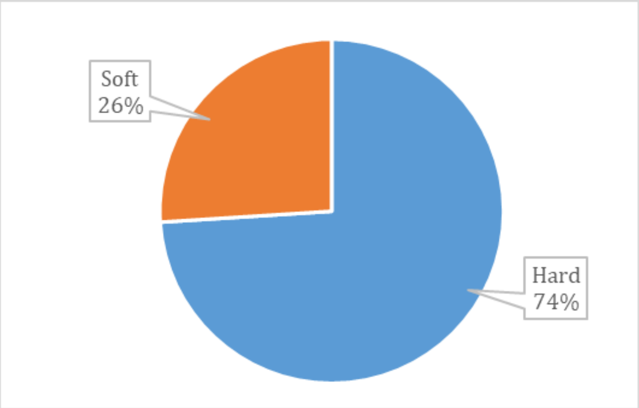


Figure 5 - Hard vs Soft Exoskeletons

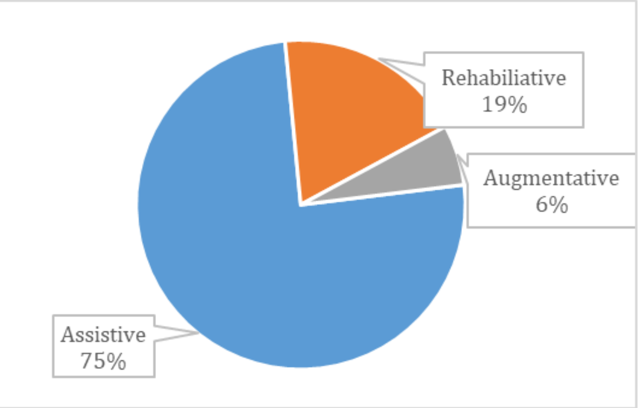


Figure 6 - Exoskeleton Type Proportions

2.6.1 Material Considerations

Exoskeletons tend to be made of metals such as steel and aluminium, plastics such as thermoplastics, and elastomers such as synthetic rubber. One way of reducing initial cost could be to make use of FDM (Fused Deposition Modelling) Printing to print test components with PETG, PLA, ABS, etc Plastics, allowing for components to be made and tested within a matter of hours, at the cost of them not necessarily being suitable for use in a real-world environment due to lacking strength to survive large amounts of strain placed on them without snapping, there are also some examples of their usage within completed exoskeletons [45].

As defined by the project objectives, in order to keep the design of the Exoskeleton Simple and capable of supporting the wearer, a Hard Exoskeleton design with minimised interference on the wearer would be ideal. For example, using an inexpensive, pre-existing orthoses as a base such as [46] could act as a skeleton that provides attachment points for actuation and sensor equipment. Within the Literature, it was quite uncommon for papers to state the exact nature of the materials used within exoskeleton construction, although one example [47] used 7075T6 Aluminium Alloy in the construction of its basic structure, due to its low density and sufficient structural durability. Similarly, several papers mention the use of Titanium and Carbon Fibre in construction of some of their components, although this comes at very high financial cost [48]. From visual inspection however, it was common for metallic and compliant/elastomer materials to be clearly apparent when depictions of an exoskeleton structure were provided.

2.6.2 Power Requirements

Whilst not universal, the vast majority of the exoskeletons possessed active elements, therefore requiring power. For the Stationary exoskeletons, power was sourced externally through wired connections, however for those designed to move independently the only viable solution was some kind of battery storage often stored in a backpack. The most common battery type was Lithium Polymer batteries (LiPo), being lighter than Lead Acid Equivalents, making up 13 of 17 instances where power sources were identified with examples such as the 2800mAh Prolite Battery [49] used in [50]. The next most common choice being larger wired power supplies such as the 35W E3630A Power Supply [51] used in [52].

2.7 Actuation Technologies

Actuation is the most significant and energy intensive part of many Exoskeletons. There are different actuation solutions that each must consider energy efficiency vs weight vs costs, and so on.

2.7.1 State of the Art and Discussion

Actuator data was collected from the 127 eligible papers concerning the development of exoskeleton technology and the analysis of the actuators used. This consists of actuator characteristics such as weight, power usage, and cost, and concludes attempts to rank them qualitatively and quantitively on the low-cost concepts discussed previously. The Actuator Types Identified were Rotary Electric Motors, Rotary Electric Motors + Ball Screws, Linear Electric Motors, Hydraulic Cylinders, Pneumatic Cylinders, Pneumatic Muscles, Cables, and Compliant Actuators. Rotary Electric Motors were found to be the most used implementation, the proportions of each are seen within Figure 7.

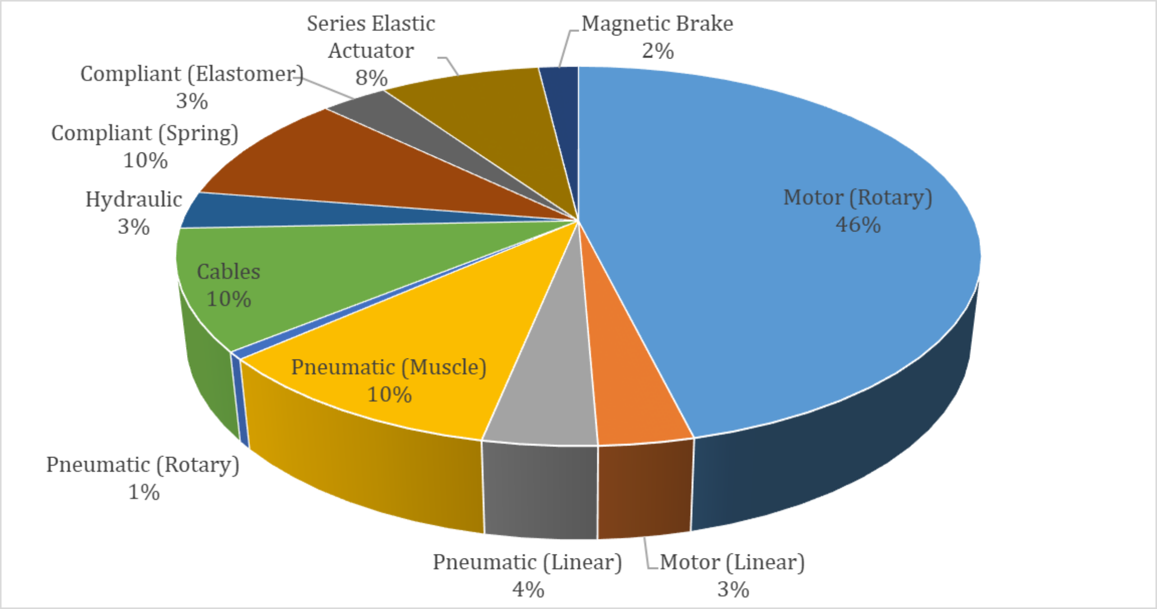


Figure 7 - Actuator Type Appearances (Note total number of appearances is larger than total number of papers as multiple actuator types may appear in one paper, in such cases, the two most prominent types are used)

For qualitative analysis theoretical implementations that display the overall costs, weights, power usage, and effectiveness of each actuator type have been created and are displayed in Table 1 to be compared against each other by collating data provided by those papers listed as “Fully Eligible” (EG: Provided enough information to identify the specific Actuator used, allowing for its cost to be found, as well as those costs for any necessary peripherals). For actuator types where less information was available such as hydraulics, an effort was made to find some examples with available characteristics. These implementations will consist of the actuator itself, in addition to any peripherals it is stated to require. Averaged characteristics are averages of several components of a similar type (such as electric servo motors) while estimated are only of a small (3 or less) number of components.

While in reality actuator costs will vary wildly depending on specifications, availability, actuated joint target, and many other factors, the following table can still be considered a possible baseline for such actuator characteristics. The Benefits and Disadvantages of each actuator type, as well as further examples of specific actuator technology costs has been covered in [53] [54].

Table 1 - Actuator Type Characteristic and Cost Breakdown [53] [54]

Actuator Type	Component	Characteristics				Characteristics Total			
		Power (W)	Weight (g)	Force (N/Nm)	Cost (£)	Power (W)	Weight (g)	Force (N/Nm)	Cost (£)
Rotary Electric Motor	Averaged Motor	150 W	415 g	0.32 Nm	£375	160W	1165 g	48Nm ~130 N [4]	£1005
	Averaged Encoder	0.25 W	50 g	-	£80				
	Averaged Gear Set	-	400 g[1]	150:1	£300 [1]				
	Averaged Driver[2]	+5%	300 g	-	£250				
Rotary Electric Motor + Ball Screw	Averaged Motor	150 W	415 g	0.32 Nm	£375	160W	1665 g	~500 N	£1125
	Ball Screw	-	900 g	4mm lead	£370				
	Averaged Encoder	0.25 W	50 g	-	£80				
	Averaged Driver[2]	+5%	300 g	-	£250				
Linear Electric Motor	Averaged Linear Motor	200 W	1000 g*	60 N	£700*	210W	1350 g	60 N	£1030
	Averaged Encoder	0.25 W	50 g	-	£80				
	Averaged Driver[2]	+5%	300 g	-	£250				
Hydraulic Cylinder	Averaged Cylinder	-	600 g*	1800 N	£40?	370W	3600 g + 2000 g	1800 N	£1565
	Estimated Valves/Solenoids[2]	5 W	500 g	-	£150*				
	Estimated Pump Motor[2]	350W	1000 g	0.8 Nm	£500				
	Av. Pump Motor Encoder[2]	0.25 W	50 g	-	£80				
	Est. Pump Motor Driver[2]	+5%	770 g*	-	£725*				
	Estimated Piping[2]	-	200 g*	-	£20				
	Averaged Fluid Container [2]	-	500 g	-	£30 [3]				
	Estimated Fluid[2]	-	2000 g	-	£20				
Pneumatic Cylinder	Estimated Cylinder	-	750 g*	1800 N	£150	1105 W	2070 g + 27 kg[5]	1800 N	£765
	Estimated Valves/Solenoids[2]	5 W	120 g	-	£100				
	Estimated Piping[2]	-	150 g*	-	£15 *				
	Estimated Gas Container[2]	-	1000 g	-	£150				
	Estimated Compressor[2]	1kW	27,000 g[5]	-	£350				
Pneumatic Muscle	Est. Fluidic/Inflatable Muscle[1]	-	200 g* [1]	100 N* [1]	£100* [1]	35 W	2470 g	100 N*	£615
	Estimated Valves/Solenoids[2]	5 W	120 g	-	£100				
	Estimated Piping[2]	-	150 g*	-	£15*				
	Estimated Gas Container[2]	-	1000 g	-	£150				
	Estimated Compressor[2]	30 W[1]	1000 g	-	£250				

[1] Values can range very considerably, this value is a middle ground, in practice, it may either be much more or less than this.

[2] This example assumes one of each component is purchased, meaning if multiple components are purchased, the ratio of this component to other components may not be consistent. This may be because only one is required for the entire exoskeleton, or that the component can be expanded.

[3] This is an estimate to the price of a 3L unpressurised tank as seen in [55, p. 10], a pressurised tank would be both heavier and more expensive.

[4] For the purposes of comparing this torque value to the linear force values of the other actuators, it has been assumed that the motor is rotating a 20cm shaft, with a force reading taken perpendicularly from its end, as this is a rough estimate to the length to the CoM of an adult's upper leg.

[5] Pneumatic Compressors capable of providing enough pressure to power cylinders seem to be too large to fit on the exoskeleton, therefore their weight would not count towards it, however their presence should still be considered. As such, for the purpose of scoring, pneumatic cylinders will have a normalised weight of 1.

* This information is a "Best Guess" based on limited source components, in some cases as little as one

Rotary Electric Motors average out to be the lightest of the actuator types, with all electric motor types benefiting from not requiring any sort of fluid that must be contained and used for force application which is the primary source of weight for pneumatic and hydraulic actuators. This comes with the downside of a potentially high cost, especially due to the varying expense of gearing. Rotary motors with ball screws would not have to deal with as expensive gearing due to a higher reliance on high RPM to rotate the ball-screw, the less expensive gearing seems to counter most of the extra cost of the ball screw itself. The more numerous implementations of a motor and ball screw in place of traditional linear motors may be due to a superior force and lower weight and power usage. Hydraulic is the worst in all categories with the exception of force, which while on average tied with pneumatic cylinders can be considered superior due to a higher precision in application. However, due to a high-power usage, weight, and cost it struggles to offer much relevance outside of industrial exoskeleton technologies.

While Pneumatic Cylinders average out to be both low-power and low-cost, they are let down by a high weight cost as a result of the necessity of the compressor, without which the system would have no way to maintain pressure within the air tank or the rest of the system. In all examples within the literature, this component is external, which would be infeasible for an exoskeleton seeing actual usage. The only benefit is that there would only need to be one of these components that would then power the rest of the system. Other than this, pneumatic systems are relatively cheap, low power, and low cost, as such still may find some use in softer applications which may be able to use a lower power compressor for lower forces, although still suffer from the considerable downside of potentially requiring a compressor to function.

The normalised values displayed in Figure 8 in addition to the Force to Weight Ratio, Impact, and Precision. These will further rank the actuator types in a quantitative manner, with each being weighted according to different approaches. This is a similar albeit simpler financial analysis to what is seen in [56].

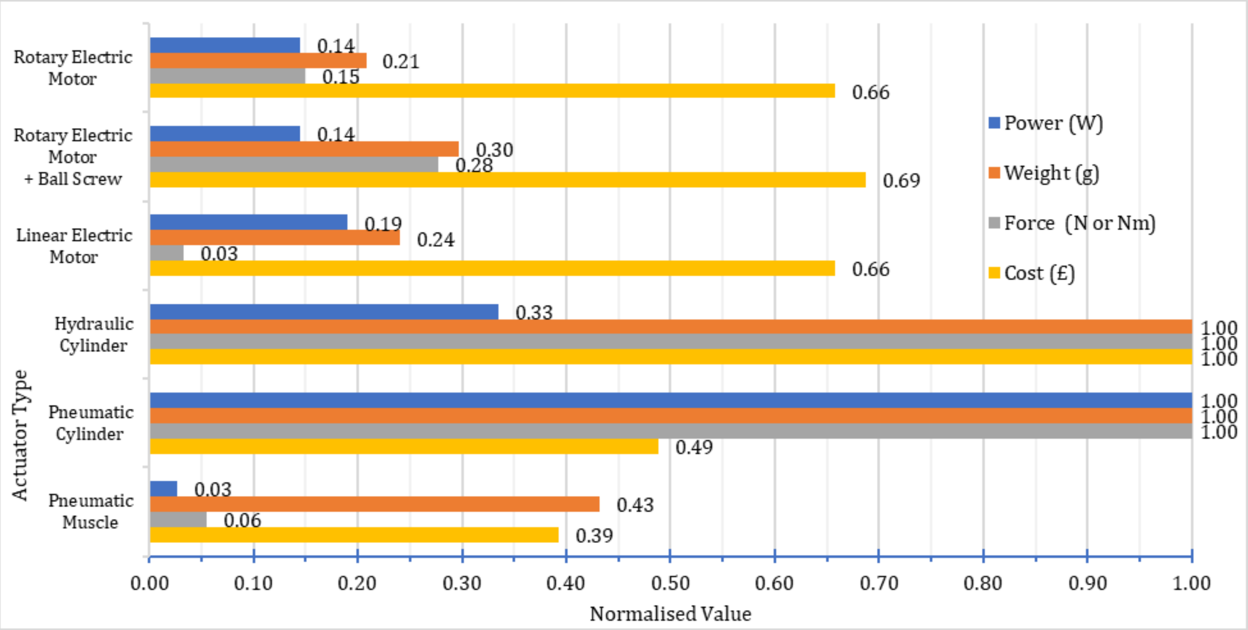


Figure 8 - Normalised Actuator Characteristics, note that for Power, Weight, and Cost higher values are worse, whereas for Force, lower values are worse.

Table 2 - Impact and Precision of Actuator Types.

	Rotary Electric Motors	Rotary Electric Motors + Ball Screws	Linear Electric Motors	Hydraulic Cylinders	Pneumatic Cylinders	Pneumatic Muscles
Impact	V.Low (1)	V.Low (1)	V.Low (1)	Medium (0.5)	High (0.25)	Low (0.75)
Precision	V.High (1)	V.High (1)	V.High (1)	V.High (1)	High (0.75)	Medium (0.5)

Impact describes the effect the exoskeleton has on the surrounding environment, such as from spillages, noise, or repairs. For this, purely electric systems score well, whereas hydraulic and pneumatic systems score poorer due to their usage of fluids, compressors, and so on providing constant noise, leak risk, and weight in pump peripherals. Precision describes how well the actuator follows the movements of the wearer and can accurately be set to a precise position, Electric and Hydraulic systems do well due to the ability to easily achieve any specific position, whereas pneumatics tend to struggle due to more unreliable compressive gas outside of end-to-end movements without additional hardware [57]. The values in brackets represent normalised approximations varying from 0 to 1 with a higher value being "better", as actual values do not appear to be readily available it was deemed impractical to calculate. With Impact ratings of V.Low (1), Low (0.75), Medium (0.5), High (0.25), and Very High (0), and Precision ratings of V.High (1), High (0.75), Medium (0.5), Low (0.25), and Very Low (0).

Table 3 - Weights of Approach Criteria.

	Power Usage	Actuator Weight	Actuator Force	Actuator Cost	F/W Ratio	Impact Reduction	Precision
Equal-Weights Approach	0.142	0.142	0.142	0.142	0.142	0.142	0.142
Industrial Augmentative Exoskeleton Approach	0.100	0.125	0.200	0.125	0.225	0.125	0.100
Rehabilitative Exoskeleton Approach	0.150	0.175	0.100	0.125	0.150	0.150	0.150
Assistive Exoskeleton Approach	0.150	0.125	0.150	0.100	0.150	0.150	0.175
Low-Cost Approach	0.175	0.150	0.125	0.200	0.150	0.150	0.05

The weights within Table 3 are defined based on the perceived priorities of different exoskeleton types and as demonstration for quantitative analysis. The Equal Weights Approach exists as an unbiased control value to provide values with no special priorities given to them, in a sense showing how well rounded an actuator type may be.

The Industrial Augmentative Exoskeleton Approach prioritises high power to weight ratios and force, sacrificing precision and power usage to achieve this, as the exoskeleton is only required to follow the already healthy user and reduce strain, allowing for bulkier designs and higher power usage to support stronger actuation.

The Rehabilitative Exoskeleton Approach aims to aid a user capable of some ambulation to improve their gait pattern, it is therefore important that the exoskeleton does not detract from movement through imprecise movement or heavy weight, additionally as this device should be used in more home environments, it must be more power efficient and less impactful on the wider environment as to avoid disruption. It does not however require high force as the user is still capable of most of the walking motion themselves.

The Assistive Exoskeleton aims to provide walking functionality to those that would not be able to walk otherwise, and to therefore assist in the walking process. As with rehabilitative exoskeletons it therefore requires high precision, although its weight is less of a concern as the user's gait does not contribute heavily to the walking process. Instead, the system requires stronger actuators in order to move for them. Due to its specialized nature, this exoskeleton does not need to reduce costs as much.

The Low-Cost Approach is the focus of this thesis and seeks to provide an alternative to the Rehabilitative Exoskeleton Approach that focuses on reducing financial cost by removing the need for high precision actuators and reducing power usage through using the active elements of the actuator as little as possible.

Other approaches could also be designed to focus on whatever exoskeleton aspects would be desired by the user, for example rather than focusing on a type of exoskeleton it could instead focus on an exoskeleton purpose, e.g. a sit-to-stand exoskeleton would weight force-to-weight ratio highly, but may weight power usage and precision low as the exoskeleton is designed to assist in sporadic but intense generalised movements. As opposed to continual minor adjustments.

Table 4 - Normalised Actuator Scores for Different Approaches

	Rotary Electric Motors	Rotary Electric Motors + Ball Screws	Linear Electric Motors	Hydraulic Cylinders	Pneumatic Cylinders	Pneumatic Muscles
Equal-Weights Approach	0.9966	**1.0000**	0.8080	0.8953	0.8861	0.8321
Industrial Augmentative Exoskeleton Approach	0.9445	0.9934	0.6829	0.9967	**1.0000**	0.6884
Rehabilitative Exoskeleton Approach	**1.0000**	0.9844	0.8149	0.8171	0.7932	0.8114
Assistive Exoskeleton Approach	0.9836	**1.0000**	0.7942	0.9233	0.8744	0.7851
Low-Cost Approach	**1.0000**	0.9967	0.7928	0.8329	0.8929	0.9654

Table 4 contains the normalised results of the data from Table 1 and Table 2 for the different approaches described in Table 3. For this calculation, the Power, Weight, and Cost were normalised to the lowest value rather than the highest as displayed in Figure 8, this was so that the highest values were always "better", as lower power usage, weight, and cost are more desirable.

The result of the table show that Rotary Electric Motors are, on average the most effective actuator, which is consistent with it being the most commonly used, either without or in combination with ball screws. Rotary Electric Motors in combination with ball screws saw higher scores than standard Linear Electric Motors which may also be why they are so comparatively uncommon as any time where a linear actuator would be applicable, the rotary electric motor + ball screw has a higher score. Meanwhile, there is a considerable gap in score between electric motors and Hydraulic and Pneumatic Actuators.

For Hydraulic and Pneumatic Cylinders, this is as a result of their very high weights, which plummet their overall score by significant margins, they did however see prominence in Industrial Exoskeleton Approaches, being the highest scoring methods, although while pneumatic was marginally higher, it assumes a method of easily carrying the separate compressor that would be required for extended function. If this method was not present, hydraulic would be superior due to being more easily self-contained. Pneumatic muscles meanwhile saw little significance in most categories due to a weight score crippling otherwise promising low power usage and costs.

These values were made with only a single actuator in mind, however as several components would only need to be purchased once such as motor drivers, or hydraulic pumps, or would not scale directly in some capacity if multiple actuators were needed such a solenoids or which can have multiple ports in one block, or fluid piping which would only need to connect between the new actuator and pump, components that may only need to be counted once or at a reduced rate are referenced in Table 1, under "2". Furthermore, it is unlikely all actuators will be functioning simultaneously or continuously, reducing power usage for electric motors and fluid throughput requirements for hydraulic and pneumatic implementations.

This becomes difficult to calculate, and as such could be a topic for future study. It can however be speculated that Electric Motor types would suffer considerably as the vast majority of their components are needed for each new actuator, with only the controller being able to be used across multiple. Hydraulic and Pneumatic actuators meanwhile could benefit from a single centralised pump and cheaper individual cylinders or muscles for each actuator, translating into reduced cost, although not necessarily reduced weight.

When considering the implementation of compliant elements or transmission systems such as springs, there will be a minor increase in weight from additional hardware, and a potential loss of precision due to uncontrolled movement from these elements. Each presents different, less quantifiable benefits as outlined in their relevant sections, cables allowing for the relocation of actuators to more favourable locations, for example, moving actuators from locations on the leg to a backpack worn by the user, therefore reducing the effect of inertia on walking. For compliant elements a level of feedback prevention is provided as compliant elements can absorb impacts as well as store energy.

Series Elastic actuators and other similar implementations are the most common implementation involving multiple different types of actuators being used within the same exoskeleton, often taking the form of a spring and rotary electric motor, theoretically a similar design could be made with multiple types of active actuator. A simple example of such an implementation can be imagined for an industrial exoskeleton. It would make use of hydraulic or pneumatic actuator for joints that would experience high force loads, such as hip, knee, and ankle flexion/extension, however for degrees of freedom that may see less load such as ankle abduction/adduction or hip rotation, geared electric motors could be used due to their smaller volume. This could reduce complexity and weight due to the need for less hydraulic fluid capacity and cabling.

As seen in Table 4, both rotary electric motors and hydraulic cylinders score highly for an Industrial Augmentative Exoskeleton Approach, using them together could be analysed quantitatively by average the two scores $\left(^{0.9967\,+\,0.9445}\!/_{2}\ = 0.9706\right)$.

To place an estimate for individual components lifetimes [58] predicts an electric motor to function for ~20,000 hours of runtime in ideal conditions, which is in line with the estimates given by Maxon for their motors [59], although they also state 1000-3000 hours is more expected in average conditions. Pneumatic Cylinders meanwhile have been estimated to last up to 10's of millions of cycles [60], which is somewhat in line with a prediction made by the Bimba Manufacturing Company predicting its cylinders could function for a total travel distance of 3,000 miles [61], if a cycle equated to ~40cm, this would equate to ~12 million cycles. While it does not seem unreasonable to assume hydraulic cylinder and pneumatic cylinder data would be similar, both are likely to suffer in terms of lifetime as a result of other peripheral components, notably leaks in piping as well as wear and tear in pumps and compressors which will cause gradual degradation of actuator quality, necessitating regular replacement or maintenance.

2.7.2 Actuator Summary

The analysis on actuator characteristics reveals rotary electric motors both with and without ball screws score well for the most types of exoskeleton approaches, which is in line with them being the most common actuator type within the reviewed dataset. How these scores would be affected by considering multiple actuators however becomes difficult to calculate. None the less, it seems that they alongside compliant actuators would make for the ideal implementation within a Low-Cost Exoskeleton in the form of Series Elastic Actuators, in so doing the Rotary Electric Motor provides relatively lightweight and power-inexpensive actuation whilst the Compliant element allows for a motor that does not need to be as immediately reactive or precise, as the elastic nature of the Series Elastic Actuator would give immediate response to movements made by the wearer, and so effectively provide an extended "reaction time" for the motor, as well as leniency as to its precision as the elastic element would make up for minor differences in any desired angle and actual angle difference caused by motor imprecision.

Whilst the Actuators are a core component of the Exoskeleton, they alone are useless without a method of data input to inform the actuators how they should move. Therefore, the next section will review the Sensors used within Exoskeleton implementations, what sort of information they provide, and how such information may be useful.

2.8 Sensors

For an exoskeleton's actuators to properly function, they require a control system to drive them and sensors to provide input data. These sensory inputs can be used to detect the intentions and status of its user and also detect the environment around the exoskeleton to assist in balance and avoiding obstacles. As such several types of sensors will be defined in this section:

- "Body Sensors", those that measure bodily processes (EMG, EEG).
- "Force Sensors", those that measure the application of force upon them (Force Sensor, FSR, Load Cell, Torque sensor, Strain Sensor, Force Plates).
- "Motion Sensors", those that measure the direct movement of the individual (IMU, Motion Sensor).
- "Pressure Sensors", those that are used to monitor the pressures of actuators that use gasses/fluids.
- "Rotary/Linear Sensors", those that measure rotation / displacement in a particular dimension or direction (Encoder, Potentiometer, Goniometer).

Examples of each will be described in the following sections with their relevance discussed within wider literature and their potential usage within a Low-Cost exoskeleton system, along with examples and costs of specific components. Within the Literature Review, most of the reviewed papers possessed some form of sensor, with the notable exceptions all being compliant or passive designs such as [62] and [63].

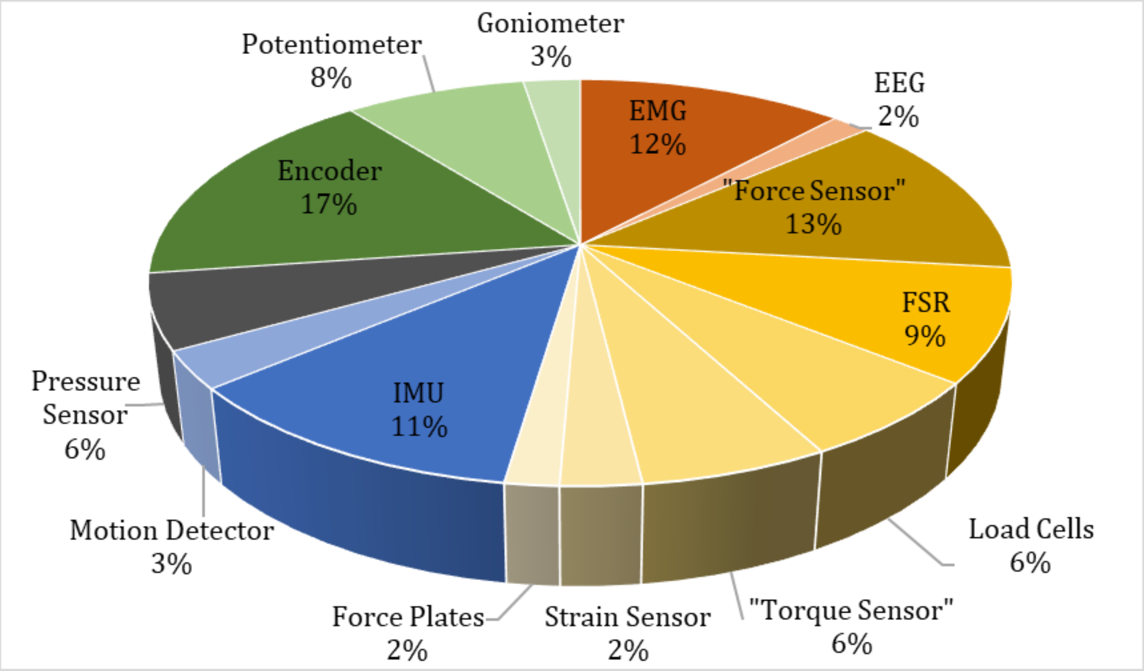


Figure 9 - Proportion of Sensor Types, separated into subtypes.

As seen within Figure 9, the most commonly used sensor type within the Literature review were Force Sensors with 75 Instances, followed by 63 Rotary/Linear Sensors Instances, 33 Instances of Motion Sensors, 32 Instances of Body Sensors, and finally 14 Pressures Sensors. The most common individual sensor was the Force Sensitive Resistor [64]. Most exoskeletons used more than one sensor type.

2.8.1 The Role of Sensors

Sensors provide the input data for the exoskeleton control system. They provide information to the controller on the user's intent or current condition via the movement of their legs, the electrical impulses of their muscles/nerves, their joint angles, or so on. This can be used to manoeuvre the exoskeleton in the way the user desires. For example, if knee flexion is detected through a decrease in knee joint angle, or force applied to a rear tibia pressure pad, or a contraction of the hamstrings, or some other method, then the control system can process this information and command the actuators to move with the wearer if it is within healthy motion parameters. Similarly, if the user were to be detected to be entering into an unstable state, such as through sudden changes in joint angles, a control system may be able to recognise this and act in a way to try and avoid falling.

Sensors may also be used to detect the occurrence of key gait moments, such as the heel strike and toe off events commonly used in gait detection, which could be achieved through the use of Force Sensors placed on the foot of the user, Motion sensors picking up the jolt of foot to floor contact, or so on. Finally, sensors may provide information about the surrounding environment, such as whether any objects would block the path of the wearer when moving, this is especially useful in heavier, rehabilitative exoskeleton designs which may take over a large proportion of the walking effort from the heavily impaired user, and therefore has to consider proper balancing and collision avoidance, these concepts will not be as relevant for this project however, as the intended wearers are still the primary drivers of movement and so capable of these actions on their own.

2.8.2 Body Sensors

These Sensors measure bodily process. With exoskeletons the most common of these were Skin-Surface Electromyography (EMG) (28 instances) and Electroencephalography (EEG) (4 instances) with both having examples being used both for the driving of and testing of exoskeletons. Other examples of sensors in this category that were mentioned included Heart Rate sensors, however these did not aid in controlling the exoskeleton and were only used for test purposes and so are not included. Table 5 provides a list of identifiable EMG and EEG Sensor Components within the Literature Review.

Table 5 - Body Sensors within the Literature Review

Study	Sensor Type	Sensor Name	Purpose	Cost
[65] [52] [66]	EMG	Delsys 8-Sensor EMG System	Testing / Data Collection	£15,000 [67]
[68]^1 [69]^2	EMG	Biometrics SX230^1 + DataLOG^2	Testing / Data Collection	£310^1, £3,385^2 [70]
[71]	EMG	BTS FreeEMG300	Testing	-
[72]	EMG	Sichiray sEMG	Data Collection	-
[73]	EMG	MCRS EMG	Controls Knee Joint	[74] (no cost)
[69]	EEG	Brain Products Acticap	Exoskeleton Movement	-

1,2: Numbers relate to relevant product

Skin-Surface Electromyography (EMG)

Skin-Surface Electromyography detects the electrical signals of the nervous system from the activation of muscles. These sensors are used within the medical field, such as in the HAL exoskeleton [10] as a way to predict user intents, this method results in minimal delay between muscle movement and machine movement as often these signals will be detected before the limb actually begins to move.

This method is however limited by being easily disturbed by sweat or Electromagnetic noise [24], as well as requiring the electrodes to be either stuck uncomfortably onto the skin or in a more invasive version implanted within the skin (which may bypass surface issues such as sweat but can only be done in a medical setting). Additionally, quadriplegic, or paraplegic patients with spinal cord injuries will likely be unable to use this method as the nervous connection between their legs and brain has been damaged, and so signals might not be sent. Alternatively, if the patient has muscle spasms such as with some variations of cerebral palsy or muscle fatigue from exertion, then erroneous signals or unexpected signals may be sent making it unhelpful for following movements.

Within the literature, the majority of examples of EMG usage were for test purposes as opposed to being the primary method of driving the exoskeleton. In effect the EMG sensors were used for making sure the exoskeleton did not affect the body’s natural motion. The Delsys Trigno [75] was the most commonly used EMG system with an estimated full cost of ~$15,000 [67] for an 8-Sensor setup, with similar systems used in [65] and [52], although as in many cases this component is used for testing rather than as part of the exoskeleton. There are some examples of EMG being used directly in the control of an exoskeleton, such as in [73] where a bespoke EMG device named MCRS used Hamstring and Quadricep muscle group signals to control a knee joint actuator. Similarly [76] used the Root Mean Squared input of the EMG signal to drive DC Motors at the Hip and Knee Joints.

Whilst EMG can provide highly detailed information regarding the state of the wearer’s muscle activity, for the purposes of a Low-Cost implementation it may have potential usage within testing, but less suitable as an exoskeleton sensor component.

Electroencephalography (EEG)

Conceptually similar to Skin Surface Electromyography, rather than measuring neural activity from the muscles, Electroencephalography is a form of Brain Machine Interface (BMI) that measures neural activity from the brain itself. It most commonly uses a cap of electrodes placed over the head that is capable of measuring brain activity, such as the Acticap used in [69]. Although more invasive, there are also permanent procedures where the electrodes can be implanted into the brain. This implementation is very uncommon, with only 4 examples (2%) within the Literature Review, of which most were used for Testing Purposes rather than driving the exoskeleton. In theory this implementation could benefit patients suffering from paralysis due to taking commands directly from the brain, however for a Low-Cost implementation it is likely not appropriate due to unneeded complexity and cost.

Summary

Body Sensors, whilst capable of providing very accurate and immediate data, are likely not suitable for the Low-Cost Exoskeleton, due to their often invasive or inconvenient nature of being required to be directly attached to the wearer's skin, as well as having high cost. For this reason, they have not been considered for this Exoskeleton Implementation.

2.8.3 Force Sensors

Force Sensors are the most commonly used sensor method within the Literature Review, with 91 Instances. Force Sensors most commonly took the form of Force Sensitive Resistors, Strain Gauges, Load Cells, and Force Plates although there were many instances of more difficult to identify sensors typically referred to as either "Force Sensors" or "Torque Sensors", or "Force-Torque Sensors". Their most common roles were the detection of foot to ground contact as a way of signalling gait phases and detecting the user pushing against the exoskeleton to move it. Pioneered by BLEEX [8], this method detects the force applied between the human limb and exoskeleton. Essentially, as the patient moves their leg it pushes against the exoskeleton, the exoskeleton would detect this movement and move in the direction the leg applied force in, thereby moving with the leg in order to support it. This method has more latency when compared with EMG or other methods that take direct body signal data significant enough amounts of movement must occur before the sensor would be capable of detecting it, although will be less susceptible to false or difficult to read signals due to functioning on the more macro-level of physical leg movement [77]. The method still suffers from not being very effective if the patient cannot move their own legs. For the purposes of this project however this is less of an issue, as it is assumed that the wearer has at least a marginal amount of walking ability. Table 6 provides some examples of identifiable Force Sensitive Resistors, Load Cells, Strain Gauges, and Force Plates from the Literature Review.

Table 6 - Force Sensors within the Literature Review

Study	Sensor Type	Sensor Name	Purpose	Cost
[78] [47]	FSR	Interlink FSR 402	Change State / Insoles	£6.30 [79]
[80] [81] [82]	FSR	Tekscan A201	Ground Reaction Force	£66.60 [83]
[84]	FSR	Tekscan A401	Gait Phase	£80.54 [85]
[86]	FSR	Sparkfun FSE-SEN-09376	Insole Force Sensing	£10.27 [87]
[88]	Load Cell	Phidgets 50kg Micro Load Cell	Strain on Cable	£5.75 [89]
[90]	Load Cell	Futek FSH03904	Strain on Cable	£737 [91]
[90]	Strain Gauge	Omega MMF003129	Torque Applied to Ankle	£121.64 est. [92]
[93]	Force Plates	Bertec 6 DoF Force Plate	Testing	£10k - £16k [94]

Force Sensitive Resistors

Force Sensitive Resistors (FSR's) detect force using a conductive film, which is deformed through imparted force such that it touches a substrate layer below it, altering the resistance of the film which can be measured to determine the force applied to it. Force sensitive resistors benefit from being simple, small, thin, and inexpensive and were the single most commonly used sensor within the literature review with 21 instances. Many Force Sensitive Resistors seen within the Literature review were Interlink FSR 400's and Tekscan A200's, such as the Interlink FSR 402 costing ~£6.30 [79] or Tekscan A201 costing £66.60 [83]. This low price however comes at the cost of imprecise readings, being non-linear and having as much as 25% inaccuracy [95]. In addition to this Force Sensitive Resistors in such insole environments have been seen to plastically deform due to the force applied to them, skewing results over time [96]. As with all sensors, there are more expensive implementations as well, although they did not see much use.

The most common FSR implementation consisted of one or more FSR's placed on a foot plate under or within the wearer's shoe to form an insole, allowing the force the wearer applies to the ground, as well as gait phases such as the heel of the foot touching the ground during heel strike, or the toe leaving the ground during toe-off to be noted, this can be used to determine the gait phase of the wearer, examples of which are listed in Table 6. While only used once within the literature review, Force Sensitive Capacitors are a similar alternative to Force Sensitive Resistors, and in some instances have been seen to produce more linear outputs less prone to plastic deformation [97].

Load Cells

Load Cells are more advanced methods of detecting imparted force and are seen in many Industrial Applications. These can be in the form of Hydraulic or Pneumatic Load cells that measure the change in pressure of their fluid/gas when force is applied upon them as well as more traditional solid blocks which function very similarly to Strain Gauges with structures distorted by force to alter their electrical resistance, although they tend to be in the form of blocks that tension/compression are applied to as opposed to thin strips applied to a surface. Load Cells appear have 15 Instances within the Literature Review, and have quite a significant variance in cost, from the Phidgets 50kg Micro Load Cell used in [88] costing ~£5.75 [89] , to the Futek FSH03904 Load Cell used in [90] costing ~£737 [91]. The Most common usages of Load Cells were to measure the force applied by an actuator or a component of an actuator, with both prior examples measuring the tension applied upon Bowden cables, although the Forsentek FL-25 Load Cell used in [68] measured force applied by an actuating spring.

Strain Gauges and Flex Sensors

Strain Gauges are similar to FSR's in having a variable resistance due to applied force, although do so via bending rather than compression to cause a change in resistance within its conductive material. These resistance differences are however usually incredibly small, and as such multiple strain gauges can be used on a single strained object (usually on the inner and outer side of the object) and connected to form a Wheatstone bridge circuit that amplifies the resulting output. This Wheatstone bridge implementation is seen in several examples such as [47] and [90], the latter using them to measure torque across the ankle.

Strain Gauges were also occasionally referred to as Flex Sensors, which seemed to have a similar purpose such as the Bend Labs Flex Sensor used in [98] costing £120 [99]. Strain Gauges were far less commonly used compared to FSR's, with only 6 instances appearing within the Literature review, their primary usages being measuring the torque or bend across a joint.

Force Plates

Force Plates were the least common force sensor within the literature review with only 4 Instances. Unlike the prior sensors these are external to the exoskeleton, and so primarily used for testing purposes. These Force plates are used exclusively for collecting Ground Reaction Forces in multiple axes during exoskeleton testing, such as in [100] where four Kistler Force Plates measured the Ground reaction force of the wearer's feet and crutches in 6 Degrees of Freedom.

Information regarding cost of these components was limited, although as they are external and used only for testing, they would not be needed for each individual exoskeleton, another example [93] used a Bertec Force Plate costing £10,000 - £16,000 [94].

Summary

For a Low-Cost Implementation, Force Sensitive Resistors, Strain Gauges, and Load Cells all have potential usage within designs. For the development of the Low-Cost Exoskeleton, Force Sensitive Resistors were considered for usage in Actuator Force and Ground Force detection.

2.8.4 Motion Sensors

Motion Sensors were the third most common exoskeleton sensor type with 33 Instances. They can be broken up into Internal, exoskeleton mounted sensors of which Inertial Measurement Units were the overwhelming majority, and external Motion Detection or Analysis Capture devices. Detecting the Motion of the wearer is a useful component in creating and testing simulations, for example, in comparing how the wearer's gait changes with and without the exoskeleton. They can also be used to track the current location of the wearer's legs in space, as opposed to just the direction or amount of movement that is occurring. Some examples of Motion sensors from the literature review are listed in Table 7.

Table 7 - Motion Sensors within the Literature Review

Study	Sensor Type	Sensor Name	Purpose	Cost
[65]	IMU	APDM Opal Sensor	Foot-Mounted Gait Speed	£3500 [101]
[102]	IMU	Adafruit LSM9DS0^1/1^2	Acceleration/Rotation Data	£12^1 £20^2 [103]
[98]	IMU	Sparkfun 9DoF Razor M0 IMU	Shank Movement	£30 [104]
[100] [105]	Motion Capture	Mac3D Motion Cameras	Motion Capture	£16,420 [106]
[107]	Motion Capture	Kinect	Image Capturing	£35 [108]

1,2: Numbers relate to relevant product.

Inertial Measurement Units

The Inertial Measurement Unit consists of an Accelerometer, Gyroscope, and Magnetometer. With Accelerometers measuring the Acceleration of the device within a specific dimension, the Gyroscope measuring the Roll, Pitch, and Yaw of the device, and Magnetometers measuring the direction and strength of magnetic fields. Most IMU's contain one of each device for the X, Y, and Z dimensions, allowing for the movement, rotation, and orientation of the device to be known, to some precision. Although often requiring recalibration and protection from non-earth-caused Magnetic fields to reduce drift and error. It was a relatively common sensor within the Literature Review with 26 Instances.

IMU's are used to measure kinematic data of walking, being placed upon the foot, shank, and thigh joints such as in [109] [110], for recognising gait motion or phases. Other papers such as [111] used the IMU for balance or posture control.

Within the literature while there were some high precision Inertial Measurement Unit devices such as the Tech IMU v4 used in [110] or the APDM Opal Sensor used in [65] costing ~£3500 [101], there were also several examples of far lower costed MEMS (Microelectromechanical Systems) devices such as the LSM9DS0/1 used in [102] costing £12-£20 as part of a breakout board [103], these devices have a small form factor, power requirement, and cost, and would be useful in situations where absolute precision is not required.

Motion Capture Devices

Motion Capture Devices, typically motion tracking cameras were used for primarily testing purposes in 7 Instances within the Literature Review. These tended to make use of multiple cameras that surrounded a wearer who would sometimes be fitted with identifying markers to aid in tracking for example, both [100] and [105] made use of 6 Mac 3D Motion Analysis Cameras. Being Test components they would not need to be purchased for every exoskeleton, although still present an initial cost of ~£16,420 ($20,000) [106] in total, Other papers made use of similar Vicon Cameras or a Kinect. These devices combined with markers allowed for accurate trajectory tracking of the wearer allowing later analysis of how their gait may have been affected by the use of an exoskeleton.

Summary

The low financial, weight, energy costs of the MEMS Inertial Measurement Unit make them an ideal fit for a Low-Cost Implementation to measure the motion of the wearer. Larger IMU's and Motion Capture devices were commonly used for testing functionalities or validation which would likely be unnecessary for simpler implementations.

2.8.5 Pressure Sensors

Possessing no subtypes, Pressure Sensors were sensors used alongside Hydraulic and Pneumatic Actuators to measure the pressure within them both as a safety feature to shut the system down in situations of excessive pressure build up and also to provide an indication as to the force the actuator is applying, as this would be proportional to its internal pressure. Therefore, this pressure sensor can be used to control the amount of force the actuator provides; if the sensor provides a pressure value less than desired pressure can be increased, and vice versa. Some examples include the Honeywell 100PGA5 Pressure Sensor costing £41.64 [112] used in [113]. As was concluded in [53], Hydraulic and Pneumatic Actuators were generally not appropriate for Low-Cost Implementations due to unacceptably high weight and reliability costs, and by extension therefore neither would these sensors be appropriate.

2.8.6 Rotary/Linear Sensors

The Second most common Sensor within the Literature Review, with 63 Instances. Rotary/Linear Sensors consist of methods by which Rotation or Linear motion amounts could be measured. Their specific component subtypes consisting of Encoders, Goniometers, and Potentiometers. Their universally common usages include measuring Joint Angles and the amount of extension of linear actuator designs. Joint Angles specifically allowing for an exoskeleton control system to know the position of the wearer's legs, significantly aiding in detecting Gait Phase and State. Some examples are listed in Table 8.

Table 8 - Rotary/Linear Sensors within the Literature Review

Study	Sensor Type	Sensor Name	Purpose	Cost
[114]	Rotary Potentiometer	Bourns 6637S-1-502	Ankle Joint Angle	£41.96 [115]
[116]	Rotary Potentiometer	Bourns 3382	Joint Angle	£2.85 [117]
[93]	Linear Potentiometer	Midori Precisions LP-100F	Joint Position	£372.28 [118]
[81]	Goniometer	Delsys S700 Goniometer	Joint Angles	£1500 [119]
[48]	Goniometer	Biometrics SG110	Angle Joint Angles	£520 [70]
[102]	Rotary Encoder	Netzer DS-25	Joint Angles	£550 [120]
[121]	Rotary Encoder	1024ppr MILE	Motor Position	£150 [122]
[123]	Rotary Encoder	Avago HEDL-5640	Motor Position	£48.96 [124]
[123] [125]	Linear Encoder	Renishaw RGH-24	Actuator Position / Torque	£552.83 [126]
[127]	Linear Encoder	Renishaw RGH-41	Linear Actuator Position	£373 [128]
[102]	Linear Encoder	Maxon ENX16	Compliant Element Length	£88 [129]

Encoders

Several different types of Rotary/Linear Encoders, those being Incremental, Absolute, as well as Optical and Magnetic encoders were noted within the Literature. Incremental Encoders measure their change in position and are as such unaware of their starting position, requiring realigning after each use.

Absolute Encoders meanwhile recognise changes in position even whilst unpowered, although tend to be more expensive. Optical Encoders and Magnetic Encoders are ways by which the prior encoder types may be implemented. With optical making use of light through a coded disk, and Magnetic using a separate magnetic disk of north/south pole slits and a latch that flips to indicate a change in magnetic field with each north/south change. The Majority of the 39 Encoder Instances within the Literature Review were Optical Incremental Encoders, many were directly embedded within or otherwise a part of an actuator, typically a rotary electric motor, and so used to directly measure the movement of that actuator, for example in [121] the electric motor is directly controlled by an attached 1024ppr MILE Maxon Incremental Encoder, or the Linear Motor in [127] controlled by a Renishaw RGH41 Linear Optical Encoder . Other examples of their usage include measuring joint angle such as the Netzer DS-25 in [48] and compliant element length such as the Maxon ENX16 Encoder costing ~£88 [129] in [102], similarly to a Rotary/Linear Potentiometer equivalent. The Output of an Encoder may not be immediately readable by a control system, and they are sometimes paired alongside some kind of interface that converts the Encoder’s output into easily readable data. A Common example used in papers such as [123] was the PC/104, costing around £380.

Potentiometers

Rotary/Linear Potentiometers function as a resistor that changes resistance when rotated or extended/contracted with the change in resistance either being linear or logarithmic. These incredibly simple devices are somewhat common within the literature review with 19 Instances. Rotary Potentiometers are used almost exclusively to measure the change in angle at the Thigh/Knee/Ankle joints, such as the Bourns 6637S-1-502 costing ~£42 [115] used to detect the Ankle Joint angle in [114] or Bourns 3382 costing £2.85 [117] used to detect Joint angles in [116]. Linear Potentiometers also saw use to measure leg position around a joint such as the hip [130] as opposed to angle, as well as being used to measure the length of a spring as part of a Series Elastic Actuator to determine its compression [131]. In General, Potentiometers have low sizes, weights, and power usages, therefore making them excellent for Low-Cost Implementations.

Goniometers

Only having 5 Instances within the Literature Review, Goniometers are specifically used to measure angles, and as such were exclusively used for measuring Joint Angles. Within exoskeletons Digital Goniometers appear to be entirely specialist in nature, with the Delsys Goniometer and Adaptor costing £1500 [119] used in [81] and Biometrics Twin-Axis Goniometer costing £520 [70] used in [113], both used to measure in two orthogonal planes at once, with a design consisting of two attachment points and a wired connection. Whilst this device is relatively small and form fitting, weighing as little as 20 grams, its cost is quite high.

Summary

For A Low-Cost Implementation, Potentiometers are a clear standout of being inexpensive, small, and possessing low power requirements, however both Goniometers and Encoders of at the lower-end of cost have potential for Low-Cost as well, especially the encoder due to their close connection to the commonly used Rotary Electric Motors. As proven by the ubiquity, Rotary Sensors likely would be appropriate for Low-Cost Implementations.

2.8.7 Sensor Summary and Justification

From reviewing the Sensors commonly used within literature, Force Sensitive Resistors, Inertial Measurement Units, and Potentiometers seem to be ideal for a Low-Cost implementation. Allowing for an effective combination for detecting the state of the wearer and/or the gait stage they are currently in when walking. Force Sensitive Resistors could be placed both on the feet where they may detect whether the foot grounded or not, whilst potentiometers can be used on the knees and ankles to determine how the wearer is standing, allowing a control system to detect if the wearer is standing, sitting, or crouching, or possibly if they are in an unstable position.

When reviewing the usage of sensors within the Literature Review as a whole, Sensor usage is very diverse regardless of the type of actuator used within the exoskeleton. Some obvious exceptions exist, such as Pressure Sensors being overwhelmingly more common within Hydraulic and Pneumatic Implementations.

Table 9 and Table 10 describe the total number and distribution of sensors per actuator type, in situations where an exoskeleton had multiple actuator types such as a Series Elastic Actuator consisting of both Electric Motors and Compliant, the sensor was listed for both. Sensors are far less comparable that actuators due to there being less of a unifying metric of comparison between sensor types that specialise in detecting different stimuli, as such an objective comparison will not be attempted as it was for the actuators.

Table 9 - Total Number of Sensors per Actuator Type

Actuator Type	Cable	Compliant	Hydraulic	MagBrake	Motor(L)	Motor(R)	Pneumatic
Total Sensors	27	41	10	6	5	149	49

Instead, As stated prior, as the exoskeleton that will be designed in this thesis to be more passively assistive than actively rehabilitative, it will instead be considered which sensor types and subtypes would be most appropriate for the goal of the exoskeleton. Firstly, external sensors for collision avoidance and automatic balancing have been considered unneeded as these are more appropriate for exoskeletons designed to fully guide the movement of a patient who may not be able to walk on their own. Otherwise, the sensors seen as most effective for a Low-Cost Implementation are Force Sensitive Resistors, Inertial Measurement Units, Encoders, and Potentiometers as each of these devices is capable of being inexpensive, small, and low-power whilst collectively allowing for a large amount of information about the user, such as when they take steps, the nature of their movement, and the angle of their joints, which alone would be enough to simulate their walking pattern accurately.

The Final section of the Literature Review will summarise the Control System, which acts as the intermediary to take in the sensor's provided inputs and use them to control the exoskeleton's actuated outputs.

Table 10 - Percentage Usage of each sensor per Actuator Type

Sensors	Cable	Compliant	Hydraulic	Magbrake	Motor(L)	Motor(R)	Pneumatic
EMG		15%		33%		11%	14%
EEG		5%				3%	
Force Sensor	7%	17%	30%	16%		11%	17%
FSR	11%	10%				10%	8%
Load Cell	15%	5%			40%	7%	2%
Torque Sensor		2%		17%	20%	7%	2%
Strain Sensor	7%	5%				2%	6%
Force Plates	4%	2%	10%				2%
IMU	22%	5%		17%		13%	12%
Motion Camera	15%	2%				3%	2%
Pressure Sensor	4%		40%			1%	17%
Encoder	15%	22%	10%	17%	20%	21%	6%
Potentiometer		10%	10%		20%	9%	6%
Goniometer						2%	6%

2.9 Control Systems

The act of keeping an exoskeleton balanced quickly became the topic of importance when exoskeletons were first used in a rehabilitative function. Initial research into areas such as zero moment points were introduced in 1968 by Miomir Vukobratović [132] and stated that there is a point where the foot contacts with the ground such that the horizontal force component is counteracted by friction between the foot and ground. This horizontal force component consists of the weight of the individual applied horizontally through the foot to ground contact as a result of this contact point not being vertically aligned to the individual's centre of mass, in addition to horizontal movement of the foot itself due to walking. This concept therefore helps to make sure that the foot does not overstep to the point where it would slip or otherwise trip over. This, along with other concepts such as the haptic feedback used with the Hardiman robots were some of the earliest examples of control system concepts which go along with the earliest examples of exoskeletons themselves [7].

Unlike the prior Literature Review Sections, Control Systems are far more than the hardware that they run on; therefore, this section will summarise common examples of concepts implemented within control systems in addition to the hardware that drives them, including examples of such implementations noted within the Literature Review. At least some amount of Control System Hardware and Implementation information was identified within 70 of the 127 papers within the Literature Review.

2.9.1 Control System Components

Control System hardware can be split into four primary groups, those being "Controllers", "Microcontrollers", "Peripherals", and "Software". Both Controllers and Microcontrollers may perform the same function, that being controlling the exoskeleton, however the Microcontroller being a notably smaller and less powerful device, Microcontrollers may also be used as a way of collating data from sensors that it then sends back to an off-exoskeleton processing device such as a laptop which may act as the primary Controller. Peripherals consist of other components that may aid in the processing of data, this may be boards that control motor actions or decode sensor data such as the Addi-Data APCI1710 Encoder Board used in [127] to decode Encoder data for use in the control system, they may also allow for wireless communication between the exoskeleton and an external processing device such via as a Bluetooth connector. Software is anything that is run on the Controllers, either for managing the exoskeleton, collecting data, or so on.

Controllers and Microcontrollers

There were several examples of both the larger and more powerful Controller devices and smaller and less powerful Microcontroller devices within the Literature Review. These devices run programmed software or simulations in order to control or monitor the movements of the exoskeleton, in some cases feeding sensor data to an offboard device for further processing. Table 11 and Table 12 give examples found within the Literature Review.

Table 11 - Controller examples from Literature Review

Study	Name	Details (Power, Processing, Weight)	Cost
[80] [88] [123] [133] [134]	PC/104	5V/5.4W, 1-2GB SDRAM + 1.1GHz - 1.6GHz CPU, 212g [135]	£370 (varies) [136]
[121] [131] [137]	NI CRIO-9082/ 9074/ 6259	30V/75W, 2x256kb Cache + 4MB Cache + 1.33GHz-2.44GHz CPU, 3.1kg. [138]	No longer available
[90]	SpeedGoat	12V/15W, 4GB DDR3 RAM + 1.6GHz Intel Atmo x5-E3940, 800g. [139]	~$2,000 (est.)

Table 12 - Microcontroller examples from Literature Review

Study	Name	Details (Power, Processing, Weight)	Cost
[52] [72] [98] [140]	Arduino Uno/Mega Microcontroller Board	5V, 8kb SRAM + 16MHz ATMega328/ ATMega2560, 25g/37g	~£20 [141]
[142]^1 [143]^2 [144]^3	ATMega16^1/128^2/328^3 Chip	3V-5V, 1kb^1/4kb^2/8kb^3 SRAM + 16MHz Processor, negligible weight.	£5 [145]^1, £2.20 [146]^3, £12 [147]^2
[78] [50] [148]	TMS320F28335 Micro Controller Board	3.3V, 68kb SARAM + 150MHz C28x Processor, ~28g	£11.75 [149]
[130] [150]	Beaglebone Black Microcontroller Board	3.3V, 512MB DDR3 RAM + 1GHz Cortex-A8, 40g	£41 [151]
[150] [152]	STM32F7 Chip	3.3V, 256kb SRAM + 216MHz Cortex M7, negligible weight	£10 [153]
[86]	Huzzah32 Feather Microcontroller board	3.3V, 520kb SRAM + 240MHz ARM Cortex-7, 6.8g	£16.50 [154]
[155]	Teensy 3.6 Microcontroller Board	3.3V, 256K SRAM + 180MHz ARM Cortex-M4, Unknown Weight [156]	£24.21 [157]

1,2,3: Number refers to associated product

From the Literature Review the most common implementations were all Embedded Systems, either the PC/104 Controller Board, of which the Diamond Systems Aurora PC/104 and Cool RoadRunner LX800 were the most commonly used. For Microcontrollers the Arduino Uno Development board and ATMega Chip were the most common. The Limited space of the exoskeleton design sees many exoskeletons that employ small and inexpensive Embedded Microcontrollers as opposed to larger controller designs.

For Stationary Designs, such as [90] the controller is not required to be on board the exoskeleton itself, here a SpeedGoat is set offboard the exoskeleton running a Real-Time MATLAB Simulink Controller that tracks the joint torque of the wearer, comparing the actual torque value to a desired torque determined based on an iteratively learning ideal profile and the current percentage position of the wearer's stride calculated from average recent stride time. For the purposes of a Low-Cost Methodology, the smaller Microcontrollers seem the more appropriate choice, as the Exoskeleton design is intended to be mobile such a control device would ideally be capable of performing all necessary functions onboard.

For the use of Prototyping, the ubiquitous Arduino would seem an ideal choice, however its comparatively low processing power may limit what control methods can be explored. Within the Literature the usages of the Arduino or its ATMega microprocessor included Sensor Data Collection that would then be sent wirelessly to another device, such as in [72], as well as relatively simple control functions such as in [140] where the Arduino took in IMU, Load Cell, and Pressure Sensor data to turn on and off the solenoid valves to a Pneumatic Muscle in line with the flexion and extension of the wearer's knee to aid in moving the ankle joint.

Size could be further minimised in comparison to Development Boards by using the component Chips themselves as part of a custom-made PCB, and thereby only using any required functions, although the simplicity and convenience of the development board makes it a superior choice for prototyping as making additions and removals has little to no cost as features are more readily available.

With the exception of the specific languages related to development boards, such as Arduino on the Arduino Uno or the EspressIF IDF for the Huzzah32 Feather, MATLAB's Simulink was the most commonly used software component that ran simulations upon the control system hardware or was used in the designing of a control system that did, such as in [80] where Simulink was used to design an Adaptive Network-Based Fuzzy Logic Controller made of a Neural Network and Feedback System. The software that runs on these hardware components will likely follow some of several of these common Control System Concepts identified within the Literature Review.

2.9.2 Control System Concepts

How exactly an exoskeleton functions, and the protocols which it makes use of can vary wildly depending on the purpose of the exoskeleton as well as the maker [158]. Control systems exist to govern how the exoskeleton itself will function, both in low-level movement of actuators and high-level decision making of specifically which movements may need to be made in the future or in response to external events. The level to which to control system controls the exoskeleton is dependent on its purpose, a light, assistive exoskeleton is not required to respond to the minutia of exact foot placement and counterbalance strategies that a full, rehabilitative exoskeleton designed to support those with paraplegic SCI would. None the less, there are several concepts which both may make use of. The Universality of these concepts makes them all feasible to use within a Low-Cost Exoskeleton Implementation, although not requirements.

Hierarchical Systems

A design methodology that splits control systems into Task-Level, High-Level, and Low-Level [159] systems. Task-Level acts as the highest level intent for the control system such as rehabilitation of the user by providing assistive force when necessary; the Task Level controller may therefore adjust the maximum amount of assistance allowed to be given based on the overall performance of the wearer, affecting the assistance levels given by all lower levels, it may also decide to completely change how assistance is given based upon set trigger condition, such as if detecting a fall.

The High-level controller is an implementation of a specific strategy for achieving a task such as applying an impeding force to the user if their gait strays too far from the ideal model. It then manages the level of impedance based on the task-level's long term performance suggestions as well as based on the user's current performance (e.g. if they are currently very far from the ideal model, use more impedance).

The Low-level controller directly controls what occurs in the exoskeleton such as the movements of the actuators themselves, such as through PD (Proportional Derivative) and PID (Proportional Integral Derivative) position controllers. Most complex exoskeleton designs within the Literature Review implemented some sort of Hierarchical system.

Finite State Machines

A Finite State Machine consists of a number of different states that can be switched between if conditions are met via the state machine's inputs, with each state having a varied effect on how the control system controls the exoskeleton. States connect with Transition Functions, determining the condition that needs to be fulfilled to go from one to another. For example, in [78] the Finite State machine shifts between a "Lifting" and "Lowering" motion depending on the Rising/Falling edges of the Torque Sensor and Inertial Measurement Unit. There were at least 10 confirmed uses of Finite State Machines within the literature review.

Each state in the state machine does not have to lead to every other state, if it is assured that the machine cannot get stuck in a state, then a state machine could look more like a loop than a web, as each state leads into the other in a predictable fashion. For example in [48] the Finite State Machine has 9 states representing different phases of the gait cycle, going through 2-9 whilst active in order before repeating from the start again (stand, shift left, half step right, full step right, lead right foot, shift right, half step left, full step left, lead left foot) with states shifted between when the centre of mass of the wearer and exoskeleton shift appropriately.

PID Controllers

PID Controllers, or Proportional Integral Derivative Controllers are used to correct error based on closed loop feedback. The PID Controller will compare the input sensor "real" value to a desired value and attempt to correct error between the two. The Proportional Component applies a proportional change to the input dependant on the size of the error, the Integral considers the sum of the error from the present and recent past to reduce the effect of fall-off that a proportional component will have due to slowing down as it reaches a desired value, and the Derivative considers the rate of change of error into the future to counteract the overshoot caused by the integral component if the rate of change of error ends up becoming too large. The values of each component (which may be weighted differently by a designer to modify the significance of their effects) are summed together and applied onto the input, to create an output. This output is then used as part of a feedback or feedforward loop to see how close this new modified output is to the desired output. If a component is weighted 0, it is disabled and typically omitted from the PID acronym (making a "PD" controller for example with no Integral component).

A Higher control system level may adjust the gains of the PID controller during the course of the exoskeleton as a way of optimising an exoskeleton's responses to its wearer.

If the PID is used to control physical hardware these will have upper limits as to the rate they are capable of correcting themselves. If the PID would push the hardware beyond these limits in trying to correct itself it would built up a "fake" error where the PID sees itself in error for longer than it expects, leading to massive overshoot when it arrives at the ideal value whilst it burns off this fake error. Clamping PID values to a maximum can resolve this and is typically seen most in the integrator component. The derivative component meanwhile can risk overvaluing high frequency noise values, resulting in error. To prevent high frequency noise causing issue, a frequency cut-off can be used such as a low-pass filter, to attenuate any frequencies above a set value set by the designer.

In the Literature Review, PID, or subsets of these controllers were seen quite commonly, with at least 20 confirmed uses within the 70 papers where Control System information could be identified. There was one example of a P Controller, [90] where a motor proportionally adjusted cable lengths dependant on the desired force to be applied to them, although generally PI, PD, and PID were more common. Examples for PI Controllers include [100], [160], and [161] with a total of three implementations, PD Controllers such as [162], [163], and [164] with a total of seven implementations, and finally PID Controllers with 9 implementations such as [165], [166], and [167].

Ideal Models and Neural Networks

An Ideal Model is a method of creating a target goal for a control system to reach. Consisting of an idealised or "healthy" stance or gait that can then be compared to the gait of the user. The model may cover the entire body, or part of it and will usually only be appropriate for a particular action. A common example of this is a healthy gait model, this will be recorded using the same or a similar exoskeleton to the one being used, using a similar array of sensors to collect data. This data is then saved and key elements such as the different gait stages identified. These same elements can then be identified by the control system during actual use, and relevant sensor data between the ideal and actual results compared. For example, if the knee angle of the wearer is more flexed than it should be in the healthy model, this can indicate to the control system that it needs to try and make the knee extend more, as well as by how much in order to reach a healthier gait. A PID System controller may act as a method to provide the appropriately proportional push or pull values to reduce error. A Model could also be used to predict how the wearer will move in the immediate future by comparing the user's current position to the likely immediate future of the ideal model. Examples of uses for ideal models are described in 2.9.3.

The model does not need to be a singular set of data representing an ideal gait, for example it could be a Neural Network trained on a large amount of ideal gait data that then uses the wearer's sensor inputs to predict a likely future value of their gait. For example in [76] a Neural Network is used that takes in Hip and Knee angle data that outputs a modifier to a matrix generated from EMG data on the wearer's movement to account for the user's limb posture, which is then used to predict future motion.

As Described by Caldas R et al in [168] up until 2016 some of the most common Machine learning Implementations within exoskeletons were either Artificial Neural Networks (ANN's) or Hidden Markov Models (HMM's). Common use cases were determining Gait Events such as heel strikes and toe off as well as swing/stance positions, Spatiotemporal Parameters such as Stride and Step/Stance Times as well as Gait Velocity and Cadence, and Joint Angles [169]. Earlier examples of Joint Angle prediction and measurement using machine learning can be seen in [170] which made use of a General Regression Neural Network (GRNN) to estimate the Hip, Knee, and Ankle Angles based off of motion sensors at the hip, knee, and ankle joints. Meanwhile Sun et al [171] describes other examples of Machine Learning up to 2020, such as Reinforcement Learning [172], Support Vector Machines [173], etc.

Convolutional and Recurrent Neural Networks are two types of Neural Network that appear to be quite commonly used for neural network-based control systems within exoskeletons. Examples for CNN's include [174] which separately implemented both a Convolutional Neural Network and a Multi-Layered Perceptron (MLP) onto an ESP32 Microcontroller, collecting data using a Shank Mounted IMU to predict foot angular velocity roughly 200ms into the future.

For RNN's, many examples run on high-power computers, laptops, or dedicated controllers, with regards to implementations, A type of Recurrent Neural Network, the Long Short-Term Memory Model (LSTM) as seen in [175] used multiple LSTM Hidden layers each of hundreds of units, whilst allowing for improved accuracy it also increased the size of the model, as Microcontrollers are limited for memory space a reduced model size was a necessity. LSTM's improve upon base RNN's through reducing the effect of vanishing gradients that cause older information during training to significantly decrease in effective influence over the end result compared to newer information [176].

A more recent Neural Network known as a Transformer, first proposed in 2017 in [177]. There have been some examples of the use of Transformers within exoskeleton Control Systems, such as [178] and [179]. By default, Transformers do not suffer issues of attenuation of data that RNN's have that would result in older information having less effect on predictions. However this is achieved through processing information simultaneously, which is more computationally expensive, and has the risk of removing temporal significance, which is essential for time series data. Although this is not a hard limitation, which may be solved such as via encoding the arrangement of information into input data itself.

In terms of papers that use Neural Networks to predict the wearer's knee angle, [180] used Shank and Thigh Inertial Measurement Unit (IMU) Data provided to an ANN model to predict walking speed and knee angle, whilst [181] used IMU and Electromyography (EMG) data provided to an RNN model run on an STM32F4 Microcontroller (costing £16.40 [182]) to predict joint torque and moment, although did not use LSTM/GRU due to technological limitations. For Sensor usage paired with control systems, papers such as [183] used EMG and Knee angle data collected from sensors via a ATMega328p Microcontroller and processed on a laptop. Meanwhile, [184] make use of an LSTM provided data by Inertial Measurement Units (IMUs) placed on the lower leg to predict lower-limb joints, which alongside Potentiometers are commonly used, inexpensive sensors in exoskeletons.

A summary of such Neural Network methods is seen in Table 13. Noted is a high proportion of EMG and IMU Sensor-Based implementations. This may be as these implementations less directly measure wearer movement than direct Joint Angle Measurements, which may make kinematic simulation more difficult.

Table 13 - Summary of Various Neural Network Implementations.

Study	Model	Hardware	Sensors	Prediction Purpose
[170]	GRNN	~	Xsens Motion Trackers	Gait Kinematics
[172]	Reinforcement	Intel I7-8700	EMG	Hip/Knee/Ankle Joints
[173]	Support Vector	~	EMG	Motion Intention
[174]	CNN + MLP	ESP32	IMU	Hip/Knee/Ankle Joints
[180]	ANN	Microcontroller	IMU/Encoders	Knee/Ankle Joints
[181]	RNN	STM32F4	IMU/EMG	Knee Joint
[183]	LSTM	Macbook Pro	EMG/Encoder	Knee Joint
[184]	LSTM	Intel I7	IMU	Hip/Knee/Ankle Joints
[178]	Transformer	~	EMG	Hip/Knee Joints
[179]	Transformer	Intel I7 16GB RAM	IMU/EMG	Motion Intention

2.9.3 Control System Implementations

The Control System primarily represents the Task- or High-Level implementation methods that guide how an exoskeleton controls actuation in the aim of providing assistance to the wearer. There are a considerable number of implementations, and accounting for all would be beyond the scope of this Thesis but are explored further in various review papers on the subject such as [158], [185], and [186]. Instead, some notable control systems seen within the Literature Review and elsewhere will be looked into further.

Gait Phases/Events

As Described in 3.2.1 the Human Gait can be split into several different phases. These can be identified through Gait Events such as the Heel Strike and Toe off that might be detected by a Ground Reaction Force sensor such as a Force Sensitive Resistor such as [84] which used two FSR's to detect the Stance and Swing Phases based off of foot placement. [187] meanwhile used FSR's to detect four states (heel-contact, toe-contact, heel-off, and toe-off) and used a Finite State Machine to decide how the exoskeleton would act during them.

A Control System could also use identifiable patterns in walking such as the swing of the knee in the Swing Phase detected via a Potentiometer measuring knee angle, or both foot and joint angle methods could be used together for additional reliability such as [188], which used four modes determined by a Finite State Machine in a High-Level Controller using input data from Ground Reaction Force, Knee Angle, and an Inertial Measurement Unit, with a low-level controller to move the actuator dependant on the phase. [189] meanwhile determined three phases: Stance, Early Swing, and Late Swing, and assisted except during Early Swing using a Finite State Machine to determine the thresholds for when each phase occurred.

This concept is commonly used as the basis for control systems that wish to provide assistance to the wearer during a specific portion of the gait cycle by identifying the phase in which this assistance takes place whilst it is occurring and then applying assistance. Within the Literature Review, there were 11 identifiable instances of Gait Phases being used as part of the control system.

This implementation likely would be appropriate for a Low-Cost Implementation, some of the designs seen within the literature review capable of reliably detecting Gait changes using low-cost components such as Potentiometers, Force Sensitive Resistors, and Inertial Measurement Units, allowing for a system that provides help when needed. It may however struggle if only limited to walking movements, for example if the wearer were to be transferring from a standing to sitting position or vice versa they would likely benefit from assistance in making this movement which may require separate "Movement State" identification to give more general usage and allow an exoskeleton to function in a more diverse environment.

Challenge Based and Assist as Needed Algorithms

Whilst an Assistive Exoskeleton's primary focus is on the improvement of motor function, in other words adapting a user's current gait to that of a more preferable, healthier one, if it merely attempts to fix all errors within the user's gait this may lead to a level of passive allowance in the wearer, or a subconscious expectation that the exoskeleton takes on the burden of correcting innate gait deficiencies.

Two methods of addressing this could be to deliberately introduce acceptable errors within the system for the wearer to resolve, such as seen in [190] which introduced deliberate haptic disturbances during tasks. This "Challenge-based" control algorithm effectively allows an introduction of forced difficulty into a control system. It may for example make a user have to put in more effort in order to move or force the used of impaired limbs by making use of unimpaired limbs harder, such as a steering wheel that resists if the unimpaired limb is used.

An Alternative but similar implementation known as "Assist-as-Needed" or "Performance Based" Control Systems have a varying level of assistance based off of the user's current walking quality. Such a method having some form of "ideal" gait of which the control system compares the user to, if the user emulates this gait well it provides little aid, but if they struggle it provides more. For example, through the use of haptic feedback the exoskeleton could naturally guide the user along a path of a healthy gait, applying the feedback dependant on how much the user's gait differed from its model of a healthy one, encouraging the user's leg motions to change in accordance with it.

An example from the literature review, [166], effectively created a tunnel of ideal values in the control system that formed an "ideal path" of movement, as long as the wearer followed the ideal path no assistance was provided, however if they began to deviate from it significantly a gradual assistance would be provided to bring them back in line.

Another example of this is Lokomat [18], which makes use of a "Forgetting Factor" which allows an exoskeleton to provide less assistance if several gait cycles go by with little error, with its value set between

0 and 1 determining the "difficulty", or rate of forgetting, of the system. At its most basic level it follows a formula seen in (1):

$$\boldsymbol{P_{i+1}} = fP_i - ge_i \qquad \text{(1) - Generic Forgetting Factor}$$

Where P_i = Control Parameter, i = iteration number, e_i is the error of that iteration, f = forgetting value (constant), g = gain value (constant). Effectively seeing the Control factor reduced by the forgetting factor over time. Other examples such as [191] and [192] expand on this by looking into the future such as through an Iterative Learning Controller. The Forgetting factor can be changed over time to increase or decrease the difficulty of the walking process as the user continues walking for longer periods of time.

Both of these methods benefitting from implementing additional peripherals, such integrating rehabilitation with VR games to aid in engagement in the form of "gamifying" the rehabilitative process, this has been seen to be especially effective with children as seen in [193] and [194].

When making rehabilitative devices it is important to make sure that the user drives the device and not the other way around, otherwise the user may simply develop a dependence on the exoskeleton. However, if the user has a deformed gait, then the exoskeleton must still have the ability to actively correct gait issues otherwise no rehabilitation will have actually occurred.

Impedance and Admittance

Impedance and Admittance controllers make use of an ideal gait model from which they compare the current walking gait to. Actuated joints then change the amount by which they impede or admit movement from the wearer proportional to the expected error of this movement from the ideal gait. For example An impedance-based system proportionally impedes movement that would take the user further away from ideal movement (usually once outside an allowable variance band). The exact thresholds and rates of impedance will depend on what domain the exoskeleton measures, be it elapsed time, spatial position, limb velocity, muscle activity, and so on. Some example of their use within the Literature Review include [195], [196], and [197]. Springs were commonly used as part of the Impedance control method, due to being able to immediately and passively offer impedance proportional to the force applied to them.

Admittance is effectively the opposite of impedance, providing more assistance or ease of movement whilst ideal movement is followed and providing less when it isn't. Admittance control appeared less commonly than impedance, although in one example [47] the exoskeleton would react to a detected loss of stability due to an undesired extrapolated centre of mass value and reduce appropriate joint impedances to allow for easier movements to correct the instability.

A Variance on this is known as "Triggered Assistance" where the system allows the user to move unassisted unless a specific error threshold is reached, in which case assistance is quickly introduced, such as a sudden impedance to prevent a person taking a dangerous step. Both Impedance and Admittance are in effect one implementation of an Assist as Needed control system, with their impedance or admittance effects often conditional to the state of the wearer.

2.9.4 Control System Justification

As outlined by the Aims and Objectives in 1.2.1, a Low-Cost Exoskeleton Implementation should prioritise a generalised simplicity in order to avoid the need for costly hardware or have a control system that only functions in specific circumstances, such as only whilst the wearer is walking. For example, a generalised Ideal Movement model that the exoskeleton can use as a baseline to predict the wearer's current state (walking phase, sitting/standing, etc) or next move, which can then be used to provide specific help depending on their state. Aid could then be provided only when necessary if the wearer is seen to be moving in a non-ideal manner for their state.

In effect, all of the discussed concepts and implementations within the prior two sections would be theoretically feasible to implement within a Low-Cost Control System. As such a next step would be to investigate hard and soft prototyping of an exoskeleton concept to see which would be most appropriate.

2.10 Summary

The Intent of the Literature Review was to review many real examples of developed exoskeleton technologies to understand the state of the art. To this end after reviewing the Actuator, Sensor, and Control System choices of 127 papers several conclusions can be made to potential ideal solutions for a Low-Cost Implementation.

For Actuators, Rotary Electric Servo Motors were by far the most commonly used actuator as well as the most effective candidate from semi-objective observations, these, either in their standard form or in a linear-actuated form with the aid of a ball screw, alongside some kind of compliant element to form a Series Elastic Actuator seems the clear choice for a Low-Cost active actuator.

For Sensors, there were a variety of commonly used sensors such as Potentiometers, Inertial Measurement Units, Force Sensitive Resistors, and Encoders which would likely represent the ideal Low-Cost Components. Most exoskeleton examples within the literature review did not rely solely on one of these components, due to the variety of different inputs they receive.

Therefore, for a Low-Cost Exoskeleton it would seem ideal to include Potentiometers for Joint Angles, Encoders for Motor Position, Inertial Measurement Units for Lower-Limb Movements, and Force Sensitive Resistors for actuator force and/or Ground Reaction Force.

For Control Systems on the hardware side, Lower-Cost commercially available microcontroller boards such as the Arduino would be most appropriate as a starting point for prototyping, with the potential to move on to more powerful, yet still relatively inexpensive commercial microcontroller boards. The nature of the control system itself likely requires further experimentation and prototyping to see what is and is not possible given the limited selection of prior components.

When comparing this literature review with other Reviews of similar focuses, papers such as [40] which reviewed 215 papers similarly broke down exoskeletons into separate components such as Power Source, Control Unit, Actuator, Mechanical Structure, and Sensor, in addition to considering the Human-Machine Interface (which the paper considered as part of the Control System), and the Environment. It concluded that DC motors were the most common actuator implementation, alongside Bowden cables and Series Elastic Actuators. For sensors it similarly concluded the commonality of Encoders and Force and Torque sensors (also concluded by [24]), with more visual sensor methods more commonly used for validation.

With the Completion of the Literature Review, it is clear the next step is to use the conclusions reached within it to formulate a Novel Exoskeleton Structure, Actuator, and Sensor, as well as potential control system software which can be used to aid a wearer in reducing Gait Degradation. This may be done such as to test the feasibility of the Low-Cost Methodology in a more practical manner. This will be the focus covered within the next chapter.

Chapter 3: Theoretical Development

3.1 Introduction

Using the conclusions of the Literature Review regarding the preference for rotary electric motors, low-cost sensors such as potentiometers and inertial measurement units, and the potential of a generalised movement model for a control system, this chapter seeks to perform the initial theory and prototyping to create an exoskeleton design capable of reducing gait degradation. As such, a set of Core and Additional Requirements for the exoskeleton developed during this research project shall be created, based off those initially described in 1.1.1:

Core Requirements that all exoskeletons must adhere to:

- Adaptability with a wide range of user proportions and movement/walking patterns, such as those that are over or underweight. The device must not exceed joint angle limits or considerably restrict freedom of movement. Joints of the Exoskeleton are properly aligned to the joints of the wearer.
- Sufficient comfort and support to allow for continued usage without strain, harm or undesired pressure.
- Sufficient Energy efficiency and battery life for continued usage, energy source is safe in proximity or sufficiently protected to avoid potential harm, such as shocks.
- Minimised exoskeleton weight, with necessary weights placed as to reduce inertia and avoid strain of wearer. Wearer is not made off balance by wearing the exoskeleton and wouldn't be crushed or experience considerable additional harm as a result of the exoskeleton should they fall over.

Beyond this, there is the set of Additional Exoskeleton Objectives the proposed design will aim for:

- Abide by the Low-Cost Methodology, that was outlined in 2.4.
- Be mechanically simple and inexpensive.
- Be capable of supporting and reducing degradation of users with crouch gait or other pathological gait disorders, or otherwise be capable of theoretically demonstrating this capacity.
- Be capable of assisting in knee flexion/extension and possibly other degrees of freedom, or otherwise be capable of theoretically demonstrating this capacity.

This chapter will focus on the theoretical concepts behind a proposed exoskeleton actuator design. Subsequently this chapter will propose a novel actuator design which along with an exoskeleton structure will be conceived via mathematical modelling to decide certain optimal design elements, then simulated in software using a MATLAB Simulink simulation to test kinematic feasibility as well as prove the robustness of said mathematical model by comparing it to simulated reality. This theoretical framework will be used to build a practical exoskeleton design within subsequent chapters.

3.2 Human Movement

The Kinematics of the Leg are important to consider when designing an exoskeleton. [25] recognises 8 primary degrees of freedom, 3 in the hip, 2 in the knee, and 3 in the foot, although simplifies it to 6 for kinematic models. Details of Joint Angles are summarised in Table 14.

The Hip has three degrees of freedom, consisting of:

- Flexion and Extension: Forwards/Backwards motion. Limits are 120^{o} flexion and -45^{o} Extension.
- Abduction and Adduction: Inwards/Outwards motion. Limits are 40^{o} Abduction, and 30-35^{o} for Adduction.
- Medial and Lateral Rotation: The leg's rotary motion. Limits are 15-30^{o} Medial Rotation, and up to 60^{o} for Lateral rotation. Knee Flexion can change these values.

The Knee is generally modelled to have one degree of freedom, that being Flexion/Extension:

- Flexion and Extension: Decrease/Increase in angle between thigh and shank. Limits are dependant on the angle of the hip, Flexion ranging from 120^{o} to 160^{o} depending on whether the hip is extended or flexed.
- *Medial and Lateral Rotation:* Rotation Towards/Away from the centre of the body, only possible when the leg is sufficiently flexed. Limits of Medial Rotation are roughly 15^{o}, whilst Lateral Rotation reaches maximum at 50^{o} knee flexion.

The Ankle and foot have two degrees of freedom, consisting of:

- Dorsal and Plantar Flexion: Upwards/Downwards motion of heel, limits are roughly 20^{o} for Dorsiflexion, and 40-50^{o} for Plantar flexion.
- Inversion and Eversion: Rotation of foot to bring toes closer to/further from the centre, limits are roughly 30-35^{o} for Inversion, and 15-20^{o} for Eversion.
- *Pronation and Supination:* Minor extra movement, a slight sideways abduction/adduction allowing feet to rest on angled surfaces. Limits are roughly 10^{o} for Pronation and ~3^{o} for Supination.

Table 14 - Table of Leg Angle Ranges adapted from [198, p. 4]

Joint Mobility	Range of Motion (deg)	Comfortable Range (deg)	When this occurs
Hip/Thigh	113 / -45	39.55 / -15.75	Default Standing 0^o
(Flex/Extend)	90 / - 30	31.50 / -10.50	When knee is flexed 90^o
Knee/Shank	113 (Standing)	39.55	
(Flex)	125 (Prone)	43.75	Hip is Neutral 0^o
	159 (Kneeling)	55.65	
	80 (Standing)	28.00	Hip is flexed 90^o
Ankle/Foot	38 / -35	13.30 / -12.25	Knee Neutral 0^o
(Dorsal/Plantarflex)	36 / -33	12.60 / 11.55	Knee Flexed 90^o

3.2.1 Movement Kinematics and Modelling

An exoskeleton must however be able to do more than simply move within and up to the constraints of the body. Ideally it should be able to move with the body while causing as little restriction as possible while the body itself is performing a movement action such as walking, crouching, standing, or ascending/descending stairs. It may also have to be able to deal with changes in terrain during these motions, or combinations of them, such as if a person is walking over uneven terrain, then a pre-set gait pattern will inevitably fail due to not being able to react to unpredictable changes in the amount of movement the foot will take or angle with which is it in contact with the ground. Common actions can be broken into two types, standing actions, such as standing up and sitting down where a user does not move much from the spot they start in, and moving actions, such as walking and ascending/descending stairs, where the user must be able to consistently place one foot in front of the other. These actions require far more control of one's posture and therefore assuring that their centre of mass stays within a region of stability [199].

Walking is a very efficient method of movement for a healthy person, with minimal metabolic cost for unit distance travelled. At a basic level, walking can be defined as consisting of a double support phase (both feet on the ground), and a single support phase (one foot is on the ground), additionally each leg can individually be in either a Stance or Swing phase. Stance and Swing are defined as points during a stride when the foot's heel first touches the floor (heel-strike/initial contact), and when the toe leaves the floor (toe-off/pre-swing). As seen in Figure 10, from heel-strike to toe-off the foot will be on the floor and so in stance phase; from toe-off to heel-strike the foot will be off of the floor and so in swing phase. The stance and swing phases of each leg roughly oppose each other, such that one leg is roughly in the middle of its swing phase (mid-swing) while the other is in the middle of its stance phase (mid-stance), and therefore under single support phase for the majority of gait, with times around either end of each where both feet are on the floor, a minority double support phase [200].

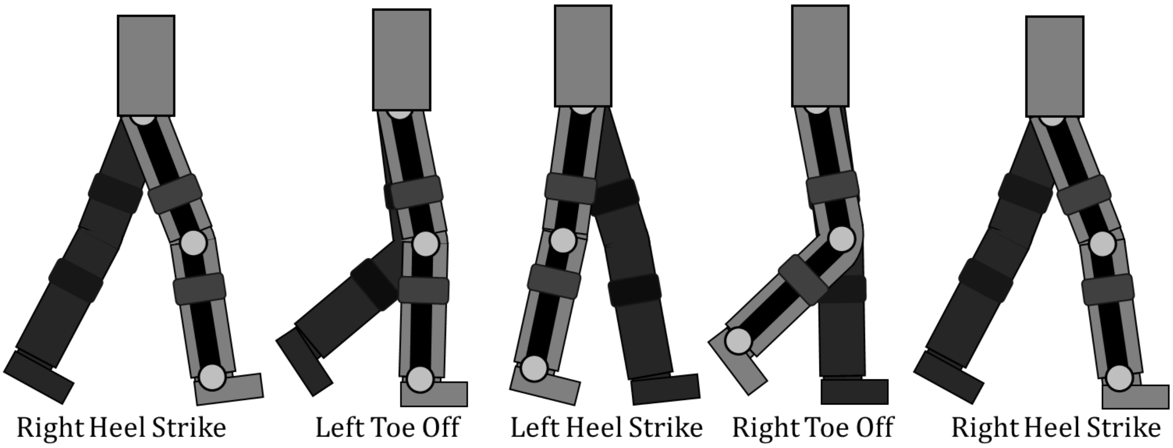


Figure 10 - Gait events over one cycle.

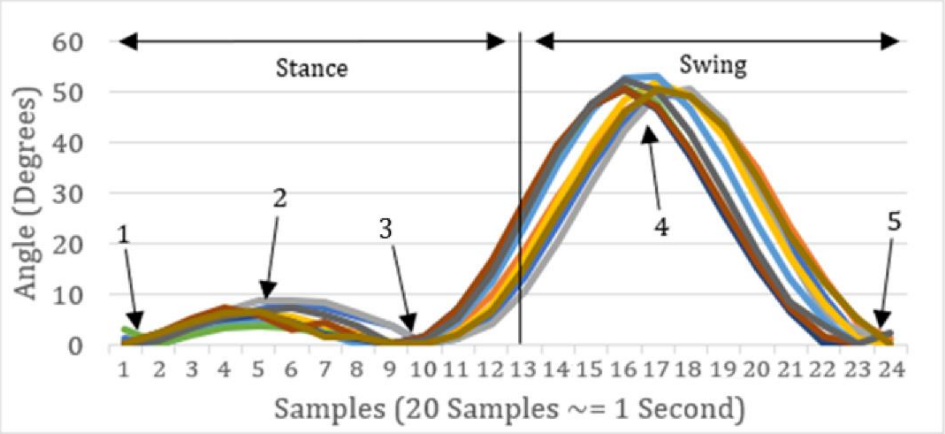


Figure 11 – Knee Angle changes over 10 Gait cycles. 20Hz sampling rate.

During this process, the angles of the hip, knee, and ankle will vary over the course of the stride in a manner that is consistent across multiple step cycles (presuming a consistent walk), as seen in Figure 11 the process consists of the Right Heel Strike (1), Left Toe Off (2), Left Heel Strike (3), Right Heel Strike (4), and finally looping back to Right Heel Strike (5). The Knee Angle seen in Figure 11 follows a consistent loop of Stance (1-3) and Swing (3-5) phases. Note: 0 Degrees represents standing straight. This cycle is however delicate, and as such walkers with disabilities may quickly find losses in efficiency.

3.2.2 Differences between healthy and pathological gait

As seen in Figure 12 [201], from a patient with paralysed quadriceps, there is far more variation in joint angles and sharper changes as the walker is unable to properly extend their knee, they must instead flick it forwards using the hip, leading to a higher hip angle and a more violent knee angle change. The end result would require more energy from the hip as well as leading to long-term stretching of the back of the knee.

In comparison with a healthy posture, sufferers of crouch gait for example can have numerous pathological gait issues stemming from excessive flexion/extension in the hip, knee, and/or ankle [202]. During walking, the Hamstring will be overextended when compared to regular walking, leading to more strain on it. Additional strain can result in stretching and damage to the muscles, which will result in further crouch and further damage, and so on as the user's walking ability continues to degrade.

This can be visually seen as a gradual increase in knee flexion. During the Single Support Phase there is increased hamstring activity for accelerating the knee and hip forwards.

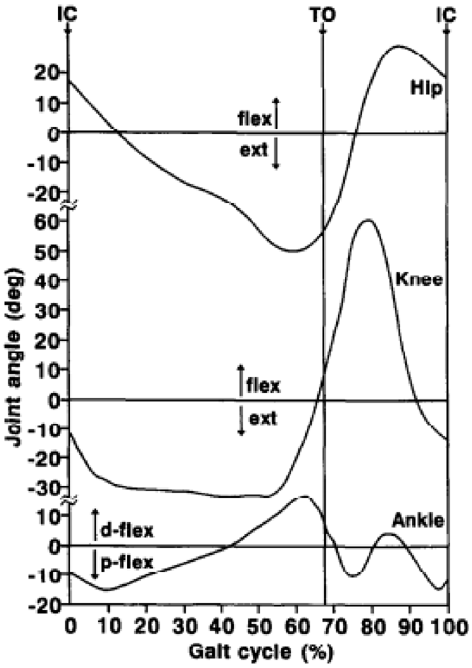


Figure 12 - Pathologic Gait [30, p. 13]

Effectively, as a result of undesired strains upon the individual's bones and muscles in ways that were not naturally intended there is a higher rate of musculature and bone degradation, to which an exoskeleton may be able to reduce by supporting the wearer (and so reducing weight upon their muscles/bones) and correcting their gait. There are a variety of defined Pathologic Gait types, such as Hemiplegic, Diplegic, Ataxic, etc [203], although this is beyond the scope of this research.

3.2.3 Summary

This section acts as a summary of common terminology within exoskeleton development as well as the typical limitations of human joints relevant to this thesis. With this information the outline of the initial aim, that being an assistive exoskeleton able to reduce gait degradation has been completed. Within the next section, research is conducted regarding the state of the art of exoskeleton technology for both hardware and software with a focus on a defined Low-Cost Methodology.

3.3 Human Proportions

Some Exoskeletons, especially those designed for younger patients must consider that wearers will not be of consistent sizes, and may experience considerable physical growth with age, as it is impractical to rebuild an exoskeleton continually to fit an exact size, it is instead preferable to design an exoskeleton to generically allow for a range of wearer sizes, isolating specific age groups and finding a natural middle point to create an "average" exoskeleton with a certain percentage tolerance. This will be due to within an age group there still being variation due to natural height differences, weight, and so on, and so some level of adjustment capability would also be ideal when creating such exoskeletons. Although, it is not feasible to consider every possible size a person could be for a specific age group due to occasional exceptional individuals. Therefore, a reasonable cut-off point can be set for a lower limb exoskeleton such that it is able to adjust to fit roughly 95% of possible candidates (values within 2 Standard Deviations of the mean leg length and width), with this example given for an age range of 6-16 years old. Leg length variation can come from numerous factors, such as diet, health, and pre-existing conditions, If a wearer suffered from a gait disorder from a young age, this may cause growth stunts from excess strain to the legs as a result of a pathological gait, and so their proportions may be lower than average especially for later ages [204]. Furthermore, leg lengths may vary on a single person, for example with Hemiplegic CP, where one leg may be shorter than the other. It should be noted, figures for males will be used in all following examples, as males tend to be both heavier and taller than females, and it is these maximum values that are of focus.

According to [205] 6-year-old male children have a mean Femur (Upper leg) and Tibia (Lower Leg) length of roughly 30cm and 23.4cm respectively, with 95% being between 26.6cm-33.6cm (Femur), and 20.3cm-26.4cm (Tibia). For 16-year-old male teenagers there is a mean Femur and Tibia Length of 45.5cm and 35.3cm respectively, with 95% being between 41.7cm-49.2cm (Femur), and 32.0cm-38.7cm (Tibia). This shows a maximum physical change of 22.6cm in Femurs, and 18.4cm in Tibias over the course of 10 years. This data is however provided by a Korean source, with Korean male adults being on average slightly shorter than UK male adults (173.8cm (South Korea) vs 178.2cm (UK)). Additionally, this measures the length of the bones, not the total leg length (the foot would not be included in the lower leg length for example).

Additional information collected by M.O.T.I.O.N Interreg [206] which was collated from several sources ([207], [208], [209]), reported average upper leg lengths of 31.8-38.7cm, and lower leg lengths of 30.0-36.2cm for children between 8 and 12 years old (male and female).

Upper and lower leg width must be considered as well as length in order to make a properly fitting device. There however seems to be much less documentation on average leg widths, this factor is primarily tied to average weight, which is the final necessary known quality in order to calculate force requirements for actuators. For designing an exoskeleton to account for a variety of leg widths, the use of softer attachment methods such as belts with multiple locking points is relatively common and allows for a simple implementation that covers most reasonable leg widths.

According to the NHS [210], UK Male children weigh between 16.5-27.5kg at 6 years old, and between 45-86kg at 16 years old (2nd-98th percentiles). It also predicts total heights between 106-126cm at 6 years old, and 158-189cm at 16 years old. The weight of individual leg sections can be found as a proportion of the total body weight, for an adult the proportions of the upper (thigh) and lower (shank + foot) legs are 14.78%, and 6.10% respectively, or 7.39% and 3.05% per leg [211]. It should however be noted that the leg length and therefore weight in relation to total height tends to increase with age from child to adult, especially around puberty.

However, by comparing the Body Height (BH) to Leg Length (LL) ratio of an adult with that of a child we can estimate the appropriate leg weight proportions of the child. According to [209], men aged 21 have an average BH/LL Ratio of 1.06, 16-year-old children have a ratio of 1.045, and 6-year-old children have a Ratio of 1.18. From these results an 11% decrease in leg proportion between the 6-year-old and adult values can be observed, as such it can be presumed that for this age group, the average weight proportions of the upper and lower legs would be 13.27% and 5.48% respectively (6.64% and 2.74% per leg). These values are displayed in Table 15.

Table 15 - Averaged Leg/Body Length and Weight for 6- and 16-year-olds (Sitting Height from Buttocks to Head)

	Upper Leg		Lower Leg		Body (Sitting Height)	
Age	Length (cm)	Weight (kg)	Length (cm)	Weight (kg)	Length (cm)	Weight (kg)
6	26.6-33.6	1.1-1.75	20.3-26.4	0.45-0.75	60.0-70.0	13.41-22.34
16	41.7-49.2	2.98-5.71	32.0-38.7	1.23-2.36	84.0-100.0	36.56-69.88

These values can be used as a baseline for children and young adults in future calculations. However, prior to this a basic model would be created to outline how the wearer's dimensions and leg positions would affect the amount of weight the exoskeleton would theoretically need to be able to support.

3.4 Concepts

This section will cover the mathematical theory developed for the research project, consisting of a simplified mathematical model of a potential wearer within the sagittal plane, as well as the initial design concept of the Exoskeleton, the actuation technique behind it, and the arithmetic required to calculate the ideal actuator position using the wearer's current state.

3.4.1 Centre of Mass (CoM)

To find the centre of mass, one can model the lower leg, upper leg, and body as a three-part inverted pendulum (Figure 13), with the centre of rotation being at the point where the foot (modelled as an infinitely small point) touches the floor. Assuming the leg/body length and weight values are known and remain constant, the Horizontal (Equations (2), (3), and (4)) and Vertical (Equations (5), (6), and (7)) Coordinates of the wearer's centre of mass can be estimated per body part. As this is a Lower-Body model, it does not consider the upper body, assuming the arms and head remain in line with the body. Similarly, the model assumes the wearer does not angle themselves outside of the sagittal plane. Note this model measures forward/down as negative and backwards/up as positive with zero as the Centre of Rotation. The Horizontal Coordinates for Lower leg X_l, Upper leg X_u, and Body X_b CoM's are:

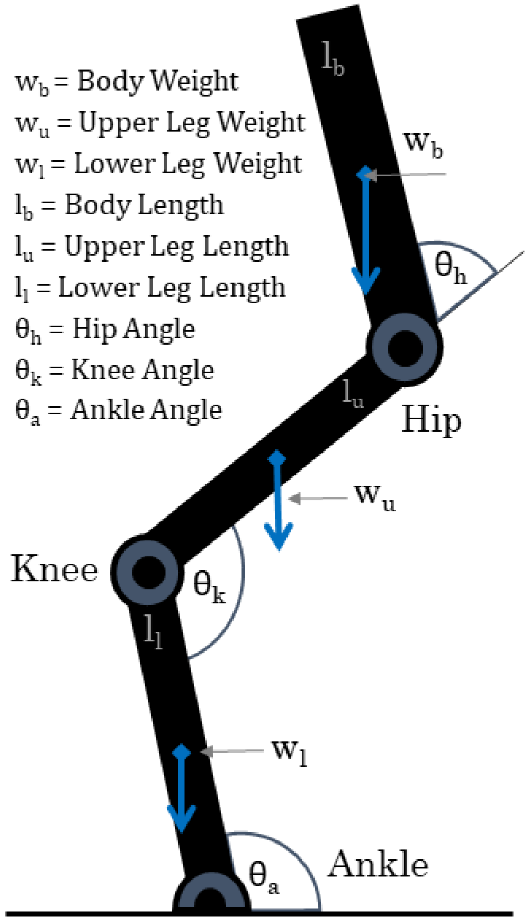


Figure 13 - Sagittal Plane Linear Inverted Pendulum Diagram

$$\boldsymbol{X_l} = \frac{l_l}{2} cos(\theta_a)$$

(2) - Lower Leg CoM X Position

$$\boldsymbol{X_u} = l_l\, cos(\theta_a) + {}^{l_u}/_{2}\, sin(\theta_k + (\theta_a - 90))$$

(3) - Upper Leg CoM X Position

$$\boldsymbol{X_b} = l_l\, cos(\theta_a) + l_u sin(\theta_k + (\theta_a - 90)) + {}^{l_b}/_{2}\, sin(\theta_b + \theta_k + (\theta_a - 90))$$

(4) - Body CoM X Position

The Vertical Coordinates for Lower Leg (Y_l), Upper leg (Y_u), and Body (Y_b) CoM's are:

$$\boldsymbol{Y_l} = \frac{l_l}{2} sin(\theta_a)$$

(5) - Lower Leg CoM Y Position

$$\boldsymbol{Y_u} = l_l\, sin(\theta_a) + {}^{l_u}/_{2}\, cos(\theta_k + (\theta_a + 90))$$

(6) - Upper Leg CoM Y Position

$$\boldsymbol{Y_b} = l_l\, sin(\theta_a) + l_u cos(\theta_k + (\theta_a + 90)) + {}^{l_b}/_{2}\, cos(\theta_b + \theta_k + (\theta_a + 90))$$

(7) - Body CoM Y Position

The lower leg is directly connected to the rotation point, the upper leg connected via the end of the lower leg (and therefore its orientation angle is dependent on the angle of the lower leg angle), and the body further connected to the end of the upper leg (and therefore its orientation angle is dependent on both the upper and lower leg angles).

The position of the Centre of Mass Horizontal and Vertical Coordinates of each part (lower leg, upper leg, and body) are biased by their weights to determine the total Centre of Mass. (10) and (11) show the full CoM Calculation considering both the left and right legs (denoted with "*l*" and "*r*" subscript notation) as well as body position. *O* represents any offset between the connection to the ground in the X/Y plane for the right leg and left leg, with the left leg considered the origin "zero" point. Offset *O* can be found either by physically measuring this distance or calculating it ((8) and (9)).

$$\boldsymbol{O_x} = \left(sin\left(\theta_{h_r} + \theta_{k_r}\right) - sin\left(\theta_{h_r}\right)\right) - \left(sin\left(\theta_{h_l} + \theta_{k_l}\right) - sin\left(\theta_{h_l}\right)\right)$$

(8) - X Axis Offset of non-centred leg

$$\boldsymbol{O_y} = \left(cos\left(\theta_{h_r} + \theta_{k_r}\right) - cos\left(\theta_{h_r}\right)\right) - \left(cos\left(\theta_{h_l} + \theta_{k_l}\right) - cos\left(\theta_{h_l}\right)\right)$$

(9) - Y Axis Offset of non-centred leg

$$\boldsymbol{X_{CoM}} = \frac{\left(\left(X_{l\_l} * w_{l\_l}\right) + \left(X_{u\_l} * w_{u\_l}\right) + \left(X_{b\_l} * \frac{w_b}{2}\right)\right) + \left(O_x + \left(X_{l\_r} * w_{l\_r}\right) + \left(X_{u\_r} * w_{u\_r}\right) + \left(X_{b\_r} * \frac{w_b}{2}\right)\right)}{w_{l\_l} + w_{l\_r} + w_{u\_l} + w_{u\_r} + w_b}$$

(10) - Centre of Mass X Coordinate relative to Centre of Rotation

$$\boldsymbol{Y_{CoM}} = \frac{\left(\left(Y_{l\_l} * w_{l\_l}\right) + \left(Y_{u\_l} * w_{u\_l}\right) + \left(Y_{b\_l} * \frac{w_b}{2}\right)\right) + \left(O_y + \left(Y_{l\_r} * w_{l\_r}\right) + \left(Y_{u\_r} * w_{u\_r}\right) + \left(Y_{b\_r} * \frac{w_b}{2}\right)\right)}{w_{l\_l} + w_{l\_r} + w_{u\_l} + w_{u\_r} + w_b}$$

(11) - Centre of Mass Y Coordinate relative to Centre of Rotation

Model Limitations

The Model seen in Figure 13 has several limitations innate to it that shall be expanded upon. None the less it will continue to be used within future chapters.

- This model defines all objects as having all mass located at a single point-sized CoM (Centre of Mass), assuming an unchanging density throughout the object that results in the CoM being directly located at the centre of the object, rather than a mass distributed continuously throughout the object relative to any varying densities. This is done to retain simplicity.
- This model considers all mass above the hips as a uniform "Body" mass, and so changes to the distribution of the centre of Mass due to the movement of the arms and head are not considered. As this model is a Lower-Limb Exoskeleton, and is therefore limited to the lower-limbs, there would not be any way of measuring the movement of the arms, and as such the model reflects this limitation.
- This model assumes there is always enough friction in connection with the foot to the ground such that the user would not slip, effectively meaning that the foot is fused to the floor and that all force is directly applied by the foot to the ground without creating additional movement. Similarly, it implies both feet are touching the ground. Considering these factors would increase model complexity beyond necessity.
- This model only considers the weight of the user, and not any additional weight they may be carrying. Negligible examples could include clothing, technological peripherals, etc. Less negligible examples include backpacks or the to-be designed exoskeleton itself. Known weights such as the latter could be factored into final calculations as additional body/leg mass when they are known.
- This model only considers force due to gravity, and not any additional force due to muscle strain or torque/inertia from movement such as the user deliberately flexing/extending, or an exoskeleton actuator doing the same, or an external force applied to the user. This is deliberately not considered as part of this model which aims to find the base centre of mass. Effectively, the model only considers gravity, and later considers actuator counterforce for a single instant assuming that the model was entirely static prior to this point and that any initial inertia is ignored.
- This model only considers the sagittal plane, and as such assumes the user's legs are not angled to the side (abducted/adducted) or that the surface the user is standing on is not angled outside of the sagittal plane. Considering other planes would make the model more complicated. The exoskeleton design will also be focusing on the Knee Joint, which only has Sagittal plane flex/extension movement.
- This model assumes that the body, upper leg, and lower leg cannot collide with each other, as the ability for the upper leg to rest upon the lower leg such as during a high knee flexion squat reduces the torque of the upper leg due to gravity [212].

With these limitations considered, this Linear Inverted Pendulum Model can be used to simulate the supporting force a knee-mounted linear actuator design could provide.

3.4.2 Theoretical Actuator Force

The Core exoskeleton concept is the support of the wearer's upper leg by providing a way to rest it upon a compliant linear actuator, easing the work of the natural movement of the Quadriceps Femoris and Hamstring Muscles during walking and crouching. When simplified and viewed in the sagittal plane as seen in Figure 14, the body and leg can be described as an Inverted Pendulum with a linear actuator connecting from the lower leg to the upper leg applying a force "F" to the upper leg. As with the prior model in 3.4.1 this too consists also of an attachment point to the ground representing the Ankle and two further joints representing the Knee and Hip. This model considers the applied counterforce to be an infinitely small point whose angle can always be perfectly calculated by knowing the knee angle. In practice this will not occur due to bending of the linear actuator and attachment points, resulting in a slight uncontrolled deviation.

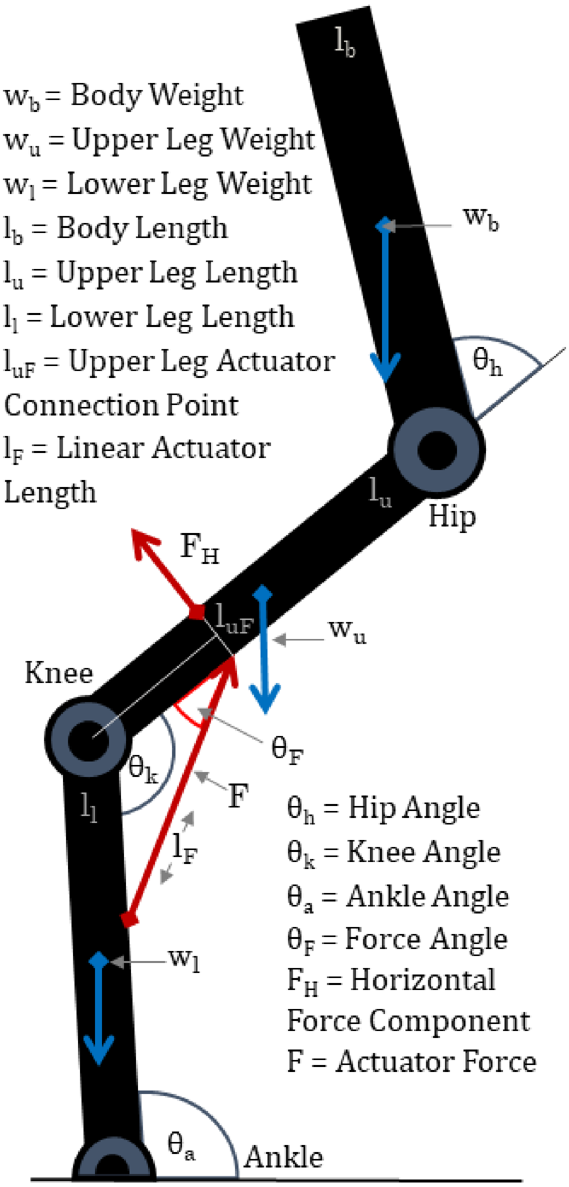


Figure 14 - Sagittal Linear Inverted Pendulum with Actuator

The Length values l_l, l_u, and l_b represent the lower leg, upper leg, and body lengths respectively, with l_{uF} representing the length between the knee joint and linear actuator connection point where force is applied to the leg. l_F represents the effective length of the linear actuator, which varies with knee angle.

The Weight values, w_l, w_u, and w_b represent the weights of the lower leg, upper leg, and body respectively. They apply a constant torque about the Centre of Rotation due to gravity.

The Angle values θ_a, θ_k, and θ_h represent the joint angles at the Ankle, Knee, and Hip respectively, with θ_F representing the angle at which Actuator Force is applied to the Upper Leg (The "Force Angle"). Within Figure 14 it is part of a triangle with θ_k and may be derived from it.

Finally, F Represents the force applied upon the upper leg by the linear actuator, which would connect between upper and lower leg connection points. This Actuator acts to support the wearer's knee, with the force it provides acting to counteract some percentage of the torque applied by the upper leg and body weights due to gravity (it is assumed the user is not deliberately flexing their knee and attempting to crouch or applying some other kind of force, as otherwise the resistance of the actuator would be undesired). Only the Horizontal component of this force perpendicular to the upper leg value F_H will be useful in supporting the knee. The vertical component meanwhile is parallel with the upper leg and so not aiding in rotation (Horizontal and Vertical are used relative to the upper leg in a standing position). For example, if the wearer was standing perfectly straight (θ_k = 180, θ_F = 0), F_H would be zero as the force and leg weight would be perfectly in line with the lower leg and the ground, and as such would entirely consist of its vertical force component. The proportion of useful force varies with θ_F, at 90^{o}, the linear actuator would be perpendicular and so all force would be applied usefully. In later sections the value of θ_F dependant on θ_k will be modified to provide more desirable angles at times where more force is needed.

Modelling The Force Applied to Upper Leg

To calculate the value of F at any value of θ_k the model can be simplified for ease of calculation. By assuming that θ_a = 90 one can display the three forces acting upon the upper leg as a series of infinitely small points applying torque about the knee, which acts as the Centre of Rotation (CoR). (Figure 15).

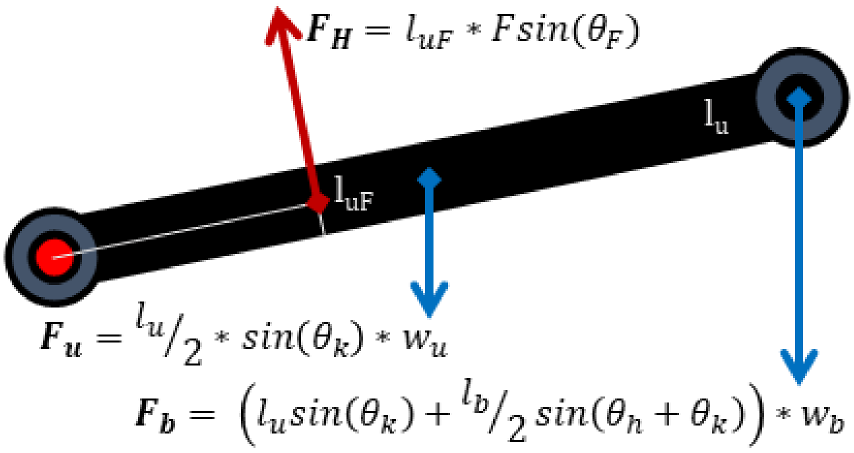

$F_H = l_{uF} * F\sin(\theta_F)$
l_u
l_{uF}
$F_u = {}^{l_u}/_2 * \sin(\theta_k) * w_u$
$F_b = \left(l_u\sin(\theta_k) + {}^{l_b}/_2 \sin(\theta_h + \theta_k)\right) * w_b$

Figure 15 – Moment about the Knee

These equations assume positive values as backwards and negative as forwards.

- F_u = Torque about knee due to Gravity from Upper Leg Weight
- F_b = Torque about knee due to Gravity from Body Weight
- F_H = Torque about knee due to Linear Actuator Force Perpendicular to Upper Leg

In a worst-case scenario, assuming that the wearer is entirely resting upon the actuator and providing no muscular force, then the equation $F_H = F_u + F_b$ would result in a stable system where the force provided by the actuator balances the weight of the body and upper leg. Substituting F_u, F_b, and F_H in (12):

$$\left(\boldsymbol{l_{uF}sin(\theta_F)}\right) * \boldsymbol{F} = \left({}^{l_u}/_2\, sin(\theta_k)\right) * w_u + \left(l_u sin(\theta_k) + {}^{l_b}/_2\, sin(\theta_h + \theta_k)\right) * w_b$$

(12) - Balanced Equation where Linear Actuator Counterforce matches Torque from Gravity about knee.

Which can then be Re-Arranged to find F to form a simplified equation for calculating the required supporting force described in (13).

$$\boldsymbol{F_H} = \frac{\left({}^{l_u}/_2\, sin(\theta_k)\right) * w_u + \left(l_u sin(\theta_k) + {}^{l_b}/_2\, sin(\theta_h + \theta_k)\right) * w_b}{l_{uF} * Fsin(\theta_F)}$$

(13) – Re-arranged Horizontal Force Component Calculation

To make a model that also considers the position of the lower leg and ankle, each term must consider the effect of the lower leg length on its horizontal length component, and the angle of the ankle on all future joint angles. As such, each Equation gains the term "$l_l \cos(\theta_a)$" and adds $(\theta_a - 90)$ to each subsequent joint angle. This creates the full Equations (14), (15), and (16).

$$\boldsymbol{F_u} = \left(l_l\, cos(\theta_a) + {}^{l_u}/_2\, sin\left(\theta_k + (\theta_a - 90)\right)\right) * w_u$$

(14) - Upper Leg Torque on Knee due to gravity that considers ankle angle.

$$\boldsymbol{F_b} = \left(l_l\, cos(\theta_a) + l_u sin\left(\theta_k + (\theta_a - 90)\right) + {}^{l_b}/_2\, sin\left(\theta_h + \theta_k + (\theta_a - 90)\right)\right) * w_b$$

(15) - Body Torque on Knee due to Gravity that considers ankle angle.

$$\boldsymbol{F_H} = (l_l\, cos(\theta_a) + l_{uF} sin(\theta_F)) * F$$

(16) - Horizontal Component of Actuator Counterforce that considers ankle angle.

If the F_u, F_b, and F_H sub-equations are combined as was previously, they can be re-arranged to find the required Actuator Force F to counteract the weight of the upper leg and body, assuming no other provided forces such as the muscles. Effectively making it the "worst case" scenario. If considering two legs, the force of the weight of the body would be split across each leg proportionally, this proportion would vary dependant on the position of each leg, if the legs occupied the same sagittal position then this split would be 50/50, if one leg was supporting the body whilst the other did little the proportion would be closer to 100/0, the equation for this proportion value is seen in (17). This proportion is shown in the equation as p_L or p_R for left or right knee respectively and would vary between 0 to 1 and is calculated as a ratio of the force applied by the body upon the left or right knee (F_{b_L} or F_{b_R}) over the sum of the forces applied to both knees. The Full equation finding the actuator counterforce F is shown in (18).

As has been stated, these equations are designed for a model that only considers the sagittal plane, if the wearer were to move their body to the side to press more force on one leg than another or abduct/adduct their legs these factors would not be considered. Similarly, Force F is only considered in a single instant, prior inertia is not considered.

$$\boldsymbol{p}_{\mathbf{L}\vee\mathbf{R}} = \frac{F_{b_{\mathrm{L}\vee\mathrm{R}}}}{F_{b_L}+F_{b_R}}$$

(17) - Proportion of Body weight on left and right knees.

$$\boldsymbol{F}_{(\mathbf{L}\vee\mathbf{R})} = \frac{\left(l_l\,cos(\theta_a) + {}^{l_u}/_{2}\,sin\big(\theta_k + (\theta_a - 90)\big)\right) * w_u + \big(l_l\,cos(\theta_a) + l_u sin\big(\theta_k + (\theta_a - 90)\big) + {}^{l_b}/_{2}\,sin\big(\theta_h + \theta_k + (\theta_a - 90)\big)\big) * w_b * p_{\mathrm{L}\vee\mathrm{R}}}{l_l\,cos(\theta_a) + l_{uF} * sin(\theta_F)}$$

(18) - Full Equation for Calculating Linear Actuator Counterforce on Left (F_L) or Right leg (F_R)

Applying the Model to Worst-Case and Reasonable-Case Squats

While the model cannot provide predictions for the force applied to the actuator during movement as it does not take into account joint torque or inertia from movement, only the effect of gravity, as this would considerably increase complexity (examples of such derivation are seen in [213]). It instead can estimate a "worst case" rest force scenario for if the wearer were to rest their entire weight upon the actuator and provide no assistance at the knee as with a squatting position. From there it can be assumed that all "real" values will be lower than this worst-case, assuming that the wearer was not performing an active crouching motion or some other deliberate act.

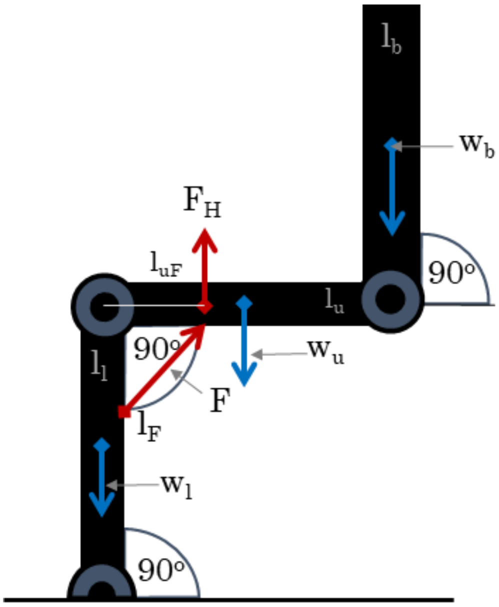


Figure 17 - Worst-Case Squatting Position

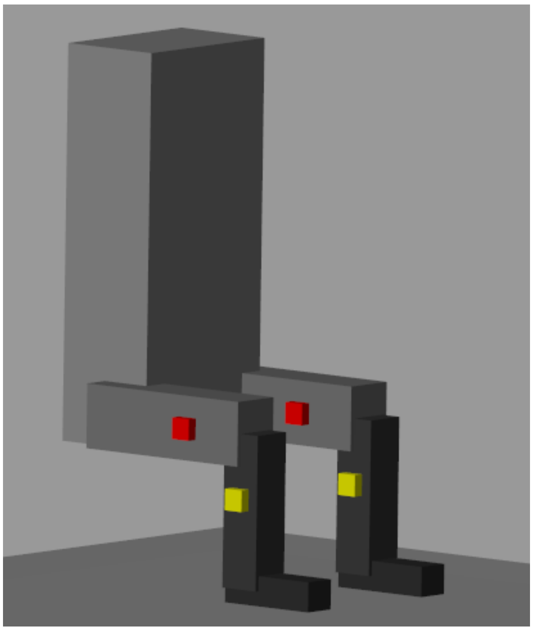


Figure 16 - MATLAB Simulink simulation of worst-case model. (red: upper actuator connection point, yellow: lower actuator connection point)

A Worst-case squatting position shown in Figure 17 is defined as the position with the highest horizontal moment about the knee, which is when the upper leg is horizontal and lower leg vertical. The Weight and Length Values of the Upper Leg and Body will be drawn from Table 15's 16 Year old average values (w_u = 4.22kg , l_u = 45.45cm Lower Leg, and w_b = 53.22kg, l_b = 92cm Body & Head). The length of the actuator counter force from the knee joint is defined as 1/3 upper leg length (l_{uF} = 15.15cm), both legs are assumed to be in the same sagittal position (body weight is split equally across both legs, so p_l = 0.5). (19) displays the calculation process.

(19) - Calculating Counterforce for Worst-Case Scenario

$$\boldsymbol{F_u} = \left(0 + {}^{0.4545}/_{2}\, sin(90+0)\right) * 4.22g = \mathbf{9.4077\ N}$$

$$\boldsymbol{F_b} = \left(0 + 0.4545 sin(90+0) + {}^{0.92}/_{2}\, sin(90+90+0)\right) * 53.22g = \mathbf{237.2891\ N}$$

$$\boldsymbol{F_H} = 0 + 0.1515 * Fsin\left(\frac{180-90}{2}\right) = \boldsymbol{F_u} + \boldsymbol{F_b} * \boldsymbol{p_l}$$

$$\boldsymbol{F} = {}^{\left(9.4077 + (237.2891 * 0.5)\right)}\Big/_{0.1515\sin(45)} \rightarrow \mathbf{F = 1195.335164\ N}$$

Assuming the Lower Leg Actuator Connection Point is an equal length from the knee joint as the Upper Leg Actuator Connection Point is, the distance between the upper and lower linear actuator connection points l_F will be 21.43cm calculated via Pythagoras theorem. By creating a triangle from the knee angle and upper/lower connection lengths, with the linear actuator length as the hypotenuse. If the Linear Actuator is modelled as a spring of unloaded active length l_{s_u}, loaded active length l_s, and active spring constant k, Equations (20) and (21) can be formed to find the unloaded Spring Length. The Loaded Spring length will either be equal to l_F or derived from it (32):

$$\boldsymbol{F} = k(l_{s\_u} - l_s) \rightarrow \boldsymbol{l_{s\_u}} = {}^{F}/_{k} + l_s$$

(20) - Calculating Unloaded Spring Length

Substituting l_F = 21.43cm into l_s, F= 1195.43, and k= 5000 (arbitrary spring constant).

$$\boldsymbol{l_{s\_u}} = \left({}^{1195.34}/_{5000}\right) + 0.214 \rightarrow \boldsymbol{l_{s\_u}} = 0.453068$$

(21) - Calculating Unloaded Spring length in Worst-Case Scenario

These calculations are appropriate for a 65.5kg individual that is around 172.85cm tall, values derived from Table 15.

Using these values, a simulation was created in MATLAB Simulink (Figure 16) to test the validity of the Mathematical model. This Simulink model had dimensions and weights equal to those used in the prior equations. Using the previously stated $k$value to represent the spring constant, the ideal spring length was calculated to be 0.453 metres (21). A Spring force of this length was placed between the red and yellow boxes shown in the figure, with the knee joint using a free-moving revolute joint. When this simulation was run the instantaneous force registered by the spring was 1193.58 N at 90.00 Degrees, with the force after 10 seconds of simulation stabilising at 1198.919 N at 90.54 Degrees. These results, being within 0.3% of the predicted value of the model were considered sufficiently accurate, with rounding error considered the likely reason for this minor difference, with a slightly higher force resulting in a slightly higher knee angle.

As shown in Figure 18, the Simulink Model consists of a Main System component for initialising gravity and the World Frame, this connects to two leg components each consisting of three revolute joints. The Foot of each leg is directly connected to the world plane and immovable, the ankle joint is locked at a pre-set value, the hip joint is weighted such that the body always remains vertical, and the knee joint is allowed to move freely. A Spring Force connects between the lower and upper leg components that is preset to a certain length prior to the simulation.

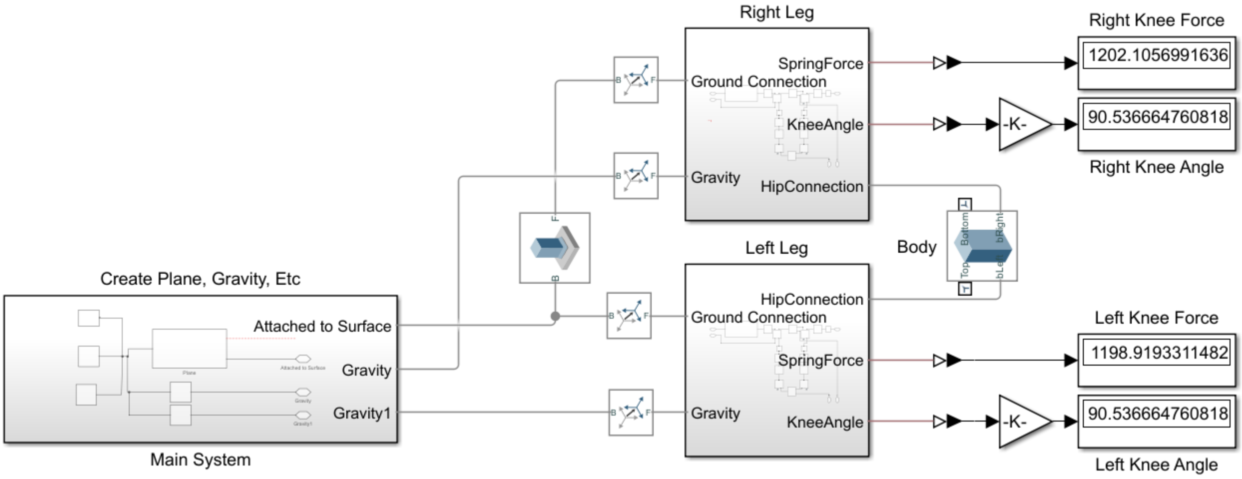


Figure 18 - MATLAB Simulink initial lower-leg model

This basic model will be expanded later in 4.3 to further optimise the system. Currently much of the force F applied to the leg is lost within the Vertical Component. Ideally, the moment about the upper leg would be increased. Theoretically increasing the value of the upper actuator connection length l_{uF} and so placing the Horizontal Force Component F_H further away from the centre of rotation would increase the torque applied by it due to improved leverage. In reality unless the lower actuator connection point of F was also increased to match on the lower leg this would cause a reduction in the Force Angle θ_F, which would result in the Horizontal Force Component F_H being a lower proportion of force F overall, therefore requiring the actuator force to be increased to counteract this, additionally it would increase the linear actuator length l_F, increasing its weight and size. This force inefficiency can be seen with the already existing example in Figure 17, as whilst F was equal to 1195.42N, the actual usable force, F_H was only ~70.7% of this (845N), with the rest of the force effectively being lost due to the fact that the force was not perpendicular to the lower leg. As such, the Vertical Force Component can be lowered by increasing the Force Angle θ_F, however as it is innately tied to the Knee Angle θ_k it would instead be desired to make it such that a high θ_F would be most prevalent during the times where the supporting force would be most useful, such as when the wearer was squatting or flexing/extending their knee during walking, where the moment about the knee will vary in both stance and swing phase [214].

3.4.3 Ideal Actuator Placement and Force Transfer

Not all torque the linear actuator applies to the upper leg will be in its horizontal component, the exact proportion of horizontal to vertical component forces is innately dependant on the knee angle and the position of the lower and upper linear actuator connection points as these determine the force angle "θ_F". These Linear Actuator connection points can be adjusted vertically (parallel to the leg) to be placed closer or further from the knee joint, as well as horizontally (perpendicular to the leg) by shifting them out of line with the leg and on to "Stickouts" set out perpendicular to the legs at known lengths, examples of both can be seen in Figure 20 with (B – Vertical Adjustment) and (C – Horizontal Adjustment) both being examples of increasing the Force Angle by altering the actuator connection points. The distance between the Upper and Lower actuator connection points and the Knee Joint is represented as l_{uF} and l_{lF} respectively. When creating a Stickout, these force connection lengths themselves consist of a Horizontal and Vertical Component; l_{uH} and l_{uV} for Upper Leg, or l_{lH} and l_{lV} for Lower Leg. l_F represents the Length of the Linear Actuator.

For the purpose of finding the efficiency of useful force transfer the exact pose of the upper and lower leg are irrelevant for this calculation, only the knee angle between them. Although it is assumed that any force applied to the lower leg by the linear actuator would not cause it to be pushed away. As with prior model examples this model looks at a single instant of force without considering inertia or prior torque.

Whilst the Force Angle is still affected by the Knee Angle, it can no longer be assumed that the Force Angle will be exactly half of ($180-\theta_K$), as the upper and lower actuator connection points lengths may be different, or not directly attached in line with the upper/lower leg. Therefore, first it is required to calculate the distance between the Linear Actuator Connection Points l_F. The length of the linear actuator across the knee can be estimated via the knee angle using the Cosine Rule, as l_{uF}, l_{lF}, and l_F form a triangle. The Stickouts are assumed to be perpendicular to the exoskeleton itself, as seen in Figure 19.

If not already known directly, l_{uF} and l_{lF} are calculated via Pythagoras theorem, The Angles between these lines and their leg sections would be θ_u and θ_l, these would be used to alter the Knee Angle, which, due to the horizontal Stickouts would no longer represent the angle between the Actuator Connection points, the new angle representing the angle between the connection points would be θ_{kF}.

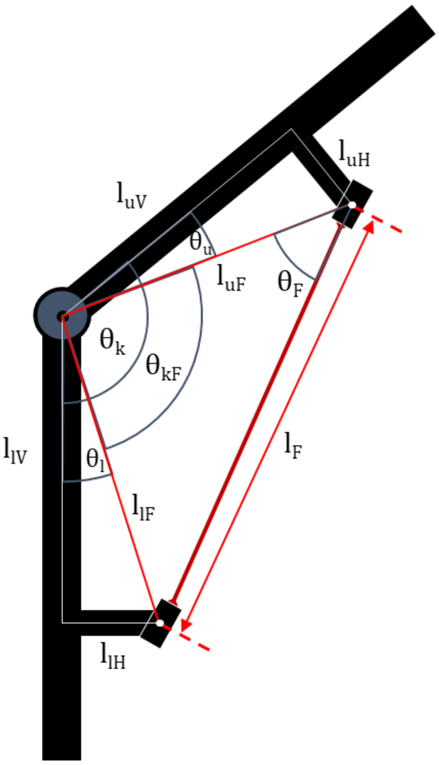


Figure 19 - Calculating Linear Actuator Length "L_F" and Force Angle θ_F

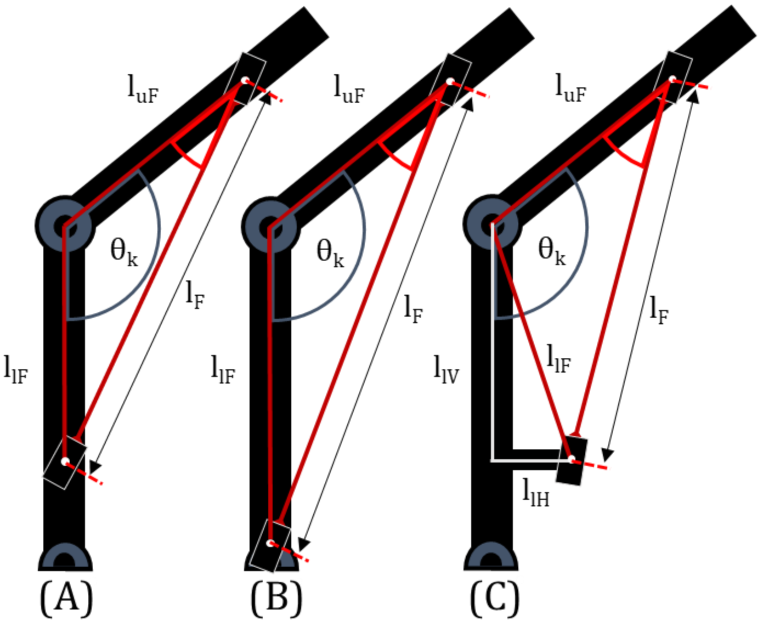


Figure 20 - Adjusting θ_F by moving Actuator Connection Points. (A: None, B: Vertical, C: Horizontal (Stickout))

Based on this, a new set of equations can be made to calculate the length between the knee joint and actuator connection point l_{uF} and l_{lF}. As well as the angle between them and the knee itself θ_u and θ_l.

Find length between Upper/Lower Actuator Connectors and Knee Joint ((22) and (23)):

$$\boldsymbol{l_{uF}} = \sqrt{l_{uH}^2 + l_{uV}^2}$$

(22) - Length from CoR to Upper Actuator Connection Point

$$\boldsymbol{l_{lF}} = \sqrt{l_{lH}^2 + l_{lV}^2}$$

(23) - Length from CoR to Upper Actuator Connection Point

Find Angle Between Actuator Connectors by altering Knee Angle ((24), (25), and (26)):

$$\boldsymbol{\theta_u} = tan^{-1}\left(\frac{l_{uH}}{l_{uV}}\right)$$

(24) – Angle between upper leg and connector

$$\boldsymbol{\theta_l} = tan^{-1}\left(\frac{l_{lH}}{l_{lV}}\right)$$

(25) - Angle between lower leg and connector

$$\boldsymbol{\theta_{kF}} = \theta_k - (\theta_u + \theta_l)$$

(26) - Adjusted Knee angle to consider connector angles and stickouts

Hence, Actuator Length is calculated via Cosine Rule (27):

$$\boldsymbol{l_F} = \sqrt{l_{uF}^2 + l_{lF}^2 - (2 * l_{uF} * l_{lF} * cos(\theta_{kF}))}$$

(27) - Length of Linear Actuator

Additionally, using Actuator Length and the Sine Rule, the Force Angle θ_F can be found (28):

$$\boldsymbol{\theta_F} = sin^{-1}\left(\frac{l_{lF} * sin(\theta_{kF})}{l_F}\right)$$

(28) - Force Angle between Upper Leg and Linear Actuator

The Force Angle, the Angle at which the Linear Actuator applies force to the leg determines the proportion of "useful" force within the horizontal (perpendicular) component that may rotate the leg and "wasted" force in the vertical (parallel) component that is pushed up the leg and provides no assistance in knee motion (29).

$$\boldsymbol{F_H} = F\sin(\theta_F) \qquad \textit{(29) - Horizontal Force Component}$$

Whilst F_H does not appear Horizontal to the leg, as seen within Figure 21 it is tangent to an arc that when rotated to be in line with the leg would be perpendicular. Applying force to the Stickout rotates the effective pose of the leg. With the Force Angle θ_F now able to be calculated, the ideal Stickout design would be one that would maximise the Horizontal Force Component F_H for the most useful angular period, such as during squatting or walking.

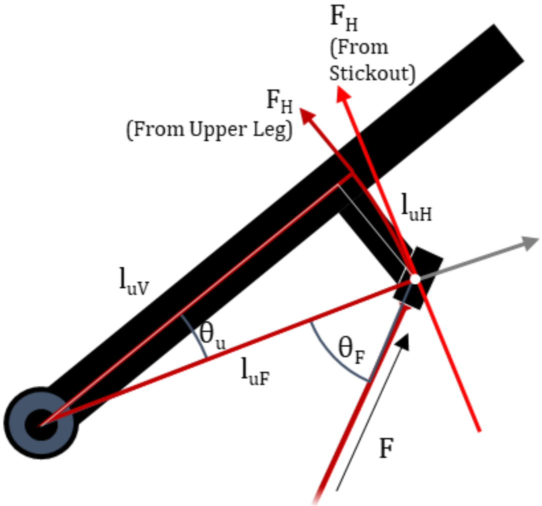


Figure 21 - Upper Knee Force Components Example

Four different designs were considered to try to find the ideal position of the actuator connections and Stickouts. For each of these, the proportion of force contained within the Horizontal Force Component was calculated as a percentage for all angles between 180° (Knee is Straight) and 0° (Knee is folded in on itself). The four general designs displayed in Figure 22 were No stick-out (a), only bottom stick-out (b), both stick-out (c), and top inverted stick-out (d). The horizontal force component would then vary with knee angle, and so each was calculated and graphed.

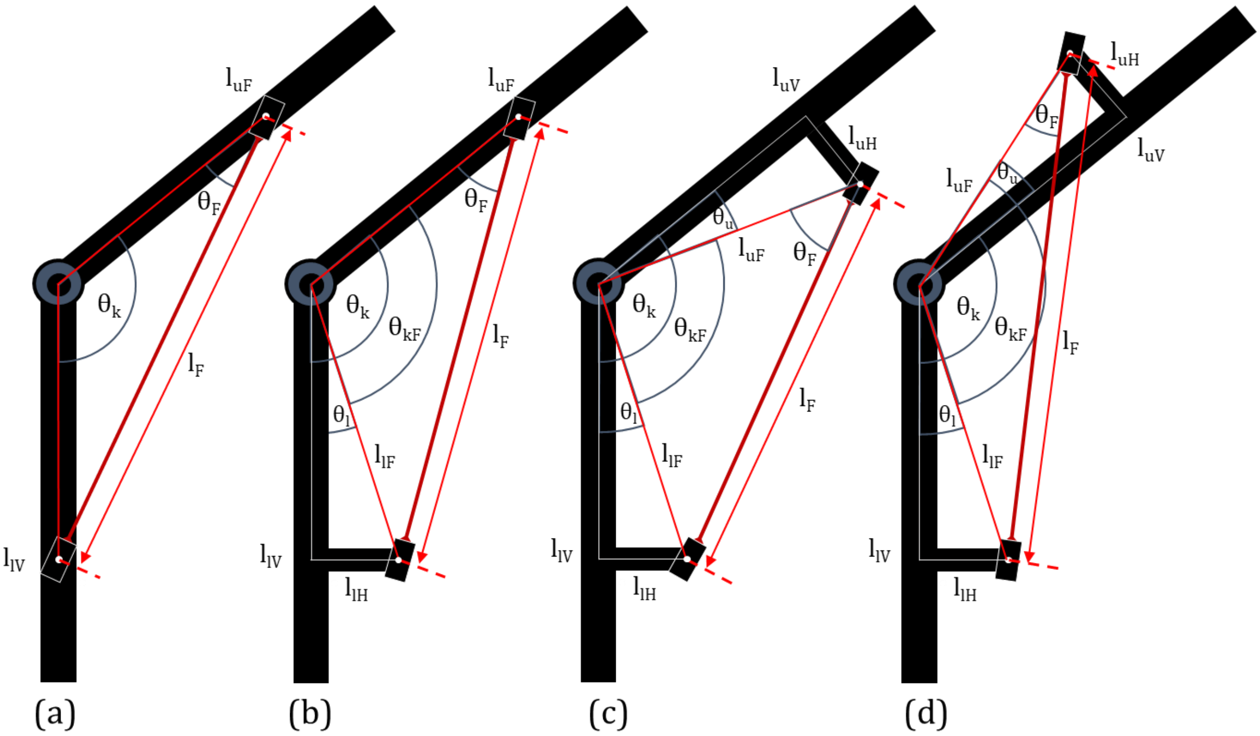


Figure 22 – Actuator Connection and Stickout Designs.
a: No stick-out, b: Bottom stick-out, C: Both stick-out, d: Top inverted stick-out

Theory and Initial Prototyping

There are four variables concerning the Horizontal and Vertical Actuator Connection Distance Components, these being l_{uV}, and l_{lV} for how far the actuator connection Stickout are from the knee joint, and l_{uH} and l_{lH} for how long it is; these would be tested at several variations to look for an optimal value. The resulting "Best" values were determined based off of calculating the summed total force F required to support the wearer at all angles between 0- and 180-degree knee angles during a crouching motion. The crouching motion calculated was an idealised crouch where the lower leg remained perpendicular to the ground, and the hip angle would be the inverse of the knee angle, such that the body remained vertical during knee motion (eg: when knee angle was 180^{o}, body would be 0^{o}).

These calculations followed the model created in this chapter. A Python program was created to iterate through each possible combination of l_{uV}, l_{uH}, l_{lV}, and l_{lH} incrementally. The length between the knee and Upper leg actuator connection point l_{uV} varied between 4cm and 20cm, and Lower leg connection point length l_{lV} between 4cm and 16cm, these being half of the minimum average lengths of a theoretical 16-year-old participant from Table 15 and used in prior calculations (Figure 17 and (19)). Meanwhile the stickout lengths l_{uH} and l_{lH} varied between 0cm and 16cm. Each value would be tested initially with 1cm increments, resulting in a total of 49,152 combinations with 8,847,360 total iterations.

The program would calculate the force profile for each of these combinations between a range of knee angles between 180 to 0 degrees and retain the "best" value based on the finite integral calculated between this range. The Lower the integral value, the lower the total required force and such it was theorised the more efficient the overall force transfer would be across the entire crouch movement. As this process was effectively O(n^4), or O(n^5) if considering knee angle, more detailed ranges could not reasonably be performed without resulting in excessive processing times. Instead, a broad test would be performed first before then later tests with more limited ranges but more precise increments would define exact values. The full Finite Integral for the simplified crouch is seen in (30).

$$\boldsymbol{F_{sum}} = \int_0^{180} \frac{\left({}^{l_u}/_2\, sin(\theta_k)\right) * w_u + \left(l_u sin(\theta_k) + {}^{l_b}/_2\, sin((180-\theta_k)+\theta_k)\right) * w_b * 0.5}{l_{uF} * \left(\frac{l_{lF} *\ sin(\theta_k - (\theta_u + \theta_l))}{l_F}\right)} d\theta_k$$

(30) – Finite Integral of all Force values over set Knee Angle Range

During this crouching motion, it would be possible for the Stickout sections to "collide" if the knee angle were to decrease below certain values that would cause the Stickouts the be in the same position as each other. This would result a in sharp spike of force values (that would in a continuous integration go to infinity) and the force values to become negative afterwards as it would be "pulling" on the upper leg to support its weight as opposed to "pushing" to return to the point of collision. This would not be possible in the reality and so to de-incentivise these values such that there would be as large a region as possible before collision when calculating the summation of forces, negative values were made positive and multiplied by a multiplication value, effectively making negative values proportionally worse than equal positive values as lower values were preferred. Several different multiplication values were tested, between 1 and 10 discussed later in Figure 28 and Figure 29.

All values were capped to a maximum of 2000, such that the large collision "spikes" would not overly bias results negatively as well as prevent integrals getting stuck on tending to infinity, with 2000 being slightly higher than the largest expected value (~1600). The results, seen in Figure 23, are the force profiles of a simplistic crouch model where exoskeletons of ideal Horizontal and Vertical Stickout length values are supporting all of the weight of the user (the same weights used in Figure 17's equations). In both cases Negative "Collision" Values are displayed as their actual calculated values rather than their modified disincentivising values. Note that in Figure 23 and Figure 24, the four numbers after the name represent l_{uV}, l_{uH}, l_{lV} and l_{lH} respectively as defined in Figure 20. The ideal lengths were:

- No Stickout: l_{uV} = 20cm, l_{uH} = N/A, l_{lV} = 16cm, l_{lH} = N/A
- Low Stickout: l_{uV} = 20cm, l_{uH} = N/A, l_{lV} = 16cm, l_{lH} = 12cm
- Both Stickout: l_{uV} = 20cm, l_{uH} = 3cm, l_{lV} = 16cm, l_{lH} = 12.25cm
- Inv. Stickout: l_{uv} = 20cm, l_{uH} = -9.75cm, l_{lV} = 15.5cm, l_{lH} = 16cm

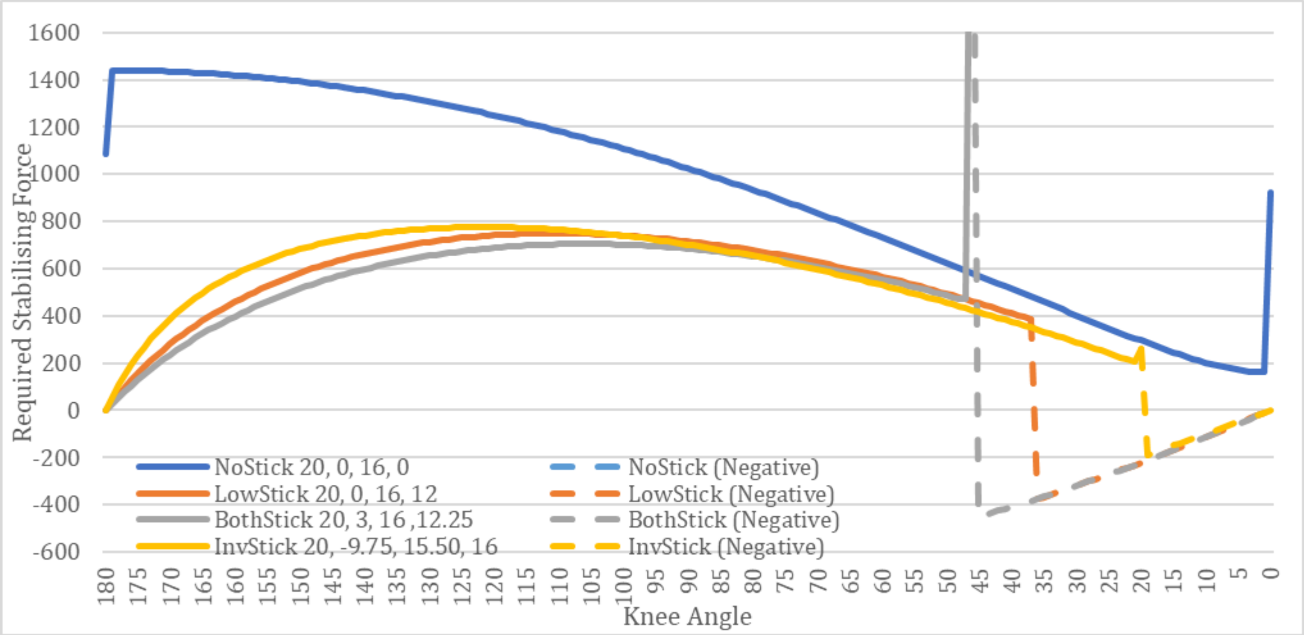


Figure 23 - Force values required of ideal Stickout designs for knee angles of simplified crouch.

The similarities between the Stickouts included a clear preference for the largest upper vertical component and lower horizontal component (*No Stickout* does not have a lower Stickout horizontal component, and so was zero), which was expected as maximising these values would increase the moment about the upper leg and universally increase the horizontal force component respectively, leading to better force transfer. *No Stickout* was a clear outlier, which was expected, having a notable increase in required force at higher knee angles as opposed to all other designs. While higher knee angles would require a lower amount of force due to more of the user's weight being directed down their legs as opposed to rotating their knee joint as torque, there would also be a decrease in the percentage of the horizontal force component of the actuator out of its total applied force. For *No Stickout* this would approach zero percent useful force applied at 180^{o} knee angle shown in Figure 24, therefore it would tend to infinity up until zero force is applied (not shown in graph due to resolution). This would not be the case for any design with a lower leg Stickout.

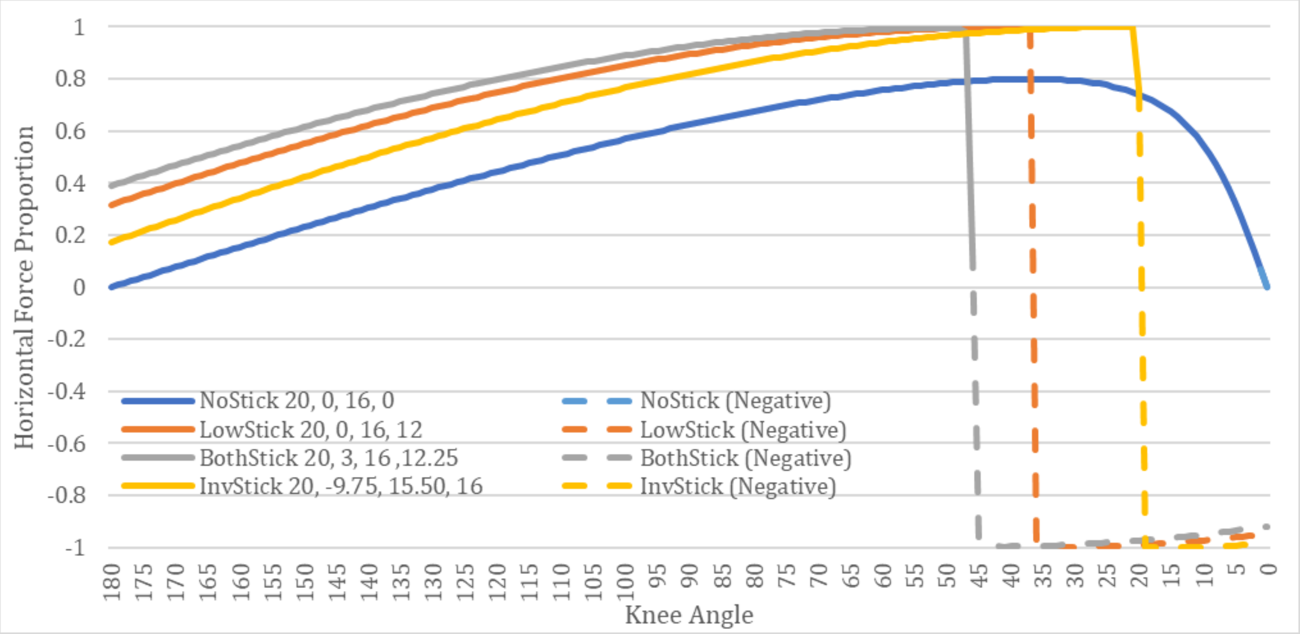


Figure 24 - Ideal Stickout designs Horizontal Force Component Force Proportion.

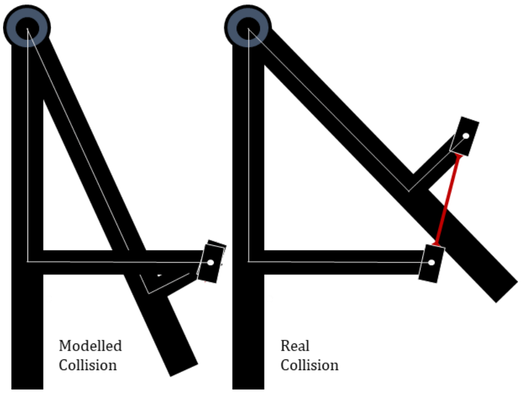


Figure 25 - Inverted Stickout Modelled Collision vs Realistic Collision positions.

Other designs, possessing a lower leg Stickout of some length retain a horizontal force component at 180^o, and therefore tend to zero force at 180^o as assuming the body is also vertical there would be no rotational force about the knee as all the wearer's weight would be directly downwards through the leg. Each of these other designs however encounters a collision at some point earlier than 0^o, noted by the force profiles when switching from positive (pushing force) to negative (pulling force).

As seen in Figure 23 and Figure 24 while *Both Stickout* has a more efficient transfer of force at higher angles, and therefore an overall lower force requirement, *Lower Stickout* and *Inverted Stickout* possess longer ranges before collision whilst maintaining similar force profiles. From this experiment it was deemed that *Lower Stickout* was likely the most ideal option, as whilst Inverted Stickout possessed a mildly larger range before collision, this would be under the assumption that collision would only occur when the actuator connection points collided (See Figure 25), in reality the lower leg Stickout would collide with the exoskeleton itself before this point which would limit its range of motion to being worse than or equal to that of *Lower Stickout*, which also possessed slightly superior force profiles. Only *Both Stickout* possessed a notable spike in values near to collision as was seen in testing (See Figure 27). It is possible that the program prioritised profiles with a lack of a spike, and therefore an overall lower summed force value, with some results not having a spike due to limited resolution. It is unknown whether the spike in force would have any real world effect, although it can be reasonably assumed to not due to it occurring immediately prior to collision and likely represents the Stickouts not perfectly passing through each other during collision but rather than being slightly ajar from each other, causing a brief period where the direction of the linear actuator would rapidly turn as it continues facing between the actuator connection points during theoretical collision. At a single instant this would cause it to face away from the knee joint while being perfectly in line with it, resulting in infinite required force as the horizontal force component would be zero. Figure 26 shows an example of this. This infinite force does not appear during summations due to the deliberate limitations put in place to prevent divide by zero errors within the code and excessive values added to force summations, with any force value above 2000 set to a value of 2000 and any calculation that would result in a divide by zero error skipped.

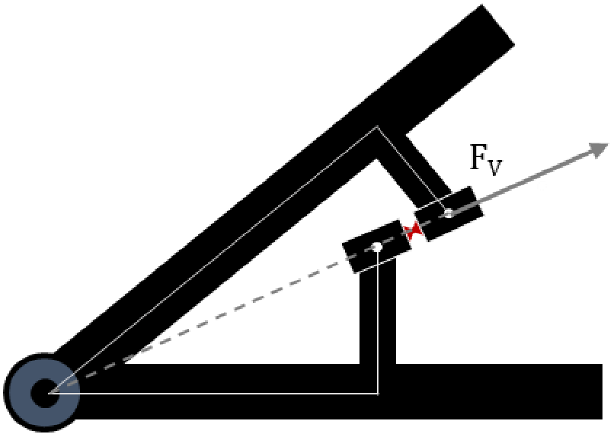


Figure 26 - Example "Collision" that would result in infinite required force due to no Horizontal Force Component, further flexion beyond this would result in a negative force and true collision with exoskeleton.

Seen as a "Spike" in required force within force profile graphs, such as Figure 27.

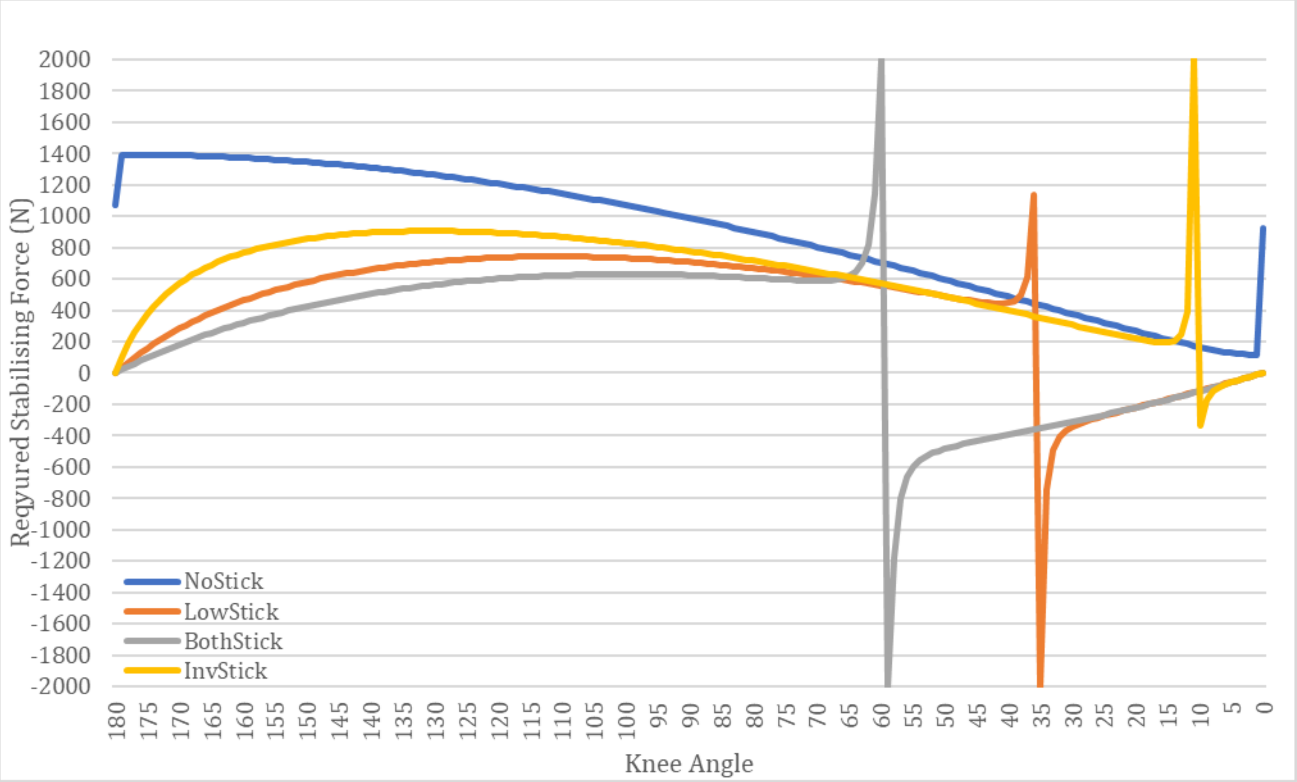


Figure 27 - Example of force profile (20/9/17/12) with "spike" around collision. Force Profile graphs only display one value for each 1°, as such the spike may sometimes be missed if it occurs closer to the middle of 1° increments.

As was discussed prior, negative, or post-collision values were deliberately given a modifier that made them worth a larger amount (EG: A Force value of -500N would be converted to +2000N). As the program summed all force values and considered lower sums to be preferable, this acted as a primitive negative reinforcement that de-prioritised negative values. This implementation would allow for there to be a trade-off between minimising the Force Profile and Maximising the Knee Angle range before collision.

To test which multiplier would be most appropriate, this trade-off was visualised by testing multiplier values of 1 to 10. This formed the graph seen in Figure 28, with higher multiplier values resulting in the "ideal" force profile having a smaller negative section, and therefore a wider knee angle range before collision. They however paid for this with a less efficient transfer of force at higher knee angles (see Figure 29). It was clear that, given an infinite multiplier, the system would trend to a profile that collided at exactly 180^{o}, in effect creating an identical profile to that of the "No Stickout" Implementation. As this was also not ideal, the chosen multiplier would be between the two extremes. As there was not a strictly "Optimal" value, nor a way to decide such a value that would not be equally arbitrary, a multiplier of 4 was chosen as its collision point of ~26^{o} was roughly halfway between 0 degrees knee angle and the collision point of the Force Profile produced with a multiplier of 1.

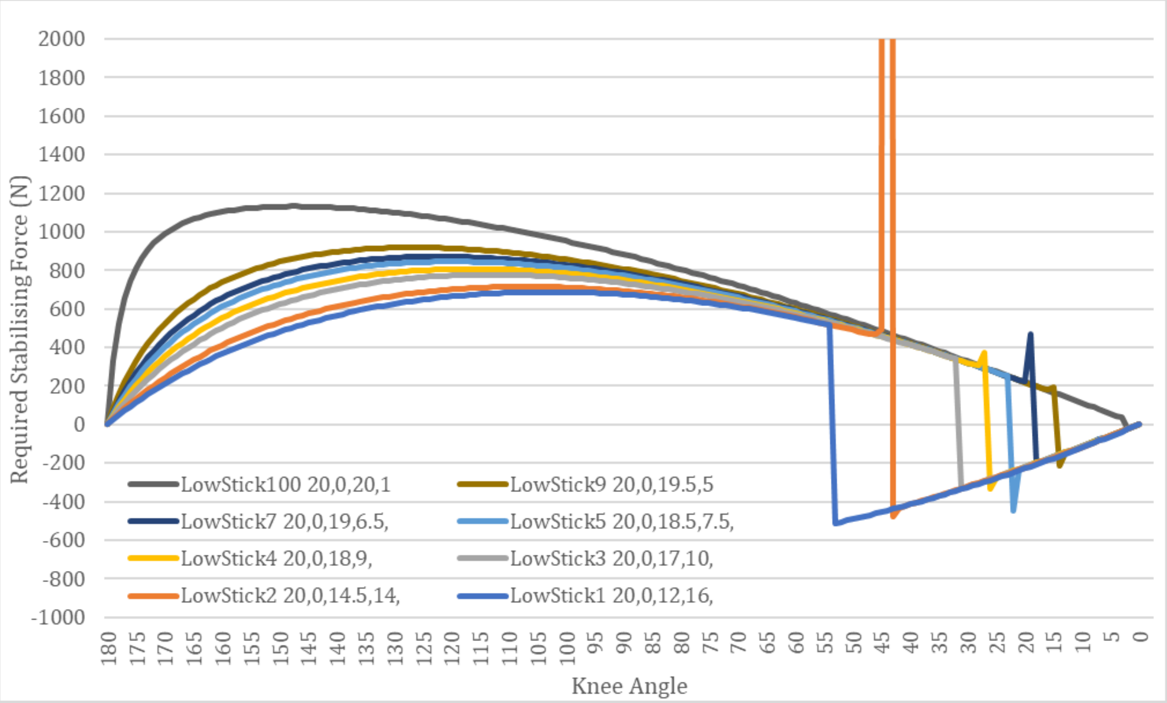


Figure 28 - Change in Lower Stickout Ideal Force Profile depending on Negative Force Multiplier Value

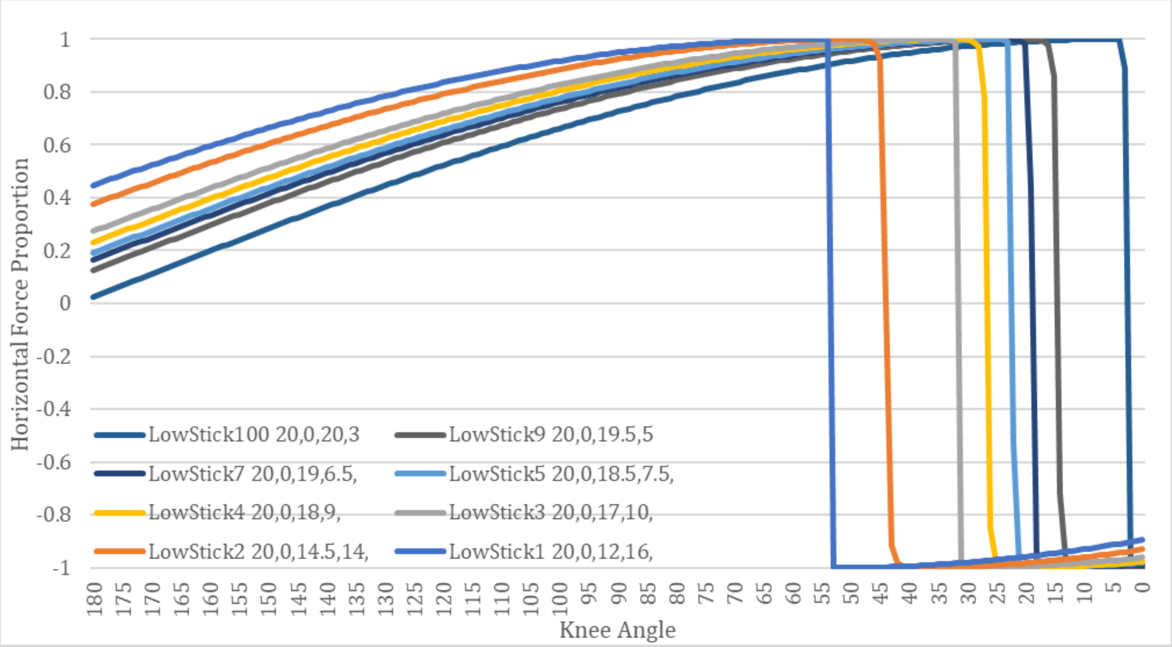


Figure 29 - Change in Lower Stickout Ideal Change in Force Proportion depending on Negative Force Multiplier

With this design considered, it was implemented into the previously developed MATLAB Simulink model seen in Figure 18. This model was modified to perform the simplified crouch manoeuvre. With a static ankle angle of 90° and a freely moving knee joint supported by a Simscape multibody prismatic joint and hip joint whose angle would move relative to the knee joint such that it maintained a vertical position. The model was given the Stickout dimensions matching the ideal Lower Stickout (l_{uV} = 0.2m, l_{uH} = 0, l_{lV} = 0.16m, l_{lH} = 0.2m) and the prismatic joint was set with a force calculated from the force profiles to support all of the model's weight at a knee angle of 90°.

The Model was then set to a different starting knee angle to observe whether it would settle at 90° over time. Additionally, the full model calculation was set to predict what force the prismatic joint would theoretically be providing based off of the model's actual joint angles. If the model successfully stabilised at 90°, and the predicted force matched the actual force, the model and theory would be considered successful. A Second test would then connect this predicted force to the prismatic joint, which would ideally stabilise the model at whichever angle was selected.

The force at 90° for the model using the specified Stickout dimensions and weights used in Figure 17 was theorised to be 715.834N, the model started at a knee angle of 120° and as was hoped the model stabilised at 90.002° with a predicted force of 715.839N at this angle. During the second test, started again at 120° with the predicted force connected to the prismatic joint force, the model stabilised almost immediately at 119.989° with a supporting force of 741.104N, similar to the 741.0859N predicted in theory.

This was considered proof enough that the theoretical model was appropriate for predicting force values within the simplified reality it was designed for, that being only considering static positions within the sagittal plane. As such, the next task would be to further develop the theoretical exoskeleton design based upon this model. For further resources regarding the development and testing of this model, see the listed addendums at the end of this thesis.

3.4.4 Initial Exoskeleton Design

Minimising Financial, Weight, and Energy Costs whilst maintaining effectiveness represents the core concept of the exoskeleton. The First implementation theorised the concept of an actuator design that did not need to provide assistance constantly, but instead only adjust a passive compliant force providing element when a threshold was reached. This idea effectively allows the actuator to remain unpowered for the majority of the exoskeletons use-time, minimising energy cost. Additionally, if a passive element could be the primary provider of assistance that would merely be occasionally adjusted by the actuator, the actuator would not necessarily need to be of high precision, as the compliant nature of the passive element would naturally adjust to minor inaccuracies by providing the leeway seen in Series Elastic Actuator designs. This implementation is similar to others seen in [215], [216], and [217], with the actuator very similar to a Jack Spring.

Figure 30 displays this initial actuator design. Consisting of a rotary electric motor and a spring that acts similarly to the ball screws on other exoskeleton designs that used rotary electric motors to provide linear actuation.

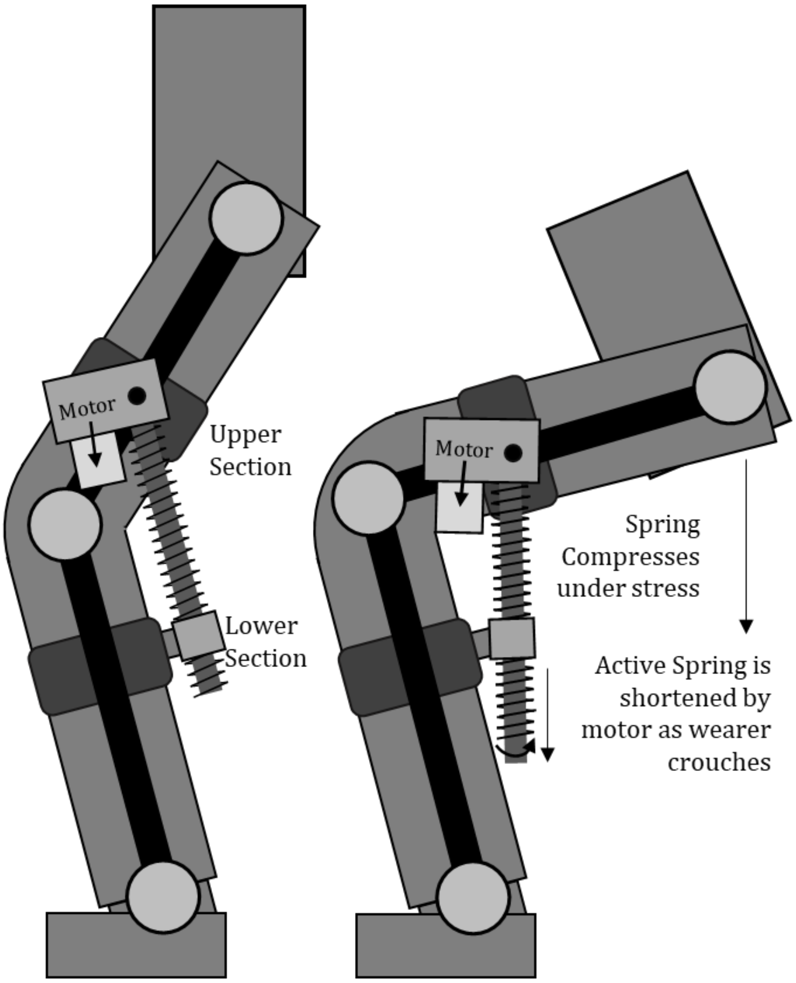


Figure 30 - Initial Exoskeleton Actuator Design

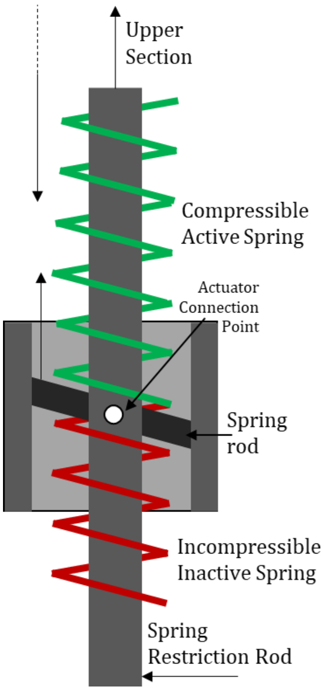


Figure 31 - Lower Section of Actuator Design

The spring, unlike the ball screw would be compliant allowing for compression and extension as the user flexed or extended their joints (The Knee, in this example), with the spring resting on a dowel-like rod slotted through the Lower Section called the Spring Rod (see Figure 31). Similarly to a Ball Screw, the Rotary Electric Motor could then rotate this Spring to increase or decrease its effective length, with the spring being split into an "Active" side between the Upper and Lower Section that would be compressible as the Upper Section and Spring Rod that the spring rests upon would be forced together during flexion. The "Inactive" section beyond the Spring Rod where the spring would have no object to force its compression meanwhile would remain uncompressed. The Central Spring Restriction Rod through the spring would be required to prevent the spring from bending, which would considerably reduce the applied force as well as potentially apply unwanted stress upon the Upper Section as it would too be bent, making rotation difficult.

Effectively, the system could compare an "Unloaded" Spring Length and "Loaded Spring Length" to determine the force applied to the spring. If this force exceeded or decreased beneath set values, the system could adjust to decrease or increase the amount of Active Spring. The Unloaded Spring Length would be calculated using a servo motor encoder to determine the length of the spring based on the known values of the spring's starting unloaded length, its unloaded pitch, and position of the servo motor at time of calculation. Essentially, a closed loop motor could counts each time it completes a full 360° rotation, adding one to the count if it rotates anticlockwise (increasing spring length) and subtracting one if rotating clockwise (decreasing spring length). As the amount of active spring changes with the rotation of the motor, a full 360° is one unloaded pitch-length of the spring either added or subtracted from the Active Spring's total length, allowing one to know roughly how long the spring should be. This is more reliable with closed loop motors, as using open loop motors runs the risk of steps being skipped and the motor going permanently out of sync if its movement were to be resisted until recalibrated, as an open-loop motor would not be aware of its own absolute position.

The Equation for calculating this theoretical Unloaded Spring length would vary depending on the nature of the motor used. But, assuming a motor currently at position m that had M possible positions per 360 Degree Rotation and that could count infinitely positively or negatively to represent multiple full anticlockwise or clockwise rotations, in tandem with a spring of unloaded pitch L_{sp} and an initial Active Spring starting length of L_{s_i} it would take a form similar to (31)

$$\boldsymbol{l_{s\_u}} = \left(\frac{m}{M} * L_{sp}\right) + L_{s\_i}$$

(31) – Unloaded Active Spring Length calculation using Motor Position

The Loaded Spring length has already been calculated, being equal to l_F from (27) Although l_F assumes that the spring passes through or stops directly in line with the actuator connection point as seen in Figure 31. If this is not the case, then this excess distance between the start of the spring and actuator connection points (L_{us} and L_{ls}) would need to be removed to achieve a true Active Spring Length l_s (32).

$$\boldsymbol{l_s} = l_F - (L_{us} + L_{ls})$$

(32) - True Active Spring Length

Based on these final values by comparing the loaded "True" Active Spring Length (l_s) and the Unloaded Active Spring Length (l_{s_u}), one can find how much force is theoretically being applied to the spring based on the difference in compression between unloaded and loaded lengths. In addition to the two previously mentioned values, the spring constant k of the spring used would vary due to the changing length of the Active Spring. This change would be inversely proportional to the changes in Active Spring length (if the length of the spring doubles, the spring constant will halve as there are twice as many coils that can be compressed/stretched). The full Unloaded Spring consisting of all usable active and inactive length would have a static length of L_s and Spring Constant of K and would be compared to the Active Spring with variable length l_{s_u} and spring constant k, shown in (33).

$$\boldsymbol{k} = \mathrm{K} * \frac{L_s}{l_{s\_u}}$$

(33) - Calculating dynamically changing Spring Length due to changing length of Active Spring

Finally, Hooke's Law can calculate the force applied upon the spring via the difference between the loaded Spring Length and Unloaded Spring Length multiplied by the Active Spring's Spring Constant in (34). This finds *F*, the force applied to the upper leg, whose Horizontal Component supports the upper leg against the wearer's weight.

$$\boldsymbol{F} = -k * (l_s - l_{s\_u})$$

(34) - Calculating Force applied by Linear Actuating Spring to Upper Leg

The Difference between unloaded spring length l_{s_u} and loaded spring length l_s, as well as the provided force F would act as the basis for the Low-Level Control System of the device. If the value of F became too high, such as going beyond an "expected" value calculated based off of the positions of the wearer's legs and body and the force moments that would naturally impart, this may mean that the user wishes to deliberately flex their legs, and as such the system would then reduce the length of the Active Spring to allow this process. Similarly, if F became too low, or even negative due to the user attempting to extend their legs, the system would then increase the length of the Active Spring to both prevent the system from restricting extension movement and provide more assistive force. In Summary, the final equations are as follows:

Equation (35) represents the calculation of expected Actuator Force within the theoretical model using ankle, knee, and hip joint angles in the sagittal plane.

$$\boldsymbol{F} = \frac{\left(l_l\, cos(\theta_a) + {}^{l_u}\!/_2\, sin\big(\theta_k + (\theta_a - 90)\big)\right) * w_u + \left(l_l\, cos(\theta_a) + l_u sin\big(\theta k + (\theta_a - 90)\big) + {}^{l_b}\!/_2\, sin\big(\theta_h + \theta_k + (\theta_a - 90)\big)\right) * w_b * p}{l_l\, cos(\theta_a) + l_{uF} * \left(\frac{l_{lF} * \ sin(\theta_{kF})}{l_F}\right)}$$

(35) - Full Force Equation considering position of Lower Legs, Upper Legs, and Body

Meanwhile (32) and (33) represent the actual length of the spring based upon the current knee angle, and the unloaded length of the spring predicted by the current motor position and known starting length respectively.

Finally, (34) represents the force produced by the spring by comparing the difference between actual and unloaded spring lengths, if the required conditions of the theoretical model are met, this "Real" Force value will be the same as that seen in Equation 38. This functions effectively as the "Low-Level" of a potential exoskeleton control system, by calculating the current force at a leg position.

A Higher-level system could then provide a predictive or identifying element, such as predicting a knee angle likely to be reached within a short period of time, a current gait phase, or so on, such that the control system could pre-emptively support the wearer as they moved or provide the right kind of support for the wearer's state, rather than working entirely reactively. However, as the current control system and model exists only as mathematical theory and as simulated models, steps were taken to test effectiveness in reality. The first implementation of this was deliberately crude, both because it was designed as a bare minimum proof of concept and because chronologically it was constructed before the mathematical and simulated model were completed.

3.4.5 Key Variables

This Section will summarise the key variables of the mathematical exoskeleton model.

Basic Model Values:

- Basic Lengths: l_l, l_u, *and* l_b represent the lengths of the lower leg, upper leg, and body
- Basic Angles: θ_a, θ_k, and θ_h represent the ankle, knee, and hip angles.
- Basic Forces: F_u, F_b: The Force applied by the Upper Leg and Body about the knee due to gravity.

Actuator Values:

- F, The Counterforce applied by the linear actuator upon the upper leg,
- F_H *and* F_V, represent the useful Horizontal and useless Vertical Force Components of F.
- l_{uF}, and l_{lF} represent distance between the ankle and the linear actuator's upper and lower contact points
- l_{uH}, l_{uV}, l_{lH}, and l_{lV} represent the Horizontal and Vertical Components of l_{uF} and l_{lF}.
- l_F represent the distance between the contact points of the linear actuator and the upper and lower legs. This may be different from the active length of the linear actuator itself (l_s).
- l_{s_u} the unloaded active length of the spring, the length of the active spring if no force was being applied to it. L_s represents the length of the entire spring, both active and inactive, as described in Figure 31.
- θ_F Force Angle, represents the angle at which the Linear Actuator meets the upper leg contact point.

3.5 Summary

This Section contained the summation of work done prior to the full development of the Low-Cost Exoskeleton design. Whilst more work would be needed to understand the intricacies of the control system and structure, further work would be done in situ with exoskeleton construction itself.

It consisted of the development of a mathematical model that took the form of an inverted pendulum that could simulate the starting effects of gravity upon a wearer within the Sagittal Plane. From this, an applied force to the knee representing an ideal actuator could be calculated that would support the wearer's upper leg and body. The accuracy of this model compared to reality was tested within Matlab, with the resultant kinematic Matlab simulation independently producing results nearly identical to those produced by the mathematical model. With this proven, the ideal upper and lower connection points of the linear actuator and exoskeleton were calculated, considering a full range of knee rotation from 0 to 180 degrees to maximise useful force applied perpendicularly to the leg, as well as maximising the range of motion of the knee before this theoretical exoskeleton would collide with itself.

The Ideal position of "Stickouts" represented the maximisation of applied force by the linear spring actuator upon the wearer during the most useful range of knee angles. As a No-Stickout design would provide little to no assistive force during times where a wearer was standing up due to the linear actuator being in line with the wearer's own leg, therefore seeing all assistive force lost to the Vertical Component (In effect, just pushing up the leg, rather than rotating it about the knee). Therefore, by adjusting this angle at which the linear actuator meets the knee, a higher proportion of useful, Horizontal force could be applied at these angles. Allowing better assistance whilst the wearer was standing up or walking.

The final actuator design took the form of a rotating spring that took inspiration from Rotary Electric Motor and Ball screw designs seen commonly within the Literature Review, in addition to Jack Springs.

These simplified models acted as the initial testbeds for proving the feasibility of the design, however the actual implementation of the Low-Cost Methodology in the construction of an actual functional exoskeleton was considered essential for establishing the legitimacy of the concept. The work of developing the physical exoskeleton structure and hardware and its accompanying high- and low-level control systems in ways that best follow the outlines of the Low-Cost Methodology occupied the majority of the work of the research project. The next sections will detail this development.

Chapter 4: Exoskeleton Development

4.1 Introduction

The Development of the Full Exoskeleton and the control system governing it were some of the major contributions of this thesis. This section will document the development of the physical hardware, structure, and Low-Level Control System of the Exoskeleton with which the exoskeleton would perform basic movement functions, with Chapter 5: covering the development of the High-Level Neural Network-based Control System which would predict the movement of the wearer to allow the exoskeleton to move proactively. The First "Advanced" prototype structure design was a conservative iteration on the initial prototype discussed in Section 4.2.1 using 3D Printed PETG Structural components built around a pre-existing structural knee orthosis, with the High-Level Control System consisting of a Convolutional Neural Network that would identify "Movement States" in the wearer based on collected sensor data, then adjust the actions of the actuator dependant on these states. This would lay the groundwork for a more advanced exoskeleton and control system that made use of a servo motor based compliant linear actuator guided by a High-Level Control System that would instead use a Recurrent Neural Network to predict future knee angle motion of the wearer. This second implementation would become the final, successful design.

4.2 Initial Prototype

One of the primary focuses on the research project is the constructing of a functioning prototype. As such, before deciding on ideal values, a rough initial prototype was constructed to test the developing theory. This "Version 0" prototype, due to being a proof of concept, had dimensions, components, and component placements decided either arbitrarily or based on what hardware could be found without unnecessary expenditure. The intent of this was to build a device that was capable of functioning, and thereby proving at least some of the theory could be relied on. After this, simulations could be made to better optimise for a more final design.

4.2.1 Initial Knee Prototype

This prototype consisted of a first draft design of the linear series-elastic actuator design, which was attached to a simplistic knee analogue (two 4.5x4.5x40cm wood struts connected via a hinge joint) that could have additional force applied via manual effort in order to adjust the angle or simulate a person flexing or extending their legs.

This prototype sought to answer three questions: “How would it work?”, “Does it actually work?” and “If it does work, what is the best implementation?”. Therefore, first, there was a necessity for a reliable way of threading a spring such that it can be split into the compressible “active” section and incompressible inactive section, as well as a method to adjust the proportion of active to inactive spring automatically via an actuator. The spring used was a steel spring of 2cm outer diameter, ~5mm pitch, 18cm free length, with a spring constant of ~2700N/m, these were not specifically chosen requirements, but as with all components of the initial prototype, was chosen from limited pre-existing available hardware without requiring unnecessary expenditure.

The solution consisted of a copper pipe section of a slightly larger internal diameter than the spring’s outer diameter, allowing the spring to move freely through it, and a single stainless-steel rod (a repurposed nail) that would enter through a hole in the side of the copper pipe at an angle such that it would emerge ~5mm higher, as such being in line with the spring such that the spring would rest upon it vertically. This would mean the rungs of the spring could pass harmlessly by the Spring rod if rotated, however if compression was applied upon one side of the spring (the active side) it would rest on the Spring rod without innately bending due to improper resting. This Spring rod was then soldered in place to prevent it from moving. See Figure 31 for visual design and Figure 32 and Figure 33 for initial design.

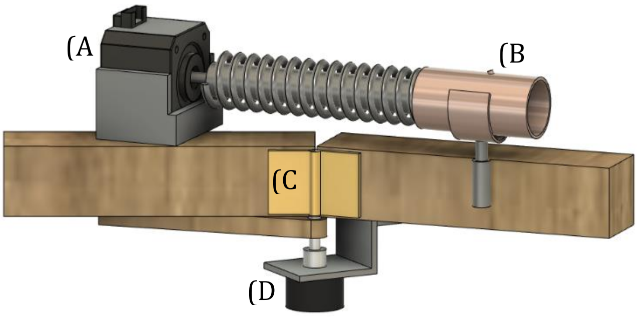


Figure 32 - 3D Mockup of final design: (A) Motor Upper Section. (B) Spring Rod Lower Section in Pipe. (C) Hinge "Knee" Joint. (D) Potentiometer.

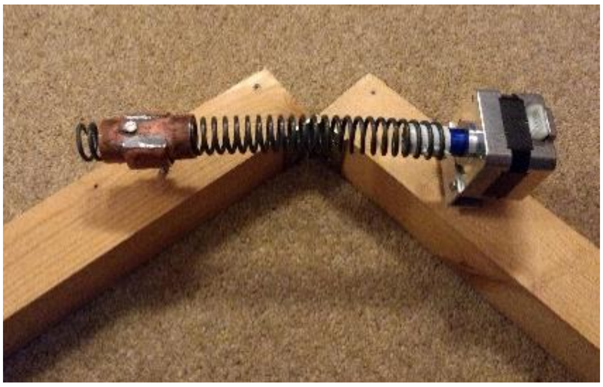


Figure 33 - Initial Prototype Actuator

A 12V 350mA NEMA-17 Stepper Motor [218] would be attached to the spring through its centre via a locking cap that kept the spring in place via friction between it and the locking cap, such that rotating the motor would rotate the spring without slippage. This Stepper motor would rotate the spring through the copper tube, if it rotated in the same direction of the spring’s spiral would result in more spring being on the inactive side and less on the active side thus reducing the stiffness and pushing force of the active side as well as making the active side shorter. The motor was attached to a metal bracket which was itself attached to the upper “leg” via a revolute joint analogue (a screw) in a way that allowed it to rotate freely.

Similarly, the copper tube was attached to the lower leg via a revolute joint analogue and also allowed to rotate freely. This would allow the system to change its angle with the bending of the knee.

This produced a system that, when the leg analogue was flexed, the spring would compress in resistance, if the active side was larger, the spring would provide more resistance. This system had several flaws, most notably that the spring itself would bend much more than desired, considerably reducing the amount of force it provided over if it stayed perfectly straight. In order to reduce the ability of the spring to bend, a metal restriction rod was implemented that sat inside the spring and attached to the motor shaft, this rod would attach to the centre of the motor locking cap in line with it, and extend the length of the spring and out the end of the copper pipe, being at a slight angle as to be to one side of the Spring rod within the copper pipe. This resulted in a much better application of force, with little bending of the spring, as the rod would prevent the spring bending more than a minimal amount. Additionally, the lower attachment point of the actuator was moved such that it was attached to the inner side of the leg, this had the effect of increasing the horizontal component of the force the spring applied to the upper leg and spring as described in in the above section.

This resulted in an altered Knee Angle "θ_{kF}" value with a 9^o difference to the true Knee Angle. The exact positioning of the upper and lower actuator parts on the leg was experimented with in simulation in a later section.

4.2.2 Initial Prototype Sensors and Control System

In order to direct the motor and keep track of the knee's angle, sensors were required. Initially, it was believed that three sensors would be appropriate, that of a Potentiometer to measure the knee angle, an Inertial Measurement Unit to measure knee movement, and a Force Sensitive Resistor to measure the force the spring applied to the lower section of the knee. The sensors chosen for the device were an a RS Pro P25 250Ω Rotary Potentiometer [219], LSM9DS1 Inertial Measurement Unit Breakout Board [220], and an IEE 15.25mm Strain Gauge (Force Sensitive Resistor) [221].

The Potentiometer was placed in line with the knee joint, to measure the Knee Angle, with the upper section of the Potentiometer connected to the upper leg and the lower rotating section connected to the lower leg, such that the movement of the leg would rotate the potentiometer. The Inertial Measurement Unit was placed in a static position at the upper ankle to measure the position of the lower leg and its movement. The FSR was placed just above the lower leg connection of the counter force spring to the leg.

This connection was free to pivot slightly, the sensor itself was placed in between a thin metal wedge as to spread force applied to it by this pivot, which will naturally occur as force is applied to the spring when it is stressed. Each sensor was connected to a breadboard, and further to a Hobby Components 2560 [222] Microcontroller which initially ran a simple program to output all sensor values to a serial monitor. (The Potentiometer, FSR analogue readings, and the Accelerometer, Gyroscope, and Magnetometer X/Y/Z Values).

From this a basic control system was developed that was designed to follow the diagram shown in Figure 34. Built with a level of forward planning for future prototypes, the initial prototype would consist of the low-level, short-term adjustment of spring length in response to knee angle in a passive manner. With long-term, rehabilitative efforts handled by some kind of recognition/prediction system or state machine that had yet to be decided upon. The Control system was programmed in Arduino, with the control system reading in data directly from the potentiometer which ranged from 0-1023 and map it into its relevant angle 1023-0 → 180°-0°, where 180° was standing straight up. During this Initial Implementation, it was found that the Potentiometer alone could provide sufficient information to control the movement of the motor, following the calculations made in 3.4. The NEMA-17 Stepper Motor used in hardware ran at 200 steps per revolution, as such every 100 steps, or half a revolution, the system would recheck the length of the spring in order to see what changes in knee angle had been made in the meantime, adjusting what position the spring would move to.

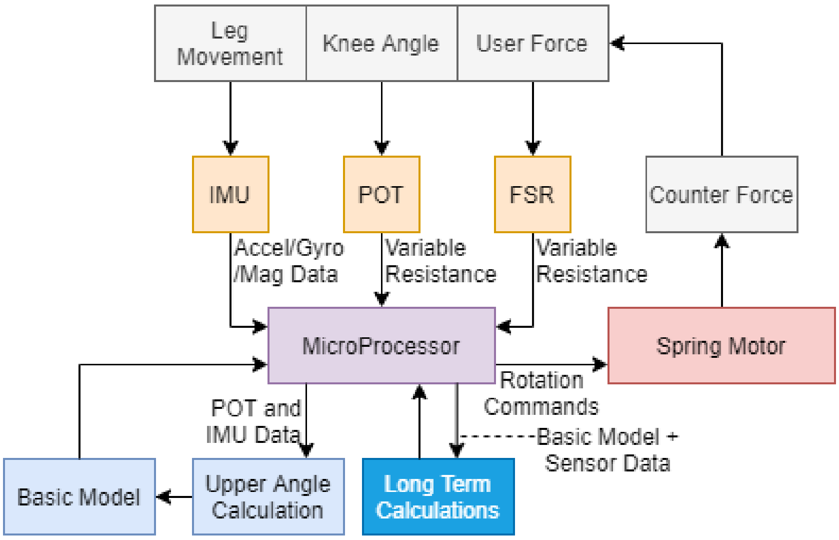


Figure 34 - Theorised Prototype Control System Diagram

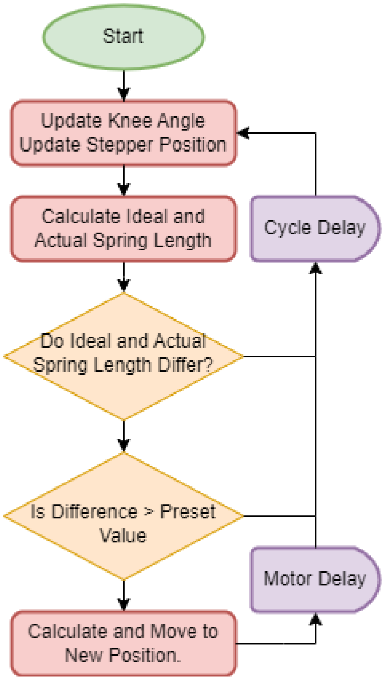

Start
Update Knee Angle
Update Stepper Position
Calculate Ideal and Actual Spring Length
Cycle Delay
Do Ideal and Actual Spring Length Differ?
Is Difference > Preset Value
Motor Delay
Calculate and Move to New Position.

Figure 35 – Basic Control System

The Control System consisted of a simple code loop described in Figure 35. The System would read in the analog value of the potentiometer and convert this to the appropriate knee angle value, as well as using the "Stepper" Arduino Library to read the current position of the Stepper Motor relative to a known starting position. These would be used to calculate the Unloaded/Ideal and Loaded/Actual Spring Length values as per (31) and (32). These would then be compared to calculate the difference in length between them, if this difference in length was greater than a preset value then the system would command the servo motor to move to a position where the difference in length between the ideal and actual lengths was zero, calculated by re-arranging (31) to find m, the new position, and then calculating the difference in steps between this new m and the current position of the motor. The Stepper Motor would then move to this location, and the control system would repeat once the entirety of this movement was completed.

While somewhat slow, this provided a functional knee exoskeleton (Figure 36) that would react to the knee being flexed by reducing the active spring length and increasing it if extending the knee.

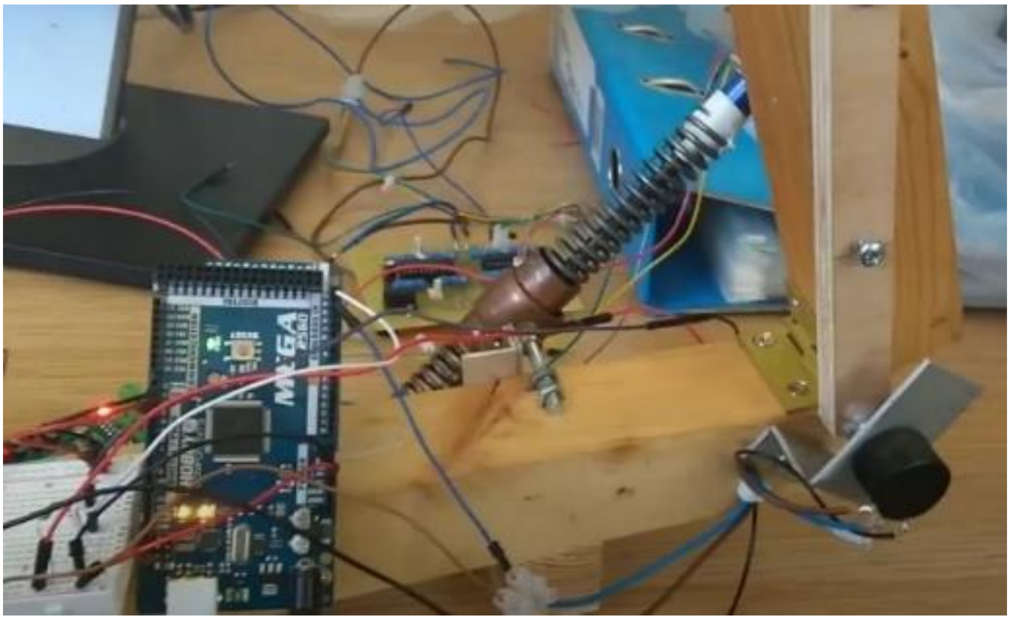


Figure 36 - Functioning Initial Prototype

4.2.3 Summary of Initial Prototype

While deliberately simple, the initial prototype was capable of reacting to movement, rotating the spring to lengthen or shorten when the "knee" was moved. It however struggled if overstrained, getting stuck and losing its position such that it would be permanently out of sync until reset; this was as a result of the open-feedback nature of the stepper motor. The Force Sensitive Resistor was found to not be of much use, while theoretically allowing the system to know when to shorten the spring during knee flexion, it was of little benefit during extension due to providing no information as to the extend to extension, and did not offer much accuracy as to how much movement was needed by the spring due to a limited range of values. The Potentiometer alone provided all necessary information, with its values being able to be used to predict exact required force values in combination with motor rotation.

Other limitations of the design were primarily as a result of its simple nature, such as a weak spring, or it being directly driven by the motor as opposed to via a gearbox, or not being practically attachable to a knee. The core design was however considered feasible enough that it could be adapted into a better prototype, which would have a more accurate way of calculating the length of the spring, and in general be more feasible for usage.

In summary, the prototype successfully answered two of the three questions it set out to achieve, those being "how would it work?" and "does it actually work?". Answering "If it does work, what is the best implementation?" was now worthwhile and to be achieved with more advanced prototypes. However, it was first considered prudent to consider the most ideal placement of upper and lower actuator connections as well as minimum hardware requirements such as the base Spring Constant of the Spring, and the Torque of the Motor, which would be done within simulation.

4.3 Simulink Simulation of Actuator

A MATLAB Simulink simulation was previously made in order to provide additional proof to the mathematical model developed in Sections 3.4.2, 3.4.3, and 3.4.4. This section will go into further detail regarding its functions, in addition to its effects on the chosen actuator Stickout dimension values beyond the previously considered "ideal" values discussed in 3.4.3.

4.3.1 Simulation Details

This simulation of the simplified model discussed in previous sections consisted of two sets of three revolute joints representing ankle, knee, and hip flexion. During Simulation (Figure 37), the Ankle joint was set to a static value whilst the hip joint was indirectly connected to the ankle via a cartesian joint that meant that its orientation was always parallel to that of the ankle, and that the hip angle was always the inverse of the knee angle $\theta_h = (180 - \theta_k)$.

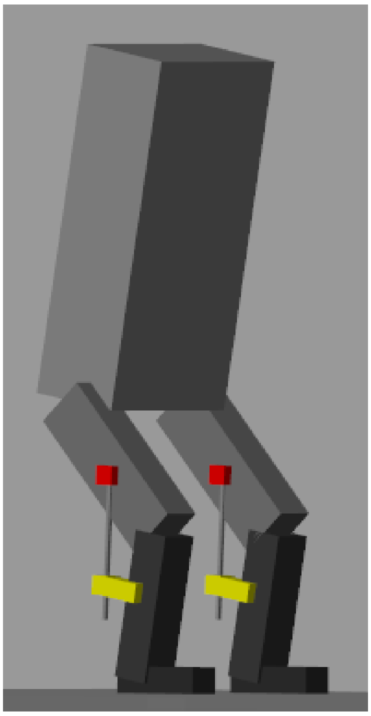


Figure 37 - MATLAB Multi-body Simulation.

Finally, the Knee joint was left to move freely, being supported by a prismatic joint visibly represented by the rod connected between the lower (yellow) and upper (red) Stickout actuator connection points.

Based on the conclusion of section 3.4.3 that the Lower Stickout format would be the most effective, the Stickout design can be seen in Figure 37. The initial dimensions being those found as optimal within that section: l_{uY} = 20cm, l_{uX} = 0cm, l_{ly} = 16cm, and l_{lx} = 12cm. (l_{uX} was set to a value very close to zero to avoid divide by zero errors). With the lengths and weights of the lower leg, upper leg, and body being those used in Figure 17 and its accompanying equations: Lower Leg: 0.387cm, 2.36kg; Upper Leg: 0.4545cm, 4.22kg; Body: 0.92cm, 53.22kg.

This model was attached directly to the ground plane, and as such the feet were unable to move. This, along with the limitations of the simulation to only the sagittal plane meant that this simulation would not accurately represent reality but would act as a "close enough" estimation for the simplified mathematical model it was designed for. The Full Model Diagram is displayed in Figure 39. Consisting of a Main System, two Leg Systems, and the Spring force Calculation Systems.

The Prismatic Joint force would either be controlled by a static force value separately calculated for a specific ankle, knee, and body joint angle position, at which case assuming the ankle joint was preset to its appropriate value the system would naturally settle at the appropriate knee and hip angle that this force was meant to support. Alternatively, the system could be set to act as a feedback loop, with the force calculated based on the system's current knee angle fed into it, meaning that the system would settle at whatever the starting angle was set to.

4.3.2 Evaluation

Creating the model allowed a glimpse as to the theoretical exoskeleton's real design implementation, which led to considering its real-world usability with the currently theorised Stickout vertical and horizontal dimensions. At their current length, they required a linear actuator/spring of a maximum length of, at minimum, 36cm, this was considered potentially too long, as when the user would perform a crouching manoeuvre, especially if they possessed shorter than average legs, there was a high likelihood of the linear actuator/spring touching or brushing against the ground, limiting further movement as seen in Figure 38. Such a spring would also have a not-insignificant weight factor, and could potentially interfere with the wearer's ability to sit down (which would have been potentially be entirely impossible if the "Both Stickout" implementation was used, as the upper Stickout would have extended past the user's leg and into any chair they sat on). Similarly, a large lower Stickout length could limit the user's ability to bend their legs, with the current implementation allowing flexion between 180^o (standing straight up), and ~36^o for the proposed "Lower Stickout" implementation as was shown in Figure 23 and Figure 24.

Shortening the Stickout lengths and positions was considered preferable to improve usability, at the sacrifice of some amount of efficiency. This way allowing the Stickout and linear spring actuator to be less disruptive to normal movement behaviour.

It was decided that the exact length of the Stickout dimensions would be finalised during exoskeleton development based on its structural requirements. The Original Ideal structure maximised distance from knee joint in the Y Axis for both upper and lower leg sections (20cm and 16cm), had no upper leg X Value (0cm), meanwhile having a lower leg X Value ¾ that of its Y value (12cm), and so this would be considered.

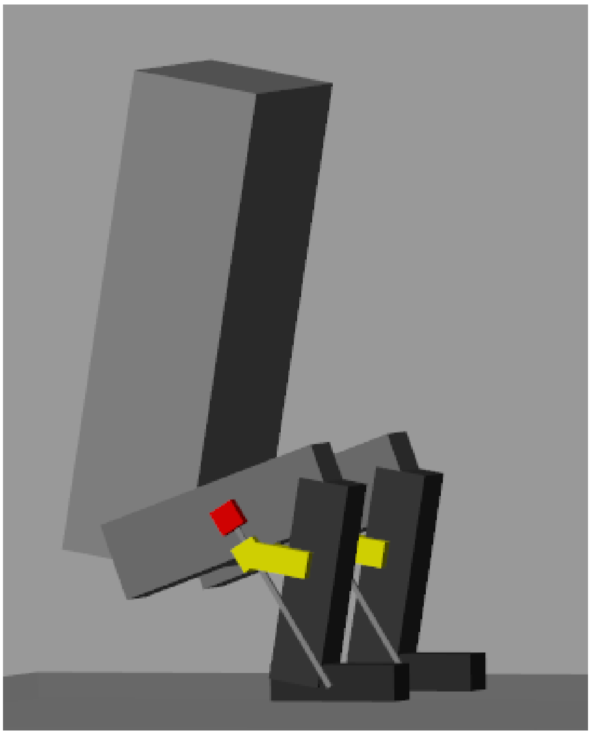


Figure 38 - Crouching Design shows Linear Actuator Close to touching the ground.

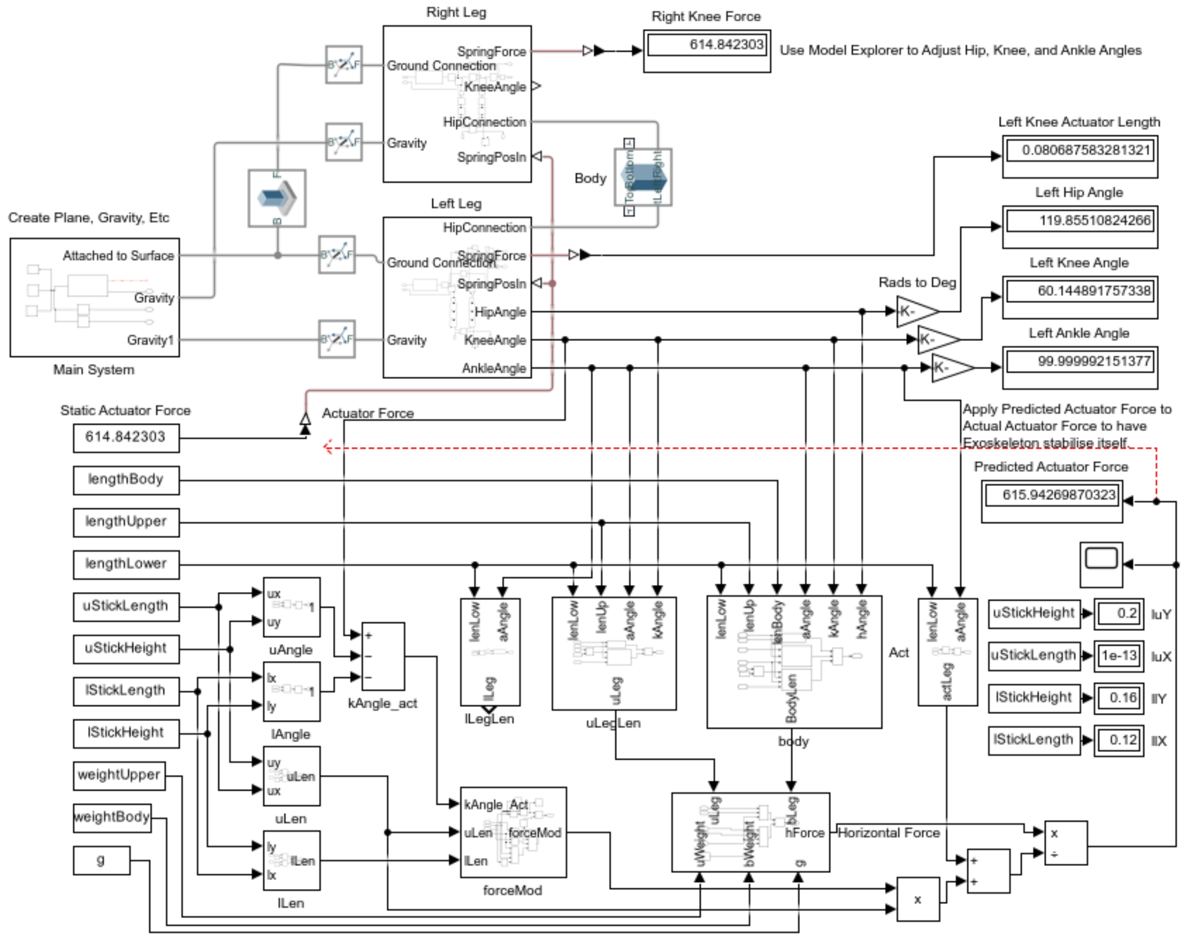


Figure 39 - Full Simulink Model

4.4 Structural Design

Based upon the developments of the Initial Prototype, work could now begin on the Advanced Prototype version 1. The intent would now to be to create a more practically usable design that performed all desired features. The primary components would be the Physical Structure, Actuator, Low-Level Control System, and High-Level Control System. The first of these to be designed would be the Physical Structure and Improved Actuator.

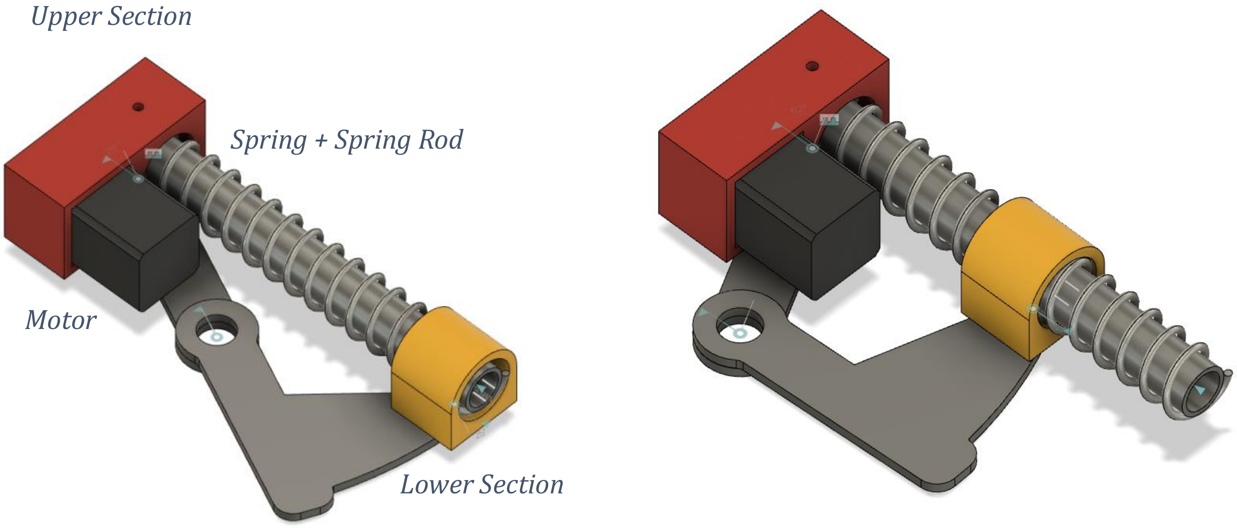


Figure 40 – Conceptual 3D Design, motor would rotate spring to allow for range of flexion/extension to change.

4.4.1 First 3D Designs

A 3D design of the basic knee actuation system was created in Autodesk Fusion 360 to act as a blueprint for future development (Figure 40), displaying the basic elements of the actuator. The Upper section would contain a motor and gearbox connecting to the spring. The spring would then pass through the lower, rounded section which would contain a Spring Rod. The decision to separate the motor from the spring via a gearbox instead of directly driving it was made to both reduce stress on the motor from force applied to it via the spring, as well as to enable the possibility of sacrificing speed for additional torque or vice versa through gearing.

The True Actuator Design was made to attach to a pre-existing knee orthosis (The ROM Adjustable Knee Brace [46]) seen in Figure 55. The Additional components were designed to attach directly to this orthosis via the pre-existing slots used for holding the soft Velcro-held leg straps in place. As such, the top and bottom actuator sections had undermounted Hardpoints that fitted directly into these slots and could be screwed in place (Figure 42) The basic design of which was that of two slot shaped boxes and would fit through the slots of the knee orthosis and held in place with 2mm screws. To these would attach via screws a flat upper plate that would both act as a Stickout for the lower section and flat hardpoint for the upper section to rest on. These are displayed in Figure 42 and Figure 41.

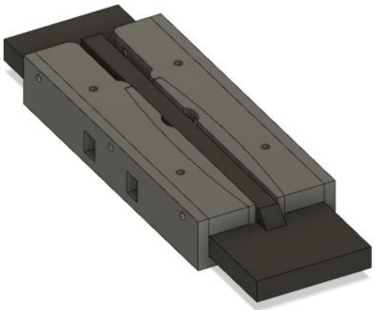


Figure 42 - Exoskeleton Hardpoint slots (Pre-existing Orthosis in black)

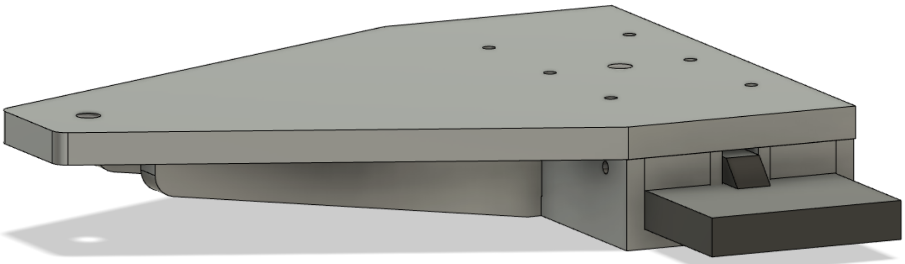


Figure 41 – Exoskeleton Hardpoint Top/Stickout (Pre-existing Orthosis in black)

Upper Section v1

The upper section (Figure 43 and Figure 45) consisted of a block of several double helical gears attached to the motor. Gearing was designed using “GF Gear Generator”, an addon for Fusion 360. The Stepper motor, a NEMA-17 Type motor (Motech MT-1704HS168A [223]), had 2.5:1 Gearing (2.5 Rotations of stepper motor = 1 rotation of spring) to apply more force to the spring.

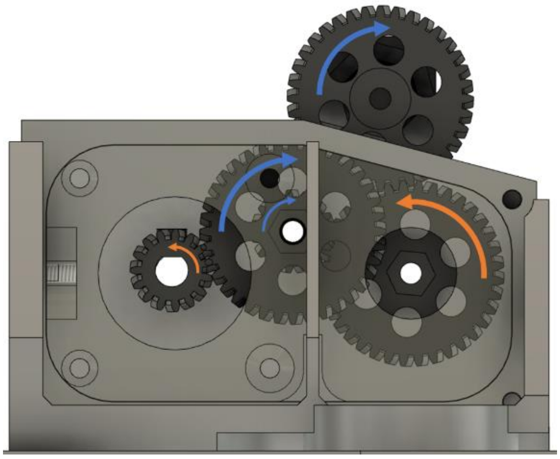


Figure 43 - Upper Leg Actuator version 1 (Front)

An ACE-123 Rotary Encoder [224] would measure the current motor position, this absolute encoder was chosen due to being inexpensive and having the ability to be rotated indefinitely, as well as retaining its position data even if the system was depowered. It provided 128 states over 360 degrees, (~2.815° per state) which was more than sufficient for this implementation, as the motor did not directly control the position of the knee. This upper leg actuator design would attach to the upper leg hardpoint via a 4mm screw, with enough looseness to act as a revolute joint

The spring, a 30cm SODEMANN Spring was chosen due to being reletively inexpensive and being of the desired dimensions was attached to a 3D printed spring connection section, with a restriction rod running its length to hold the spring in place and prevent it bending. This rotating spring attachment (Figure 44) consisted of three sections, the main section the spring attached to (1) which was of an outer diameter equal to the spring's inner diameter allowing the two to stay put via friction, an inner part (2) held in place by a hole through which a screw would go to keep the spring being pulled out, and an outer slot (3) which could freely rotate via a loose screw connection between it and part 2. (4) refers to the spring restriction rod that prevents the spring bending during compression.

Here, a long 3D printed hollow cuboid shape would run the length of the spring, going either side of the spring rod and acting as a spring restriction rod. This acted as a replacement for the metal restriction rod to prevent the spring from bending under compression. Although in reality due to being made of 3D printed plastic, it mainly acted as a stand-in for something more durable made of metal.

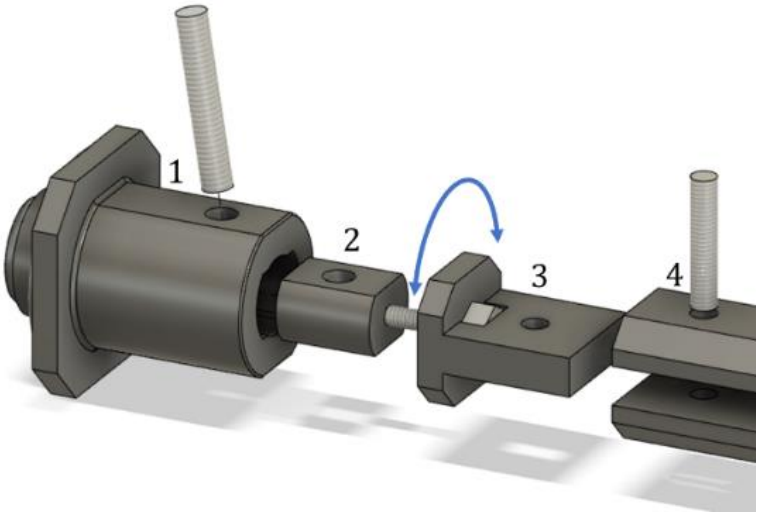


Figure 44 - Rotating Spring Attachment. 1: Spring Attachment point, 2: inner connector to allow rotation of restriction rod, 3: Restriction Rod Connection, 4: Restriction Rod

The encoder allowed for the actual position of the motor to be reliably tracked with little risk of it falling "out of sync". The Encoder used 8 digital inputs for a potential of 256 values, of which 128 were used. The encoder used a grey code system, such that the output 8-bit value of the encoder did not itself represent the current rotary position, instead, the input value was received from the 8 digital inputs, summed as a binary number and stored as an integer, then compared to a lookup table array to find the actual rotary position from 0° to 360°. The Lookup table and other data regarding the encoder were found within its datasheet [224].

Knowledge of the absolute position of the spring, and thus both its starting length and change in length over time allowed for the full implementation of Force Calculation seen in 3.4.4. With this full implementation, a higher-level control system making use of a Convolutional Neural Networks was designed (described in 5.2). While quite slow, this full implementation was capable of reliable tracking of a wearer's knee angle. As seen in Figure 46 with the full 3D Printed Implementation.

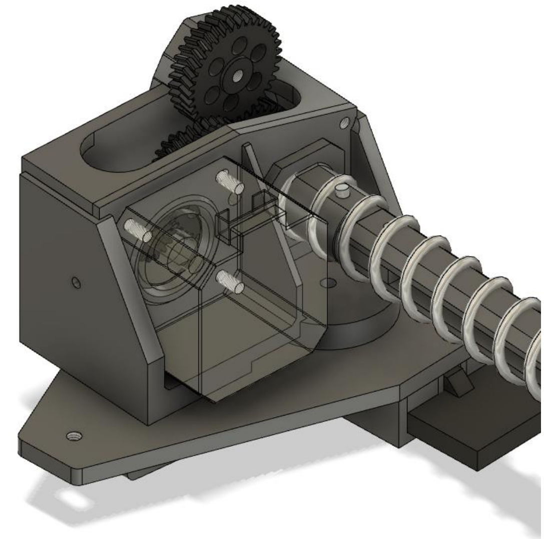


Figure 45 - Upper Leg Actuator, Version 1 (full)

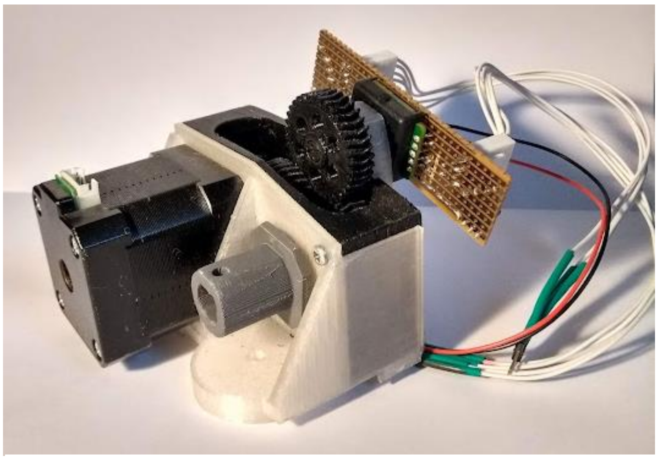


Figure 46 - 3D Printed Upper Leg Actuator Version 1.

Knee Section v1

The First Version of the Knee Joint Section displayed in Figure 47 consisted of a 3D printed gear attached to the knee joint of the orthosis by a tight screw such that it was held in place and moved with the movement of the upper leg, meanwhile a separate attachment on the lower leg held the potentiometer (P25 250Ω Rotary Potentiometer) and gear with a 2:1 Gear Ratio to the larger knee gear. As the Larger gear was capable of moving ~120^{o} with the movement of the upper leg, as the change in knee angle between the upper and lower leg occurred the knee gear would rotate the potentiometer gear, altering the potentiometer value. Converting the 120^{o} range to a 240^{o} range in the potentiometer gear. The Potentiometer gear is held in place by a lock screw to the potentiometer itself, which had one side of its round surface sanded down to a flat point, such that there would not be any slippage.

While the central knee gear was held tightly enough that it would not move on its own due to the force of the potentiometer gear applied to it during rotation, with sufficient manual force it could be adjusted to help with recalibration. The Potentiometer would then have a wired connection to the exoskeleton's microcontroller.

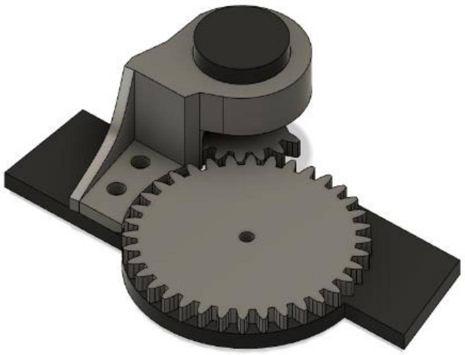


Figure 47 - Exoskeleton Knee Section Version 1

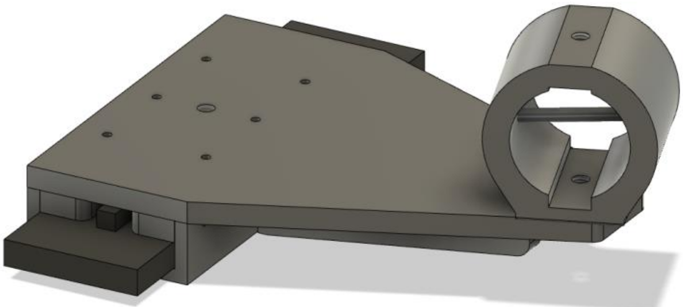


Figure 48 - Exoskeleton Lower Section Version 1

Lower Section v1

The Lower Section displayed in Figure 48 had no electronic components, and simply consisted of a plate similar to the one in the upper section, but raised such that it was in line with the upper section. The hollow cylindrical section which the spring would pass through would hold a metal spring rod that the spring would rest upon when compressed, when rotated the spring would then slot past it, with the spring beyond the Spring Rod not being able to be compressed, and so considered "inactive". The Lower section had to be well aligned with the upper section such that the spring did not apply any undesired abducting or adducting force, as such both the upper and lower connection points would be able to rotate such that they could always face each other, as well as be on the same plane.

4.4.2 Exoskeleton Dimensions

As the exoskeleton frame consisted of several slots where sections could be attached at set locations, it was not feasible to align the upper and lower sections to the ideal distances from the centre of rotation as was defined by theory in 3.4.3. In addition to the consideration of keeping the spring of a reasonable length, the positions of l_{uV} =17cm and l_{uH} = -1.5cm, from centre of rotation for upper section and l_{lV} =12.5cm and l_{lH} = 9.25cm, for lower section were chosen, as close estimates to the ideal values that kept the spring shorter. The force profile and proportions of the used profile in comparison with the ideal profile are seen in Figure 49 and Figure 50.

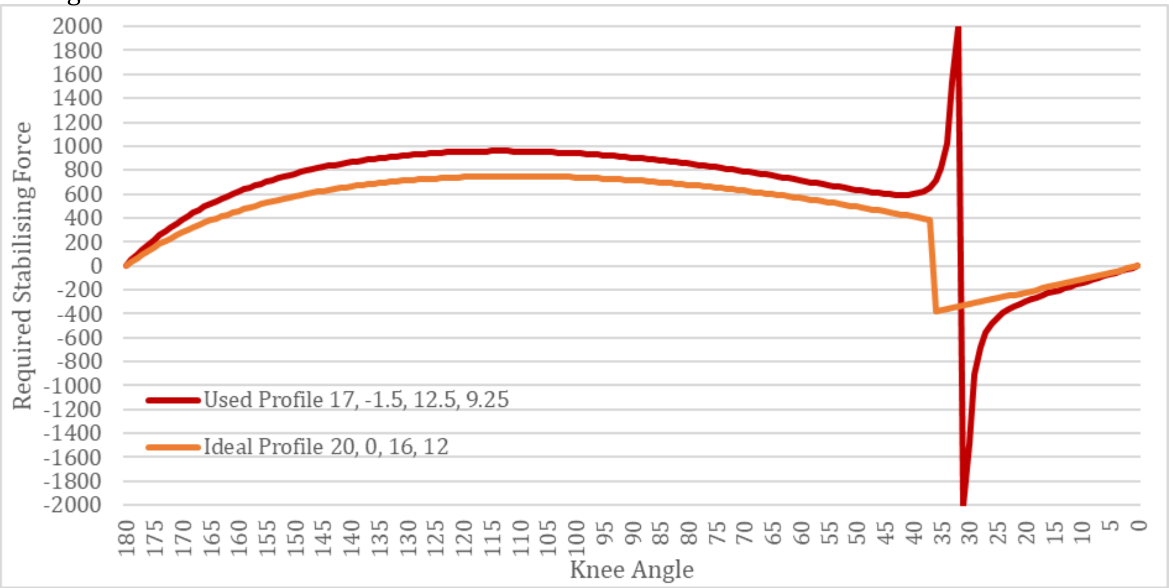


Figure 49 - Used Force Profile vs Ideal Force Profile

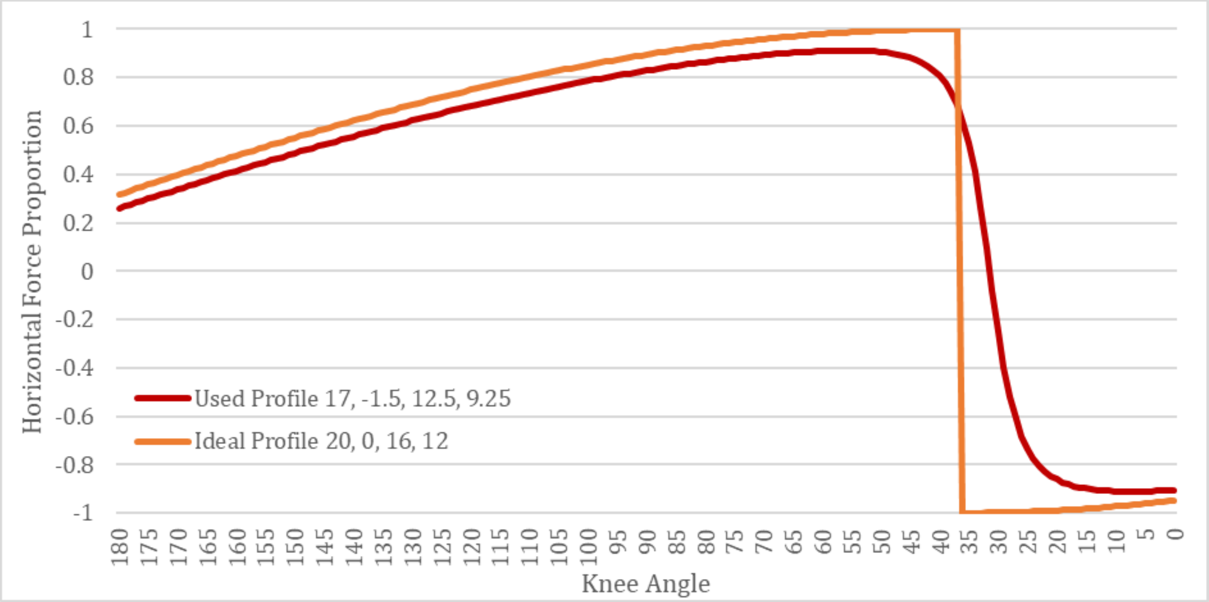


Figure 50 - Horizontal Force Proportions of Used Profile vs Ideal Profile

These values were deemed close enough to the ideal that their difference was an acceptable exchange for ease of implementation. This was a potential innate limitation of pre-constructed structural components. Whilst inexpensive and simple, they risk not allowing for an exact ideal implementation. While the Used Profiles show a larger angle range than the Ideal Profiles, as per Figure 49 the real collision point would be prior to this, due to the small negative (-1.5cm) upper section length value. The real collision point ends up being at ~36 degrees, the same as the ideal profile. In effect, the Used profile has the same angle range, but slightly worse force profile and a lower maximum spring length.

4.4.3 Review of Effectiveness

While the Stepper Motor implementation did successfully demonstrate the basic low-level control system as was outlined within theory, its slow speed and lack of sufficient torque made it unsuitable for any real-world usage. As was described within the literature review, stepper motors saw little to no usage within exoskeletons, with Servo Motors being the majority of all actuator implementations. A Stepper Motor was originally chosen simply due to it being freely available for test purposes, which while appropriate for such test purposes, had delayed effective implementation. As such, during the Structural Redesign, it would be replaced to abide by the conclusions of the literature review and therefore the Low-Cost Methodology.

4.5 Improvements and Redesigns

The new implementation of the exoskeleton structure would aim to fix the primary issues of the first iteration. These being primarily that it was far too slow and underpowered for any practical usage. This would be achieved by finally replacing the stepper motor with a servo motor.

4.5.1 Structural Redesign

As stated, a servo motor replaced the stepper motor used in the initial prototype. The Rhino Motion RMCS-2206 [225] was chosen for this, while relatively large it was also lightweight (180g, same as the much smaller NEMA-17 Motor previously used) as a result of an aluminium frame and ran at an acceptable 600RPM whilst maintaining high torque. Additionally, an internal encoder provided an innately closed system, such that the motor knew its own position without requiring external hardware. This Servo Motor connected to a ESP32 Microcontroller via a USB-to-UART protocol Serial connection, allowing data to be printed and read from the motor. The Servo motor could be commanded to change maximum speed and position (either relative or absolute), and also be asked what the current speed and position were.

The Arduino Microcontroller's limited usability had necessitated its replacement with an Adafruit HUZZAH-32 Feather. Due to its multiple cores, Sensor data receiving, motor angle calculations, and serial communication with the motor could be performed simultaneously, rather than one having to wait until another was complete. The ESP-32 was compatible with Arduino programming via usage of PlatformIO, and so the already existing control system code of the prior prototype could be ported across with minimal difficulty, and then further improved via EspressIF programming features such as Real Time Operating System (RTOS) Task management to make use of the ESP32's multiple cores. Meanwhile, to support this Visual Studio Code became the primary IDE, supplanting the Arduino IDE.

Initially, while a SODEMANN Spring was used as the spring for the design, being of the correct length and possessing a high spring constant of ~2700Nm, this was later replaced with a custom-made spring with a larger pitch and lower spring constant (~1500Nm), to allow for a more responsive passive compression of the spring when force was applied, as well as being able to actively shorten and lengthen the spring faster via the motor.

4.5.2 3D Design

The New Actuator required a new Upper Leg Structure to hold it, which was again designed within Fusion 360 to fit onto the pre-existing leg orthosis. During this time, all other components were revised and updated to higher standards based on observations of their effectiveness.

Upper Section v2

The new upper section displayed in Figure 51 and Figure 52 was designed similar to the prior versions, although now sporting a larger Servo motor it had larger 3D printed Double-Helical Gears meant to easily handle the potentially high load they would carry, these gears were printed in PLA as opposed to PETG for higher fidelity. No additional gearing was present, with the two gears being of equal size and so of 1:1 ratio. As with prior upper leg actuators, it attached to the upper leg hardpoint via a M4 loose screw, acting as a revolute joint. The point at which the spring affixed to the upper actuator section was directly in line with this attachment point, such that it would not impart any notable torque on it.

The servo motor did not need a separate encoder as it had one built in, making the ACE-128 from the previous iteration of the design redundant and the design far simpler. The same Spring Connection method discussed in Figure 44 was re-used here.

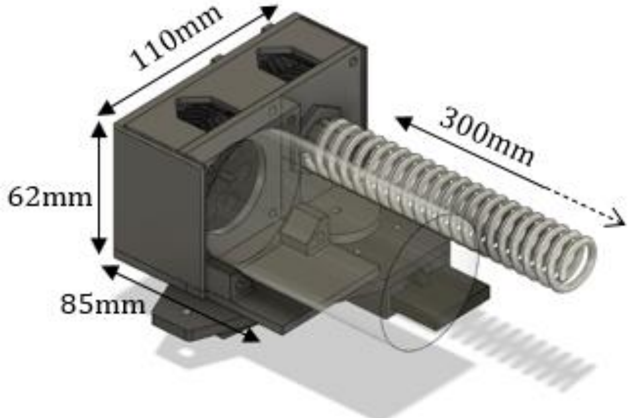


Figure 51 - Upper Leg Actuator, Version 2.

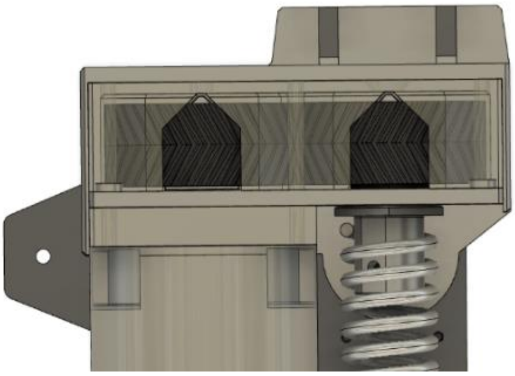


Figure 52 - Upper Leg Actuator, Version 2 (Top-Down View)

Knee Section v2

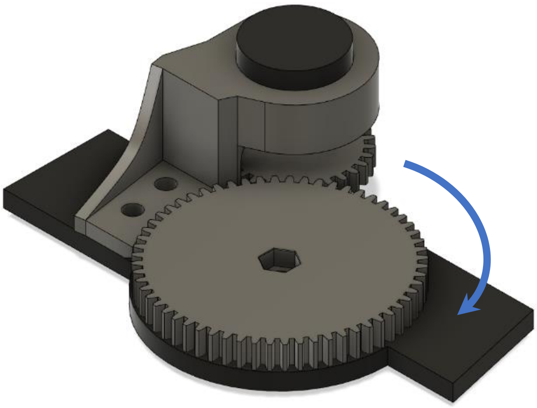


Figure 53 - Exoskeleton Knee Section Version 2

The Knee section displayed in Figure 53 was nearly unchanged from its prior version with its primary update coming in the form of a new set of gears of a smaller tooth size to improve accuracy and slot to allow for a screw nut to better hold the gear in place. The primary knee gear was also thicker, such that the larger motor could comfortably slide across it. These gears were also printed in PLA Plastic.

The Potentiometer used by the knee joint was unchanged, although as with the Knee gear, a new gear with a smaller tooth size was used. As before it was in a 2:1 ratio with the larger knee gear (60 teeth to 30 teeth), such that one turn of the knee gear would result in two turns of the potentiometer gear.

Lower Section v2

The Lower Section displayed in Figure 54 had to be adjusted as a result of the position of the spring increasing in height when compared to prior designs. As such to make sure that the spring did not bend in undesired directions the lower actuator connection point was raised such that both would be in line. However despite this it retained a functionally similar design to its prior iteration.

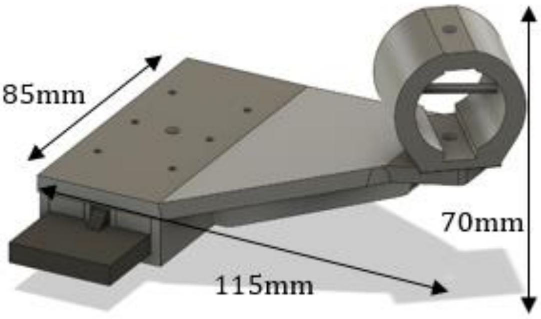


Figure 54 - Exoskeleton Lower Section Version 2

Most 3D printed components were made in PETG plastic apart from the gears, which were made in PLA.

Other Components

The ESP32 microcontroller was placed at the top-most part of the orthosis, wired into a Veroboard with additional ports for Ground and 3.3V wires. The LSM9DS1 Inertial Measurement Unit was placed at the bottom of the orthosis near the ankle, where it would best be able to collect movement data. Both of these were given protective boxes that attached directly to the exoskeleton frame, with ports to allow for easy access/egress for wires, seen in Figure 56. Additionally, for later use, a battery holder with in-built fuse was constructed, to allow the exoskeleton to run off of Battery power as opposed to a static power supply.

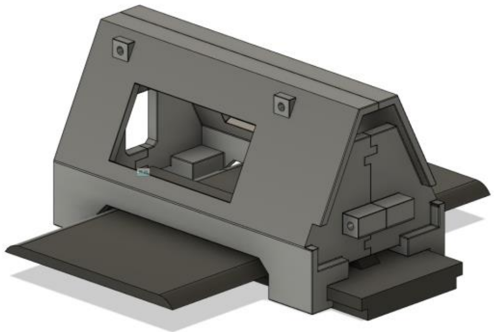


Figure 56 - Hardware Protection Box

All designs were made to allow for easy deconstruction and reconstruction, held together using screws and no adhesive materials. Once the prototype was completed, it was intended that some key components would be rebuilt in metal to improve durability. Notably the Actuator connections to the exoskeleton and Spring Rod. The Final design as it appeared in reality is displayed in Figure 55 (minus Spring Rod) and in design in Figure 57. This consists of the redesigned components assembled and attached to the exoskeleton frame. The frame attaches to the leg via two straps at the top and bottom, held by velcro, and supported by two further straps around the knee that wrap around the exoskeleton to keep it in line with the knee.

When activated, the low-level control system would rotate the motor such that the spring would always be of the correct length between the upper and lower spring connection points to apply no force. The improvement of this system is described in the next section.

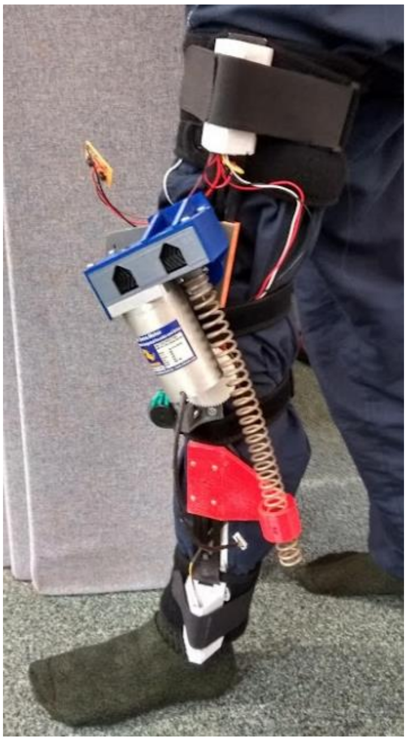


Figure 55 - Full Exoskeleton Build

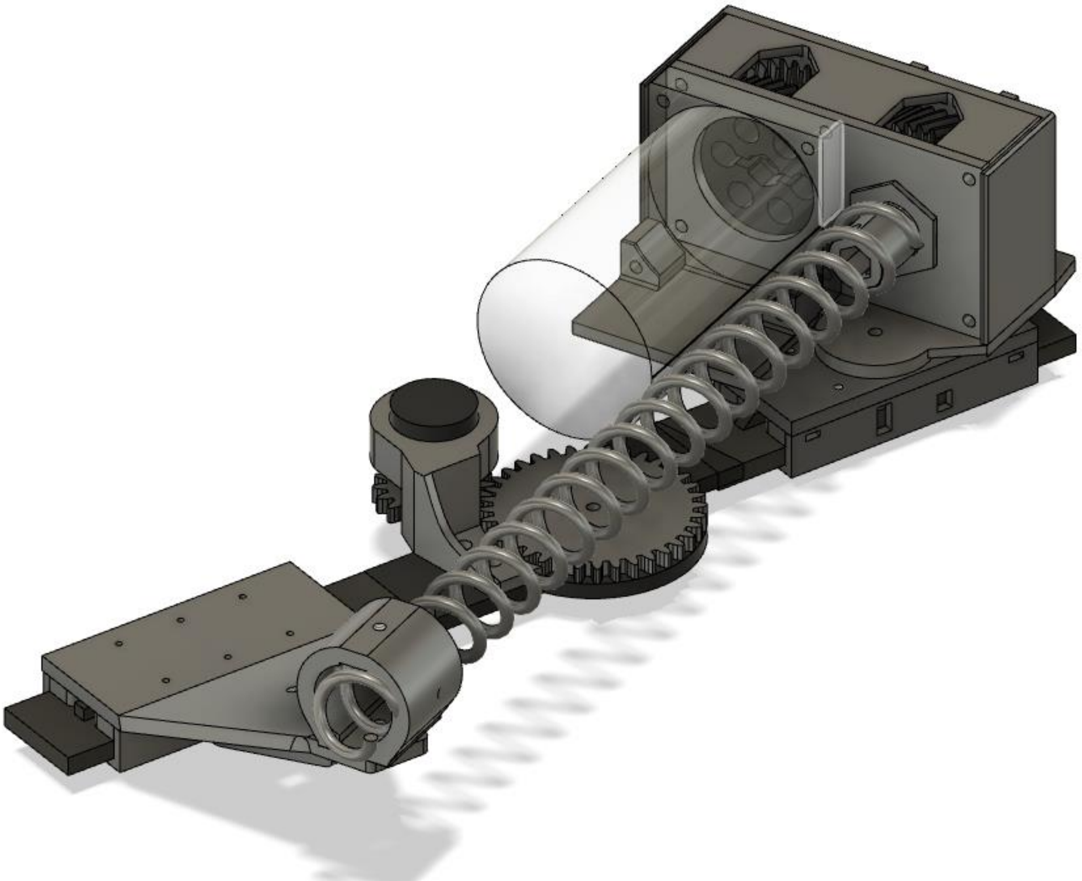


Figure 57 - Full 3D Design of Final Exoskeleton Design

4.5.3 Low-Level Control System

The Low-Level Control System would now be refined into its final form, consisting of four separate “Threads” that would run as part of a Real Time Operating System (RTOS) (see Figure 58). The first of these threads would be the primary input, the sensor input thread *readSensors*, which would be responsible for reading in sensor data from the potentiometer and Inertial Measurement Unit. This data would be sent to both the other two Threads. The Second of these Threads would contain all model-related and prediction functionality, *doModel* would take in sensor data from *readSensors* and output predictions to the third Thread, it was considered the High-Level Control System. The Third Thread *doMotor* controlled the movement of the Servo Motor and was considered the “Output”, and Low-Level Control System. The Final thread *printData* was a debug thread that would receive important information from all other threads and print it to the serial monitor. If data was to be collected this thread was disabled to prevent interference.

Each Thread had an associated setup function which would initiate some aspects of the thread prior to its running, such as finding the initial motor position, loading the model from file, and initialising the Inertial Measurement Unit Sensor. Whilst in operation the ESP32 would run the *doModel* thread on one of its cores, while handling all other threads on its second core.

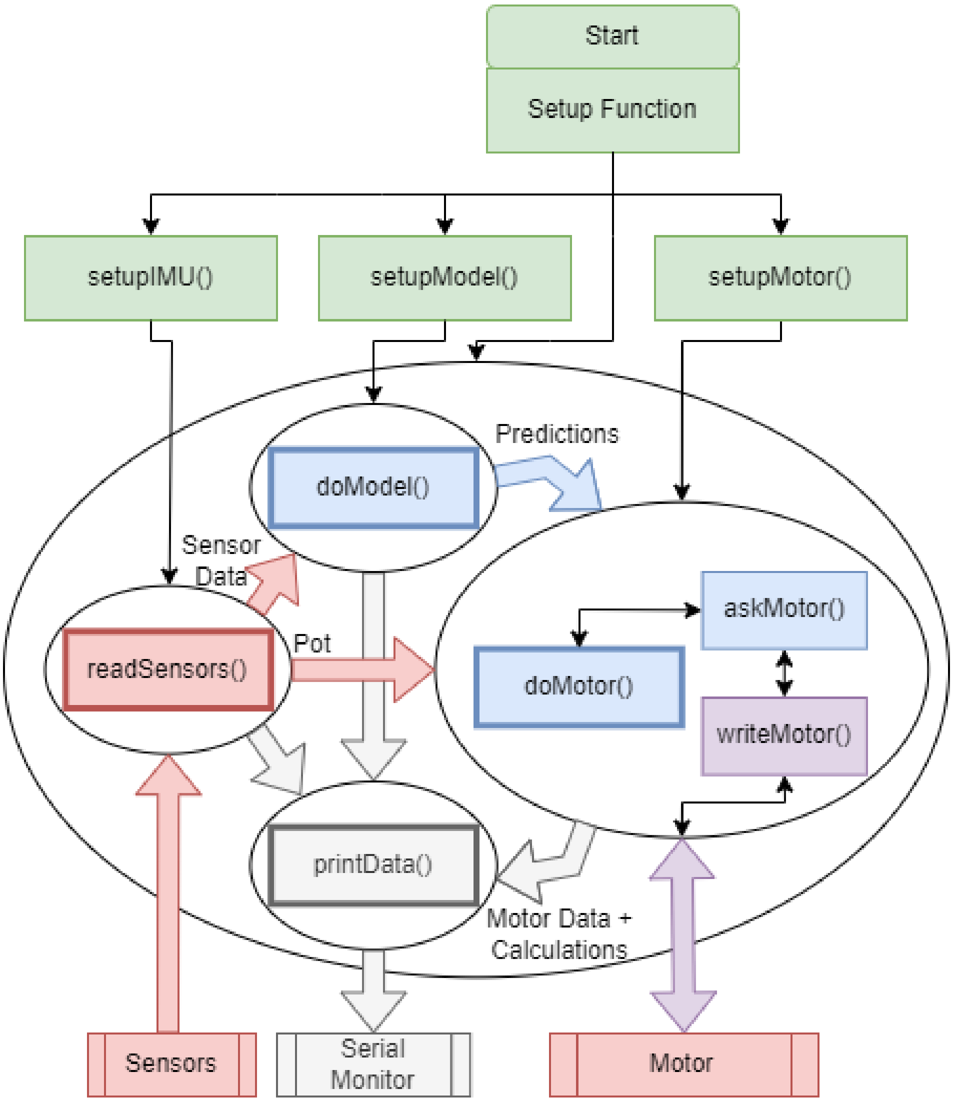


Figure 58 - Diagram of Control System function connections

readSensors() Function

readSensors was responsible for reading in sensor data from the Potentiometer and Inertial Measurement Unit. This sampled data at a rate of 100Hz, however would only record data as an average of 5 values, resulting in *readSensors* recording data at a rate of 20Hz. This allowed for the benefit of higher sampling rates, that being a reduction in noise and a higher fidelity of sampled values, however meant that the system would not need large numbers of samples in order to observe notable changes, keeping the input size of the model small. The function would provide a history of the last 20-40 samples stored as a 2D array to the *doModel* function to act as the model input, as well as providing the current potentiometer value to *doMotor* for calculating knee angle, and provide the current sensor samples to *printData* for debug output.

Each of these sample outputs would be updated after receiving a request from the relevant function in the form of a flag being set to indicate new data was required. The Function would in turn reset the flag to indicate new data had been provided. A Block diagram of *readSensor's* function is seen in Figure 59.

The System was kept in sync using the FreeRTOS ESP-IDF Component [226] to create individual tasks for each function, with each key function using *vTaskDelayUntil()* to extend the time between the start of one function loop and the start of the next to be exactly the desired time, for *readSensors* this was 10ms to allow for a 100Hz sampling rate, although for the other functions it was 50ms (20Hz) as *readSensors* would only record the average for every 5 cycles. This was a compromise to reduce the input size of the Neural Network Control System.

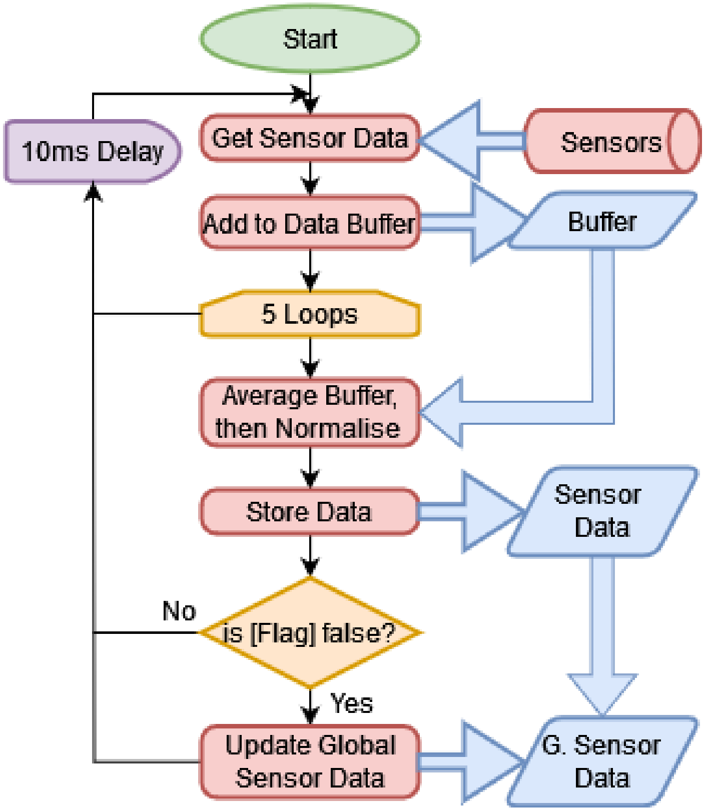


Figure 59 - readSensors() Function Flowchart

doMotor() Function

The Servo motor required a different approach to command than the stepper motor, being able to both be sent data and be requested to return it. Via the RMCS-2206 Datasheet [225], writing data used a command letter followed by a value, such as "G2700" to move to the absolute position of 2700. Meanwhile to request data from the motor, just the command letter was sent, for example sending "P" would return "P2700\n", for Position 2700. Sending and receiving using Serial Communication *Serial.print()* and *Serial.read()* respectively. It was required to read after sending any command, as the motor always responded with at least the command sent. The *writeMotor* Function was therefore created both to write commands to the motor and read the result.

As to not interfere with debug data, information sent to the motor was done along the Serial1 channel as opposed to Serial. This data would not appear in the serial monitor as it was not connected to it. Instead *writeMotor* would first print the string containing the relevent command to Serial1, then, it would wait until a response was received.

If reading was required, after something was detected within the serial buffer (or a timeout point is reached, in which case an error would be sent), the function would serially read in one character at a time and append it to a String. Once reaching a "\n", or new line character, there would be nothing left worth reading at which point anything left in the buffer would be emptied by repeatedly reading until nothing was left. It would then return the read value (or nothing if reading was not asked for).

askMotor was a function that would call *writeMotor*, its purpose was to build on *writeMotor* when specifically asking for an integer value to be returned from the motor, such as when asking for its position. *askMotor* would write a command letter via *writeMotor*, such as "P" to ask for the motor's current position, on receiving the output from *writeMotor* as a String that would take the form of *"P####\n"*, with "#" being a number, it would then convert this string number into a true integer that would represent the current position of the motor.

These two functions were used in the main *doMotor* Function (Figure 60), which primarily consisted of a loop that would first check whether a new present potentiometer value and model predictions were available from *readSensors* and *doModel*, and update if there was. After this, following (32) and (33), it would calculate the knee angle using the potentiometer value and known knee and exoskeleton Stickout dimension constants, as the input potentiometer value between 0 and 4095 on the ESP32 (or 0 to 1023 on Arduino-based systems) could be converted to the knee angle by multiplying it by 285/4095, with 285 referring to the potentiometer's angular range, acquired from its datasheet [227]. As the potentiometer's gear was in a 2:1 ratio with the knee gear of the exoskeleton, this value would then be halved (36).

$$\theta_k = 180 - ((Pot * 285)/4095)/2$$ *(36) - Calculating Knee angle from 12-Bit Potentiometer value*

As data was collected in terms of potentiometer values, knee angle graphs within this thesis will be displayed in terms of potentiometer values rather than knee angle itself. To see an example of both compared to each other, refer to Figure 71.

The Function would also calculate the current number of motor rotations based on the current motor position requested using *askMotor*. Using these values the function would calculate the actual and unloaded spring lengths, and by finding the difference between these calculate how much force was being applied to the spring (34). Based on the predictions of the *doModel* function and this known force, a motor movement amount would be calculated based upon the difference between the knee angle that the motor has rotated to support and the current knee angle, plus or minus an additional modifier applied by *doModel's* prediction. If this difference was above a set minimum the motor would be commanded to move this set amount using *writeMotor* such as to re-align the exoskeleton. If the motor did not move for a set number of cycles, it would set the motor to sleep by using *writeMotor* to set the motor speed to 0, which would prevent power draw.

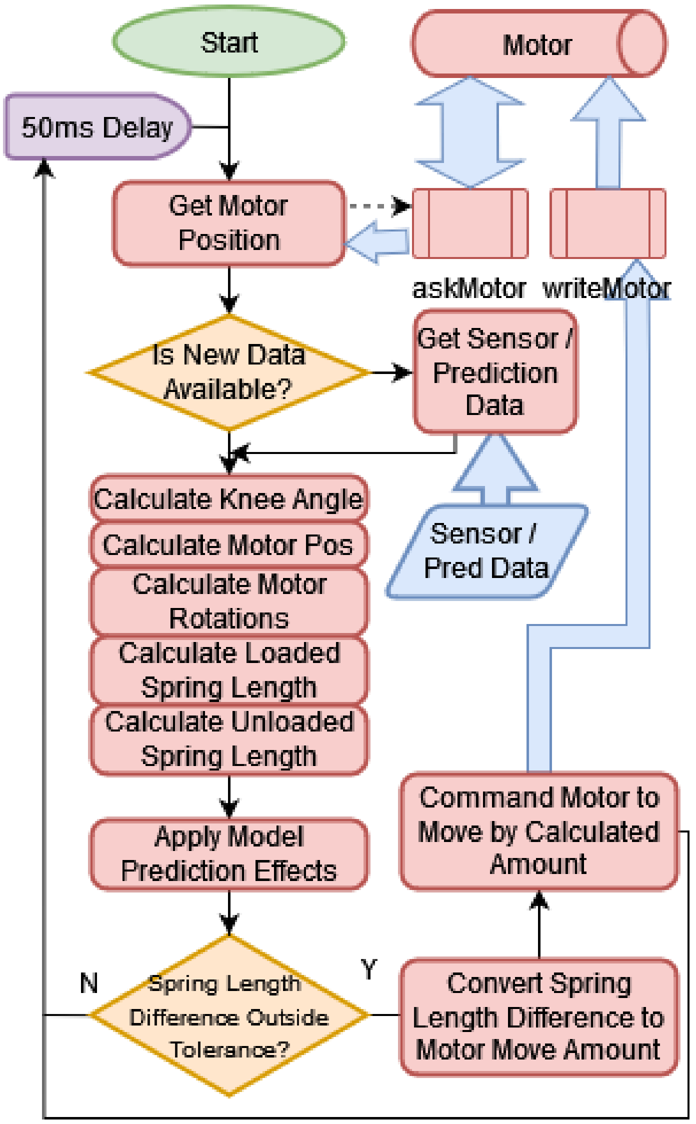
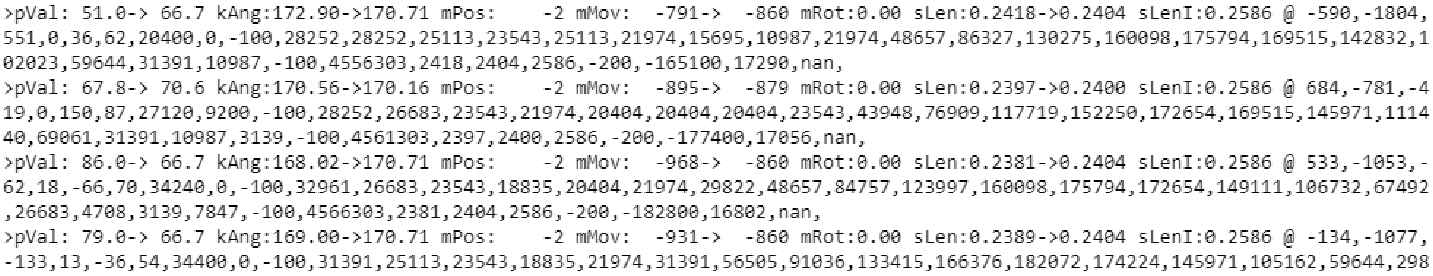

Start
Motor
50ms Delay
Get Motor Position
askMotor
writeMotor
Is New Data Available?
Get Sensor / Prediction Data
Sensor / Pred Data
Calculate Knee Angle
Calculate Motor Pos
Calculate Motor Rotations
Calculate Loaded Spring Length
Calculate Unloaded Spring Length
Apply Model Prediction Effects
Command Motor to Move by Calculated Amount
N
Y
Spring Length Difference Outside Tolerance?
Convert Spring Length Difference to Motor Move Amount

Figure 60 - doMotor() Function Flowchart

doModel() and printData()

doModel was the function that held the TensorFlow Lite Model that would act as the predictive, High-Level control system element. It would take in as an input a 2D array of the last 20-40 samples of sensor data, convert this into a 1D input array and then process this data to predict an output. This output prediction would then be sent to *doMotor* to aid in motor movement. *doModel* was the function that took by far the longest amount of processing time, due to the complexity of the prediction system, as such it was isolated to its own processor core, such that it would not be interrupted or interrupt the running of any other task. The development and implementation of the Prediction system and Neural Network model will be discussed further in Chapter 5:.

printData was a minor function that received data from all other functions. It existed to provide easily readable debug information during development by printing values to the serial monitor. The Function printed the current sample data, the current model predictions, and the current position, movement, and desired position of the motor as well as predictions about the calculated values for the spring's length and unloaded length.



```
>pVal: 51.0-> 66.7 kAng:172.90->170.71 mPos:    -2 mMov:  -791->  -860 mRot:0.00 sLen:0.2418->0.2404 sLenI:0.2586 @ -590,-1804,
551,0,36,62,20400,0,-100,28252,28252,25113,23543,25113,21974,15695,10987,21974,48657,86327,130275,160098,175794,169515,142832,1
02023,59644,31391,10987,-100,4556303,2418,2404,2586,-200,-165100,17290,nan,
>pVal: 67.8-> 70.6 kAng:170.56->170.16 mPos:    -2 mMov:  -895->  -879 mRot:0.00 sLen:0.2397->0.2400 sLenI:0.2586 @ 684,-781,-4
19,0,150,87,27120,9200,-100,28252,26683,23543,21974,20404,20404,20404,23543,43948,76909,117719,152250,172654,169515,145971,1114
40,69061,31391,10987,3139,-100,4561303,2397,2400,2586,-200,-177400,17056,nan,
>pVal: 86.0-> 66.7 kAng:168.02->170.71 mPos:    -2 mMov:  -968->  -860 mRot:0.00 sLen:0.2381->0.2404 sLenI:0.2586 @ 533,-1053,-
62,18,-66,70,34240,0,-100,32961,26683,23543,18835,20404,21974,29822,48657,84757,123997,160098,175794,172654,149111,106732,67492
,26683,4708,3139,7847,-100,4566303,2381,2404,2586,-200,-182800,16802,nan,
>pVal: 79.0-> 66.7 kAng:169.00->170.71 mPos:    -2 mMov:  -931->  -860 mRot:0.00 sLen:0.2389->0.2404 sLenI:0.2586 @ -134,-1077,
-133,13,-36,54,34400,0,-100,31391,25113,23543,18835,21974,31391,56505,91036,133415,166376,182072,174224,145971,105162,59644,298
```

Figure 61 - Debug Information Example

Collectively, the Control System was Hierarchical. Firstly, consisting of the Low-Level immediate calculations of *doMotor* which would reactively keep the length of the Actuating Spring of the ideal length for any particular knee angle such that it would neither push nor pull against the wearer. Secondy, consisting of the High-Level predictive system of *doModel* which would modify this Low-Level action to proactively move with or slightly ahead of the wearer to the knee angle it predicted, thereby applying an assistive force to the wearer's knee.

4.5.4 Summary of Improvements

These improvements upon the initial implementation acted to bring the hardware side of the exoskeleton in line with the results of the literature review and better suited the Low-Cost Methodology by improving effectiveness with as little increase in further costs.

The RMCS-2206 servo motor's replacement of prior stepper motors for example, did not add to the weight cost of the exoskeleton, with a weight of ~180g, similar to that of the prior stepper motor. With a cost of ~£80 considered comparatively small to those commonly seen within other exoskeleton designs. The use of 3D printed components, whilst not suitable for real usage, was considered satisfactory for testing purposes and gathering data.

4.6 Exoskeleton Effectiveness

To determine whether the exoskeleton was well suited for wear, the form of the exoskeleton was tested qualitatively through requesting test participants to review their experience wearing the exoskeleton for extended periods whilst it was not fully activated (whilst no Spring Rod was present, and so preventing the Linear Actuating Spring from applying any force). Quantitatively, the maximum speed and accuracy of the exoskeleton were tested to see whether or not it could keep up with a wearer if the exoskeleton were fully activated.

4.6.1 Qualitative Effectiveness

As outlined by the Core Requirements in 1.2.1, the exoskeleton in addition to being sufficiently supportive for the wearer should also be comfortable enough to wear for extended periods of time and should not notably restrict or inhibit movement. 12 participants wore the exoskeleton for the purposes of collected sample data, as expanded upon in Chapter 5: each also completed a simple question form detailing their experience with the device during their time wearing it. Table 16 below displays the provided questions and the average score of all candidates that answered it, each result having an associated score between 1 and 5, with a higher score indicating a more positive opinion.

Table 16 - Simple Questionnaire provided to Test Participants to gauge exoskeleton comfort.

Name:	Highly Agree (5)	Mildly Agree (4)	Neutral (3)	Mildly Disagree (2)	Highly Disagree (1)
The Exoskeleton was comfortable.	25%	50%	25%	0%	0%
The Exoskeleton did not inhibit natural movement.	25%	62.5%	0%	12.5%	0%
The Exoskeleton would be wearable for extended periods of time if required.	62.5%	12.5%	25%	0%	0%
The Exoskeleton fitted to your proportions and did not sag, slip or feel excessively tight.	37.5%	50%	0%	12.5%	0%

8 of the 12 Participants completed the survey, with results generally presenting the exoskeleton as comfortable to wear for extended periods and not excessively inhibitive of natural movement. For real-world use, the exoskeleton may benefit from an external cover to protect wiring and actuator components from the environment.

With this review of current development, the next sections focus on the Improvements and Redesigns made to improve the exoskeleton to preferred functionality. The next section will cover the development of the Neural Network based Control System that would act as the High-level controller of the exoskeleton, and the challenges faced in its implementation.

4.6.2 Quantitative Effectiveness

For measuring Qualitative Effectiveness, the accuracy of the potentiometer measurements compared to the actual angle of knee. Additionally, the speed and calculated position of the motor were measured. In effect, determining how well the input data used for the prediction system would mimic reality.

To accurately measure knee angle, as the structure of the exoskeleton was not directly straight from knee joint to upper/lower actuator connection point, instead the unloaded spring length between the upper and lower connection points will be set based on knee angle, as defined by Equation (27). The results of which are seen in Table 17 based on exoskeleton dimensions l_{uV}= 17cm, l_{uH}= -1.5cm, l_{lV}= 12.5cm, and l_{lH}= 9.25cm defined in 4.4.2. Measured potentiometer values were converted to Knee angle using (36).

Table 17 - Comparison of set Knee angle vs Potentiometer measured value and calculated knee angle

Set Knee Angle	Knee Angle due to Stickout	Unloaded Spring Length	Potentiometer Value	Calculated Knee Angle	Calculated Spring Length
60^{o}	*28.54*^{o}	*8.17 cm*	3352	63.35^{o}	9.07 cm
90^{o}	*58.54*^{o}	*16.00 cm*	2520	92.30^{o}	16.57 cm
120^{o}	*88.54*^{o}	*22.79 cm*	1690	121.19^{o}	23.03 cm
150^{o}	*118.54*^{o}	*28.04 cm*	778	152.93^{o}	28.46 cm
180^{o}	*148.54*^{o}	*31.40 cm*	0	180^{o}	31.40 cm

As seen, there was only minor difference (~4.2% if excluding 180 degrees due to it being a maximum value) between the set unloaded spring length and calculated spring length, which may be attributed to some mechanical leeway in the gearing that rotates the potentiometer, in addition to inaccuracies in measurement.

Motor Speed

From observing the rate of change in length of the motor versus the rate of change in length required it was clear that the motor was not rotating quickly enough to provide the required rate of change of spring length at a healthy walking pace. The motor was connected to the spring via a 1:1 gearbox, therefore by increasing this gear ratio the length change could be increased. In leu of reconstruction this gearing was simulated (As shown in Figure 62 and Figure 63) by modifying the constant value determining the length of the unloaded spring within the code to be a larger value. This would result in the control system believing that the motor's rotation produced a greater amount of spring length change than it actually did. As the spring was not loaded, with no spring rod present during this testing period, this had no effect on the exoskeleton itself and did not cause any effect on the wearer's movement during wearing.

By doing this, different gearing ratios could be tested in quick succession, with 1:2, 1:3, 1:4, and 1:5 tested. Testing Revealed 1:4 gearing was capable of matching the average walking speed of the author, with 1:5 providing higher reliability during anomalous peaks of faster movement, and so 1:5 gearing was chosen to provide room for faster gaits.

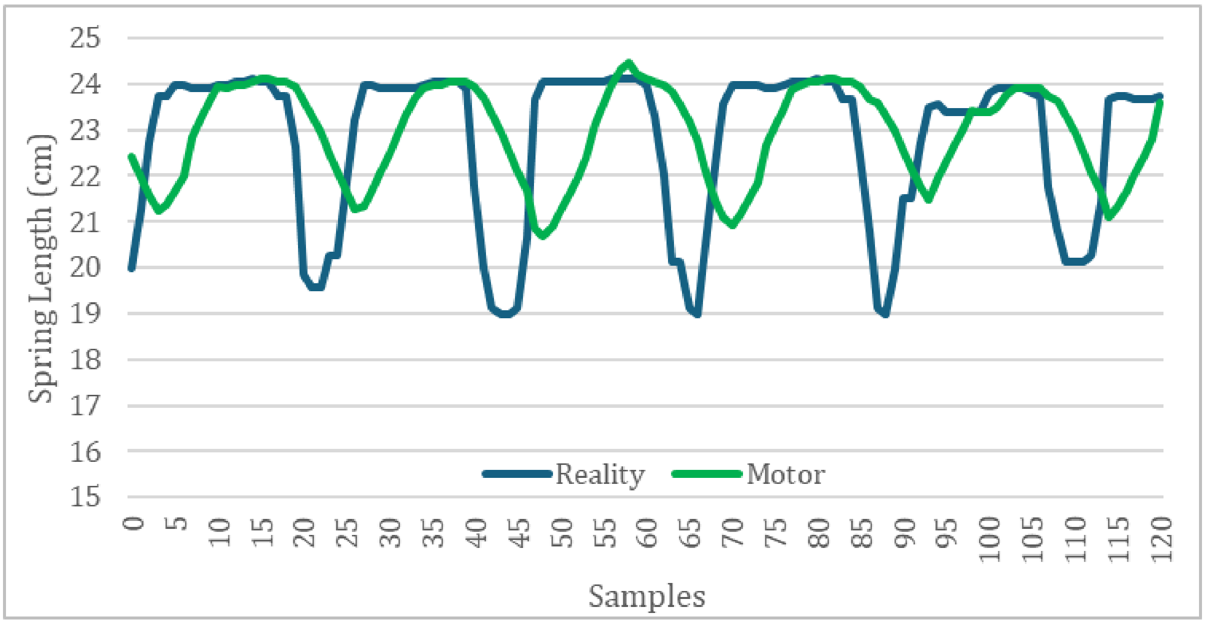


Figure 62 - Change in Spring Length between Reality and Motor. With 1:1 Gearing. Motor moves too slowly to keep up with reality. 20Hz Sampling Rate. "Reality" represents actual loaded spring length as derived from knee angle; "Motor" represents unloaded spring length according to motor position.

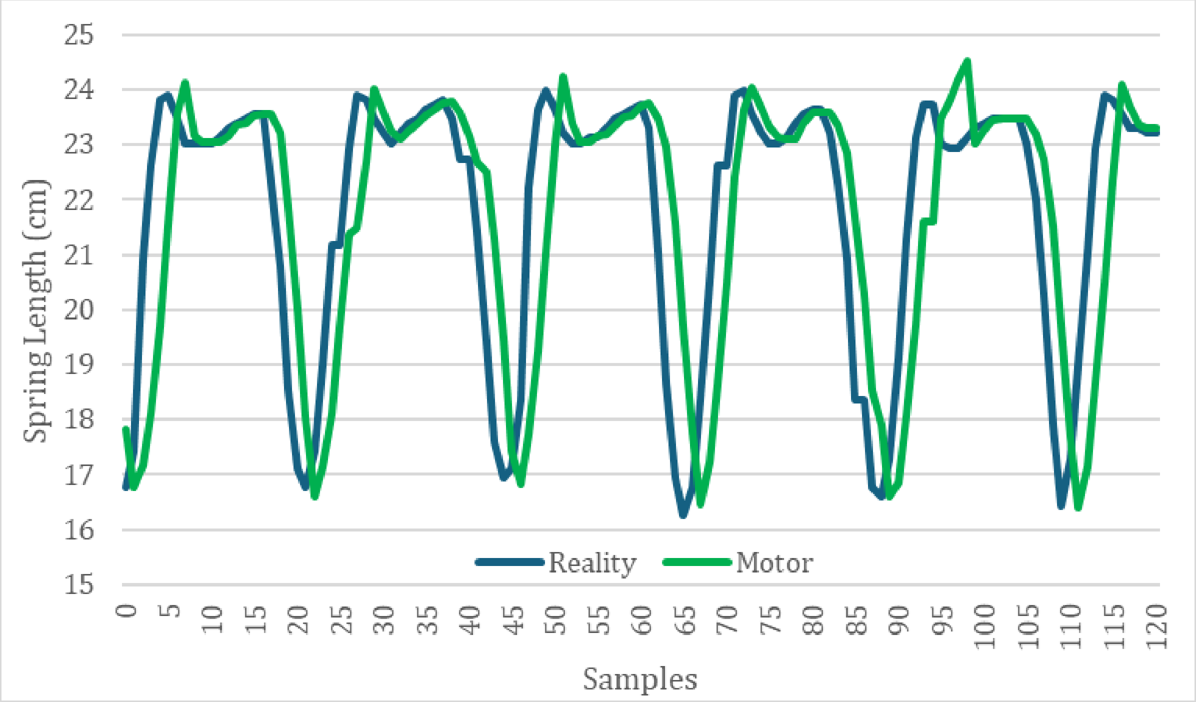


Figure 63 - Change in Spring length between Reality and Motor with 1:5 Gearing. Motor keeps up with reality. 20Hz Sampling Rate. "Reality" represents actual loaded spring length as derived from knee angle; "Motor" represents unloaded spring length according to motor position.

The result provided an output profile that better matched reality and therefore would allow the exoskeleton to keep up with the movement of the wearer.

4.7 Summary

This chapter focused on the development of the physical design and structure and low-level control software of the exoskeleton. Pulling from and realising the initial theoretical work done in the previous chapter. The final actuator design especially, taking inspiration from works such as [216] and [215] to produce a novel, Low-cost implementation designed to minimise cost by reducing the need for low-latency, high precision devices through use of a compliant linear spring element to replace the typically solid ball-screw seen in commonly used rotary electric motor and ball screw designs that were common within the literature review. Within future chapters the effectiveness of this design will be tested.

The Low-Level Control system realised the theoretical work of the prior chapter in calculating the required active spring length based off of knee angle and known upper and lower actuator connection points. With the knee angles of the potentiometer being within ~4.2% of the real angle of the knee, and the control system adjusting the spring to similarly accurate lengths to what was be calculated in theory. The system was tested both qualitatively and quantitatively, showing promise in both aspects.

Chapter 5: Prediction System Development

5.1 Introduction

Neural networks, such as those made using Keras and TensorFlow Libraries within Python, were the next step for the research project, however presented several challenges. Primarily: creating a functional Recognition system using exoskeleton data, converting this functional design to run on a microcontroller or other portable implementation, and then improving its effectiveness such that it would be capable of recognising common movement actions such as walking, standing, sitting, etc.

Within this chapter, a prediction system will be developed that is capable of reliably predicting the motion of the wearer's knee. This will then be deployed to the exoskeleton itself to drive the motion of the motor and spring actuator.

5.2 CNN Control System Prototype

Convolutional Neural Networks (CNN's) were chosen due to the precedence for their implementation on microcontroller technology, although not for the "Movement State" purpose that was intended for them within this implementation. The feasibility of any level of implementation was the first aim of the prototype. With a functional, if not fully considered recognition system developed, made capable of recognising gait states, and then implemented onto a microcontroller to test its functionality.

Convolutional Neural Networks are a deep learning method that extracts features from a dataset by sliding a filter over a dataset, with features arising from dataset patterns. These features can then be pooled to identify those features most prevalent within the dataset. While typically used to identify images through patterns of colours, edges, textures, etc, here it was used to identify a change in angular values over time. Convolutional Neural Networks were chosen as an initial benchmark due to pre-existing support for micro-controller implementations already existing and being well-documented. They were therefore a useful method of testing whether the implementation of a recognition system onto low-power microcontrollers would be feasible outside of external sources and what challenges may be faced in doing so. After this had been established, it would be considered whether a potentially superior, if less well supported network may be more suitable.

Many Neural Networks, including CNN's consist of three "layers". The first being a single Input layer, whose dimensions will match that of the input data and provides the starting point of the system. After these will be a number of hidden layers, who are the intermediate, processing steps of the neural network, with each unit, or neuron of the hidden layer receiving all inputs of the neurons of the previous layer and applying weights and biases upon them which vary with training. Finally, the Output layer's dimension will be that of the output data, this may be a single value, a series of percentages, a continuous set of values, etc.

5.2.1 Initial Data Collection

Data was collected manually using a program that ran on an Adafruit Huzzah ESP32 Feather [154] that would replace the previously used Hobby Components Mega, with the ESP32 possessing a far faster clock speed (240MHz vs 16MHz) and multi-core functionality whilst remaining low-cost. Connected to the ESP32 via a serial connection would be a laptop that would capture the input serial data and decode it back into its component sensor values. Data was collected for a series of different "movement states".

The Results were acquired through wearing the basic exoskeleton structure on the left leg, with the potentiometer attached to the knee joint and Inertial Measurement Unit attached to the outer side of the ankle. Data was collected at a rate of 50 samples per second, with the average step taking between 1.1 to 1.3 seconds (55 to 65 samples). Sensor data was normalised to between -1 and 1 such as to be equally weighted. The Inertial Measurement Unit was oriented as described in Figure 64, with the Accelerometer measuring acceleration in three dimensions, and the gyroscope measuring rotation about three dimensions (Positive = Anticlockwise Rotation).

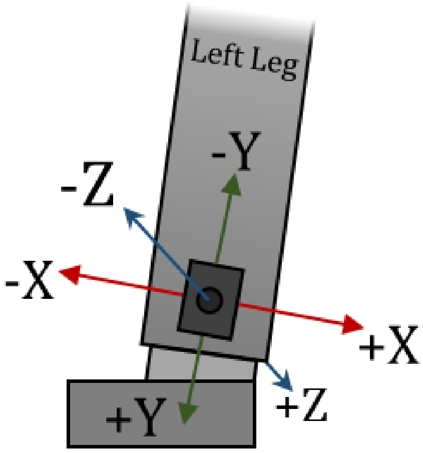


Figure 64 - IMU Orientation

The process of collecting data was completed for each of the states that the recognition system would be trying to predict. Those being "Walking", "Walking up Stairs", "Crouch Up/Down", "Standing", "Sit", and "Rest". With this, the recognition system itself could be developed. These Movement States were decided based on the most common everyday movements that a wearer would perform. Waking data is seen in Figure 66, Figure 68, Figure 67 showing Potentiometer, Accelerometer, and Gyroscope data respectively. Walking up Stairs Potentiometer data is seen in Figure 65.

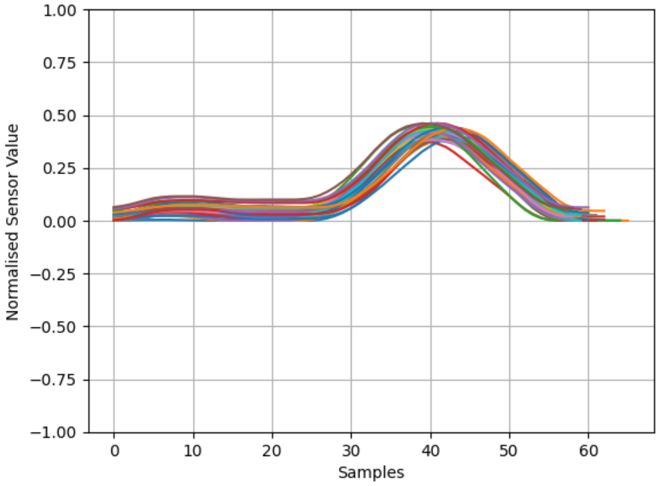


Figure 66 - 32 Instances of Potentiometer values for Walking. 0 represents 180 Degrees. Sample rate: 50Hz

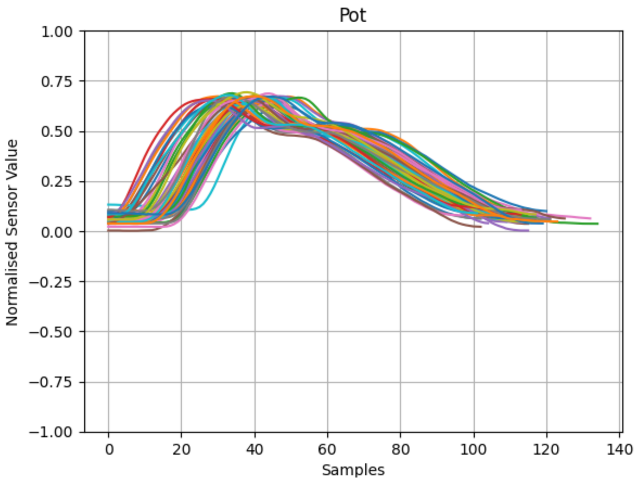


Figure 65 – 32 Instances of Potentiometer values for Walking up stairs. 0 represents 180 Degrees. Sample Rate: 50Hz

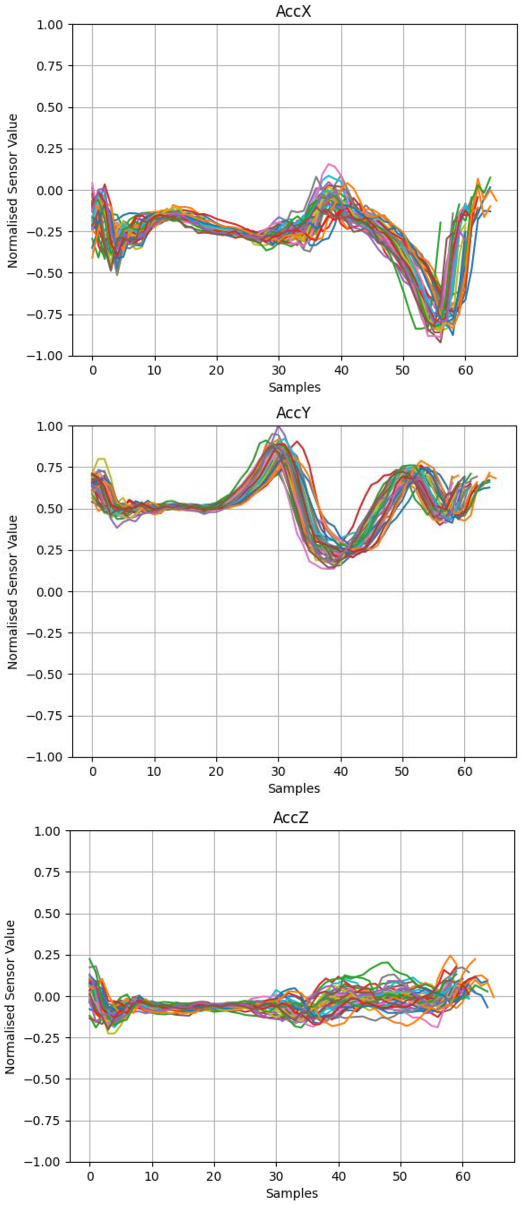


Figure 68 - 32 Instances of Accelerometer X, Y, and Z Data for Walking Cycles. Sample rate: 50Hz

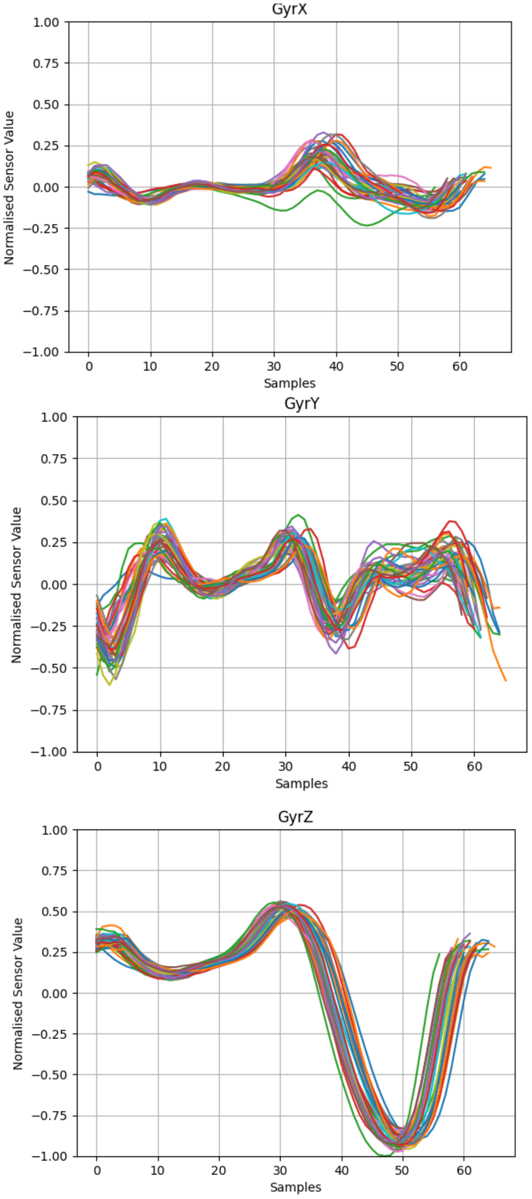


Figure 67 - 32 Instances of Gyroscope X, Y, and Z Data for Walking Cycles Sample rate: 50Hz.

Crouch was split into "Up" and "Down" states, with Crouching Up consisting of a wearer moving from a sitting position to a standing position and Crouching Down consisting of the inverse (Figure 69).

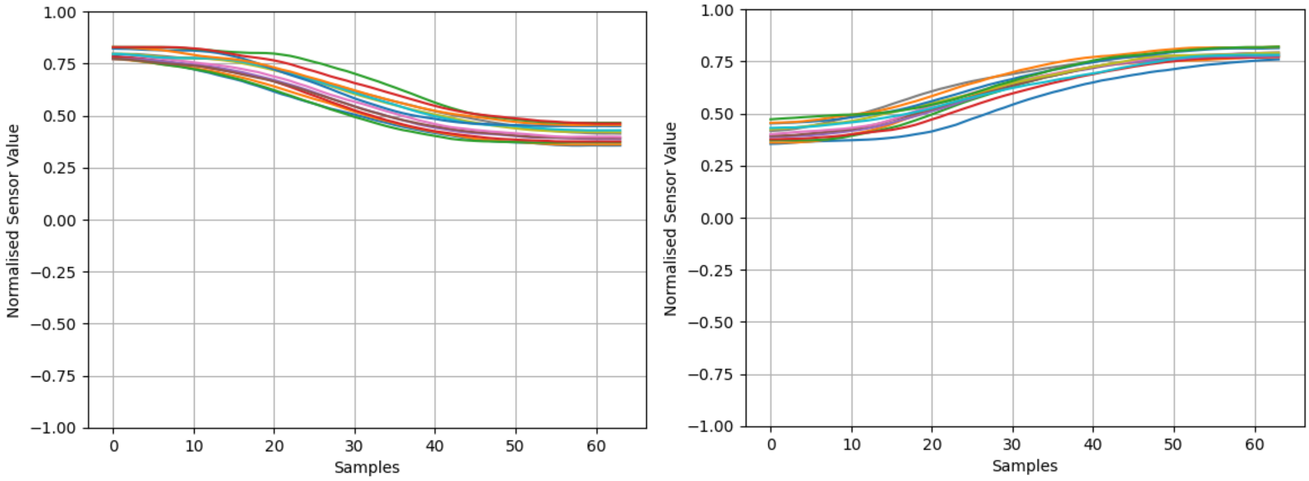


Figure 69 - 32 Instances of Potentiometer Values for Crouching Up and Crouching Down. Sample rate: 50Hz.

Standing, Sitting, and Resting were all static states, with little movement occurring. As such their profiles did not show much change, appearing as flat lines (Seen in Figure 70). Rest was specifically used as a test state, with the exoskeleton resting on its side. Differentiating it from Standing.

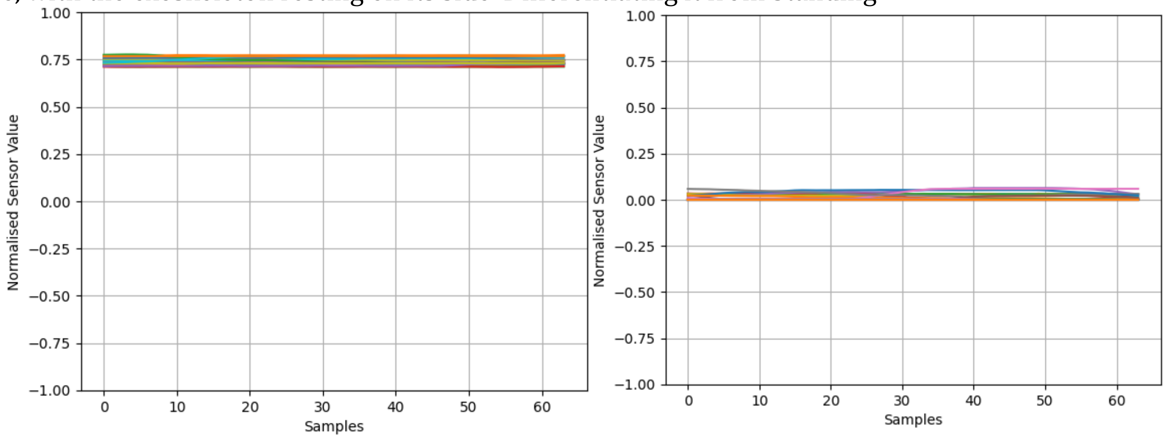


Figure 70 - 32 Instances Potentiometer Data for Sit and Stand Values. Sample rate: 50Hz.

When viewing the resultant data continuously as seen in Figure 71, walking data showed similarities to other datasets, such as seen in Figure 72 with data collected by [228]. This data was collected at a rate of one sample per 1 millisecond (1000Hz) as opposed to 50Hz.

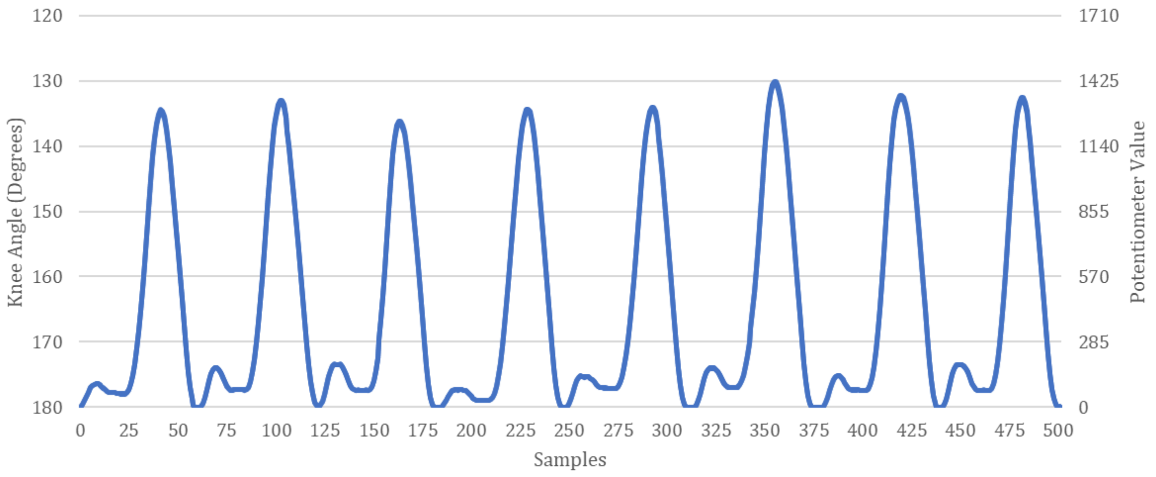


Figure 71 – Change in Knee Angle over multiple cycles (180 degrees equal standing leg). Knee Angle and Potentiometer Value compared. Sample Rate: 50Hz.

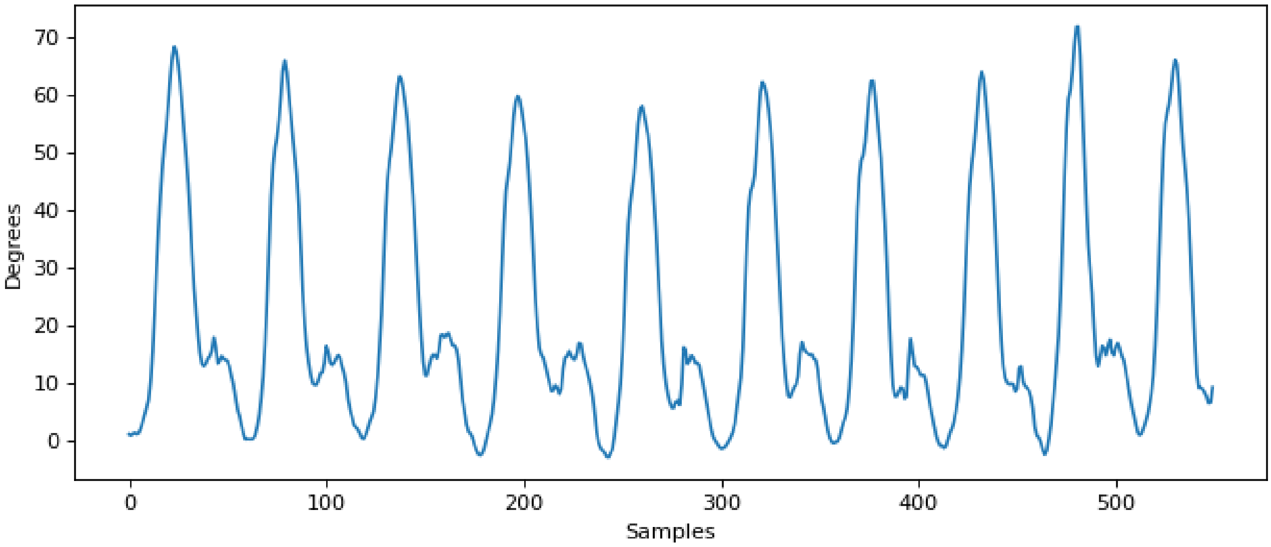


Figure 72 - Example sample data by [218], edited to show sample rate of 50Hz.

Each Sensor within the Dataset was individually normalised, (In other words, accelerometer X, Y, and Z would all be normalised together, Gyroscope X, Y, and Z would all be normalised together, and the Potentiometer normalised on its own), where 1 = highest value. This was due to sensor types that had innately higher values being erroneously given more significance if all values were collectively normalised, so, for example the potentiometer which would vary from 0-4095 would have its results hold more significance than the Gyroscope, which may only vary between 0-10. Therefore, normalising the values based on the sensor type made each sensor of equal weight.

The Recognition system would run for 100 epochs where it would have access to the training data and training states. The reduction of loss over the training epochs (Figure 73) indicating the improvement of the recognition system and recognising the desired states, although it did not necessarily mean that the model was "better" at generalised recognition. Overtraining a model could lead to overfitting, where it would only recognise the training data and nothing else unless it looked exactly like the training data. After this training completed, the model would be allowed access to the testing data, although not its answer states, instead it would try to predict these states itself, not "knowing" whether it was correct or not.

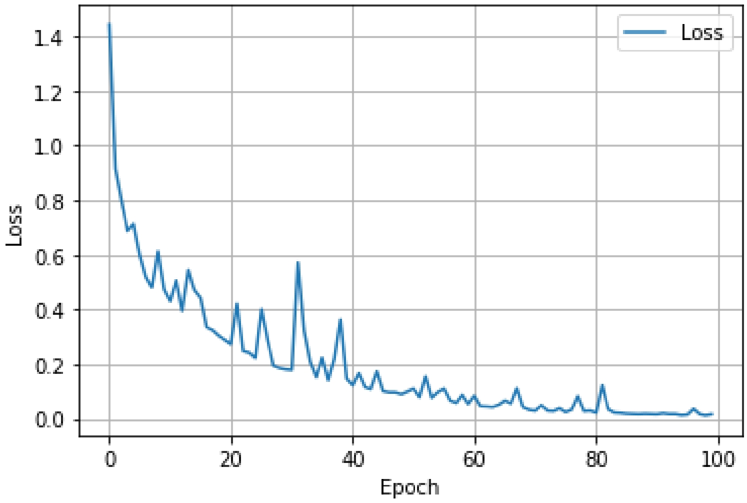
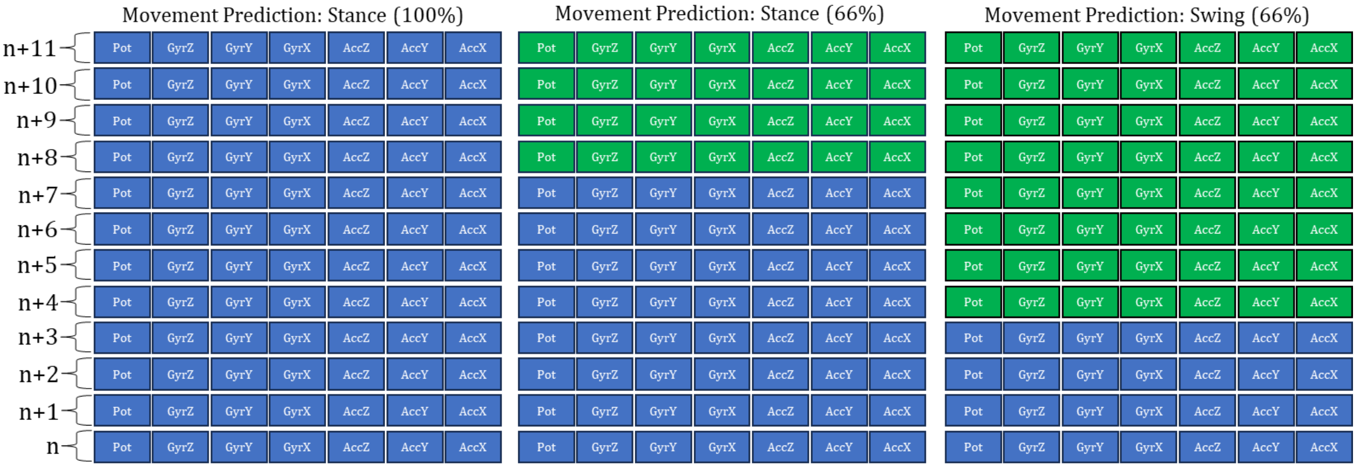


Figure 73 - Loss Reduction over training Epochs

5.2.2 State-Based Recognition Core Issues

It was found that the system took a long time to react to changes in circumstances, with it taking upwards of a second to change its prediction from one Movement State to another. This was attributed to the problem of new information taking time to “shift” through the recognition system’s input (displayed in Figure 74). With a new movement state only recognised when recognisable data of this Movement state made up the majority of the recognition system’s input.

At a bare minimum if at least 50% of the input needed to be of a particular movement state for it to be recognised, it would take 0.64 seconds, For Stance to Swing phases, this meant that the reaction time to realise a state change was about as long as the state itself lasted during average walking cycles, effectively meaning the system did not realise it was in a new state until that state had been left.


Movement Prediction: Stance (100%)
Movement Prediction: Stance (66%)
Movement Prediction: Swing (66%)
n+11
n+10
n+9
n+8
n+7
n+6
n+5
n+4
n+3
n+2
n+1
n
Pot GyrZ GyrY GyrX AccZ AccY AccX

User Switches from Stance (Blue) Phase to Swing (Green) Phase. System still believes it is in Stance Phase.

System begins to read in data it can recognise as Swing Phase, but the majority of data is still showing Stance Phase, so the system still believes it is in stance phase.

Majority of data is now Swing Phase, System now recognises User is in Swing phase, however only some time after they actually entered Swing phase.

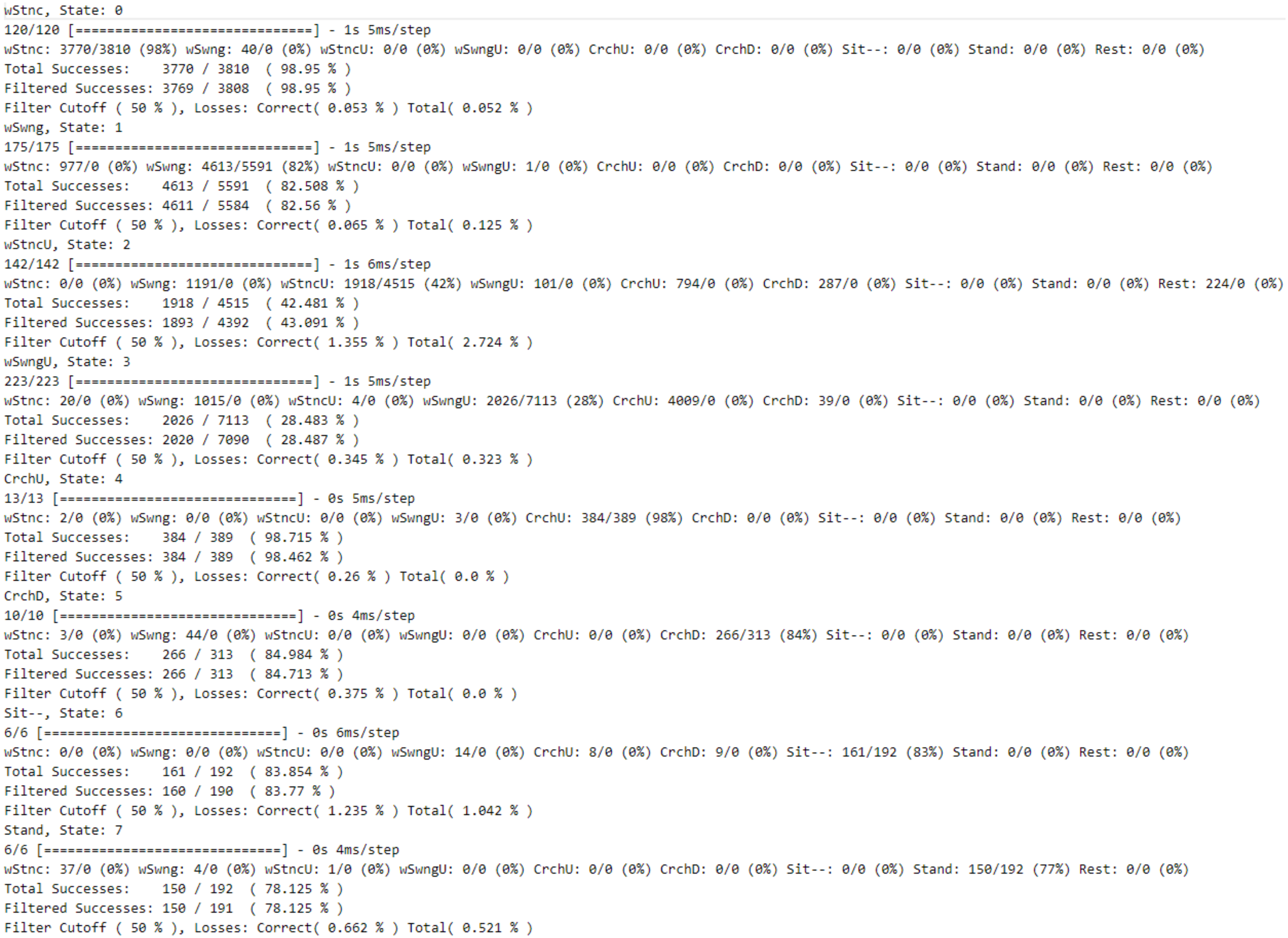
Figure 74 - Delay between changing state and its recognition, assuming perfect state recognition capacity.

Despite promising theoretical results with accuracies of >90% when using test data, there was not a translation into real-world accuracy, with the system often struggling to maintain the correct Movement State if this state had considerable changes in direction or motion. For example, whilst the system would accurately detect a wearer standing, sitting, or resting for extended periods of time, it would fail to accurately detect a walking or crouching pattern as such, often instead switching between several states that appeared more “likely” during that particular moment.

It was decided that the system’s lack of reliability made it infeasible for usage. When viewing theoretical predictions per state, accuracies varied with state, although seemingly remained high (Figure 75). However when the state prediction system was applied to data as it would be seen in reality (in other words, quickly varying between states), it failed to provide accurate predictions as seen in Figure 76.

One potential reason for this difference between theoretical and actual may be that data arrays were separated on a per-state basis before being provided as training data. In other words, the training data consisted of a “block” of Walk-Swing Data, followed by a block of Walk-Stance data, followed by a block of

Crouch Up Data, and so on. Although as Convolutional Neural Networks compute for each sample separately this was assumed not to be the case. It was decided this possibility was not worth testing and that instead a different network would be looked into that may be more appropriate for time series implementations, that being a Recurrent Neural Network.



```
wStnc, State: 0
120/120 [==============================] - 1s 5ms/step
wStnc: 3770/3810 (98%) wSwng: 40/0 (0%) wStncU: 0/0 (0%) wSwngU: 0/0 (0%) CrchU: 0/0 (0%) CrchD: 0/0 (0%) Sit--: 0/0 (0%) Stand: 0/0 (0%) Rest: 0/0 (0%)
Total Successes:    3770 / 3810  ( 98.95 % )
Filtered Successes: 3769 / 3808  ( 98.95 % )
Filter Cutoff ( 50 % ), Losses: Correct( 0.053 % ) Total( 0.052 % )
wSwng, State: 1
175/175 [==============================] - 1s 5ms/step
wStnc: 977/0 (0%) wSwng: 4613/5591 (82%) wStncU: 0/0 (0%) wSwngU: 1/0 (0%) CrchU: 0/0 (0%) CrchD: 0/0 (0%) Sit--: 0/0 (0%) Stand: 0/0 (0%) Rest: 0/0 (0%)
Total Successes:    4613 / 5591  ( 82.508 % )
Filtered Successes: 4611 / 5584  ( 82.56 % )
Filter Cutoff ( 50 % ), Losses: Correct( 0.065 % ) Total( 0.125 % )
wStncU, State: 2
142/142 [==============================] - 1s 6ms/step
wStnc: 0/0 (0%) wSwng: 1191/0 (0%) wStncU: 1918/4515 (42%) wSwngU: 101/0 (0%) CrchU: 794/0 (0%) CrchD: 287/0 (0%) Sit--: 0/0 (0%) Stand: 0/0 (0%) Rest: 224/0 (0%)
Total Successes:    1918 / 4515  ( 42.481 % )
Filtered Successes: 1893 / 4392  ( 43.091 % )
Filter Cutoff ( 50 % ), Losses: Correct( 1.355 % ) Total( 2.724 % )
wSwngU, State: 3
223/223 [==============================] - 1s 5ms/step
wStnc: 20/0 (0%) wSwng: 1015/0 (0%) wStncU: 4/0 (0%) wSwngU: 2026/7113 (28%) CrchU: 4009/0 (0%) CrchD: 39/0 (0%) Sit--: 0/0 (0%) Stand: 0/0 (0%) Rest: 0/0 (0%)
Total Successes:    2026 / 7113  ( 28.483 % )
Filtered Successes: 2020 / 7090  ( 28.487 % )
Filter Cutoff ( 50 % ), Losses: Correct( 0.345 % ) Total( 0.323 % )
CrchU, State: 4
13/13 [==============================] - 0s 5ms/step
wStnc: 2/0 (0%) wSwng: 0/0 (0%) wStncU: 0/0 (0%) wSwngU: 3/0 (0%) CrchU: 384/389 (98%) CrchD: 0/0 (0%) Sit--: 0/0 (0%) Stand: 0/0 (0%) Rest: 0/0 (0%)
Total Successes:    384 / 389  ( 98.715 % )
Filtered Successes: 384 / 389  ( 98.462 % )
Filter Cutoff ( 50 % ), Losses: Correct( 0.26 % ) Total( 0.0 % )
CrchD, State: 5
10/10 [==============================] - 0s 4ms/step
wStnc: 3/0 (0%) wSwng: 44/0 (0%) wStncU: 0/0 (0%) wSwngU: 0/0 (0%) CrchU: 0/0 (0%) CrchD: 266/313 (84%) Sit--: 0/0 (0%) Stand: 0/0 (0%) Rest: 0/0 (0%)
Total Successes:    266 / 313  ( 84.984 % )
Filtered Successes: 266 / 313  ( 84.713 % )
Filter Cutoff ( 50 % ), Losses: Correct( 0.375 % ) Total( 0.0 % )
Sit--, State: 6
6/6 [==============================] - 0s 6ms/step
wStnc: 0/0 (0%) wSwng: 0/0 (0%) wStncU: 0/0 (0%) wSwngU: 14/0 (0%) CrchU: 8/0 (0%) CrchD: 9/0 (0%) Sit--: 161/192 (83%) Stand: 0/0 (0%) Rest: 0/0 (0%)
Total Successes:    161 / 192  ( 83.854 % )
Filtered Successes: 160 / 190  ( 83.77 % )
Filter Cutoff ( 50 % ), Losses: Correct( 1.235 % ) Total( 1.042 % )
Stand, State: 7
6/6 [==============================] - 0s 4ms/step
wStnc: 37/0 (0%) wSwng: 4/0 (0%) wStncU: 1/0 (0%) wSwngU: 0/0 (0%) CrchU: 0/0 (0%) CrchD: 0/0 (0%) Sit--: 0/0 (0%) Stand: 150/192 (77%) Rest: 0/0 (0%)
Total Successes:    150 / 192  ( 78.125 % )
Filtered Successes: 150 / 191  ( 78.125 % )
Filter Cutoff ( 50 % ), Losses: Correct( 0.662 % ) Total( 0.521 % )
```

Figure 75 - Output of State-Specific training, note varying accuracies dependant on states.

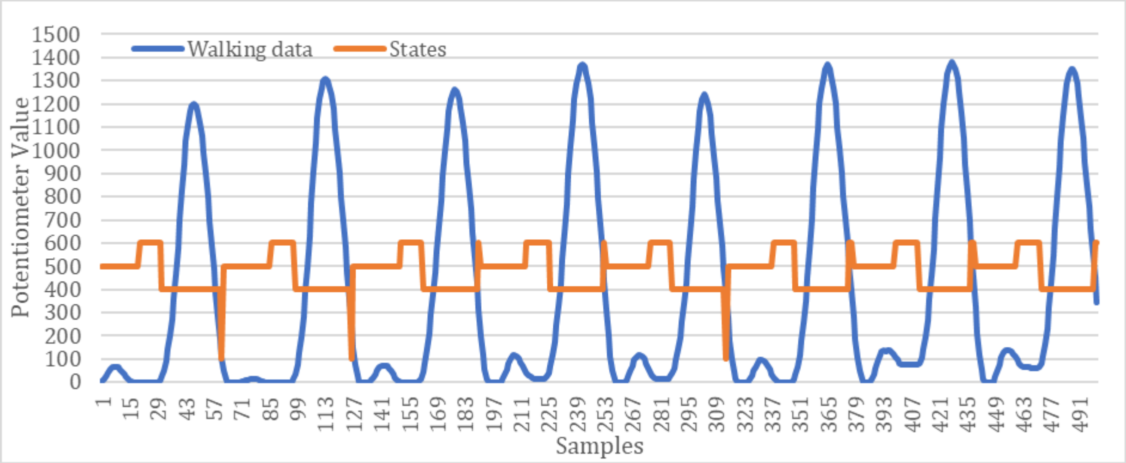


Figure 76 - Walking data with overlayed state predictions. Variation primarily between state 4, 5, and 6 (400/500/600), representing Crouch Up, Down, and Sit

5.2.3 Implementation Review

For the Recognition System, there was a notable disconnect between how effectively the system could potentially predict movement states when tested using test data and how well it actually predicted movement states when tested in reality. With a tendency to only be able to predict Movement states that remained relatively static, such as Standing and Sitting. The exact reasoning behind this was unknown, however it was decided that instead of investigating the reasoning, a different implementation would be chosen instead that would draw inspiration from several other AI-driven control system examples seen within the literature, with many prediction systems predicting a time variant value such as Knee Angle or kinematic motions, which could be considered more “Pattern-Based” classification [171].

Furthermore, the source of sample data was improved, by collecting additional movement data from a diverse set of 12 Healthy Candidates, of which 8 were Male and 4 Female between 22-58 Years old (Majority of candidates between 20 and 30 years old). Each candidate would wear the exoskeleton whilst their movement data was recorded performing several simple activities that collectively covered common “Movement States” performed by the average person, these being *Walking, Crouching, Walking Up Stairs, Sitting, Standing, Wandering, and Resting*:

- Walking upon flat ground for ~3 Minutes. (Walking)
- Sitting down and Standing up from a chair for ~2 Minutes. (Crouching)
- Performing Random Walking and Sitting/Standing actions of their own volition to simulate every day movement for ~2 Minutes. (Wandering)
- Static Sitting and Standing for ~2 Minutes (Sitting and Standing)
- (If Available) Use a StairMaster to simulate walking up Stairs for ~2 Minutes. (Walking up Stairs)

Resting, being a test case, did not require data acquisition from multiple candidates. This diversity of data was required as ultimately a recognition system was incapable of “implying” information, if soley trained on walking data for example it may predict the user to be walking even in situations where they clearly were not, as the only state the recognition system “knows” would be walking.

Such an example is seen in Figure 77, where the recognition system that will be discussed in the next section was trained exclusively on walking data, and it failing to predict the knee angle (potentiometer position) of a wearer standing completely still. Similarly, if a recognition system was trained on data from a single individual, as it had been previously, the recognition system would only effectively predict the movement patterns of that individual, which may differ from others.

While during initial development, test data was entirely collected by the author, and existed for the purpose of making sure that the developed model functioned at a basic level, it was clear that it would be beneficial to collect data from additional candidates in order to broaden the diversity of movement patterns that the model would be exposed to. Otherwise, the model would have become proficient in predicting the specific movement patterns of the author, rather than the movement patterns of a generic wearer.

There were a total of 12 Participants, including the Author, whom each performed the series of movement actions that were defined above. In addition to this a large batch of additional data was provided separately by the author, having been collected prior and consisting of extended sections of walking data collected by walking from the Author's place of residence to the University of Kent. This resulted in ~215,000 samples or ~3 hours of samples. Which was used for some initial testing of the new Recognition System.

This would form the basis of the training and testing data that would be used for the reworked control system described in the following sections.

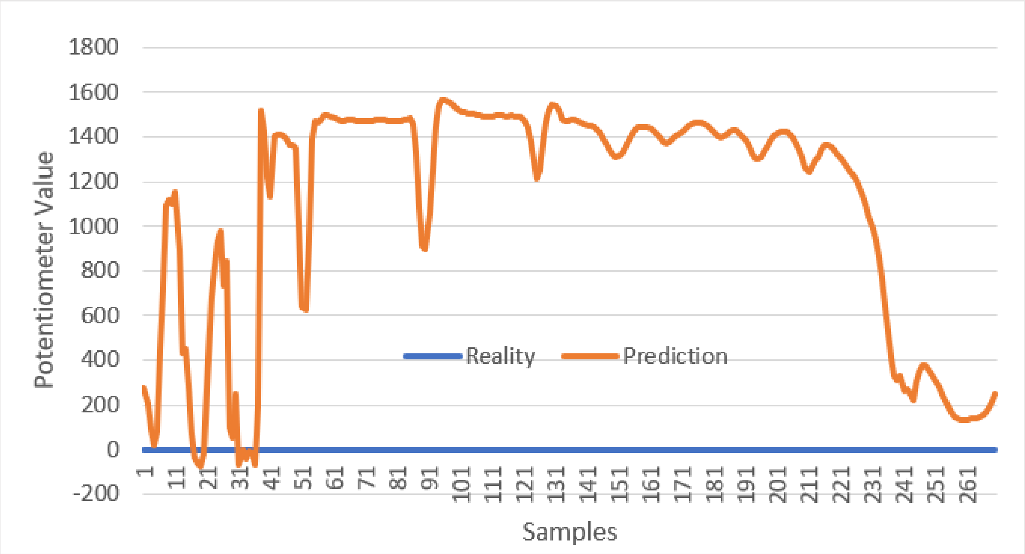


Figure 77 - Prediction System trained only on walking data knee angles struggles to predict the wearer's knee angle standing still, as it has not been trained to "know" what standing still is. 20Hz Sampling Rate.

This Data Collection Experiment was conducted following the University of Kent Central Research Ethics Advisory Group's Experimental guidelines. As to determine whether the number of participants was sufficient, an additional set of tests were performed, and are described in 5.3.6. The dataset created would be similar in nature to other exoskeleton datasets, consisting of multiple candidates performing a set of actions, with information stored separately for a set of sensors. For example, [228] collected EMG, IMU, and Goniometer data from 22 candidates performing level, ramp, stair, static, and treadmill walking.

5.3 Improvements and Redesigns

Due to the failure of the Convolutional Neural Network as a result of an inability to recognise consistent patterns of samples over several rapidly changing cycles in addition to a slow reaction time for predictions, a different method and recognition system was considered that aimed to resolve this. Recurrent Neural Networks are designed to process sequential information, consisting of recurrent cells that combine received information with prior information, identifying potential patterns. This was seen as appropriate for sequential or Time-Series Information, especially if the output can be related to the input. Similar to the previous method, the input would be in the form of an 8-width snapshot of data collected from the IMU and Potentiometer (Accelerometer X, Y, and Z, Gyroscope X, Y, and Z, Potentiometer Knee Angle, and additionally Rate of Change of Knee Angle).

LSTM's (Long Short-Term Memory) are a type of Recurrent Neural Network possessing effective Long-Term Memory capabilities, meaning that it is less prone to "forgetting" older data than some other Recurrent Neural Networks whose rates of learning gradients can tend to zero [229] over time. As the system would be provided a long series of continuous information of several distinct movement states, such as walking, crouching, and so on, it was imperative the system did not forget these states or merge them together. LSTM's have also been proven to be capable of effectively running on the Cortex-M Microprocessors used by many inexpensive microcontrollers [230]. Being directly supported by TensorFlow, a well-established Machine Learning API, with TensorFlow Lite being used in this implementation. Therefore, this is why this method was chosen, as the capacity for memory and compatibility with Cortex-M Architecture were essential features.

Some common examples within the literature of sampling rates include [231] and [232] using rates of 100-120Hz, with [233] claiming a sampling rate of 120Hz was a "minimum" when calculating spatiotemporal data for variability and sample entropy, as only more general recognition was required it was reasoned a slower sample rate could be used

When viewing effectiveness of microcontroller-based neural network predictions in terms of Average Percentage Difference from reality, [174] achieved an accuracy of ~3.75% with a CNN predicting 20-200ms into the future. Meanwhile [181] achieved an accuracy of ~4% with an RNN predicting 50ms into the future. The LSTM developed within this thesis achieves similar accuracies with predictions made up to 1000ms into the future.

The LSTM Neural Network created for this implementation did not use a State-Based prediction system as the previous Convolutional Neural Network had. Instead, it would rely on a simpler system that would predict a future knee angle based on known current knee angles. Based on this future prediction the system could then command the actuator to move pre-emptively to provide support as the wearer was moving.

Prediction System Development

This Time Series Prediction method being a common type of implementation for Recurrent Neural Networks. The system would use a combination of knee angle data provided by the potentiometer and the Inertial Measurement Unit located at the ankle, similar to [234], [235], and [236].

For the Initial Implementation, the Prediction system would sample data at a rate of 100Hz, twice that of the prior implementation. However, to keep the input window small as to reduce reaction times, it would only provide the average of every 5 samples, resulting in a 20Hz prediction and processing rate. These samples would be provided as a "Snapshot" of data consisting of the last 40 samples (2 seconds) prior to the present point in time across each sensor (Accelerometer X, Y, and Z, Gyroscope X, Y, and Z, Potentiometer, and Potentiometer rate of change). This would be stored as a [40,8] Array of data, Accelerometer, Gyroscope, and Potentiometer data would each be normalised separately to between -1 and 1 based off of known likely maximum and minimum values collected during training, which would be equal to the largest and smallest value of the training set for each sensor that was within three standard deviations of the average as to prune out anomalous spikes that would otherwise bias the dataset. This would then be compared against a [30,1] Prediction array during the training purpose, where the prediction would be the next 30 values after the end of the set of 40 that made up the training snapshot (Figure 78), or 1.5 seconds into the future.

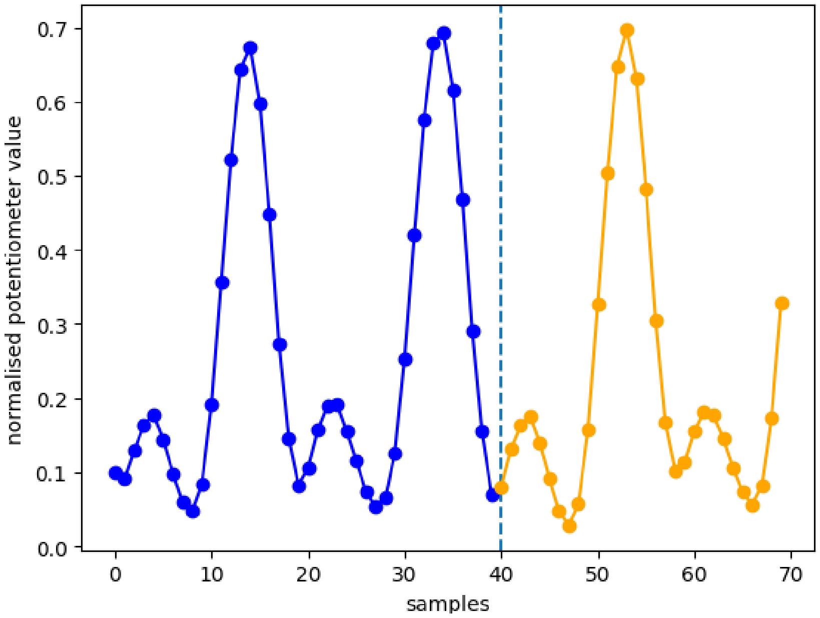


Figure 78 - Display of Input (Blue) and Prediction (Orange) Data. 20Hz Sample Rate. Normalised potentiometer values represent Knee Angle, where 0 = 180-degree Knee Angle

Much of the backend code for taking in stored data inputs was re-written, with the system pre-calculating the appropriate size of the Three-Dimensional Data Array to hold multiple [40,8] snapshots. For example, a 1000-length .csv sample file with a sliding window moving at 1 sample per snapshot for a [40,8] input and [30,1] output would provide 930 snaphots, (the length of the file minus the size of the input and output arrays). This considerably improved data processing times allowing hundreds of thousands of samples to be processed in a few seconds.

Other than this the general Sliding Window Principle used in the second iteration of the Convolutional Neural Network, where a "Snapshot" of continuous data was taken to represent one sample, before moving the window along by a set value before taking another snapshot remained unchanged (see Figure 79). This Snapshot process was also used to produce the [30,1] Predictions.

Using the Sliding Window Technique (Figure 79) on collected data, there was a total of 215,000 samples or ~3 hours of sample data. This would be randomly shuffled before being split 90/10 to Training and Testing data. Training data would be used to train the model, which consisted of a [40,8] Input, and finished with a [30,1] output Dense Layer, this would then be trained for 15 Epochs.

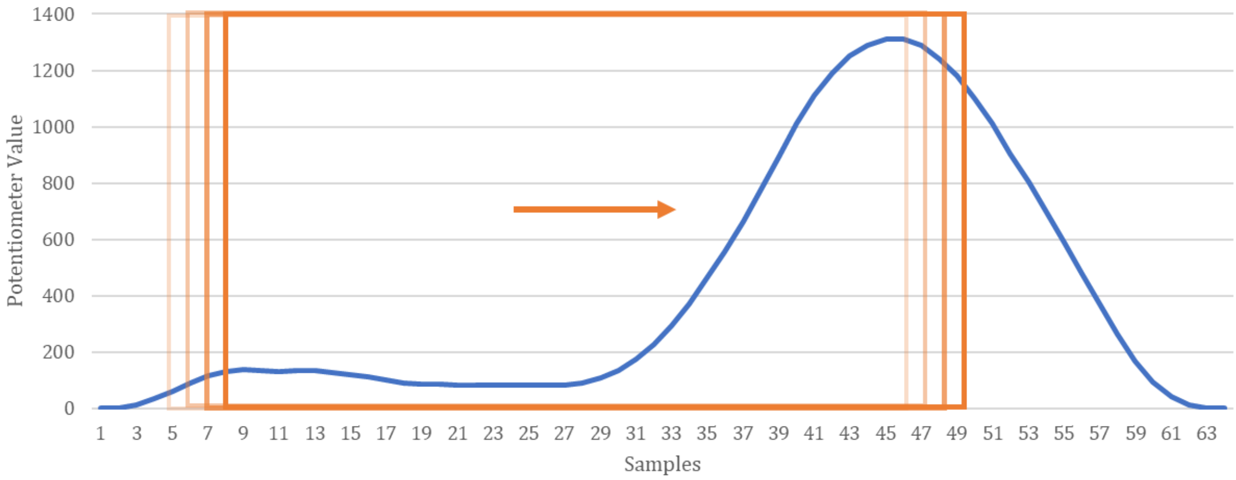


Figure 79 - Shifting Snapshots of data of a gait cycle to create a "Sliding Window".

Walking Up Stairs Data (Figure 65) was collected using a StairMaster Exercise machine to provide more consistency and allow for longer periods of consistent data collection to occur. There was no easily accessible safe alternative for Walking down Stairs for long periods of time at once, and data for it ended up not being collected.

A key benefit of both collecting potentiometer data and it being the value predicted by the Prediction system was that for testing purposes a Knee Angle Prediction could be directly compared with the actual knee angle at that point in reality to see how accurate the predictions were, a process that could be made to occur in real time to allow for prediction improvement via linear regression. The theoretical implementation achieved a Mean Squared Error of ~0.013. Linear Regression (Figure 80) showed a strong association between predicted values and real values, which in turn displayed the prediction system as capable of recognising patterns within its input samples and accurately outputting predictions similar to reality. Predictions of future potentiometer values created from the test data that the model was not trained on were recognisably similar to the actual potentiometer values, although predictions tended to exaggerate reality, later found to be due to having too much sample data from a single source, drowning out variation from other sources. After Linear Regression, values tended to look more similar to reality (Figure 81). Therefore, Linear Regression could act to reduce systematic errors in predictions.

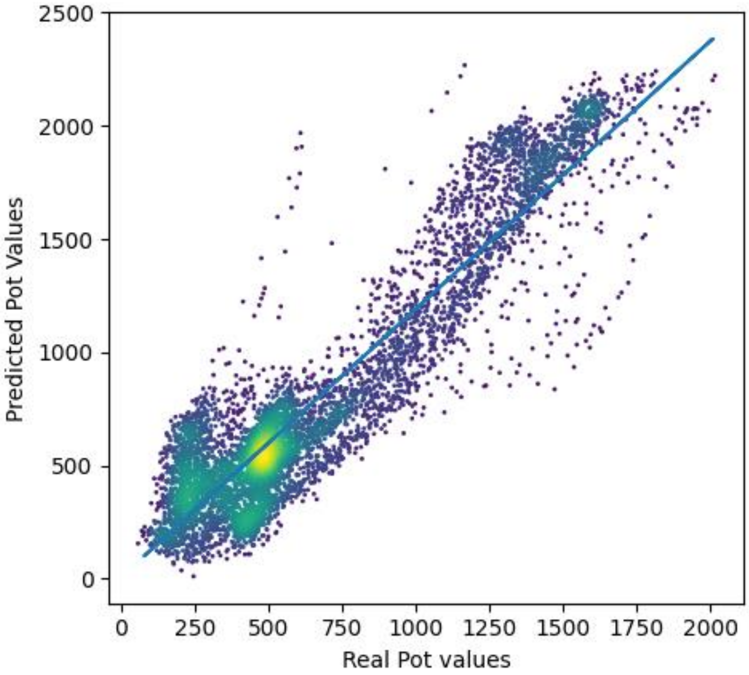


Figure 80 - Linear Regression comparing real potentiometer values to predictions. "Real" referring to sampled wearer data during walking. "Predicted" to data predicted by LSTM Neural Network.

Post regression, values tended to look more similar to reality (Figure 81), interestingly, Linear Regression was most effective when only trained on a small number of samples prior to the predicted samples that would have regression applied to them, as opposed to training a Linear Regression model for the entire test dataset. This may be due to the predictions tending to vary in how much they over and underestimate reality over the length of the dataset, and so a smaller, more immediate set was better able to counteract the errors of the present prediction. Although, this would mean that a trained Linear Regression model would only be effective for roughly a second before needing to be retrained on new data, which for early Microcontroller Implementations was infeasible.

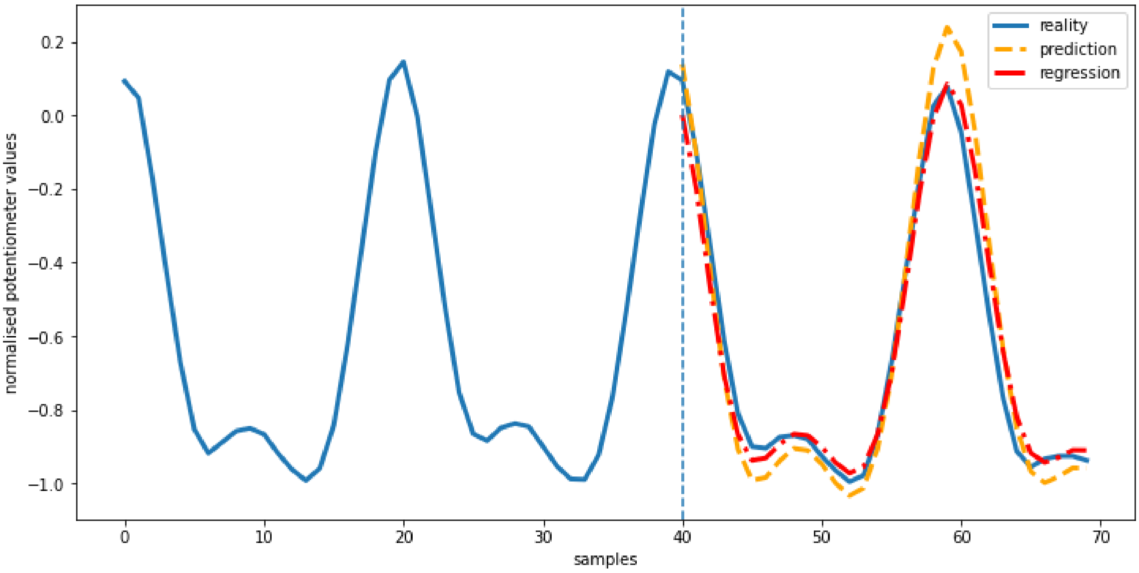


Figure 81 - Prediction (Orange) and Post-Regression Prediction (Red) vs Reality (Blue). Dividing line separates input data and output data. 20Hz Sampling Rate.

As many Examples of Regression Model Implementations made use of a Laptop to host the model, which would then be serially or wirelessly connected to a controller, both a Laptop-based and Microcontroller based implementation would be created using this initial design. The Laptop-Based version having access to higher processing power would make use of a more powerful model, and therefore test the difference in effectiveness between it and the lower power microcontroller.

5.3.1 Laptop-Based Model Implementation

To act as a comparison to the Microcontroller a prediction model was tested on a high-power laptop that received sensor data from a connected exoskeleton and returned prediction data via serial connection. This allowed for the usage of a much larger LSTM network and would act as a baseline for comparing a model designed to make full use of a computer's predictive capability versus a microcontroller, having 4 Size 128 LSTM Layers similar to [231], as well as a Dropout and output Dense Layer. The Laptop would be placed in a backpack, with a USB cable providing the serial connection between the ESP32 located on the exoskeleton and the Laptop. The ESP32 would use basic Arduino-based *Serial.print()/Serial.read()* Functions to send sensor data to and receive predictions from the Laptop, where a Python program would run to collect data, storing it in a Pandas Dataframe and return a knee angle prediction to the ESP32.

If the Laptop was set to only collect data rather than predict it, it would instead send a dummy reply back to the ESP32 to confirm that data had been received and that more data was to be sent.

Converting a model to be read by the laptop was relatively simple, using the inbuilt Tensorflow *model.save()* functions. The System would be saved as a *.pb* SavedModel file with an accompanying concrete function that stored the model's expected input dimensions and type. This could then be loaded by the similar *model.load()* function to run the model again without having to recreate or retrain it.

If the Laptop Implementation reading and recognition program was set to predict data it would prior to the beginning of sample-reading load such a SavedModel file, upon then receiving at least the minimum number of samples to form an [40,8] Dataset (or 2 seconds of samples) from the connected ESP32 it would begin predictions. This program was the final iteration of the Python serial reading program, whose basic functions are detailed in Figure 84, and iterations of it were used for both the Laptop and Microcontroller implementations. Although for the Microcontroller implementations, prediction and regression were disabled, and the program would instead receive both sensor data and predictions from the microcontroller.

For the Laptop Implementation the Program would make use of the *.predict_on_batch()* function to output a single prediction based on a normalised array containing the last 40 received data samples, the data was normalised using a set of scalars which made it such that each sensor was normalised identically to how training data was originally for the model by cutting off values outside of 3 standard deviations of the average and using a MinMax Scaler on the rest. Both predictions and sensor data were saved within the same Pandas Dataframe to be later saved to a .csv file once a set number of cycles had been completed or a manual stop command was issued.

This manual stop was issued via a button located on the exoskeleton that would send a specific serial character that triggered the software to save all collected data and stop, this button would also be used to start the act of collecting data, allowing the process to be performed by the wearer of the exoskeleton without interacting with the laptop, this allowed the laptop to be safely stored in a backpack with the program running on it without further interaction needed.

If enough predictions had been made then the program would be able to use the previous predictions made before the most recent prediction and compare them to the actual potentiometer values that had occurred at these times (EG: A Prediction made 5 cycles into the future would be compared to the actual potentiometer value that occurred 5 cycles after the prediction was made). These comparisons would form a Linear Regression model that would be used on the latest set of predictions before then being recalculated for the next, as such always relying on the most recent data. Due to the high processing power of the laptop used, despite performing 30 regressions a cycle, one cycle would take ~20-30ms to perform, which was well within the maximum time of 50ms per cycle that the system would run at.

Linear Regression would be performed on all values by comparing previously made predictions to the actual potentiometer values that were recorded at the times the predictions predicted, then applying this calculated regression value to the present set of predicted data to account for any notable errors seen within the recent past predictions that may still affect the present predictions. Testing the effectiveness of the Laptop-based implementation was done in the same way as data was originally collected, by running the system in real time whilst an exoskeleton wearer was walking and logging predictions.

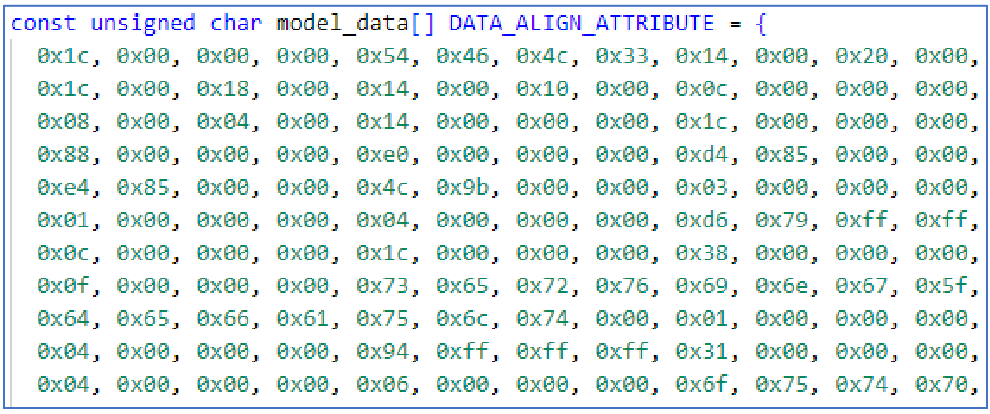
On the Microcontroller side, the system was effectively identical to Figure 58, minus the *doModel()* function, which was offloaded to the Laptop. For testing of prediction accuracy purposes, motor functionality was also disabled, with no Spring Rod present to allow the exoskeleton to move freely. This was to allow for testing a potentially imperfect prediction system without interruption from the exoskeleton.

5.3.2 Microcontroller-Based LSTM Implementation

The Process of implementing LSTM onto an ESP32 (Adafruit Huzzah32 ESP32 Feather Board) used the TensorFlow Lite for Microcontrollers library [237], specifically the *tflite-micro-esp-examples* [238] due to ESP32 specific optimisations and ease of use. Prior to September 2022, Rolled LSTM's were not fully supported, causing a "UNIDIRECTIONAL_SEQUENCE_LSTM" Error when attempted. Unrolled LSTM's were considerably larger than rolled in terms of file size and were effectively non-functional due to these size limitations severely limiting processing capacity. After the limitations of rolled LSTM's were resolved, the rough size of the tflite model was 245kb in size, as opposed to the 3.2 MB savedModel file used by the Laptop. It consisted of two LSTM layers of size 64 and 32, plus one Dropout layer to reduce overfitting and a Dense layer to format the output to a [30x1] size 1D array. This comfortably fit within the loadable file size of the ESP32, and the model processing time limit of 50ms with room to spare for other functions.

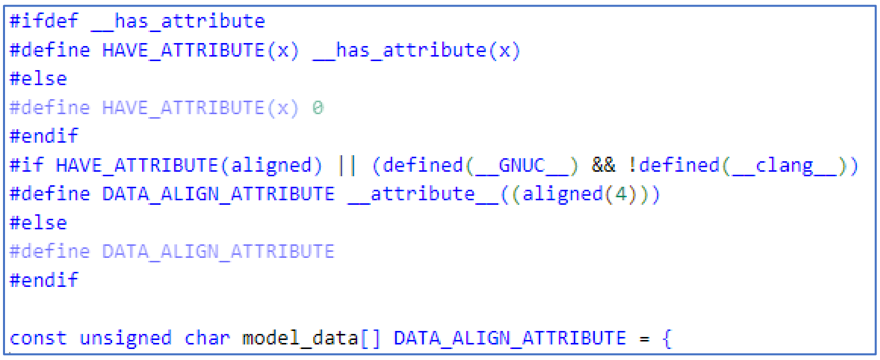
Converting a PC-based TensorFlow model to function on a Microcontroller was not the same for Recurrent Neural Networks as it was for Convolutional Neural Networks, with there being little in the way of pre-existing support for such conversions. Instead, it first required converting the model to the TensorFlow Lite Format and saving the model in the *.tflite* format. Replacing TensorFlow Operations with their equivalent TensorFlow Lite Operations compatible with Microcontroller firmware.

After being converted to a .tflite file, the file requires further conversion to a *.h* header file containing an array of hex codes that represent the model, a section of which is seen in Figure 82, to allow it to then be loaded by the tflite-micro code. This is done using Vim, and the xxd command *"xxd -i model.tflite > model.h"* within the command line. After this conversion, the following lines seen in Figure 83 must then be added to the *.h* file in order to allow for compiling on devices that do not have a file system, such as Microcontrollers. The Final line "*const unsigned char model_data[] DATA_ALIGN_ATTRIBUTE = {*" replacing the pre-existing array definition.



```
const unsigned char model_data[] DATA_ALIGN_ATTRIBUTE = {
  0x1c, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x14, 0x00, 0x20, 0x00,
  0x1c, 0x00, 0x18, 0x00, 0x14, 0x00, 0x10, 0x00, 0x0c, 0x00, 0x00, 0x00,
  0x08, 0x00, 0x04, 0x00, 0x14, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00,
  0x88, 0x00, 0x00, 0x00, 0xe0, 0x00, 0x00, 0x00, 0xd4, 0x85, 0x00, 0x00,
  0xe4, 0x85, 0x00, 0x00, 0x4c, 0x9b, 0x00, 0x00, 0x03, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0xd6, 0x79, 0xff, 0xff,
  0x0c, 0x00, 0x00, 0x00, 0x1c, 0x00, 0x00, 0x00, 0x38, 0x00, 0x00, 0x00,
  0x0f, 0x00, 0x00, 0x00, 0x73, 0x65, 0x72, 0x76, 0x69, 0x6e, 0x67, 0x5f,
  0x64, 0x65, 0x66, 0x61, 0x75, 0x6c, 0x74, 0x00, 0x01, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x94, 0xff, 0xff, 0xff, 0x31, 0x00, 0x00, 0x00,
  0x04, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x6f, 0x75, 0x74, 0x70,
```

Figure 82 - Example of Header File Hex Code Section



```
#ifdef __has_attribute
#define HAVE_ATTRIBUTE(x) __has_attribute(x)
#else
#define HAVE_ATTRIBUTE(x) 0
#endif
#if HAVE_ATTRIBUTE(aligned) || (defined(__GNUC__) && !defined(__clang__))
#define DATA_ALIGN_ATTRIBUTE __attribute__((aligned(4)))
#else
#define DATA_ALIGN_ATTRIBUTE
#endif

const unsigned char model_data[] DATA_ALIGN_ATTRIBUTE = {
```

Figure 83 - Required Additions to Model Head Files for functionality on Microcontrollers.

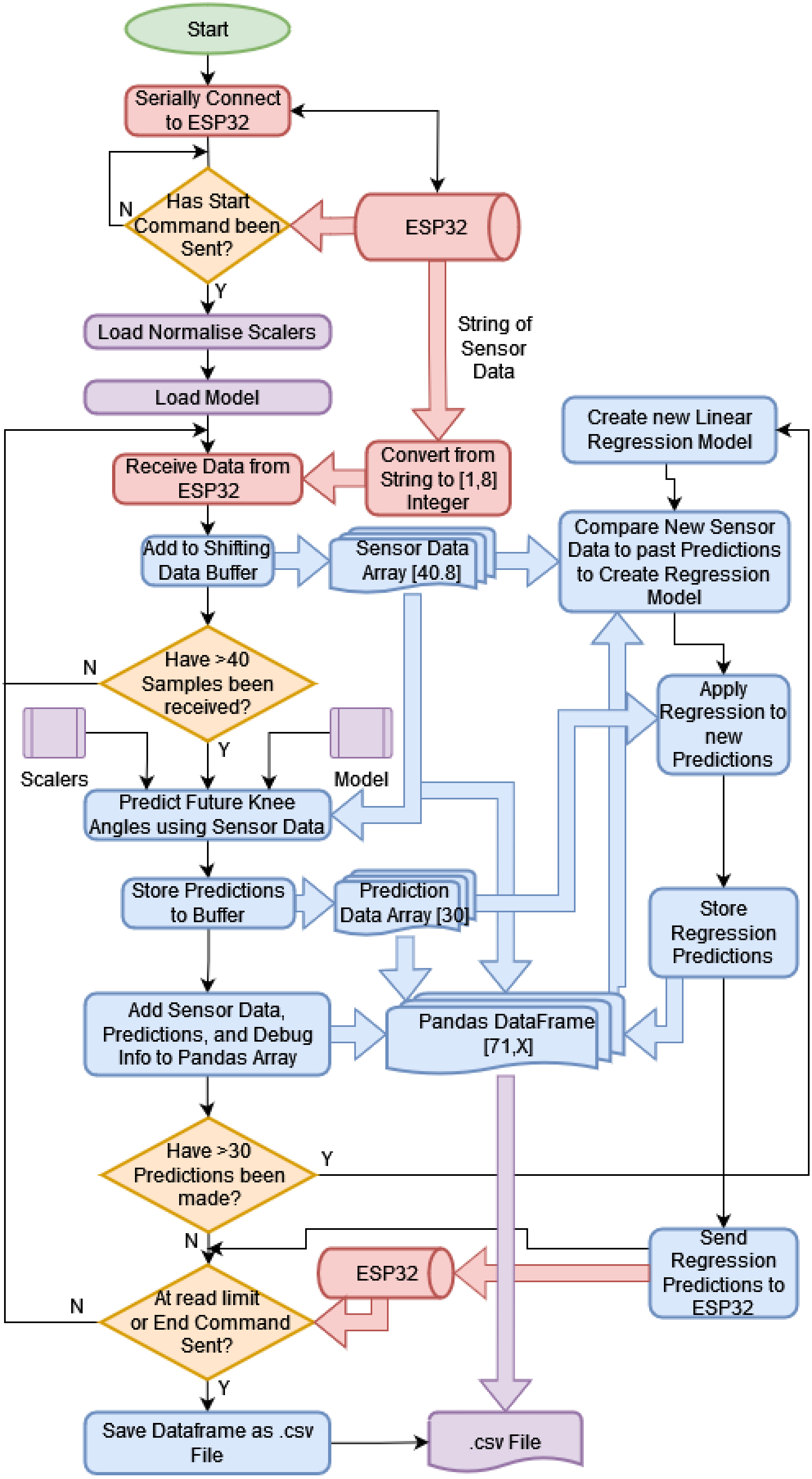
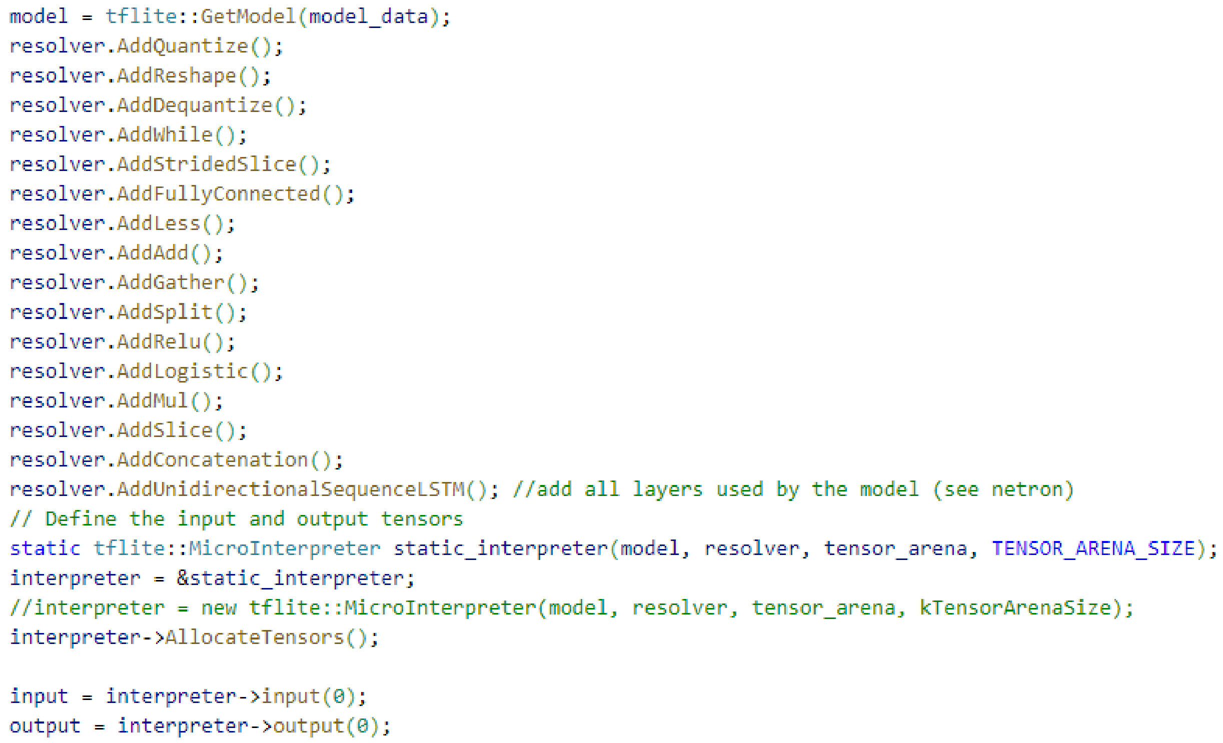

Start
Serially Connect to ESP32
Has Start Command been Sent?
N
Y
ESP32
String of Sensor Data
Load Normalise Scalers
Load Model
Receive Data from ESP32
Convert from String to [1,8] Integer
Create new Linear Regression Model
Add to Shifting Data Buffer
Sensor Data Array [40.8]
Compare New Sensor Data to past Predictions to Create Regression Model
Have >40 Samples been received?
N
Y
Apply Regression to new Predictions
Scalers
Model
Predict Future Knee Angles using Sensor Data
Store Predictions to Buffer
Prediction Data Array [30]
Store Regression Predictions
Add Sensor Data, Predictions, and Debug Info to Pandas Array
Pandas DataFrame [71,X]
Have >30 Predictions been made?
Y
N
ESP32
Send Regression Predictions to ESP32
N
At read limit or End Command Sent?
Y
Save Dataframe as .csv File
.csv File

Figure 84 - Sensor Data reading, prediction, and regression program "ReadDataPredictRegression.py" Flowchart.

This code is seen within Official TensorFlow usage within [239] as part of the TensorFlow Python GitHub within the "*convert_bytes_to_c_source()"* function. Although within the implementation for this project, the lines had to be added manually. This Model Header File could then be made available to the Microcontroller program to be used, with the initialising *getModel* function loading all layers used by the model as well as defining its inputs and outputs as displayed within Figure 85. The Model Layers were determined using the Third Party Program "Netron" [240].



```
model = tflite::GetModel(model_data);
resolver.AddQuantize();
resolver.AddReshape();
resolver.AddDequantize();
resolver.AddWhile();
resolver.AddStridedSlice();
resolver.AddFullyConnected();
resolver.AddLess();
resolver.AddAdd();
resolver.AddGather();
resolver.AddSplit();
resolver.AddRelu();
resolver.AddLogistic();
resolver.AddMul();
resolver.AddSlice();
resolver.AddConcatenation();
resolver.AddUnidirectionalSequenceLSTM(); //add all layers used by the model (see netron)
// Define the input and output tensors
static tflite::MicroInterpreter static_interpreter(model, resolver, tensor_arena, TENSOR_ARENA_SIZE);
interpreter = &static_interpreter;
//interpreter = new tflite::MicroInterpreter(model, resolver, tensor_arena, kTensorArenaSize);
interpreter->AllocateTensors();

input = interpreter->input(0);
output = interpreter->output(0);
```

Figure 85 - Microcontroller Model Initialisation

The Author made use of the *"tfmicro-esp-examples"* Library, [241] an offshoot of the Official TensorFlow-Lite library optimised for the ESP32. The VSCode IDE, alongside PlatformIO and ESP-IDF Extensions was used. Additionally, at the time, it was required to edit the flatbuffers third party library included within such that *flatbuffers/include/flatbuffers/base.h* lines 45, 46, 47, and 49 were commented out, and the line 48 *#include <utility>* was therefore always active.

Once the Microcontroller model was created, it could be used similarly to the Python Model. Predictions would be made using the Tensorflow Lite Micro Libraries' *interpreter->Invoke();* function, that would made predictions based off an [40,8] input array of sensor values, to make 20 output potentiometer predictions (1 second into the future). This was reduced from the 30 of the laptop's implementation to reduce the amount of data sent to the laptop, it had no effect on model size or speed, although sending too much data via serial print at a time slowed down the microcontroller, occasionally resulting in cycles not sending data. The program would then consistently send each prediction along with the continually collected sensor data via serial connection to be stored in a *.csv* file similarly to the Laptop Implementation.

The Parameters of the Model were determined by trial and error to discover a model that was effective within the limitations of the microcontroller's maximum allowable size and that possessed reasonable processing times. For example, more than two LSTM layers of similar size resulted in the model unable to function. As will be expanded on in the next section, the limited processing power of the microcontroller made live Linear Regression infeasible, and so for this microcontroller implementation, it was not implemented.

5.3.3 Effectiveness of Laptop and Microcontroller Implementations

As was expected the Laptop Implementation was of a similar accuracy and produced similar results to that of the freshly trained model run on a PC and had an average Processing time per sample of less than 50ms. When comparing reality to predictions and regression predictions, Mean Average Error (MAE) between reality and prediction varied between 3.63% to 8.1% for predictions made 50ms to 1000ms into the future, or 3.68% to 9.71% for regressed predictions. Both examples were averaged over 5,000 samples, a short depiction of 120 samples is seen in Figure 86.

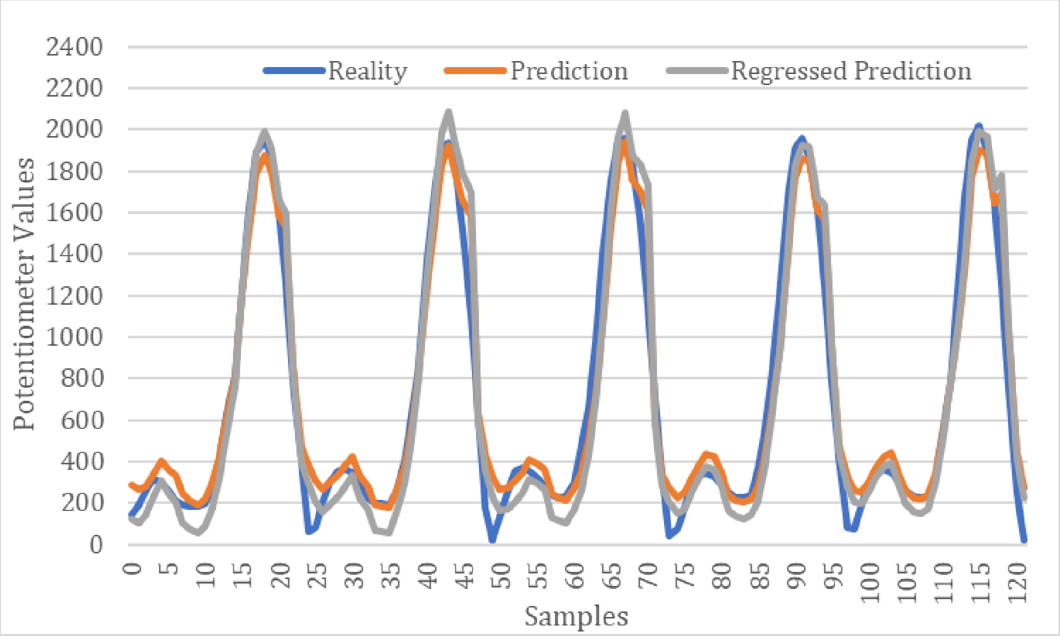


Figure 86 - Laptop Prediction vs Reality for data. Prediction made 0.25 seconds into the future. 20Hz Sampling Rate.

While this laptop implementation was reasonably accurate in predicting future values, it was limited in requiring a high-powered laptop in order to function at speed, the device used, a ROG Zephyrus M16 GU603ZX costing ~£4,000 [242] was more powerful than required however any device used would need to be capable of running TensorFlow, in addition to requiring a wired connection and backpack to carry the laptop and its power supply. In comparison, the ESP32 used in the Microcontroller implementation costed only £15.50, over 250x less expensive in addition to a smaller size and power usage. Of course, this difference in cost between the PC/Laptop Implementations and Microcontroller was countermanded by a difference in processing power. The Laptop Implementation, even whilst limited by running on CPU rather than GPU for model processing, possessed a 5GHz, 14 Core Intel I9 12900H, considerably faster than the 240MHz Dual core of the ESP32.

As a result of the smaller model deployed to the microcontroller, predictions could not be made within the 50ms sample window. Instead, predictions would take an average of ~180ms to process, allowing for a maximum lossless prediction rate of 5.56Hz, considerably lower than the desired 20Hz. If making predictions in real time as would be intended for actual usage, the Mean Average Error would be ~8.93% to 11.87% as a result of fewer predictions being able to be made and so some prediction attempts would be lost as the previous prediction had not finished processing. This resulted in a graphed output with a notably stepped or blocky appearance (Figure 87, grey "Real-Time" output).

Due to the predictions already being slow Linear Regression was not attempted, limiting corrections to predictions. It was noted that those predictions that did occur would be relatively accurate, as such the system was re-run in a way where it waited for each prediction to complete test data to allow it to run at its own pace. When performing non real-time processing, i.e. the prediction system was run on a set of pre-existing samples stored in an array that were fed in one at a time with no sample skipping, the predictions show that the microprocessor implementation was capable of reasonable accuracy, in this case the Mean Average Error between Prediction and Reality varied from ~6.1% to ~10.9%. This output, seen in orange in Figure 87 showed impressive accuracy, although still inferior to the laptop. It none the less implied that by either reducing the size of the model, or increasing the speed of the microcontroller, reasonable levels of accuracy could be achieved with a Recurrent Neural Network on a microcontroller.

As was expected when comparing the Laptop Implementation to the Microcontroller Implementation, the Laptop implementation benefits considerably from access to higher processing power by being able to run a much more powerful model much faster. With the HUZZAH32 ESP32 Feather possessing a 240MHz dual core microcontroller, compared to the 5GHz and 14 Cores of the Laptop CPU. This Necessitated the Microcontroller implementation to be much smaller to be able to produce predictions within a reasonable time frame. Therefore, when running both prediction systems on a 5,000-sample length dataset they had not been trained on, the average percentage differences between Reality and Prediction for both prediction systems were observed to be somewhat as expected. (Figure 88). With both seeing a decrease in accuracy (an increase in Mean Average Error) with a further prediction into the future.

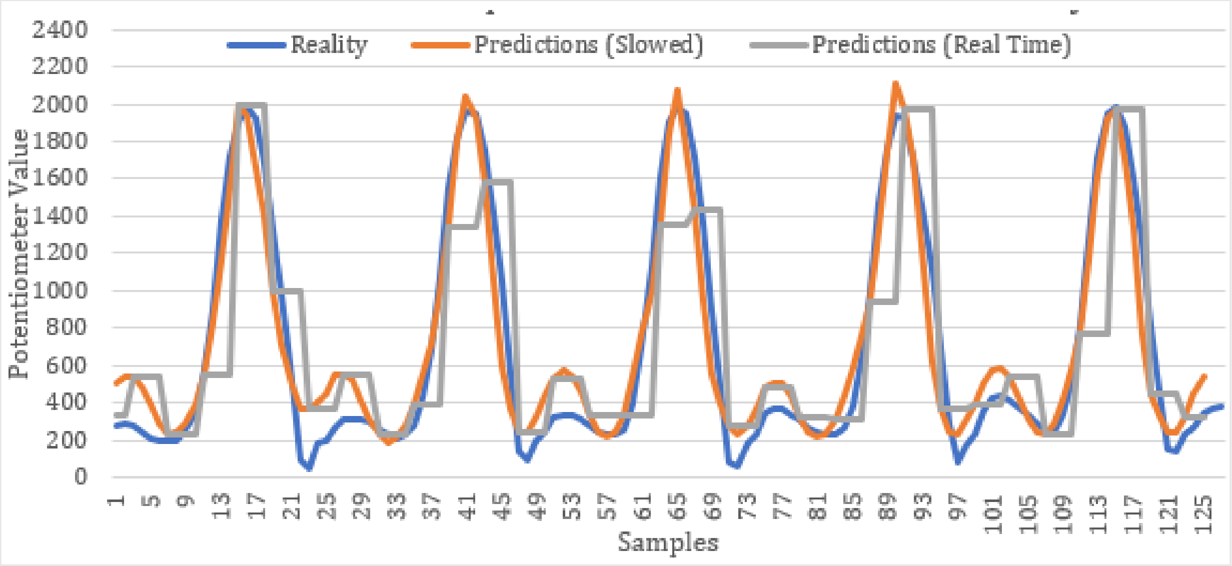


Figure 87 – ESP32 Prediction vs Reality data for Slowed and Real Time Microcontroller Implementations. Prediction made 0.25 seconds into the future. 20Hz Sampling Rate.

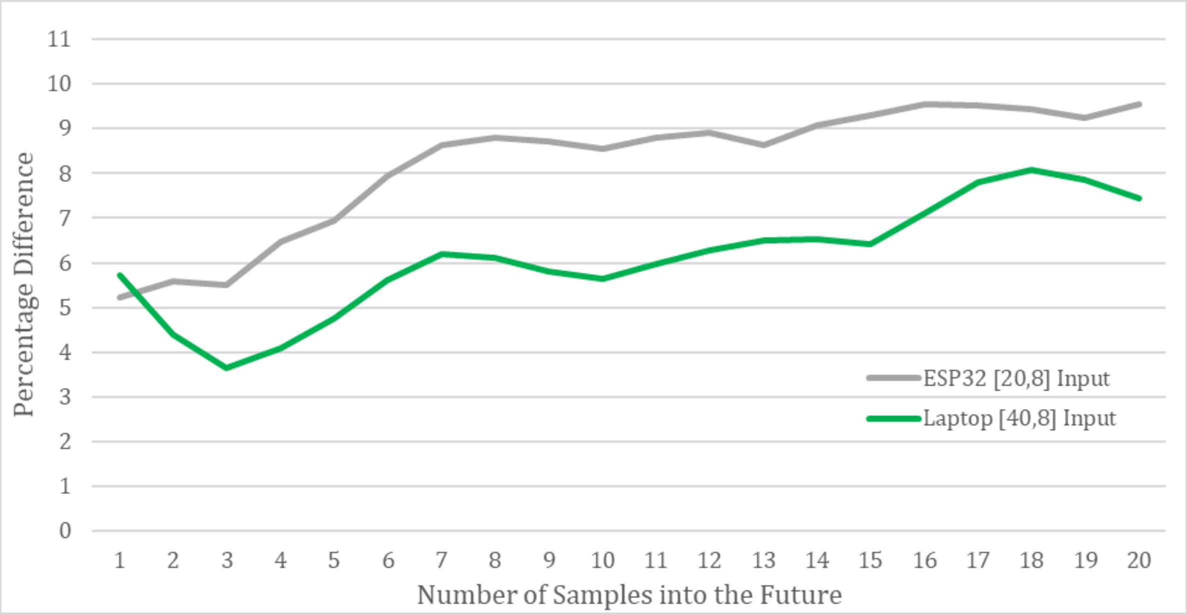


Figure 88 - Comparing ESP32 and Laptop Differences between Predictions and Reality at varying samples into the future vs the actual values that occurred at those predictions. 1 Sample = 50ms

For both the Microcontroller and Laptop Implementations, predictions followed a straighter line, with a clear linear increase in percentage difference for further predicted samples. As expected, the Laptop Implementation produced results of a higher accuracy overall accuracy and was capable of doing so at a more rapid rate, however considering the hardware required to produce these results was 250 times more expensive, the microcontroller being capable of producing technically inferior yet "almost as good" predictions was promising. These implementations showed the feasibility of the microcontroller implementation, although made apparent the issue of the lack of processing power reducing the rate at which predictions could be made and therefore decreasing the overall accuracy of these predictions. A Potential solution to this was to look for a more powerful microprocessor board that was still within a reasonable price range. Additionally, optimisations to the model could be made to reduce its size, and therefore the time taken for it to process information.

5.3.4 Improving Speed of the Prediction System

It was decided that the Prediction System would benefit from improvements to the model as well as faster processing speeds, achievable by either using a different, more powerful Microcontroller and/or reducing the size of the model. Several Tests were performed to see the impacts and are displayed separately.

Improving Training Data Quality

It was originally stated that ~215,000 samples were collected to be used in training. Much of this data was collected by the author in the form of long form walks. It was at the time assumed that a larger dataset would improve the quality of the model. The end result however of using the full 3 hours of collected data was an inferior accuracy as opposed to using a smaller subset of ~50 minutes of data, with the larger dataset predictions made against test data were consistently less than their real values, in these cases the Linear Regression of the Laptop Implementation corrected for this error (Figure 89).

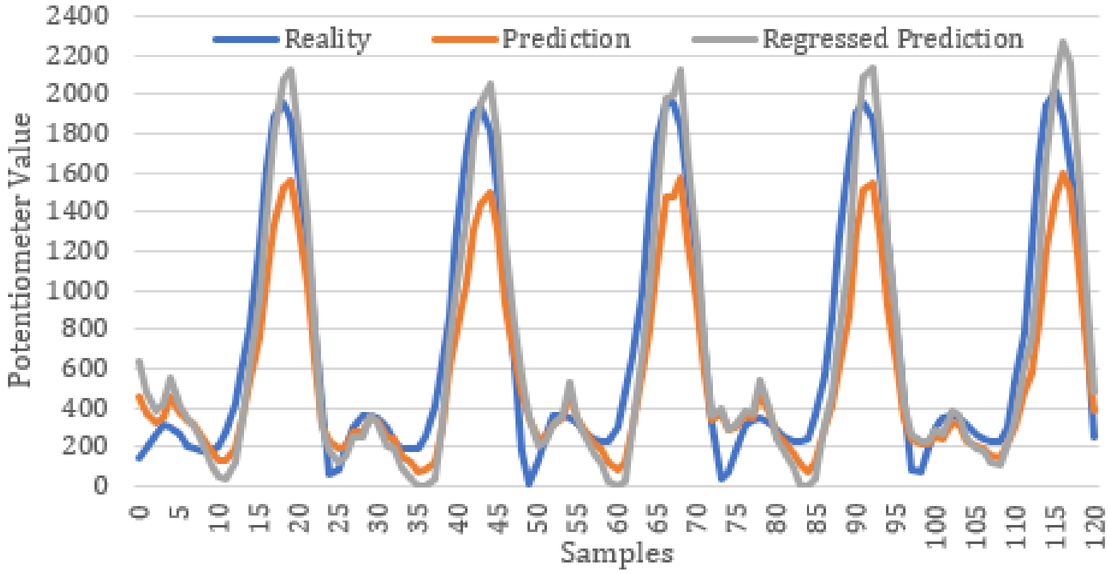


Figure 89 - Laptop Implementation that used full dataset produced values that consistently underestimated reality. 20Hz Sampling Rate.

The assumed reasoning for this was that most of this 3 hours of data was collected by the author performing relatively consistent walking, it is possible that this overwhelming majority of very consistent data polluted the dataset as a whole, by making any variety provided by test data from other participants comparatively insignificant. When retroactively corrected, as seen in Figure 86 and Figure 87, predictions were far more consistent with reality.

Reducing Input Size

For reducing the size of the model, the two most likely places to do this would be to reduce the size of the input array from [40,8], and/or reduce the size of the LSTM layers from 64/32, it was already known reducing the size of the output predictions of the model had no effect on its size of processing speed. For input size, It was found that while reducing the input size of the model did not affect model file size, it did have a proportional effect on the speed of the model, with a smaller size resulting in a faster processing time. An increase in model size of 80 (+[10x8]) resulted in a ~44ms increase in processing time (Figure 90), as such a range of models of different input sizes were run in real time to test the Mean Average Errors of their predictions vs reality. Table 18 shows the results of running each of these models for ~2 minutes whilst walking the same path. The Mean Average Error was calculated as the average of the percentage differences of all predictions vs the real potentiometer value that occurred at that time, made from 1-20 samples into the future for a particular input size, with these results shown in Table 18.

Table 18 – Variance in Mean Absolute Error for ESP32 Predictions vs Reality due to input size

Input Size	[40,8]	[30,8]	[20,8]	[15,8]	[10,8]	Laptop (40x8)
Av. % Diff.	10.56%	10.29%	8.22%	8.88%	9.41%	6.09%

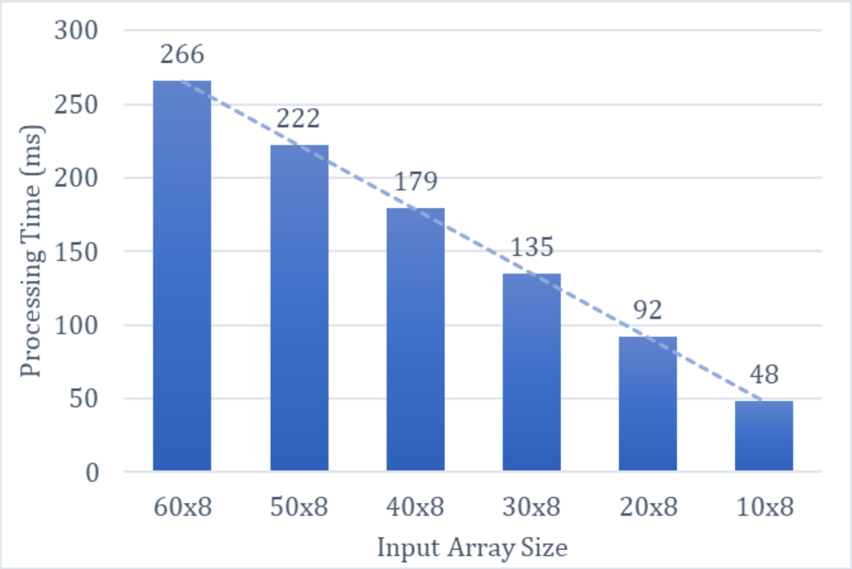


Figure 90 - Linear Relationship between model input size and average processing time in milliseconds.

As seen by the results of Table 18 and displayed continuously in Figure 91, the [20,8] input size presents the lowest percentage difference, acting as the intermediate between having sufficient input size to make an accurate prediction and being able to do so at a quick enough speed to remain useful, although this result is still ~35% less accurate than the Laptop Implementation.

The higher values, at [40,8] and [30,8], are likely of a lower accuracy due to the slow processing times (180ms and 135ms respectively) meaning that whilst having the most input data, they can only make a prediction once every three cycles (50ms), whereas the [20,8]’s 92ms Processing time allows it to predict once every other cycle, allowing for a higher level of accuracy. [15,8] and [10,8] meanwhile see worse accuracy as they also predict every other cycle. While the 48ms processing time of the [10x8] may theoretically allow it to predict once every cycle, the 2ms gap is likely not sufficient for other processes to take place. Therefore, due to its smaller input window it ends up being inferior to the [20,8] input.

This concludes that for an ESP32, a [20,8] input would allow for the highest level of accuracy over the originally used [40,8] input. An Alternative to compromising would be using a faster Microcontroller.

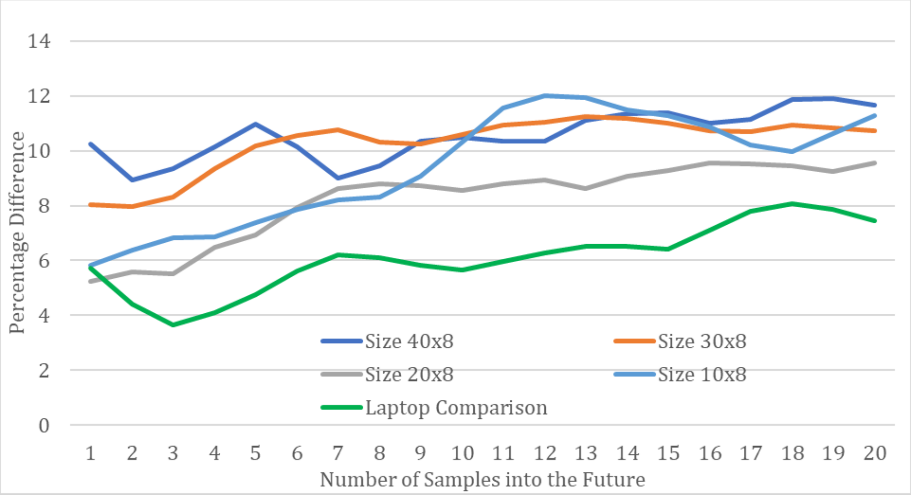


Figure 91 - Change in MAE over number of future predictions due to input size on ESP32. 1 Sample = 50ms.

Alternative Microcontroller Options

Also considered was the use of more powerful Microcontroller options, the chosen implementation, a Teensy 4.1 possessed a Dual Core 600MHz Cortex-M processor, 2.5x the 240MHz of the HUZZAH32 ESP32 Feather that had been used previously. This produced far faster results than expected, with a model of [40,8] input size having a processing time of ~35ms, over 5 times faster than the ESP32, this was lower than the 50ms program cycle time allowing the prediction system to run in real time, providing predictions every program cycle. This considerably improved the Percentage Difference of Predictions vs Reality. As such accuracy measurements were re-captured for 20, 40, and 60 sized input sizes to compare to the prior ESP32 experiments (Figure 92). [40,8] input size proved the best overall percentage difference, varying from 2.01% to 5.01%, with an average of 3.85% (Table 19), this result exceeded the accuracy of the Laptop Implementation. The Teensy 4.1 is more expensive than the ESP32 (~£33.50 [243]), and of similar size. The resultant Prediction vs Reality Graph seen in Figure 93 shows that the Teensy 4.1's predictions were very similar to reality for average walking.

Table 19 – Variance in Mean Absolute Error for Teensy 4.1 Predictions vs Reality due to input size

Input Size	[60,8]	[40,8]	[20,8]
Av. % Diff.	7.68%	3.85%	6.21%

Due to the increase processing power, Linear Regression was implemented onto the Teensy for the [40,8] input, the results for average walking showed slightly inferior accuracies, likely as a result of the predictions already being very accurate with errors not being consistent enough for Linear Regression to have any benefit, however when tested for other situations such as standing or sitting still, linear regression proved far more effective, with the prediction system consistently over or underestimating knee angles by some amount and allowing linear regression to correct this continuous error.

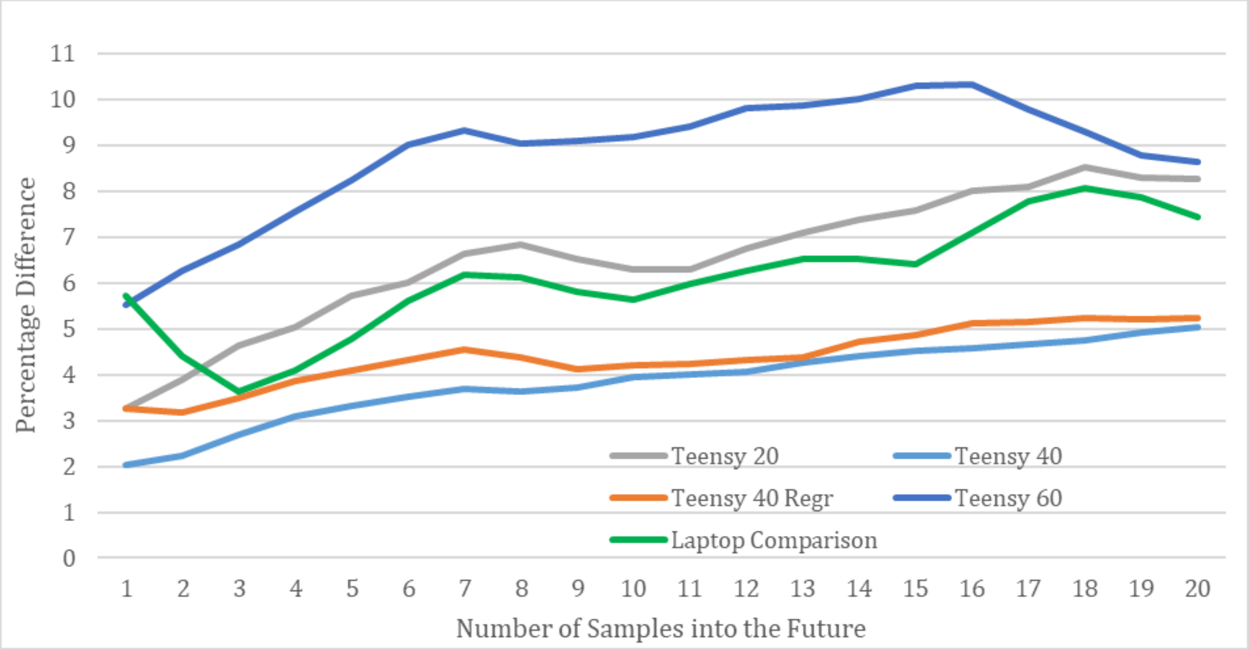


Figure 92 - Change in MAE over number of future predictions using Teensy 4.1. 1 Sample = 50ms.

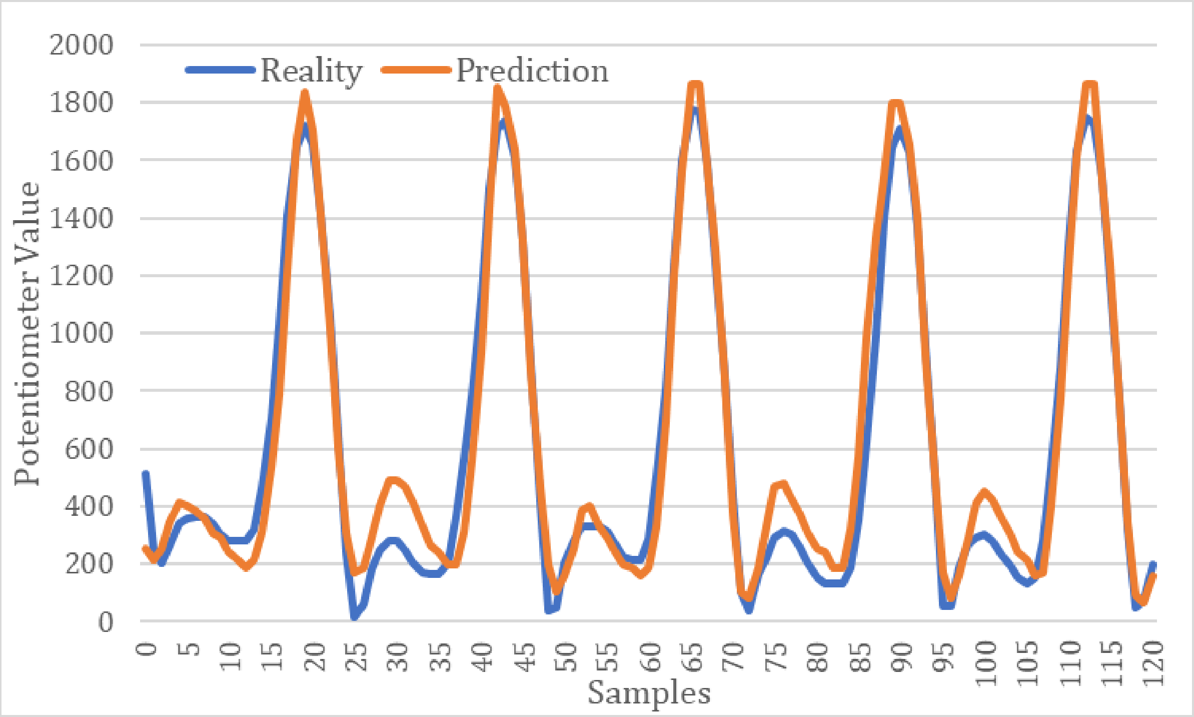


Figure 93 - Teensy 4.1 Prediction vs Reality for [40,8] input. Predictions made 0.25 seconds into the future. 20Hz Sampling Rate.

The Test to find the ideal input size for the Teensy revealed an input of [40,8] produced the best average percentage difference between prediction and reality of ~3.85%, even capable of surpassing the significantly more powerful Laptop Model. This unexpected outcome prompted re-training the Laptop and Teensy base models on the same training data and then testing them on the same machine (a desktop computer) on identical test datasets consisting of 2 sets of 2000 Samples, with results averaged. Both models consistently made each prediction below the 50ms requirement, with the smaller two-layer (64,32) LSTM model barely surpassing the average percentage difference of the four-layer (128,128,128,128) model (5.03% vs 5.11%). When running the smaller model on the Teensy, the average percentage difference was 5.44% (Figure 94), being marginally inferior.

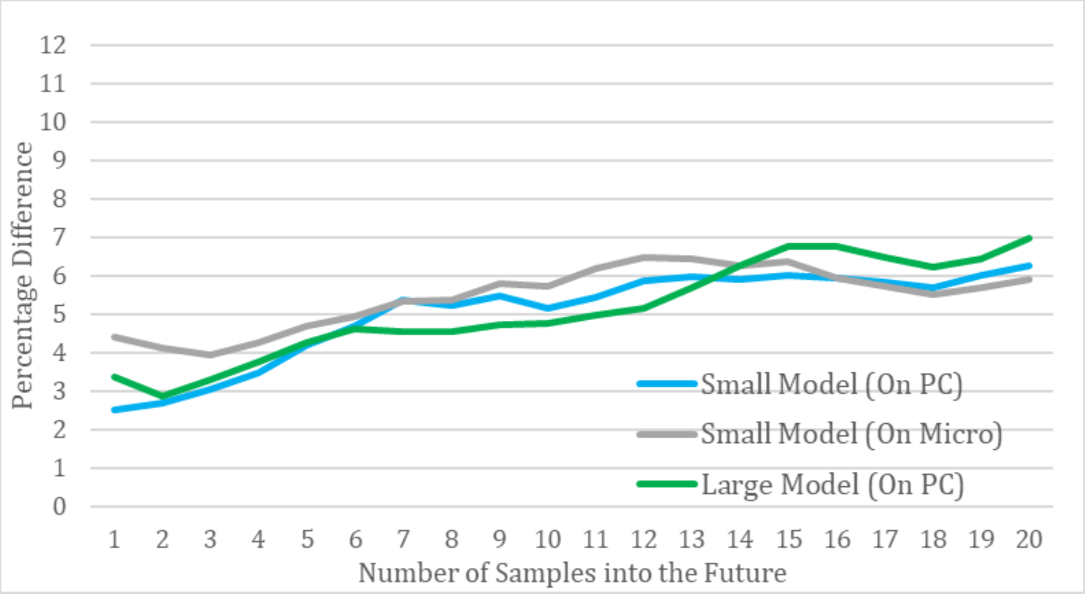


Figure 94 - Change in Percentage Difference over number of future predictions for Small Models (PC and Teensy 4.1 Microcontroller), and Large Model (PC). 1 Sample = 50ms.

It is possible that with the current method an upper limit of accuracy / percentage difference is reached as long as the model is capable of predicting every cycle. Therefore, the additional processing power of the Laptop becomes unnecessary and similar accuracies are achieved even on the microcontroller, with variance only due to differences in testing data. Therefore, this can be seen as the Microcontroller implementation being similarly "effective" as the Laptop implementation in predicting future joint angles, due to a similar Mean Average Error over the same prediction range of 50ms to 1000ms (1 to 20 samples).

As was expected when comparing the Laptop Implementation to the ESP32 Microcontroller Implementation, the Laptop implementation benefits considerably from access to higher processing power by being able to run its model much faster. With both the HUZZAH32 ESP32 Feather possessing a 240MHz dual core microcontroller and the Teensy 4.1 Possessing a 600MHz dual core microcontroller, being not fairly comparable to the ROG Zephyrus M16 GU603ZX's 5GHz, 14 Core CPU. Necessitating the Microcontroller implementation to be smaller to make knee angle predictions within reasonable timeframes (~250ms was considered the maximum "reasonable" timeframe, such that predictions were faster than the human reaction time).

Despite this, when running both prediction systems on a 5,000-sample length dataset they had not been trained on, the average percentage differences between Reality and Prediction for both prediction systems were observed to be somewhat as expected. (Figure 95). With a decrease in accuracy (an increase in percentage difference) with a further prediction into the future.

For both the Microcontroller and Laptop Implementations, predictions followed a straighter line, with a clear linear increase in percentage difference for data predicted further into the future. Whilst the Laptop Implementation produced results of a higher overall accuracy than the ESP32, improvements to the model and usage of the Teensy 4.1 were enough to produce superior or similar results despite the considerable disparity in processing power. It is possible this may be due to reaching a limit in the current method's effectiveness where further processing power does not provide additional benefit.

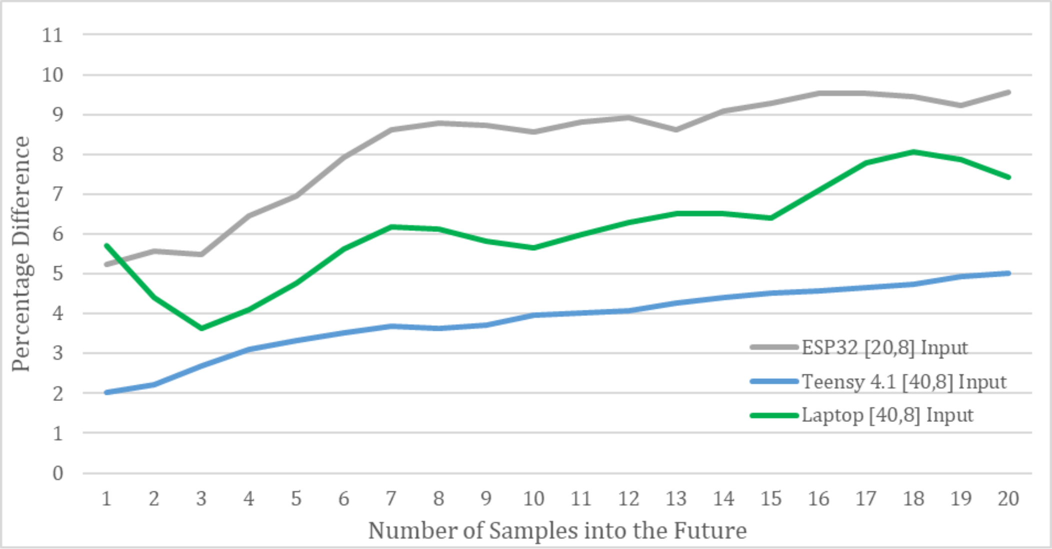


Figure 95 - Comparing Percentage Differences for the best Microcontroller, and Laptop Implementations. 1 Sample = 50ms.

In summary, there is some feasibility in implementing AI Prediction systems in Microcontrollers, although they are inflexible when presented with situations they have not been trained for or otherwise anomalous data. For a Low-Cost Implementation, while it likely would not be suitable in situations where high precision is a necessity, they could have use as providing a general guide for the likely path of a user and is worthy of study as to such potential implementations within future work. In Effect, with this method a small and inexpensive onboard microcontroller is capable of achieving very accurate predictions that are similar or greater than that of more powerful devices, thereby removing the requirement of expensive tethered components. Therefore, the Model would be made to work on a Teensy 4.1 as opposed to the up until now used ESP32.

5.3.5 Implementing Model on Teensy 4.1

As both the Teensy 4.1 and ESP32 were compatible with the Arduino Programming language, which the microcontroller Control system code was written in, the only difficulty came in replacing the ESP32 exclusive Tensorflow Lite and Real Time System libraries with Teensy compatible alternatives. The Inbuilt FreeRTOS of the ESP32 was replaced with a FreeRTOS Port to Teensy named “freertos-teensy” [244], and the TensorFlow Lite implementation was replaced with TensorFlow Lite for Microcontrollers (WCL) [245] which was installed via PlatformIO. Additionally, due to the decreasing in processing time leaving considerable breathing room for additional functionality, Linear Regression was implemented into the Microcontroller implementation to test its effect on result efficiency. As described in the prior section, it had little effect during walking, however improved accuracy during periods of little movement. This formed the final *doModel* function from the original block diagram (Figure 96).

With the Prediction System successfully implemented in a manner that would produce timely and accurate predictions, the full Higher- and Lower-Level Control Systems were collectively implemented within a single program. The format of which matched the design originally described in Figure 58.

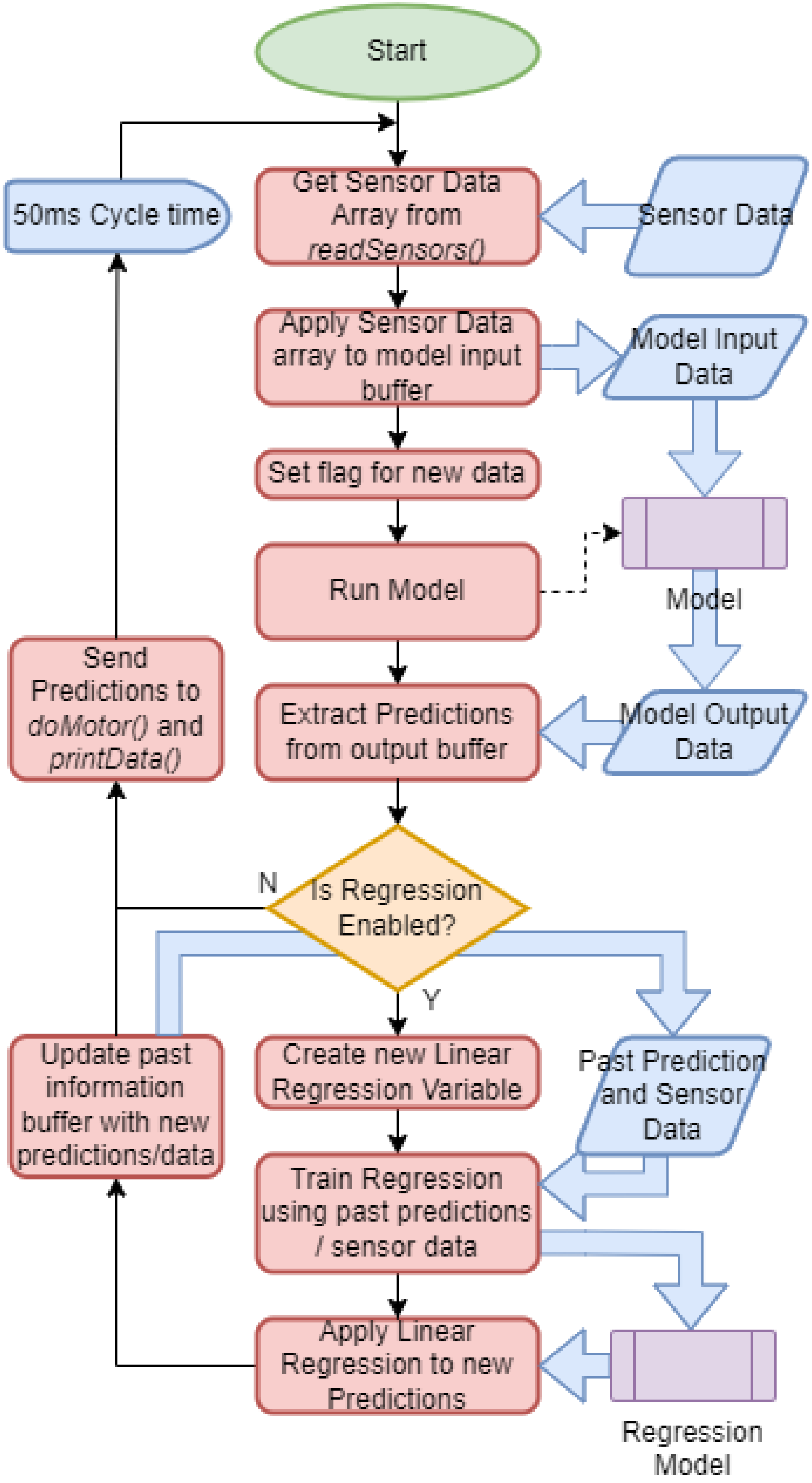

Start
50ms Cycle time
Get Sensor Data Array from *readSensors()*
Sensor Data
Apply Sensor Data array to model input buffer
Model Input Data
Set flag for new data
Run Model
Model
Send Predictions to *doMotor()* and *printData()*
Extract Predictions from output buffer
Model Output Data
N
Is Regression Enabled?
Y
Update past information buffer with new predictions/data
Create new Linear Regression Variable
Past Prediction and Sensor Data
Train Regression using past predictions / sensor data
Apply Linear Regression to new Predictions
Regression Model

Figure 96 - doModel() Flowchart

5.3.6 Test Data Legitimacy

Testing was a key aspect both during development and after its completion. During collection of test data for the predictive system, it was important to know at what point would the number of subjects be sufficiently diverse. Additionally, the limitations of exoskeleton use, such as maximum and minimum force capacities needed to be known for any potential safety concerns. Finally, the actual functionality of the exoskeleton, and its capacity to aid in movement was key to understand whether or not the exoskeleton could serve its purpose.

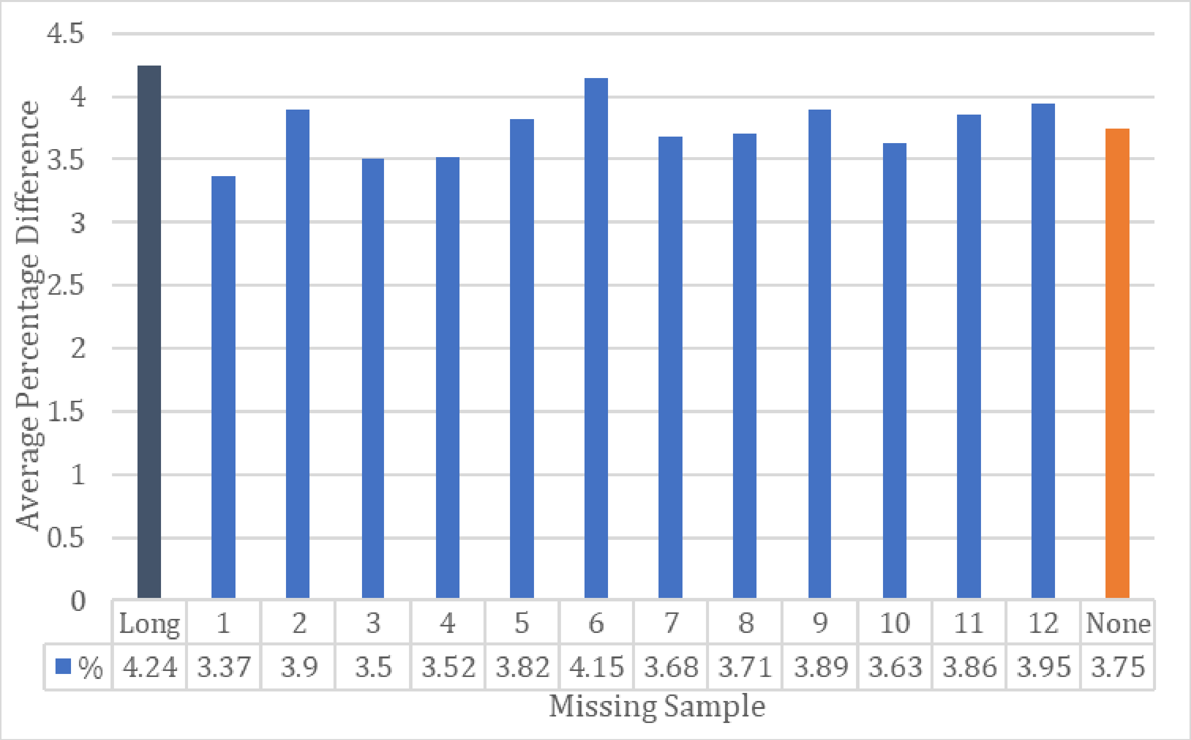
As was described in 5.3.4, the final dataset for training and testing ~58,000 samples, or ~50 minutes of data from the 12 participants, with 90% used for training and 10% used for testing. This data, consisting of the 12 participants, plus a far smaller section of the long form walking data (~5 minutes) was considered to be the final dataset from which all further models would be trained off of. Therefore, to verify the training data of the model was sufficiently diverse such that no single sample caused notable bias, the model was repeatedly trained with one such source of sample data missing, to see whether this had a notable effect on the predictive capabilities of the model. This was repeated for all 12 sample sources provided by external test subjects, as well as the remaining long form data.

Each model was tested on three sets of pre-collected test data consisting of 1000 data points each, with their accuracies recorded. If the lack of a particular participant's sample data did not cause any notable change to the model's Mean Average Error it was reasoned the dataset was sufficiently sized and varied, and therefore not limiting the predictive capabilities of the model. This same three-sample averaging method of testing predictive capabilities would be re-used at future points in exoskeleton production. The model used for the purposes of prediction would be the final Recurrent Neural Network model running on a Teensy 4.1 (5.3.4) predicting the wearer's knee angle 5 samples (250ms) into the future using their past knee angle and IMU data.

Table 20 – Model Mean Average Error when missing one Participant's samples during model training.

Missing Sample	Long	1	2	3	4	5	6	7	8	9	10	11	12	None
Test 1	3.22%	2.91%	3.58%	3.30%	2.77%	3.91%	2.98%	3.38%	2.96%	3.19%	2.99%	4.16%	4.14%	2.97%
Test 2	5.11%	4.43%	5.21%	4.45%	4.25%	4.69%	5.57%	4.86%	4.56%	4.95%	4.55%	4.51%	4.84%	4.96%
Test 3	4.40%	2.77%	2.90%	2.74%	3.55%	2.85%	3.89%	2.80%	3.61%	3.53%	3.36%	2.92%	2.86%	3.31%
Average	4.24%	3.37%	3.90%	3.50%	3.52%	3.82%	4.15%	3.68%	3.71%	3.89%	3.63%	3.86%	3.95%	3.75%

As seen in Table 20 and Figure 97, when comparing the Average accuracy of the model with no samples missing and the average of all average accuracies for each result with only one sample missing (except Long), the results are almost equal, with a Standard Deviation of ~0.222. The removal of Long Form Data consistently produced a decrease in accuracy when compared to the average, which was expected due to being a considerably larger portion of data. The Standard Deviation when including missing Long Form Data was ~0.252. This was considered reasonably small enough to conclude that the current amount of samples was sufficient and that no additional data was required.



	Long	1	2	3	4	5	6	7	8	9	10	11	12	None
■ %	4.24	3.37	3.9	3.5	3.52	3.82	4.15	3.68	3.71	3.89	3.63	3.86	3.95	3.75

Figure 97 – Mean Average Error between Prediction and Reality if one sample dropped. "None" represents the baseline Mean Average Error of the model with no missing sample sources.

It was also notable that, Participant 6, the sample whose removal caused the largest error percentage with exception of the long form data, was data collected from the author, whose walking data also made up the test data that each model was compared against. Effectively as this walking sample was most similar to the test data, its removal may have been what caused a higher-than-average increase in Mean Average Error as a result of its absence. By this extent it can be seen that Participant 1's walking could be considered the least similar to the Author's walking, as its removal reduced the Mean Average error when compared to no samples missing.

Precision

Whilst it is not possible to create a full Confusion Matrix to review the effectiveness of the model, as a time series prediction only consists of "True Positive" (predicted values that accurately match reality) and "False Positive" (predicted values that do not accurately match reality). Precision, which only requires these two values can be calculated. For this, a basis for an "Accurate" prediction would need to be defined, which was defined as a predicted value whose percentage difference from reality was less than or equal to the Mean Average Error, this would better indicate whether a minority number of very accurate or very inaccurate values were causing any sway on the end result of the Mean Average Error. Across the three "base" test samples that had no missing sample sources, precision was very consistently around 60%, with results of 59.26%, 61.57%, and 59.69% for test samples 1, 2, and 3 respectively. This rose to around 80% when measuring precision of data within twice the Mean Average Error. In other words, a slight majority of samples were more accurate than average, with only a minority of highly different values, seen graphically in Figure 98.

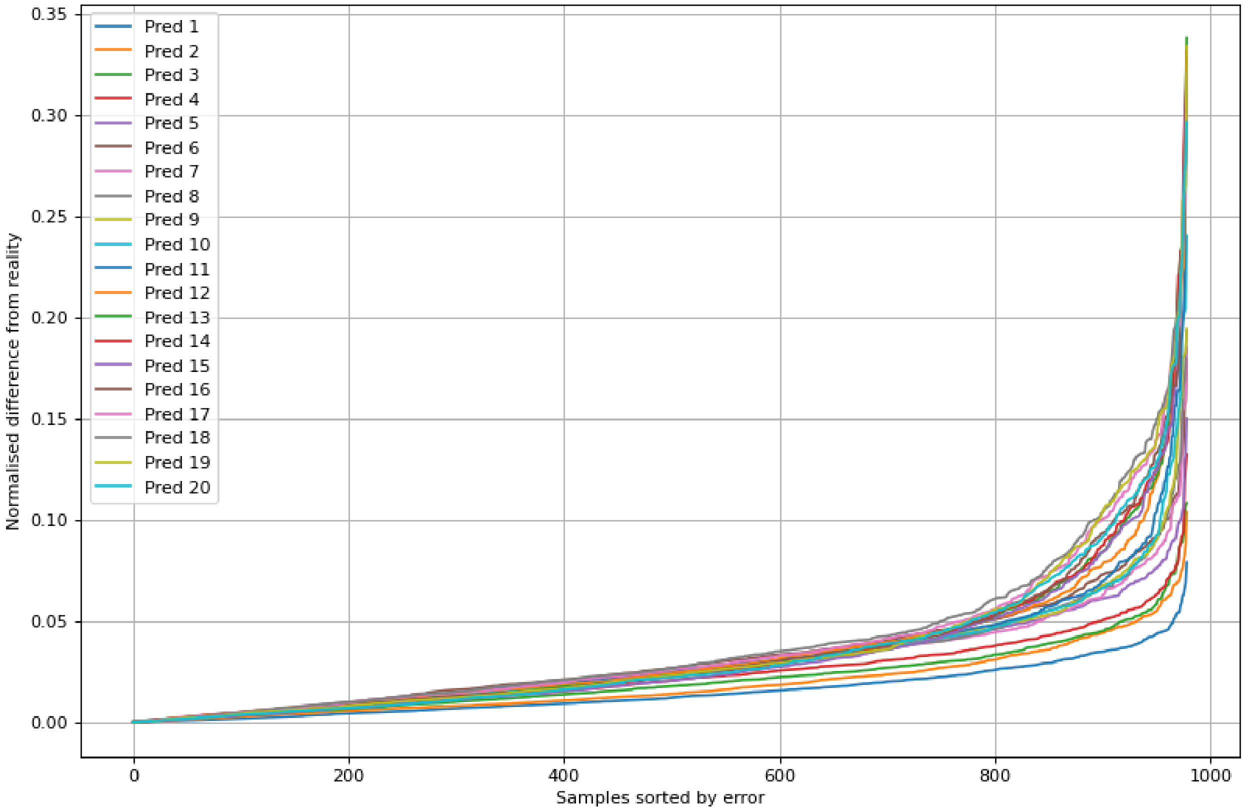


Figure 98 - Normalised Sample differences for all predictions. Sorted by error magnitude.

5.3.7 Summary

For the Prediction System the final LSTM Implementation upon the Teensy 4.1 was able to reach what was considered to be a potential maximum level of accuracy for the given sampling rate and sensor precision used. With the system able to achieve impressive levels of accuracy far beyond what was expected. While as with any Neural Network design, its predictive capacity was limited to what data it was trained on, it was proven that the test data used to form the dataset was sufficiently robust as to be minimally affected by the removal of one of the sample sources.

Similarly, the prediction capabilities of the system as a whole were confirmed to accurate and reasonably precise. With less than 20% of predictions being more than twice the Mean Average Error.

5.4 Final Implementation

Following the block diagram originally laid out in Figure 58, consisting of four primary functions that each ran within their own threads. Those being *readSensors, doModel, doMotor,* and *printData* whose functionality have already been described within 4.5.3 and 5.3.5. The predictions made by the prediction system in *doModel* as to the likely future position of the knee angle would be considered by *doMotor*, if the knee was detected to be moving at a sufficient rate (and so not almost stationary), this future knee angle would be calculated to find the appropriate amount of movement of the exoskeleton actuator that would be required to reach it using the equations described in 3.4.2 and 3.4.4 using the predicted knee angle value.

The First implementation of this system found the difference between the "baseline" motor movement value present as a result of the low-level control system to align the exoskeleton with the present knee angle, and the predicted motor movement value that would rotate the servo motor to the future knee angle. This value would then be applied as a modifier to the actual motor movement value (Figure 99).

Therefore, the command sent to the servo motor would consist of three summed values: The current motor position, the low-level movement value, and the predicted movement modifier. This command would tell the motor to move to this calculated absolute position.

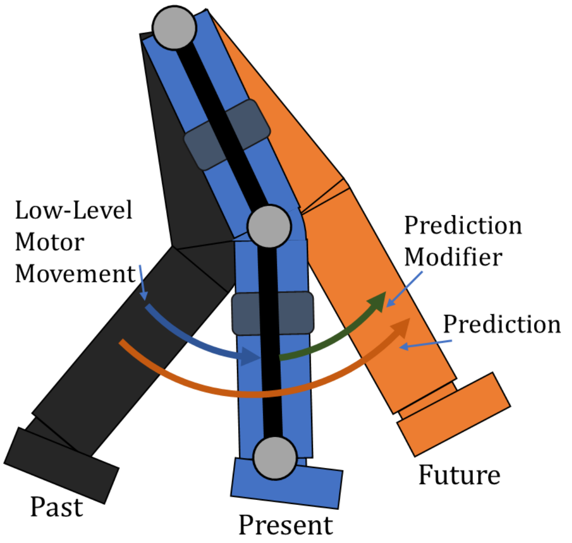


Figure 99 - Prediction Modifier adds onto baseline low-level motor movement.

The amount that the system would predict into the future when controlling the motor needed to be found. For this a Prediction modifier would be generated by looking a number of cycles into the future using the generated set of prediction data. Determining how far to look into the future would be found through more testing, at a baseline, the minimum prediction depth into the future would be determined by how long it would take for the motor to react to the commands given to it by the prediction system and move in a meaningful way. A Prediction too soon into the future would mean that the motor would move after the desired movement window had passed, a prediction too far into the future may mean that the motor would move before the actual movement that was being predicted had occurred, in addition to resulting in on average a lower accuracy prediction.

Predictions of 1 to 8 samples into the future were tested using the same set of sample data. The testing process for both the gearing level and prediction amount used pre-recorded sample data similar to the testing process of the control system, with each tested against the same set of sample data consisting of ~1000 data points to reduce variability to being only as a result of the testing input itself rather than the environment. Results of Motor position vs Reality were then recorded and aligned such as to remove time delay differences between Motor and Reality (recorded as the "Cycle Difference") to calculate Mean, Median, and Standard Deviation. The Mean was calculated based on the sum of the absolute difference between motor position and reality for all points in a sample, divided by the number of points, in other words, the Percentage Mean Absolute Error.

Table 21 - Effect on Error between knee angle and motor position based on prediction depth into the future.

Sample	Prediction	1	2	3	4	5	6	7	8
1	Mean	2.14%	2.50%	2.83%	2.93%	3.18%	3.59%	3.47%	3.37%
	Median	1.12%	1.12%	1.39%	1.43%	1.51%	1.71%	1.94%	1.98%
	Standard Deviation	8.90%	8.38%	8.92%	8.73%	8.78%	8.87%	8.82%	8.47%
	Cycle Difference	+1	+1	+0	-1	-3	-4	-5	-6
2	Mean	1.88%	1.92%	2.13%	2.40%	2.40%	2.31%	2.54%	2.53%
	Median	0.85%	0.85%	1.09%	1.09%	1.05%	1.16%	1.40%	1.32%
	Standard Deviation	8.05%	8.73%	8.69%	9.21%	12.3%	9.3%	9.2%	8.97%
	Cycle Difference	+1	+0	-1	-2	-3	-4	-5	-6
3	Mean	2.32%	2.34%	2.78%	3.30%	2.93%	3.50%	3.30%	3.51%
	Median	1.16%	0.97%	1.36%	1.43%	1.43%	1.90%	1.86%	2.05%
	Standard Deviation	9.21%	9.35%	9.54%	9.34%	9.41%	9.27%	9.43%	8.95%
	Cycle Difference	+1	+1	+0	-1	-2	-4	-5	-6
Average	Mean	2.11%	2.25%	2.58%	2.88%	2.84%	3.13%	3.10%	3.14%
	Median	1.04%	0.98%	1.28%	1.32%	1.33%	1.59%	1.73%	1.78%
	Standard Deviation	8,72%	8.82%	9.05%	9.09%	10.16%	9.15%	9.15%	8.80%

As stated, Cycle Difference represents the difference in program cycles between when a wearer made a movement in reality and when the motor reached this position. A Positive value means the motor reached the position after the wearer did, a negative means they reached it prior to the wearer (effectively moving pre-emptively). As each program cycle was 50ms, it can be seen from the table above that predictions made 1 sample into the future were too close to the present such that the motor did not have the time to move to the predicted future position before it became the present.

Predictions two cycles into the future seemingly lined up with reality, and any predictions beyond this were increasingly pre-emptive. At high levels of negative cycle difference, the motor would become out of sync with the wearer (see Figure 100), therefore lower values were preferred to prevent the motor effectively dragging the wearer along with it.

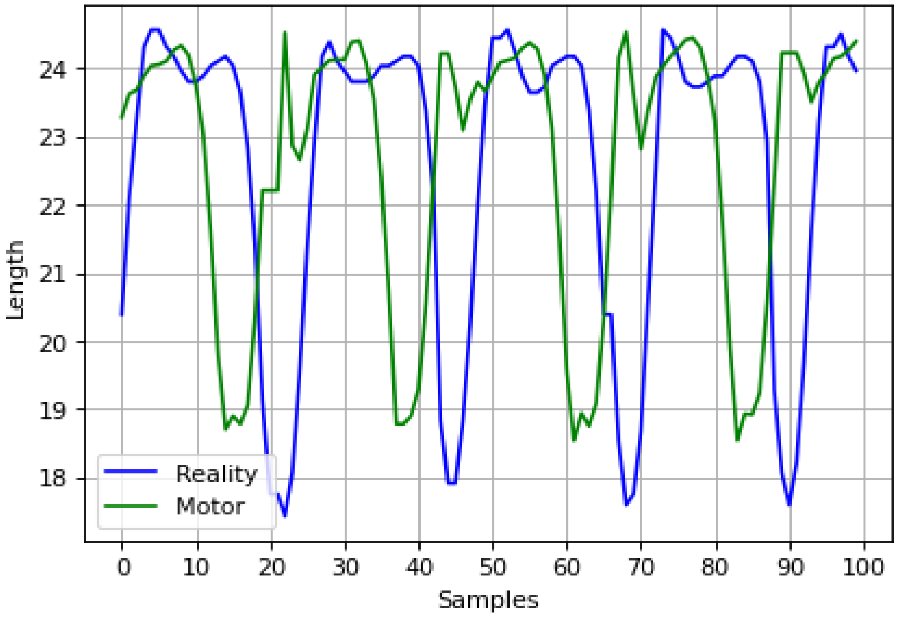


Figure 100 - A Prediction of 8 samples into the future resulted in the motor moving long before the wearer does. 20Hz Sampling Rate.

As seen in Figure 101 the percentage difference between reality and the motor position varied considerably. With regions of high accuracy during positions where knee angle did not significantly vary (primarily stance-phase), and lower during positions with higher variance (primarily swing phase).

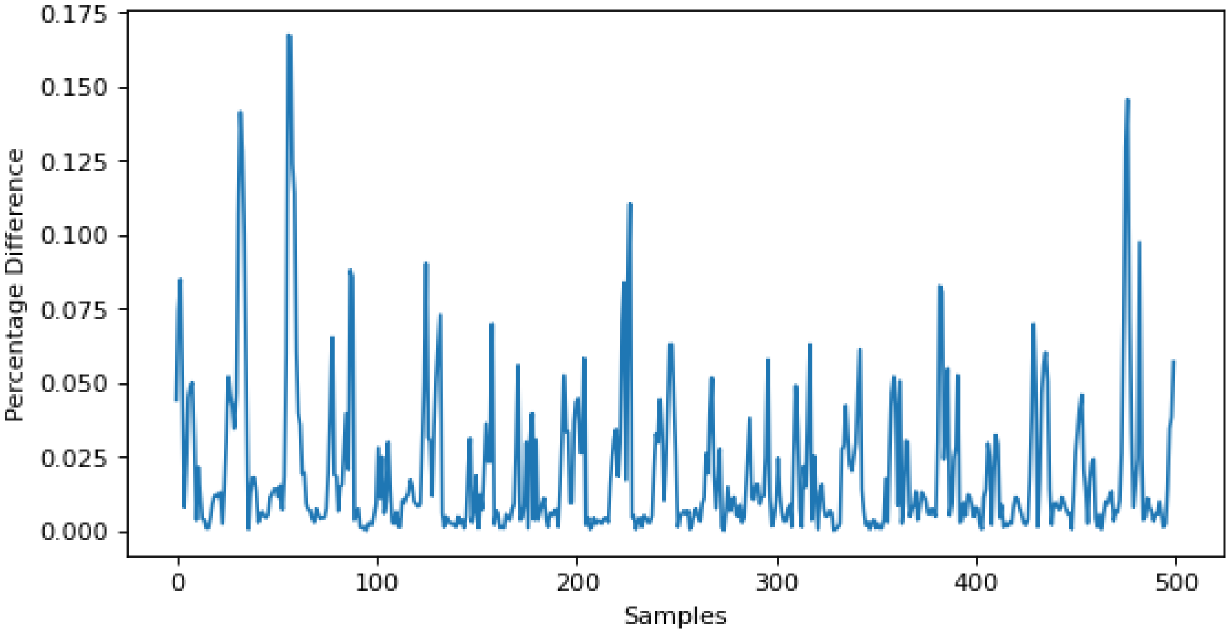


Figure 101 – Variations in Percentage Difference over time between Reality and Motor Position when predicting 3 samples into the future. High Differences lining up with swing motions. 20Hz Sampling Rate.

From this, 3 Samples into the future was considered to be the ideal level of prediction for the prediction system, as this would result in a motor that slightly pre-empted the movement of the wearer. As seen in Figure 102, the results of the movement of the motor closely matches that of the wearer's knee angle.

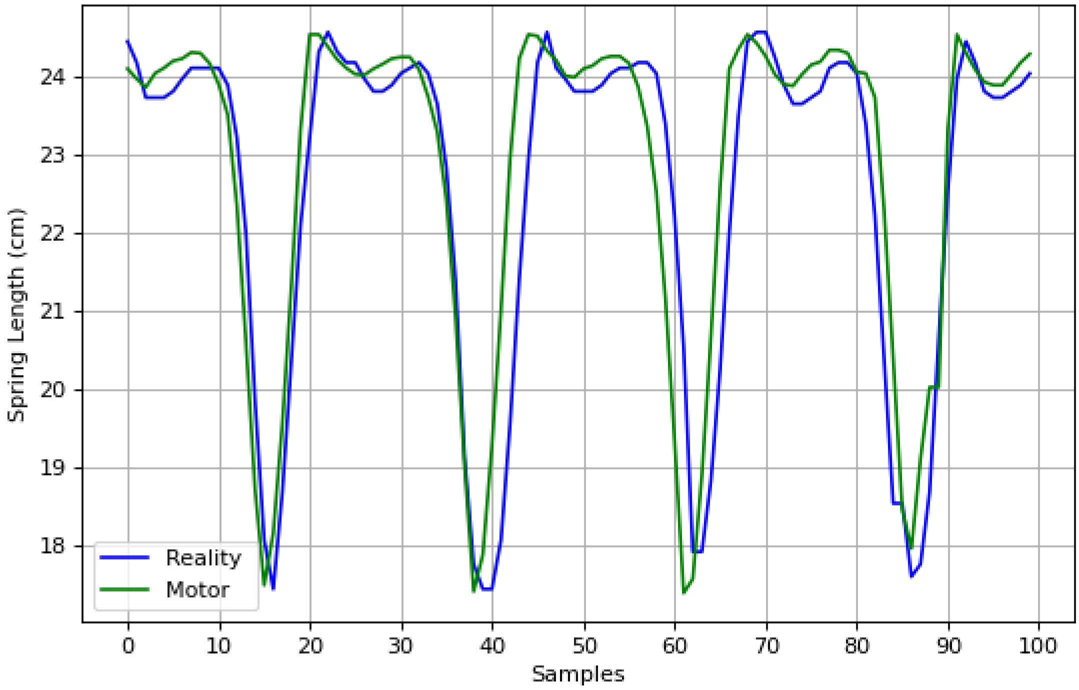


Figure 102 - Motor vs Reality Results of predicting 3 cycles into the future. 20Hz Sampling Rate.

With the basic implementation of the full design complete, how capable it was in performing a variety of different gait behaviours was observed, such as crouching, walking consistently, and walking inconsistently. These tests were performed using data collected live rather than pre-sampled, although with the actuator unable to actively affect the wearer's walking motion, due to no spring rod attached.

5.4.1 Design Effectiveness

While the exoskeleton was capable of following the wearer's walking gait with some reliability in ideal Flat-Terrain Walking scenarios, for real-world functionality it would need to show the ability to make accurate predictions in more diverse conditions. As such several simple tests were devised.

Other Movement States

As Sitting down and Standing up are common motions that often are the targets of exoskeleton assistance, the capacity for the system to be useful in these circumstances was important. To measure it, the exoskeleton would be worn whilst performing consistent in-place sitting down and standing up motions, with a brief pause in between to allow the system to settle from one state to another. Crouch Movement is achieved somewhat reliably by the system, as seen in Figure 103, although appearing to struggle slightly with sitting down. Whilst Standing up however the system was very accurate to reality.

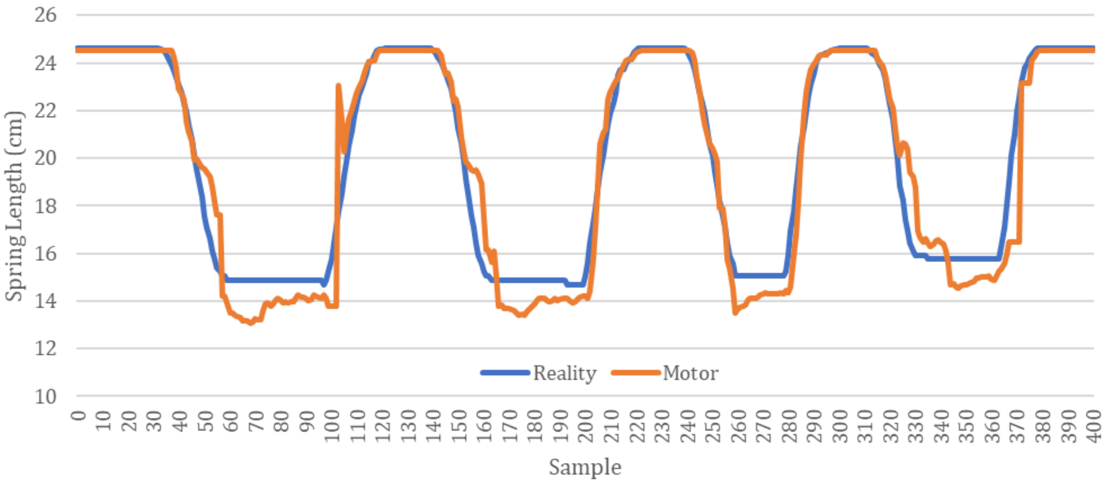


Figure 103 - Standing up and sitting down motions were somewhat accurately predicted. 20Hz Sampling Rate. "Reality" represents actual loaded spring length as derived from knee angle; "Motor" represents target position according to prediction system.

Varied Gait Speeds

As the system had been primarily trained on average speed walking, it struggled on faster and more unpredictable walking styles. To measure this, the exoskeleton would be worn whilst performing a consistent slower than average or faster than average walking pace, with the ability of the motor to both keep up with and accurately match reality measured. On a Slower than average gait, whilst the system roughly followed the pattern of the gait, it did not correctly follow the slower change in knee angle (Figure 104). The system failed completely to recognise a fast gait.

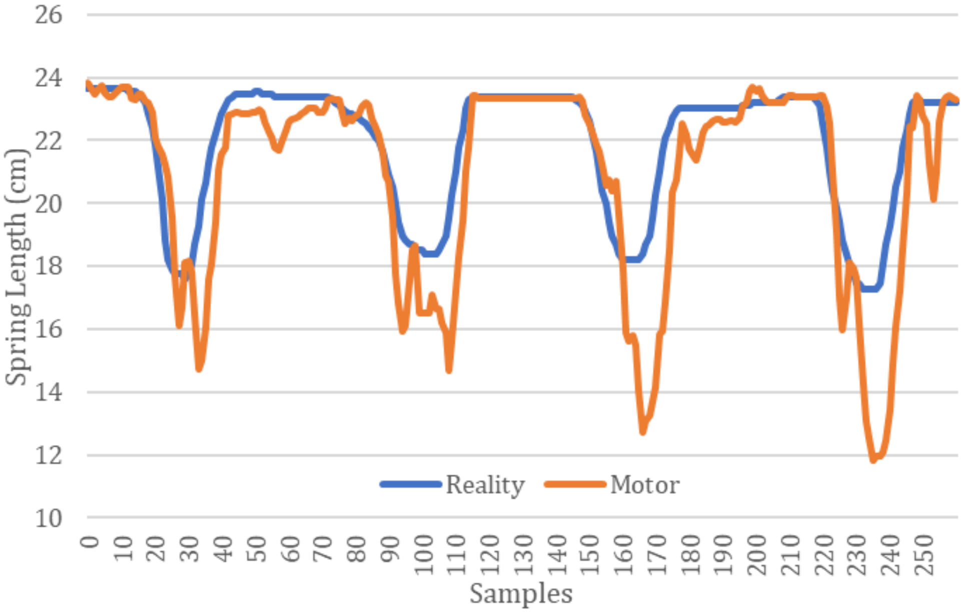


Figure 104 - Slower than average walking gait struggles to be predicted, due to lack of training data. 20Hz Sampling Rate. "Reality" represents actual loaded spring length as derived from knee angle; "Motor" represents target position according tc prediction system.

As the System was not trained to recognise Very Fast or Very Slow walking, it struggling to predict it was not unexpected. As a proof of concept, ~3000 Samples (two and a half minutes) of Very Slow Walking were collected and added to the model's list of training data, after retraining and being tested on the same set of Slow Walking data as seen in Figure 104 (Which was not included in the additional training data provided to the model), the model's capacity to predict slow walking was improved, as seen in Figure 105.

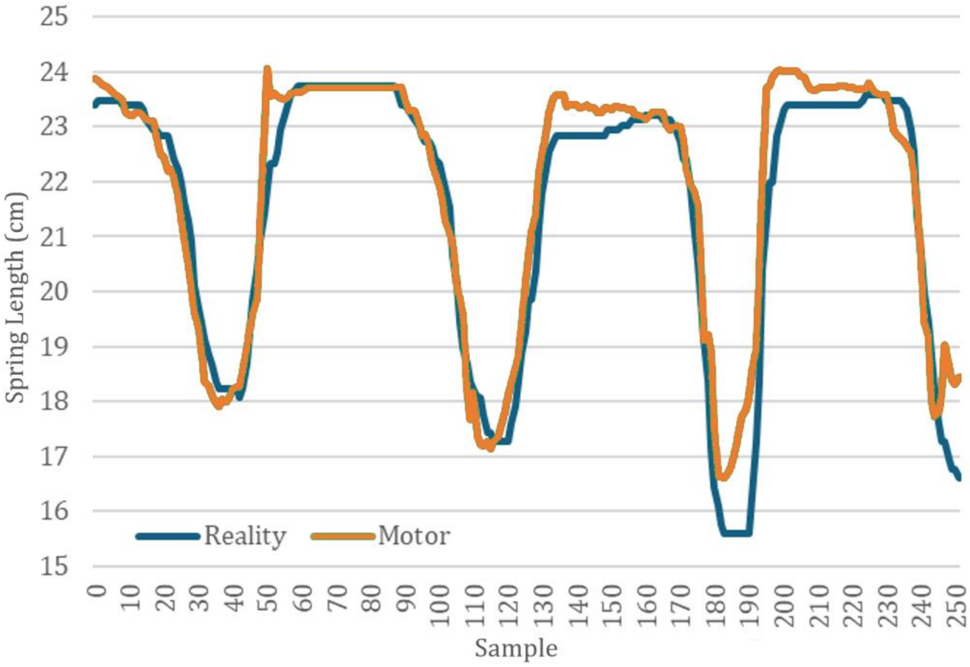


Figure 105 - Post Training on Slower Walking data, predictions more accurate to reality. 20Hz Sampling Rate. "Reality" represents actual loaded spring length as derived from knee angle; "Motor" represents target position according to prediction system.

Despite this, the motor could not be reliably tested whilst having a Spring rod attached, such that the spring would transfer force to the wearer and aid in movement due to occasional random hardware errors that would cause inaccurate results as seen in Figure 106. The most common of these errors were the system not moving in line with predictions, randomly believing itself to be in a vastly different position than it should, or repeatedly staying at the same position despite predictions requiring movement. The RMCS2206 makes use of an Incremental Optical Encoder, and potentially a second encoder could be implemented to check the results of the motor to further reduce error.

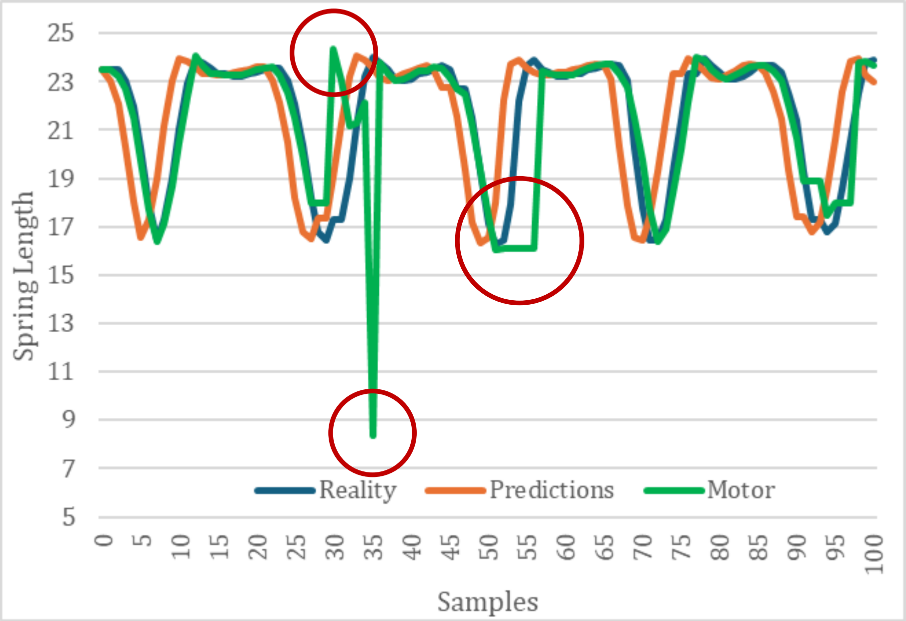


Figure 106 - Random Hardware errors in motor position despite accurate predictions. Displayed is an exceptional example where all errors occurred over a single walk cycle. 20Hz Sampling Rate.

Finally, in order to test the theoretical effectiveness of the exoskeleton, if force were being applied to a wearer's leg, the difference in position between the Unloaded Spring Length due to the motor (l_{s_u}), and the real distance between the upper and lower spring connection points calculated from the knee angle that substitutes the Actual Loaded Spring Length (l_s) can be found.

This difference was measured such at if positive, Unloaded Spring Length would be larger than the Actual Spring Length and so the wearer's leg would be pushed, whilst if negative the Unloaded Spring Length would be less than the Actual Spring Length and so the wearer's leg would be pulled.

When analysing the difference between the Unloaded Spring length and Actual Spring Length vs the change in Actual Spring Length, the two share roughly similar patterns as seen in Figure 107. With both sharing similar peaks and troughs.

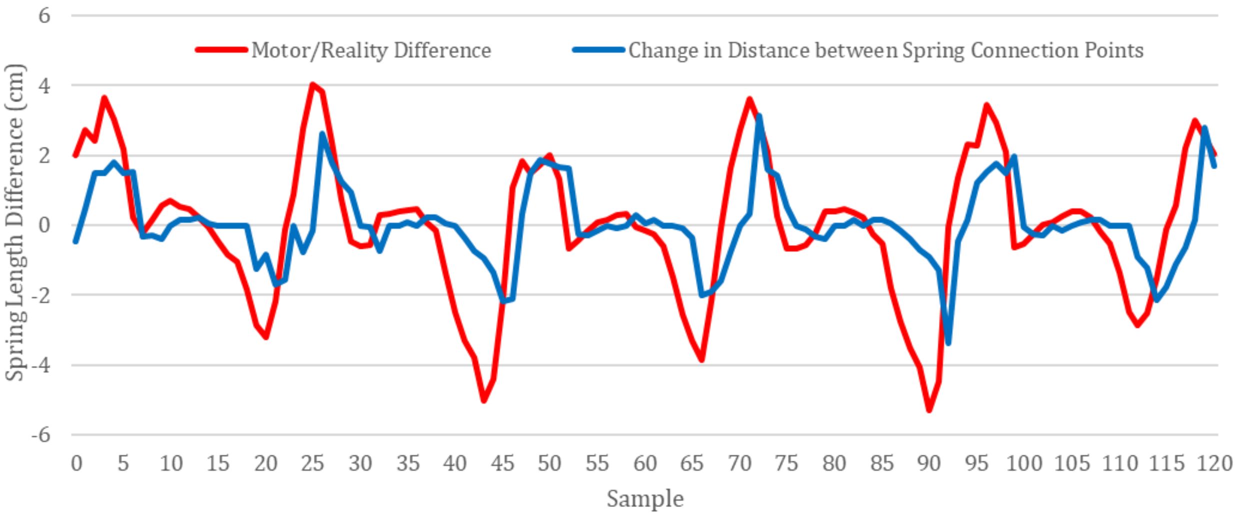


Figure 107 - (Blue) Difference between Spring Length as calculated by motor and Distance between Spring Connection points. (Red) Difference between Spring Connection Points between present and prior value. 20Hz Sampling Rate.

Testing and Determining Effectiveness

In order to define whether or not the theoretical applied force of the motor was "Useful", the proportion of time where the applied force was in the same direction of the wearer's movement can be calculated. As the wearer of the exoskeleton for these tests were healthy, the exoskeleton would only be aiding their movement rather than correcting it. As such, if the change in Actual Spring Length was positive, the knee would be extending, and so ideally the Spring Length as a result of the motor would be larger than l_F as to apply additional assistive force during this extension, and vice versa for flexion, with the Unloaded Spring Length as a result of the motor being less than Actual Spring Length.

When averaging multiple walking sessions totalling over 3000 data Samples, the motor consistently averaged a 60% "Useful" proportion. In other words, at least 60% of the time the exoskeleton was aiding the wearer to some capacity. If some leeway is given to allow instances where Unloaded Spring Length and Actual Spring Length were within 0.5cm of each other, this "Useful" proportion raises to 80%. In other words, at least 80% of the time the exoskeleton was either aiding the wearer or not notably hindering them.

5.4.2 Comparison with other Exoskeletons

The Low-Cost Methodology outlined by this Thesis prioritises minimising Financial, Power, and Weight costs, whilst maintaining effectiveness. Regarding Weights, the total weight of the exoskeleton was ~1.40kg, not including batteries. It had a total cost of ~£250. Components are summarised in Table 22.

Table 22 - Summary of Exoskeleton components

Component	Cost	Weight	Power
RMCS2206	£80	180g	9.6W – 90W (800mA – 7.5A)
Teensy 4.1 + Veroboard	£33.50	17g + ~10g	0.33W (100mA)
LSM9DS1 Breakout Board + Veroboard	£20	2.5g + ~5g	0.33W (100mA)
RS PRO P25	£6.30	~15g	1W Rating
Knee Spring	£45	40g	N/A
ROM Adjustable Knee Brace	£55	970g	N/A
PETG Plastic	~£10	~100g	N/A

When compared to other Knee-Actuated Exoskeletons such as [65], [86], and [246] weighing 1.6kg, 3.52kg, and 2.2kg respectively, all using brake systems, and other exoskeletons such as [133] and [247] that made use of DC Motors, weighing 3.2kg and 8kg respectively. The exoskeleton is, as desired, towards the lower end of exoskeleton weights, and would likely remain so even with material modifications.

In terms of power usage, it can be generalised that for many active exoskeletons, the supermajority of power usage is as a result of the actuator [248], this would be especially true of Low-Cost exoskeleton designs where control systems would be minimised to Microcontrollers and low-power sensors. For an exoskeleton designed for every-day usage there would ideally be sufficient battery capacity to allow for several hours of usage. For example, [249] describes the use of a 5000mAh battery to provide 4 hours of usage, similarly, another cable based system in [102] also aimed for 4 hours of use time with a set of two 22.2V, 1,200mAh batteries weighing just under 500g in total. Another system using a Rotary Motor described in [50] used a 7.4V, 2800mAh LiPo battery to provide 2 hours of continuous operation and weighing 127g. While difficult to measure exactly due to not being loaded, during regular usage the RMCS2206 drew on average 0-1.5A whilst unloaded, and up to ~3A whilst forcefully loaded. Using a pessimistic estimate of constant loaded walking, where 50% of the gait is at minimal load (0.8A) and 50% is at maximal load (3A), the average current draw would be ~1.9A, requiring a battery of about 7.6Ah to achieve 4 hours of continuous usage. A LiPo battery with a capacity of 7.5Ah, from basic research, weighs ~1kg [250], which is of an acceptable weight, as the battery could be contained within a backpack separate from the leg.

Regarding Model Accuracy, when comparing direct control system predictions to reality, as was defined in Table 20, the Mean Absolute Error was roughly 3.75% to predict 250ms into the future. Using a size [40x8] input, sampling potentiometer and Inertial Measurement Unit Sensor data at 20Hz, with a model consisting of one size 64 LSTM layer, one 10% Dropout Layer, one size 32 LSTM layer, and a final Dense Layer to produce a size 20 output series of predictions. As Observed from [251, p. 79]'s review of exoskeleton model implementations many models had a temporally small output window, with many looking 50-100ms into the future at most, as opposed to this model's theoretical maximum of 1 second into the future. For example, [252] used IR Camera Data from 9 optical marks collected at 100Hz and three EMG Electrodes collecting data at 2000Hz spread across all joints, were applied as inputs to separate LSTM input layers before being concatenated, and fed to a fully connected network trained for 200 Epochs. This system predicted only 50ms into the future, but achieved accurate results, with RMSE Means of 0.464 Degrees, or an average percentage different of about 1.54% (across a ~30-degree variation). Similar papers such as [253], and [254] similarly involve knee-angle predicting LSTM's, achieving accuracies of 1.104 and 5.22 degrees for predicting 60ms and 1 time-step respectively into the future.

In practice, the 1 second of prediction into the future was not necessary for this model, however was a byproduct of the low sampling rate. None of the prior examples stated that they were run on microcontrollers. When referring to prior examples of microcontroller-run models, [255] used an STM32f407 and sampled IMU and EMG data at 100Hz, using an RNN to predict up to 50ms into the future with an error of ~2.93 degrees (~4% error). [174] meanwhile used a size 20 input of IMU data from the shank and foot, into a CNN model with two CNN layers, and was deployed onto an ESP32 Microcontroller.

Table 23 - Other examples of Knee Angle Tracking using Microcomputer-Loaded Models

Work	Input Type	Samp. Rate	Input Size	Processor	Model Type	Model Size	Output Size	Pred. Depth	Average Accuracy
This One	Pot/ IMU	20Hz	40	Teensy 4.1	LSTM	64,32	20	50-1000ms	~2.5%
[255]	IMU/ EMG	100Hz	28/20	STM32	RNN	28/20, 32, 16	1	50ms	~4%
[174]	IMU	Unk.	20	ESP32	CNN	128, 64	10	20-200ms	~3.75%

Testing and Determining Effectiveness

In general, the exoskeleton control system produced here has similar results to that of other implementations, although this model was designed to be capable of much higher depths of prediction. Over shorter prediction depths, the accuracy of the model was on average superior, for example, a prediction of 150ms into the future as was used to control the motor resulted in a Mean Average Error of ~2.5%. Although the Mean Average Error across all predictions between 0 and one second into the future was ~5.44%.

5.5 Summary

The development of the prediction, and its implementation within the wider exoskeleton to act as its high-level decision-making process was the culmination of much of the work of the thesis. The final implementation, a time-series prediction method was capable of reliably predicting the knee angle of the wearer up to one second into the future, although achieved optimal predictions at 150ms into the future.

The Long Short-Term Memory Implementation was chosen due being less affected by the disadvantages of other Recurrent Neural Network methods such as a falloff in the prominence of older weights on the outcome of predictions, as well as due to its being able to be run on Cortex-M Architecture common in Microcontrollers. This method proved far more effective than the previously tested Convolutional Neural Network implementation, especially when deployed to the Teensy 4.1, which proved powerful enough to run the model within its 50ms cycle time.

The system successfully compromised on maintaining a high enough sampling rate to receive detailed information as to the movement of the wearer by sampling at 100Hz, but only processing information at 20Hz by averaging the five values collected in this time frame. This drastically increased the time available to processing predictions and therefore allowing a model with a larger input size to be deployed. This model achieved similar or superior prediction accuracy to other implementations, and even models run on far more powerful laptop hardware.

Once deployed to the Exoskeleton, it was proven that the prediction system was capable of driving the movement of the motor and spring-actuator to follow the movement of the wearer. Providing neutral-to-useful assistance force 80% of the time.

The final hardware weight and cost of the exoskeleton was 1.4kg, not including batteries, with a total cost of £250. Although this wait would likely change in a real implementation to account for covering and more durable materials.

Chapter 6: Discussion and Conclusion

6.1 Introduction

This section will review the project as a whole, considering the challenges faced during its development and any limitations as a result, as well as overview the progress and contributions made. In reference to the original Research Aims and Objectives set out in 1.2.1:

Define a methodology to minimise costs of an Exoskeleton Implementation

The Low-Cost Methodology outlined in 2.4 consisted of minimising the costs of an exoskeleton, both financially but also in terms of weight and power usage. Although not at the cost of effectiveness. This methodology prioritised concepts also noted for their significance in other works such as [22]. This methodology became the groundwork for future developments, and was successfully implemented in the creation of the exoskeleton.

Consider the Current State of the Art of Exoskeleton Technology

The Literature Review was created to provide a comprehensive breakdown of the usage of exoskeleton hardware within the state of the art and prior implementations. This was to understand what hardware was most commonly used, and therefore potentially most appropriate for exoskeleton implementations, as well as understand the costs of this hardware to better define a Low-Cost Methodology, with 127 papers reviewed.

As outlined within section 2.7 of the Literature Review, which focused on exoskeleton actuators, the Literature surrounding actuators was sufficiently comprehensive to be compiled into a separate Academic Work [53]. The conclusions, that rotary electric motors and compliant elements were the most appropriate Low-Cost Actuators, and in 2.8 that Force Sensors, IMU's and Potentiometers were effective Low-Cost Sensors, were then used to decide these components within the exoskeleton design. Similar conclusions were also made by several other reviews upon these subjects [39] [40].

For Control Systems, Long-Short Term Memory Neural Networks were found to be a subject of interest, due to their feasibility to be implemented upon low-cost hardware and effective memory retention. Which became the groundwork for their use as the High-Level Prediction System element.

Design a Knee Exoskeleton whose components fit within the Low-Cost Methodology

Based off of the Literature Review, a Theoretical Knee Exoskeleton Actuator was designed in Chapter 4: that consisted of some of the most commonly used Low-Cost components of the Literature Review. Those being Rotary Electric Motors, Compliant Actuators, Potentiometers, IMU's, and inexpensive Microcontrollers. The end result made use of a novel linear actuator design that allowed for far more inexpensive actuators to be used whilst still maintaining high accuracy. This linear actuator's placement was optimally determined within Chapter 3, with its effectiveness successfully modelled mathematically and proved within MATLAB Simulink, with Simulink and mathematical model matching each other almost exactly.

The Exoskeleton's design and low-level control system were measured both qualitatively and quantitatively within 4.6. Qualitatively, the majority of 12 test participants considering the exoskeleton to be comfortable, not significantly inhibitive towards natural movement, wearable for extended periods, and fitting to their proportions. Quantitatively, the low-level control system that would match calculated spring with via knee angle to actual spring length at motor possessed a ~4.2% margin of error. With error reducing at longer spring lengths.

The Final Exoskeleton Weighed 1.4kg, or 2.4kg if including the 7.5Ah battery that would theoretically allow for 4 hours of continuous, active usage. As expanded upon in 5.4.2 this weight and power usage were in line with other similar Low-Impact exoskeletons, and notably lighter than many Rotary Electric Motor Implementations. The total cost of construction was ~£250.

Create a Control System that fits the Low-Cost Methodology

Similarly to the Knee Exoskeleton design, its Control System was designed to minimise expense and power usage. Developed in Chapter 5:, the final implementation ran on a Teensy 4.1 Microcontroller, it consisted of a Low-Level system that calculated the wearer's knee angle and the appropriate spring length for this angle in the present, and a High-Level Neural Network that predicted the knee angle and therefore desired spring length into the future. Then, based on the difference between the present and future values a modifier would be applied to the movement of the motor to move the exoskeleton's position to that of the present plus or minus some future prediction.

This prediction system could predict between 50ms and 1000ms into the future. While there were no known implementations using LSTM's on Microcontrollers, other similar implementations using other neural networks could not predict as far into the future, as described in 5.4.2, and when predicting at the ideal depth of 150ms into the future to be in line with the motor's movement latency, was capable of achieving Mean Average Errors from Reality of ~2.5%. An accuracy on par with or superior to similar implementations, even when a larger model was used or was deployed to superior hardware. The Model deployed to the Microcontroller was similarly capable of matching the accuracy of far more powerful models deployed to High-Powered Laptops, as described in 5.3.5. The Model also maximised its effectiveness with limited size and costs by limiting the rate of processing information to 20Hz, with a sampling rate of 100Hz producing an average of the last 5 values to send to process.

This allowed for greater changes to be seen within a smaller sample window whilst still accounting for noise, further improved by keeping input sample windows of appropriate size to reduce model processing times, and by choosing a microcontroller with high processing power for its size and cost predictions.

As described in 5.3.6, the Model made use of a diverse dataset collected from 12 participants consisting of walking, crouching, sitting, and standing motions to simulate common movement events that a wearer would perform during everyday life. It was successfully proven that this number of participants produced sufficiently varied data that removal of one sample did not have considerable impacts on the accuracy of the data. In 5.4.1 it was also shown that in cases where it was presented with a form of walking it had not been trained on, as little as two and a half minutes of training data added to the recognition system' training dataset provided notable improvements to its prediction capabilities.

In addition to accuracy, the model also maintained reasonable precision. Whilst the Time-Series nature of the prediction system prevent a full Confusion Matrix, when measuring "True" Positive as any prediction closer to reality than the average precision was 60%, and when measuring "True" Positive as any prediction closer to reality than the average precision was 80%.

Test the Effectiveness of the Exoskeleton, Ascertain the Legitimacy of the Low-Cost Methodology

As the primary purpose of the exoskeleton was to collect data and prove implementation feasibility, full clinical trials were outside its scope and would be a subject for future studies. Therefore, effectiveness was defined by the accuracy with which the exoskeleton could track and follow wearer movement such that, if the spring rod were attached such that the spring would apply force to the wearer, it would aid movement.

As previously stated, the exoskeleton achieved a Mean Average Error of 2.5%, meaning that on average it successfully predicted the future motion of the wearer with a 2.5% margin of error assuming consistent movement patterns. If the exoskeleton were allowed to apply force to the wearer, it would apply this predicted motion whilst the wearer was moving or slightly before (as seen in Figure 102). Meaning the issue of latency between wearer intention and exoskeleton motion is resolved due to exoskeleton motion being pre-emptive.

In practice, the exoskeleton was proven within 5.4.1 to provide assistance during walking roughly 60% of the time, and provided no negative effect 80% of the time.

6.2 Challenges and Limitations

Several External issues resulted in limitations applied to the project.

6.2.1 External Issues

Several external events reduced the scope of this project as a result of the limitations they placed upon it. Of these, the most notable being the Covid-19 Pandemic that considerably reduced travel and human interaction possibilities between late 2020 during the start of the project and 2022 towards its mid-point. Partially as a result of this the exoskeleton was unable to be tested with or acquire test data from patients suffering from gait deficiencies, and healthy subjects were used instead.

Secondly, a Cyber-Attack targeting the University of Kent's Computer system during 2021 [256] prevented access to computer systems and workshop access for some time. Although this did not result in any loss of work.

Thirdly, development of the Control System was at several points paralysed as a result of various bugs found within used libraries and material, such as Libraries exceeding file path lengths preventing program generation [257], and Recurrent Neural Networks being improperly supported by TensorFlow [258]. While expected that such difficulties would be met, especially as the implementation of rolled Recurrent Neural Networks upon ARM Cortex-M processors was relatively new at the time (implemented by TensorFlow Lite in September 2022 [259]), it none the less wasted months while the source of these errors were found and eventually fixed.

6.2.2 Limitations

As the exoskeleton was primarily designed for data collection and testing implementation feasibility, its parts were 3D printed to reduce cost. PETG plastic components would not have possessed the strength to support the exoskeleton supporting a human being's weight, especially if they possessed physical disabilities. Therefore, for safety the exoskeleton was tested with healthy individuals for the purposes of collecting data on gait movement and measuring in theory whether the movement of the motor would have supported them should force have been applied, by calculating spring length mathematically based off of knee angle and motor rotation, and the force that would have been applied based on the difference between ideal and actual spring length.

Much of the more diverse test data collected for the exoskeleton was done so in a relatively controlled, flat environment, although the author did collect "real world" information which made up the "Long Form" additional data samples within the final dataset. This exoskeleton was also primarily trained on walking, and independent crouching data.

6.3 Key Findings and Final Conclusions

The primarily aims of the research were the creation of a Low-Cost Methodology, as well as an exoskeleton built via this methodology to define its effectiveness. The Final costs of the Exoskeleton were, financially, under £250, with a power usage of ~85W at maximum, and a weight of ~1.40kg, considerably lower than other implementations. Although this does not consider the addition of stronger materials to replace some 3D printed components.

The Research also implemented a Recurrent Neural Network capable of functioning on inexpensive microcontroller hardware whilst maintaining high levels of accuracy in predicting the movement of the wearer whilst performing consistent walking motions. These models were also proven to maintain this reliability when controlling a physical motor. On Average achieving a Mean Average Error of a few percent despite a far lower sampling rate, model size, and processing power than many comparable implementations.

In summary, the research has achieved the following:

- Defined a Low-Cost Methodology that focused on the minimisation of Financial, Weight, and Power Cost without sacrificing Effectiveness.
- Catalogued from 127 Academic Sources the Component Actuators, Sensors, and Control Systems used within the state of the art and determined which were most effective for the Low-Cost Methodology by applying its tenants.
- Designed and implemented a novel Linear Actuator that focused on minimising power and financial costs by limiting its own usage and not relying on high accuracy. Calculated the optimal position of this actuator mathematically and in simulation. Low-Level Control System accuracy was none the less ~4.2% comparing measured angle to motor position.
- Developed a LSTM Recurrent Neural Network Control System that was capable of running upon inexpensive Microcontroller Hardware, that would predict a wearer’s future knee angle to a ~2.5% Mean Average Error. Proved the dataset provided to it was sufficiently diverse and that the Mean Average Error of the predictions made by this system were on part with far more powerful Neural Network Implementations, and with 60% of predictions being at or lower than this error.
- Combined the Linear Actuator and Recurrent Neural Network Implementations within a physical exoskeleton design that successfully proved that both the Linear Actuator and Recurrent Neural network were capable of tracking a wearer’s movements in reality and applying force that could aid them, with the exoskeleton providing neutral to positively assisting force 80% of the time.

6.4 Future Developments

There are several future developments that could be made based off of this initial implementation to expand its capabilities:

- Superior Materials: Several components, namely the actuator connections to the exoskeleton could be reconstructed with metal to improve durability. This would allow the exoskeleton to potentially be used to support more of a user's weight and provide more assistance.
- Specific User Adaptability: As each wearer of an exoskeleton will likely possess their own intricacies when walking, the ability for an exoskeleton to adapt to a user, or even take in new data to improve itself would be beneficial. Such as Reinforcement Learning shown in [260].
- Testing with Unhealthy Patients: As the exoskeleton was a simple prototype by design, it was more appropriate to test it with healthy patients. A design made of more advanced materials could be tested with patients with gait deficiencies to test the effectiveness of the recognition system both in being trained on and predicting their movements.
- Increase in Effectiveness with Expenditure: As this system highly prioritised minimising cost, it may be beneficial to observe how effectiveness improved with additional expenditure in the exoskeleton's ability to aid the wearer's movement, and if at whether there was a point where further expenditure did not produce notable effectiveness improvement.
- Alternative Actuator Usage: This Thesis prioritised the usage of Rotary Electric Motors and Compliant elements, seeing these as the most effective Low-Cost Implementations. Other Actuator types may prove themselves more effective than predicted and could be an avenue to apply the same Methodology with an exoskeleton that used such actuators.
- Alternative Control System Usage: Similarly, how would the Low-Cost Methodology apply to exoskeletons making use of different control systems, such as relatively new Transformers.

In effect, by deliberately seeking to minimise cost, it may provide a way to maximise the number of potential patients that may be aided. Therefore, there is a merit to find the ideal compromise between the absolute minimum sought by this research, and the expense often seen in many modern implementations. This is however a task left up to future research.

References

[1] P. Milia, F. D. Salvo, M. Caserio, T. Cope, P. Weber, C. Santella, S. Fiorini, B. Giacomo, R. Bruschi, B. Bigazzi, S. Cencetti, M. D. Campo, P. Bigazzi and M. Bigazzi, "Neurorehabilitation in paraplegic patients with an active powered exoskeleton," *Digital Medicine,* vol. 2, no. 4, pp. 163-168, 2016.

[2] G. A. Pratt and M. M. Williamson, "Series Elastic Actuators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, 1995.

[3] E. Swinnen, D. Beckwee, R. Meeusen, J.-P. Baeyens and E. Kerckhofs, "Does Robot-Assisted Gait Rehabilitation Improve Balance in Stroke Patients? A Systematic Review," *Topics in Stroke Rehabilitation,* vol. 21, no. 2, pp. 87-100, 2014.

[4] M. K. Vukobratovic, "When Were Active Exoskeletons actually Born?," *International Journal of Humanoid Robotics,* vol. 4, no. 3, pp. 459-486, 2007.

[5] N. Yagn, "Apparatus for Walking, Running, Jumping". United States of America Patent 420,179, 1890.

[6] M. Vukobratovic, D. Hristic and Z. Stojiljkovic, "Development of active anthropomorphic exoskeletons," *Medical and Biological Engineering,* vol. 12, no. 1, pp. 66-80, 1974.

[7] Specialty Materials Handling Products Operation General Electric Company, *Hardiman I Prototype Project,* 1968.

[8] A. Zoss, H. Kazerooni and A. Chu, "On the Mechanical Design of the Berkeley Lower Extremity Exoskeleton (BLEEX)," in *IEEE/RSJ International Conference on Intelligent Robotics*, 2005.

[9] A. B. Zoss, H. Kazerooni and A. Chu, "Biomechanical Design of the Berkeley Lower Extremity Exoskeleton (BLEEX)," *IEEE/ASME Transactions on Mechatronics ,* vol. 11, no. 2, pp. 128-138, 2006.

[10] Y. Sankai, "HAL: Hybrid Assistive Limb based on Cybernics," in *Springer Tracts in Advanced Robotics*, 2007.

[11] Grand View Research, "Exoskeleton Market Size, Share & Trends Analysis Report, By Mobility, By Technology, By Extremity, By End-use, By Region, And Segment Forecasts, 2023-2030," Grand View Research, 2022.

[12] G. Stoye and B. Zaranko, "UK Health Spending," The Institute for Fiscal Studies, 2018.

[13] T. Prendergast, "Healthcare expenditure, UK Health Accounts provisional estimates: 2022," Office for National Statistics, 2023.

[14] A. Jones, Interviewee, *Ectron EksoNR cost.* [Interview]. 21 September 2021.

[15] Centers for Medicare & Medicaid Services, "National Health Expenditures 2019 Highlights," Centers for Medicare and Medicaid Services, 2019.

[16] C. Parker, J. F. Borisoff, W. B. Mortenson and J. Mattie, "A Survey of stakeholder perspectives on exoskeleton technology," *Journal of NeuroEngineering and Rehabilitation,* vol. 11, no. 169, pp. 1-10, 2014.

[17] D. Pinto, M. Garnier, J. Baras, S.-H. Chang, S. Charlifue, E. Field-Fote, C. Furbish, C. Tefertiller, C. K. Mummidisetty, H. Taylor, A. Jayaraman and A. W. Heinemann, "Budget impact analysis of robotic exoskeleton use for locomotor training following spinal cord injury in four SCI Model Systems," *Journal of NeuroEngineering and Rehabilitation,* vol. 17, no. 4, pp. 1-13, 2020.

[18] S. Jezernik, G. Colombo, T. Keller, H. Frueh and M. Morari, "Robotic Orthosis Lokomat: A Rehabilitation and Research Tool," *Neuromodulation: Technology at the Nerual Interface,* vol. 6, no. 2, pp. 108-115, 2003.

[19] S. Federici, F. Meloni, M. Bracalenti and M. L. De Filippis, "The effectiveness of powered, active lower limb exoskeletons in neurorehabilitation: A systematic review," *Neurorehabilitation,* vol. 37, no. 3, pp. 321-340, 2015.

[20] A. Vallee, "Exoskeleton Technology in nursing practice: assessing effectiveness, usability, and impact on nurses' quality of work life, a narrative review," *BMC Nursing,* vol. 23, no. 156, p. 9, 2024.

[21] L. Morris, R. S. Diteesawat, N. Rahman, A. Turton, M. Cramp and J. Rossiter, "The-state-of-the-art of soft robotics to assist mobility: a review of physiotherapist and patient identified limitations of current lower-limb exoskeletons and the potential soft-robotic solutions," *Journal of NeuroEngineering and Rehabilitation,* vol. 20, no. 18, p. 15, 2023.

[22] A. Gonzalez, L. Garcia, J. Kilby and P. McNair, "Robotic Devices for Paediatric rehabilitation: a review of design features," *BioMedical Engineering OnLine,* vol. 20, no. 89, 2021.

[23] J. Bessler, G. B. Prange-Lasonder, L. Schaake, J. F. Saenz, C. Bidard, I. Fassi, M. Valori, A. B. Lassen and J. H. Burrke, "Safety Assessment of Rehabilitation Robots: A Review Identifying Safety Skills and Current Knowledge Gaps," *Frontiers in Robotics and AI,* vol. 8, pp. 1-18, 2021.

[24] S. Net'ukova, M. Bejtic, C. Mala, L. Horakova, P. Kutilek, J. Kauler and R. Krupicka, "Lower Limb Exoskeleton Sensors: State-of-the-Art," *Sensors,* vol. 22, no. 9091, pp. 1-17, 2022.

[25] J. L. Pons, Wearable Robots: Biomechatronic Exoskeletons, Chichester: John Wiley & Sons Ltd, 2008.

[26] R. M. Razmi, A. N. Azliza and R. M. N. N. Zuliyana, "A Frontal Plane Waist Motion With A Limit HipOffset Y Length Using Inverse Kinematics In NAO Humanoid Robot To Align The Center of Mass (COM) And Robot's Foot," *Journal of Physics: Conference Series,* vol. 1529, no. 4, pp. 1-11, 2020.

[27] K. W. Krigger, "Cerebral Palsy: An Overview," *American Family Physician,* vol. 73, no. 1, pp. 1-10, 2006.

[28] K. Vitrikas, H. Dalton and D. Breish, "Cerebral Palsy: An Overview," *American Family Physician,* vol. 101, no. 4, pp. 213-220, 2020.

[29] P. L. Rosenbaum, D. Damiano, B. Jacobsson, N. Paneth, A. Leviton, M. Goldstein, M. Bax and B. Dan, "A report: The definition and classification of cerebral palsy April 2006," *Developmental medicine and child neurology,* vol. 109, pp. 1-45, 2007.

[30] J. L. Hicks, M. H. Schwartz, A. S. Arnold and S. L. Delp, "Crouched postures reduce the capacity of muscles to extend the hip and knee during the single limb stance phase of gait," *Journal of Biomechanics,* vol. 41, no. 5, pp. 960-967, 2008.

[31] A. Kinsner-Ovaskainen, M. Lanzoni, M. Delobel, V. Ehlinger, C. Arnaud and S. Martin, "Surveillance of Cerebral Palsy in Europe," European Academy of Childhood Disability, Luxembourg, 2017.

[32] R. Palisano, P. Rosenbaum, S. Walter, D. Russell, E. Wood and B. Galuppi, "Development and reliably of a system to classify gross motor function in children with cerebral palsy," *Developmental Medicine & Child Neurology,* vol. 39, pp. 214-223, 1997.

[33] J. McDonald and C. Sadowsky, "Spinal-cord Injury," *The Lancet,* vol. 359, pp. 417-425, 2002.

[34] G. Williams, M. E. Morris, A. Schache and P. R. McCrory, "Incidence of Gait Abnormalities After Traumatic Brain Injury," *Archives of Physical Medicine and Rehabilitation,* vol. 90, no. 4, pp. 587-593, 2009.

[35] E. V. Vasudevan, R. N. Glass and A. T. Packel, "Effects of Traumatic Brain Injury on Locomotor Adaption," *Journal of Neurologic Physical Therapy,* vol. 38, no. 3, pp. 172-182, 2014.

[36] M. Volpini, V. Bartenbach, M. Pinotti and R. Riener, "Clinical evaluation of a low-cost robot for use in physiotherapy and gait training," *Journal of Rehabilitation and Assistive Technologies Engineering,* vol. 4, pp. 1-11, 2017.

[37] R. K. P. S. Ranaweera, R. C. Gopura, T. Jayawardena and G. Mann, "Development of A Passively Powered Knee Exoskeleton for Squat Lifting," *Journal of Robotics Networking and Artificial Life,* vol. 5, no. 1, pp. 45-51, 2018.

[38] Z. Lerner, G. M. Gasparri, M. O. Bair, J. L. Lawson, J. Luque, T. A. Harvey and A. T. Lerner, "An untethered ankle exoskeleton improves walking economy in a pilot study of individuals with cerebral palsy," *IEEE Transactions on Neural Systems and Rehabilitation Engineering ,* vol. 26, no. 10, pp. 1985-1993, 2018.

[39] C. F. Pana, D. Popescu and V. M. Radulescu, "Patent Review of Lower Limb Rehabilitation Robotic Systems by Sensors and Actuation Systems Used," *Sensors,* vol. 23, no. 13, pp. 1-38, 2023.

[40] M. Tiboni, A. Borboni, F. Verite, C. Bregoli and C. Amici, "Sensors and Actuation Technologies in Exoskeletons: A Review," *Sensors,* vol. 22, no. 3, pp. 884-945, 2022.

[41] D. Shakti, L. Methew, N. Kumar and C. Kataria, "Effectiveness of robot-assisted lower limb rehabilitation for spastic patients: A systematic review," *Biosensors and Bioelectronics,* vol. 117, pp. 403-415, 2018.

[42] B. Shi, X. Chen, Z. Yue, S. Yin, Q. Weng, X. Zhang, J. Wang and W. Wen, "Wearable Ankle Robots in Post-stroke Rehabilitation of Gait: A Systematic Review," *Frontiers in Neurorobotics,* vol. 13, no. 63, pp. 1-16, 2019.

[43] K. Nakamura and N. Saga, "Current Status and Consideration of Support/Care Robots for Stand-Up Motion," *Applied Sciences,* vol. 11, no. 1711, pp. 1-21, 2021.

[44] D. Chiaradia, M. Xiloyannis, L. Masia, A. Frisoli and M. Solazzi, "Comparison of a Soft Exosuit and a Rigid Exoskeleton in an Assistive Task," in *International Symposium on Wearable Robotics*, Pisa, Italy, 2018.

[45] K. Batkuldinova, A. Abilgaziyev, E. Shehab and H. Ali, "The Recent development of 3D printing in developing lower-leg exoskeleton: a review," *Mateirals Today: Proceedings,* vol. 42, no. 5, pp. 1822-1828, 2021.

[46] Amazon, "ROM Adjustable Knee Brace Support- Post Op Hinged - Universal Leg Size 67 cm Length," 2017 August 30. [Online]. Available: https://www.amazon.co.uk/ROM-Adjustable-Knee-Brace-Universal/dp/B0758CLQTT/ref=asc_df_B0758CLQTT?tag=bingshoppinga-21&linkCode=df0&hvadid=80264405731096&hvnetw=o&hvqmt=e&hvbmt=be&hvdev=c&hvlocint=&hvlocphy=&hvtargid=pla-4583863982428484&psc=1. [Accessed 2023 September 27].

[47] T. Zhang, M. Tran and H. Huang, "Design and Experimental Verification of Hip Exoskeleton With Balance Capacities for Walking Assistance," *IEEE/ASME Transactions on Mechatronics,* vol. 23, no. 1, pp. 274-285, 2018.

[48] S. Wang, L. Wang, C. Meijneke, E. v. Asseldonk, T. Hoewllinger, G. Cheron, Y. Ivanenko, V. l. Scaleia, F. Sylos-Labini, M. Molinari, F. Tamburella, I. Pisotta, F. Thorsteinsson, M. Ilzkovtiz, J. Gancet, Y. Nevatia, R. Hauffe, F. Zanow and H. Van der Kooij, "Design and Control of the MINDWALKER Exoskeleton," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 23, no. 2, pp. 277-286, 2015.

[49] Thunder Power RC, "TP2800-2SPXRX," [Online]. Available: https://www.thunderpowerrc.com/products/tp2800-2spxrx. [Accessed 2 November 2023].

[50] H. Zhu, C. Nesler, N. Divekar, V. Peddinti and R. D. Gregg, "Design Principles for compact Backdrivable Actuation in Partial-Assist Powered Knee Orthoses," *IEEE Transactions in Mechatronics,* vol. 26, no. 6, pp. 3104-3115, 2021.

[51] Keysight, "E3630A 35 W Triple Output, 6V, 2.5A and ±20V, 0.5A," [Online]. Available: https://www.keysight.com/us/en/product/E3630A/35-w-triple-output-6v-2-5a-20v-0-5a.html. [Accessed 2 November 2023].

[52] R. Chaichaowarat, V. Macha and W. Wannasuphoprasit, "Passive Knee Exoskeleton Using Brake Torque to Assist Stair Ascent," in *IEEE Region 10 Conference*, 2020.

[53] T. Slucock, "A Systematic Review of Low-Cost Actuator Implementations for Lower-Limb Exoskeletons: a Technical and Financial Perspective," *Journal of Intelligence & Robotic Systems,* vol. 106, no. 3, pp. 1-31, 2022.

[54] T. Slucock, "A Systematic Review of Lower-Cost Actuator Implementations for Lower-Limb Exoskeletons - Addendums," 2021. [Online]. Available: https://drive.google.com/drive/folders/1VbrkzbnWZbbhKiqB3QxtUqZAcasgIJnX?usp=sharing. [Accessed 17 November 2021].

[55] B. Wang, Y. Liang, D. Xu, Z. Wang and J. Ji, "Design on Electrohydraulic servo driving system with walking assisting control for lower limb exoskeleton robot," *International Jounral of Advanced Robotic Systems,* vol. 18, no. 1, pp. 1-17, 2021.

[56] S. Malkawi, M. Al-Nimr and D. Azizi, "A multi-criteria optimization analysis for Jordan's energy mix," *Energy,* vol. 127, pp. 680-696, 2017.

[57] Tolomatic, "A Technical Comparison: Performance of pneumatic cylinders and electric rod actuators," Tolomatic, 2020.

[58] E. L. Brancato, "Estimation of Lifetime Expectancies of Motors," *IEEE Electrical Insulation Magazine,* vol. 8, no. 3, pp. 5-13, 1992.

[59] Maxon, "Technology - short and to the point," November 2014. [Online]. Available: https://www.maxongroup.com/medias/sys_master/root/8815461597214/DC-Technology-short-and-to-the-point-14-EN-30-31.pdf?attachment=true. [Accessed 15 April 2022].

[60] S.-w. Woo, "Estimating the Lifetime of the Pneumatic Cylinder in Machine Tool Subjected to Repetitive Pressure Loading," *Journal of US-China Public Administration,* vol. 15, no. 5, pp. 221-238, 2018.

[61] Bimba, "Cylinder Life Expectancy," May 2012. [Online]. Available: https://airinc.net/wp-content/uploads/2014/08/Original-Line-Cylinder_Life_Expectancy-1.pdf. [Accessed 15 April 2022].

[62] B. Wiggin, G. S. Sawicki and S. H. Collins, "An Exoskeleton Using Controlled Energy Storage and Release to Aid Ankle Propulsion," in *International Conference on Rehabilitation Robotics*, 2011.

[63] R. Liang, G. Xu, S. Zhang, X. Zheng, L. Li, Y. Wu, A. Luo and X. Zhang, "Design of Rigid-Compliant Parallel Exoskeleton Knee Joint," in *IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 2017.

[64] T. Slucock, "Exoskeleton Addendum," 11 October 2023. [Online]. Available: https://docs.google.com/spreadsheets/d/1b3Rf1LE7j0U0s5ELU6dMj9uLZce3Ujvk/edit?usp=sharing&ouid=108997210331929022640&rtpof=true&sd=true. [Accessed 11 October 2023].

[65] E. P. Washabaugh and C. Krishnan, "A Wearable Resistive Robot Facilitates Locomotor Adaptions during Gait," *Resorative Neurology and Neuroscience,* vol. 36, no. 2, pp. 215-223, 2018.

[66] S. Sridar, P. H. Nguyen, M. Zhu, Q. P. Lam and P. Polyherinos, "Development of a Soft-Inflatable Exosuit for Knee Rehabilitation," in *International Conference on Intelligent Robots and Systems*, Vancouver, BC,Canada, 2017.

[67] D. E. Sales, Interviewee, *Quote for Delsys Trigno EMG research system.* [Interview]. 2023 October 05.

[68] T. Zhou, C. Xiong, J. Zhang, W. Chen and X. Huang, "Regulating Metabolic Energy Among Joints During Human Walking Using a Multiarticular Unpowered Exoskeleton," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 29, pp. 662-672, 2021.

[69] Y. He, K. Nathan, A. Venkatakrishnan, R. Rovekamp, C. Beck, R. Ozdemir, G. E. Francisco and J. L. Contreras-Vidal, "An Integrated Neuro-Robotic Interface for Stroke Rehabilitation using NASA X1 Powered Lower Limb Exoskeleton," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Buenos Aires, Argentina, 2010.

[70] Biometrics, Interviewee, *Quotes for cost of Biometrics Components.* [Interview]. 10 October 2023.

[71] Y. Bougrinat, S. Achiche and M. Raison, "Design and development of a lightweight ankle exoskeleton for human walking augmentation," *Mechatronics,* vol. 64, no. 102297, pp. 1-12, 2019.

[72] P. Wu, X. Chen, Y. He and Z. Liu, "Unpowered Knee Exoskeleton during Stair Descent," in *The 7th International Conference on Mechatronics and Robotics Engineering*, 2021.

[73] S. Yu, H. Lee, W. Kim and C. Han, "Development of an underactuated exoskeleton for effective walking and load carrying assist," *Advanced Robotics,* vol. 30, no. 8, pp. 535-551, 2016.

[74] W. S. Kim, H. D. Lee, D. H. Lim, J. S. Han, K. S. Shin and C. S. Han, "Development of a muscle circumference sensor to estimate torque of the human elbow joint," *Sensors and Actuators A: Physical,* vol. 208, pp. 95-103, 2014.

[75] Delsys, "Delsys Trigno Research+ System," [Online]. Available: https://delsys.com/trigno/. [Accessed 11 May 2024].

[76] K. Kiguchi and Y. Imada, "EMG-Based Control for Lower-Limb Power-Assist Exoskeleton," in *6th International Special Topic Conference on Information Technology Applications in Biomedicine*, Tokyo, Japan, 2007.

[77] H. Kazerooni, R. Steger and L. Huang, "Hybrid Control of the Berkely Lower Extremity Exoskeleton (BLEEX)," *Department of Mechanical Engineering,* vol. 25, no. 5, pp. 561-573, 2006.

[78] T. Zhang and H. Huang, "A Lower-Back Robotic Exoskeleton," *IEEE Robotics & Automation Magazine,* pp. 95-106, 17 May 2018.

[79] Digikey, “Interlink Electronics 30-81794,” [Online]. Available: https://www.digikey.co.uk/en/products/detail/interlink-electronics/30-81794/2476468. [Accessed 11 October 2023].

[80] U. Onen, F. Botsali, M. Kalyoncu, M. Tinkir, N. Yilmaz and Y. Sahin, “Design and Actuator Selection of a Lower Extremity Exoskeleton,” *IEEE Transactions on Mechatronics,* vol. 19, no. 2, pp. 623-632, 2014.

[81] H. Zabaleta, M. Bureau, G. Eizmendi, E. Olaiz, J. Medina and M. Perez, “Exoskeleton design for functional rehabilitation in patients with neurological disorders and stroke,” in *10th International Conference on Rehabilitation Robotics*, Noordwijk, The Netherlands, 2007.

[82] H. Wu, A. Kitagawa, H. Tsukagoshi and C. Liu, “Development of a Novel Pneumatic Power Assisted Lower Limb for Outdoor Walking by the Use of a Portable Pneumatic Power Source,” in *16th IEEE International Conference on Control Applications*, 2007.

[83] Tekscan, “FlexiForce A201 Sensor,” [Online]. Available: https://www.tekscan.com/products-solutions/force-sensors/a201. [Accessed 11 October 2023].

[84] M. Lee, J. Kim, S. Hyung, J. Lee, K. Seo, Y. J. Park, J. Cho, B.-k. Choi, Y. Shim and H. Choi, “A Compact Ankle Exoskeleton With a Multiaxis Parallel Linkage Mechanism,” *IEEE/ASME Transactions on Mechatronics,* vol. 26, no. 1, pp. 191-202, 2021.

[85] Tekscan, “Flexiforce A401 Sensor,” [Online]. Available: https://www.tekscan.com/products-solutions/force-sensors/flexiforce-a401-sensor. [Accessed 18 October 2023].

[86] T. Yamada, H. Kadone, Y. Shimizu and K. Suzuki, “An Exoskeleton Brake Unit for Children with Crouch Gait Supporting the Knee Joint During Stance,” in *International Symposium on Micromechatronics and Human Sciences (MHS)*, Nagoya, Japan, 2018.

[87] Sparkfun, “Force Sensitive Resistor - Square - SEN - 09376 ROHS,” [Online]. Available: https://www.sparkfun.com/products/9376. [Accessed 18 October 2023].

[88] A. T. Asbeck, K. Schmidt and C. J. Walsh, “Soft exosuit for hip assistance,” *Robotics and Autonomous Systems,* vol. 73, pp. 102-110, 2014.

[89] Phidgets, “Single Point Load Cell 20kg,” 12 October 2023. [Online]. Available: https://www.phidgets.com/?prodid=225. [Accessed 12 October 2023].

[90] G. M. Bryan, P. W. Franks, S. C. Klein, R. J. Peuchen and S. H. Collins, “A Hip-knee-ankle exoskeleton emulator for studying gait assistance,” *The International Journal of Robotics Research,* vol. 40, no. 4-5, pp. 722-746, 2021.

[91] Logicbus, “FSH03904,” 12 October 2023. [Online]. Available: https://www.logicbus.com/FSH03904_p_13732.html. [Accessed 12 October 2023].

[92] Farnell, “SGT-3/700-FB43 Omega, Strain Gauge, 700 ohm, 3.4 mm,” [Online]. Available: https://uk.farnell.com/omega/sgt-3-700-fb43/strain-gauge-1-9mm-700-ohm-30000um/dp/3867070. [Accessed 18 October 2023].

[93] H. Zheng, T. Shen, M. R. Afsar, I. Kang, A. J. Young and X. Shen, “A Semi-Wearable Robotic Device for Sit-to-Stand Assistance,” in *IEEE 16th International Conference on Rehabilitation Robotics*, 2019.

[94] Bertec, Interviewee, *Quote for Bertec Force Plates.* [Interview]. 13 October 2023.

[95] Interlink Electronics, “FSR Integration Guide and Evaluation Parts Catalog,” Interlink Electronics, Camarillo, Calinfornia, 2015.

[96] C. Price, D. Parker and C. Nester, “Evaluation of pressure insoles during running,” *Procedia Engineering,* vol. 2, no. 2, pp. 3053-3058, 2010.

[97] C. Price, C. N. Parker and D. Parker, “Validity and repeatability of three in-shoe pressure measurement systems,” *Gait & Posture 46,* vol. 46, pp. 69-74, 2016.

[98] M. Mioskowska, D. Stevenson, M. Onu and M. Trkov, “Compressed Gas Actuated Knee Assistive Exoskeleton for Slip-Induced Fall Prevention During Human Walking,” in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2020.

[99] Mouser Electronics, “Bend Labs 105040402-01,” 12 October 2023. [Online]. Available: https://www.mouser.co.uk/ProductDetail/Bend-Labs/105040402-01?qs=PzGy0jfpSMs7MJw77sDDeQ%3D%3D. [Accessed 12 October 2023].

[100] M. Gloger, G. Obinata, E. Genda, J. Babjak and Y. Pei, “Active Lower Limb Orthosis with One Degree of Freedom for People with Paraplegia,” in *International Conference on Rehabilitation Robotics*, 2017.

[101] A. W. Technologies, Interviewee, *Quote for Opal Mobility Lab System.* [Interview]. 06 October 2023.

[102] K. Schmidt, J. E. Duarte, M. Grimmer and S.-P. Alejandro, “The Myosuit: Bi-Articular Anti-gravity Exosuit That Reduces Hip Extensor Activity in Sitting Transfers,” *Frontiers in Neutorobotics,* vol. 11, no. 57, pp. 1-16, 2017.

[103] Adafruit, “Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board - LSM9DS0,” 16 October 2023. [Online]. Available: https://www.adafruit.com/product/2021. [Accessed 16 October 2023].

[104] ActiveRobots, “Sparkfun 9DOF Razor IMU M0,” [Online]. Available: https://www.active-robots.com/sparkfun-9dof-razor-imu-m0.html. [Accessed 18 October 2023].

[105] S. Naito, Y. Akiyama, K. Ohashi, Y. Yamada and S. Okamoto, “Development of a non-actuated wearable device to prevent knee buckling,” in *1st Global Conference on Life Sciences and Technologies (LifeTech 2019)*, Osaka, Japan, 2019.

[106] Motion Analysis, “Motion Analysis Pricing,” 15 October 2023. [Online]. Available: https://www.motionanalysis.com/pricing/. [Accessed 15 October 2023].

[107] G.-S. Huang, S.-C. Chang, C.-L. Lai and C.-C. Chen, “Development of a Lower Extremity Exoskeleton as an Individualised Auxiliary Tool for Sit-to-Stand-to-Sit Movements,” *IEEE Access,* vol. 9, pp. 1-9, 2017.

[108] Amazon, “Xbox One Kinect Sensor,” [Online]. Available: https://www.amazon.co.uk/Xbox-GT3-00002-One-Kinect-Sensor/dp/B00INAX3Q2. [Accessed 18 October 2023].

[109] Y. He, N. Li, C. Wang, L.-q. Xia, X. Yong and X.-y. Wu, “Development of a novel autonomous lower extremity exoskeleton robot for walking assistance,” *Frontiers of Information Technology & Electronic Engineering,* vol. 30, pp. 318-329, 2019.

[110] P. Felix, J. Figueiredo, C. P. Santos and J. C. Moreno, “Powered Knee Orthosis for Human Gait Rehabilitation: First Advances,” in *IEEE 5th Portuguese Meeting on Bioengineering*, 2017.

[111] A. Ganguly, D. Sanz-Merodio, G. Puyuelo, A. Goni, E. Garces and E. Garcia, “Wearable perdiatric gait exoskeleton - A feasibility study,” in *International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018.

[112] RS Electronics, “Honeywell Gauge Pressure Sensor, 100psi Operating Max, Through-Hole Mount, 8-Pin, DIP,” 16 October 2023. [Online]. Available: https://uk.rs-online.com/web/p/pressure-sensor-ics/2141505. [Accessed 16 October 2024].

[113] C. M. Thalman, T. Hertzell and H. Lee, “Toward A Soft Robotic Ankle-Foot Orthosis (SR-AFO) Exosuit for Human Locomotion: Preliminary Results in Late Stance Plantarflexion Assistance,” in *3rd IEEE International Conference on Soft Robotics*, 2020.

[114] J. A. Blaya and H. Herr, “Adaptive Control of a Variable-Impedance Ankle-Foot Orthosis to Assist Drop-Foot Gait,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 12, no. 1, pp. 24-31, 2004.

[115] Digikey, “Bourns Inc. 6637S-1-502,” [Online]. Available: https://www.digikey.co.uk/en/products/detail/bourns-inc/6637S-1-502/3534258. [Accessed 17 October 2023].

[116] V. Bartenbach, M. Gort and R. Riener, “Concept and Design of a Modular Lower Limb Exoskeleton,” in *International Conference on Biomedical Robotics and Biomechatronics*, 2016.

[117] RS, “Bourns 10kΩ Potentiometer Through Hole, 3382H-1-103,” [Online]. Available: https://uk.rs-online.com/web/p/potentiometers/7703166. [Accessed 17 October 2023].

[118] MidoriPrecisions, Interviewee, *Quote for LP-100F Linear Potentiometer.* [Interview]. 19 October 2023.

[119] Delsys, Interviewee, *Quote for Delsys Goniometer and Adaptor.* [Interview]. 5 October 2023.

[120] NetzerPrecision, Interviewee, *Quote for DS-25.* [Interview]. 8 October 2023.

[121] F. Giovacchini, F. Vannetti, M. Fantozzi, M. Cempini, M. Cortese, A. Parri, Y. Tingfang, D. Lefeber and N. Vitiello, “A Light-Weight Active Orthosis for Hip movement assistance,” *Robotics and Autonomous Systems,* vol. 73, pp. 123-134, 2015.

[122] Maxon, “MILE encoder, 1024 cpt, 2-channel,” [Online]. Available: https://www.maxongroup.com/maxon/view/product/621795. [Accessed 18 October 2023].

[123] P. D. Neuhaus, J. H. Noorden, T. J. Craig, T. Torres, J. Kirschbaum and J. E. Pratt, “Design and Evaluation of Mina a Robotic Orthosis for Paraplegics,” in *IEEE International Conference of Rehabilitation Robotics*, 2011.

[124] Mouser, “Broadcom / Avago HEDL-5640#A1,” [Online]. Available: https://www.mouser.co.uk/ProductDetail/Broadcom-Avago/HEDL-5640A13?qs=RuhU64sK2%252Bv3nPu5sOD%2FhQ%3D%3D. [Accessed 18 October 2023].

[125] H. K. Kwa, J. H. Noorden, M. Missel, T. Craig, J. E. Pratt and P. D. Neuhaus, “Development of the IHMC Mobility Assist Exoskeleton,” in *Kobe International Conference Center*, Kobe, Japan, 2009.

[126] E-MotionSupply, “RENISHAW: RGH24 RGS20 LINEAR ENCODER SYSTEM READHEAD RGH24X00Z01A,” [Online]. Available: https://www.e-motionsupply.com/RENISHAW_RG2_READHEADS_RGH24_X_00_Z_01A_p/rgh24-x-00-z-01a.htm. [Accessed 18 October 2023].

[127] J. L. Emken, J. H. Wynne, S. J. Harkema and D. J. Reinkensmeyer, “A Robotic Device for Manipulating Human Stepping,” *IEEE Transactions on Robotics ,* vol. 22, no. 1, pp. 185-189, 2006.

[128] E-MotionSupply, “RENISHAW: RGH41 INCREMENTAL ENCODER SYSTEM RGH41A50L00A,” [Online]. Available: https://www.e-motionsupply.com/Renishaw_RGH41_READHEADS_p/rgh41a50l00a.htm. [Accessed 18 October 2023].

[129] Maxon, “Encoder ENX 16 EASY Absolute, with BiSS-C interface,” [Online]. Available: https://www.maxongroup.com/maxon/view/product/sensor/encoder/ENX/ENX16EASY/ENX16EASY05. [Accessed 17 October 2023].

[130] A. Ortlieb, M. Bouri, R. Baud and H. Bleuler, “An Assistive Lower Limb Exoskeleton for People with Neurological Gait Disorders,” in *International Conference on Rehabiliation Robotics*, 2017.

[131] H. Yu, S. Huang, G. Chen, Y. Pan and Z. Guo, “Human-Robot Interaction Control of Rehabilitation Robots With Series Elastic Actuators,” *IEEE Transactions on Robotics,* vol. 31, no. 5, pp. 1089-1100, 2015.

[132] M. Vukobratovic and B. Borovac, “Zero-Moment Point - Thirty Five Years of its Life,” *International Journal of Humanoid Robotics,* vol. 1, no. 1, pp. 157-173, 2004.

[133] Z. F, D. L and T. C. Bulea, “A Lower-Extremity exoskeleton improves knee extension in children with crouch gait from cerebreal palsy,” *Science Translational Medicine,* vol. 9, pp. 1-10, 2017.

[134] C. J. Walsh, K. Pasch and H. Herr, “An autonomous, underactuated exoskeleton for load-carrying augmentation,” in *International Conference on Intelligence Robotis and Systems*, Beijing, China, 2006.

[135] Diamond systems, “Aurora PC/104 SBC with Atmo Z-Series CPU,” [Online]. Available: https://www.diamondsystems.com/files/binaries/aurora%20datasheet.pdf. [Accessed 24 May 2024].

[136] A. M. Control, Interviewee, *ACS Motion Control / Tech80 5912-3 3 Axis PC/104 Encoder Interface.* [Interview]. 10 October 2023.

[137] J. Chung, R. Heimgartner, C. T. O'Neill, N. S. Phipps and C. J. Walsh, “ExoBoot, a soft inflatable robotic boot to assist ankle during walking: Design, Characterization, and Preliminary Tests,” in *7th IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2018.

[138] NI, “cRIO-9082 Specifications,” 2024. [Online]. Available: https://www.ni.com/docs/en-US/bundle/crio-9082-specs/page/specs.html.

[139] Speedgoat, “Unit Real-Time Target Machine,” 2024. [Online]. Available: https://www.speedgoat.com/products-services/real-time-target-machines/unit-real-time-target-machine.

[140] Y. Gu, Y. Lv, X. Ma and C. Lu, “Lower-Limb Soft Orthotic Device for Gait Assistance,” in *4th Information Technology and Mechatronics Engineering Conference*, 2018.

[141] Arduino, “Arduino Uno Rev3,” [Online]. Available: https://store.arduino.cc/products/arduino-uno-rev3. [Accessed 26 October 2023].

[142] L. Wang, S. Bao and K. Wang, “Experimentation Research on Lower-Limb Power Assisted Robot,” in *World Congress on Intelligent Control and Automation*, Jinan, China, 2010.

[143] Y.-L. Park, B.-r. Chen, D. Young, L. Stirling, R. J. Wood, E. Goldfield and R. Nagpal, “Bio-inspired Active Soft Orthotic Device for Ankle Foot Pathologies,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011.

[144] N. R. S. Costa and D. G. Caldwell, “Control of a Biomimetic "Soft-Actuated" Lower Body 10DOF Exoskeleton,” in *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2006.

[145] Mouser, “ATMEGA16-16PU,” [Online]. Available: https://www.mouser.co.uk/ProductDetail/Microchip-Technology/ATMEGA16-16PU?qs=cxy41lVAGV%2FB51GbOobVWw%3D%3D. [Accessed 26 October 2023].

[146] RS, “Microchip ATMEGA328-PU, 8bit AVR Microcontroller, ATmega, 20MHz, 32 kB Flash, 28-Pin PDIP,” [Online]. Available: https://uk.rs-online.com/web/p/microcontrollers/1310277. [Accessed 26 October 2023].

[147] Mouser, “ATMEGA128-16AU,” [Online]. Available: https://www.mouser.co.uk/ProductDetail/Microchip-Technology/ATMEGA128-16AU?qs=aqrrBurbvGe184AQw0bwuQ%3D%3D. [Accessed 26 October 2023].

[148] Y. D. Li and E. T. Hsiao-Wekscler, “Gait Mode Recognition and Control for a Portable-Powered Ankle Foot Orthosis,” in *IEEE International Conference on Rehabilitation Robotics*, 2013.

[149] Texas Instruments, “TMS320F28335,” 2024. [Online]. Available: https://www.ti.com/product/TMS320F28335#order-quality.

[150] T. Vouga, R. Baud, J. Fasola, M. Bouri and H. Bleuler, “TWIICE - A Lightweight Lower-Limb Exoskeleton for Complete Paraplegics,” in *IEEE International Conference on Rehabilitaiton Robotics*, 2017.

[151] RS, “Beagleboard.org BeagleBone Black MCU Development Board BeagleBone Black,” [Online]. Available: https://uk.rs-online.com/web/p/microcontroller-development-tools/1252411. [Accessed 26 October 2023].

[152] R. Baud, A. Ortlieb, J. Oliver and M. Bouri, “HiBSO hip exoskeleton: Toward a wearable and autonomous design,” in *MESROB*, 2016.

[153] Farnell, “STMICROELECTRONICS STM32F722RET6,” [Online]. Available: https://uk.farnell.com/stmicroelectronics/stm32f722ret6/mcu-arm-cortex-m7-216mhz-lqfp/dp/2725144. [Accessed 02 June 2024].

[154] Adafruit, “Adafruit HUZZAH32 – ESP32 Feather Board,” [Online]. Available: https://www.adafruit.com/product/3405. [Accessed 26 October 2023].

[155] D. P. Allen, R. Little, J. Laube, J. Warren, W. Voit and R. D. Gregg, “Towards an ankle-foot orthosis powered by a dielectric elastomer actuator,” *Mechatronics,* vol. 76, no. 102551, pp. 1-13, 2021.

[156] PJRC, “Teensy 3.6 Development Boad,” 2024. [Online]. Available: https://www.pjrc.com/store/teensy36.html.

[157] Sparkfun, “Teensy 3.6,” [Online]. Available: https://www.sparkfun.com/products/14057. [Accessed 26 October 2023].

[158] T. Yan, M. Cempini, C. M. Oddo and N. Vitiello, “Review of assistive strategies in powered lower-limb orthoses and exoskeletons,” *Robotics and Autonomous Systems,* vol. 64, pp. 120-136, 2015.

[159] K. Anam and A. A. Al-Jumaily, “Active Exoskeleton Control Systems: State of the Art,” *Procedia Engineering,* vol. Volume 41, pp. 988-994, 2012.

[160] S. K. Banala, S. K. Agrawal and J. P. Scholz, “Active Leg Exoskeleton (ALEX) for Gait Rehabilitation of Motor-Impaired Patients,” in *IEEE 10th Conference on Rehabilitation Robotics*, Noordwijk, The Netherlands, 2007.

[161] S. Toyama and J. Yonetake, "Development of the Ultrasonic Motor-Powered Assisted Suit System," in *International Conference on Complex Medical Engineering*, Beijing, China, 2007.

[162] L. Luo, Y. Yuan and Z. Li, "Design and Development of a Wearable Lower Limb Exoskeleton Robot," in *4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, Toyonaka, Japan, 2019.

[163] C. A. Laubscher, R. J. Farris and J. T. Sawicki, "Design and Preliminary Evaluation of a Powered Pediatric Lower Limb Exoskeleton," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE*, Cleveland, Ohio, USA, 2017.

[164] J. E. Pratt, C. J. Morse, B. T. Krupp and H. S. Collins, "The RoboKnee: An Exoskeleton for Enhancing Strength and Endurance During Walking," in *International Conference on Robotics & Automation*, New Orleans, Louisiana, USA, 2004.

[165] J. F. Veneman, F. C. van der Helm and H. van der Kooij, "A Series Elastic- and Bowden-Cable-Based Actuation System for Use as Torque Actuator in Exoskeleton-Type Robots," *International Journal of Robotics Research,* vol. 25, no. 3, pp. 261-281, 2006.

[166] X. Jin, X. Cui and S. K. Agrawal, "Design of a Cable-driven Active Leg Exoskeleton (C-ALEX) and Gait Training Experiments with Human Subjects," in *IEEE International Conference on Robotics and Automation*, 2015.

[167] J. Chen, J. Hochstein, C. Kim, D. Damiano and T. Bulea, "Design Advancements toward a Pediatric Robotic Knee Exoskeleton for Overground Gait Rehabilitation," in *7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, Enschede, The Netherlands, 2018.

[168] R. Caldas, M. Mundt, W. Potthast and F. B. d. L. Neto, "A Systematic review of gait analysis methods based on inertial sensors and adaptive algorithms," *Gait & Posture,* vol. 57, pp. 204-210, 2017.

[169] H. Prasanth, M. Caban, U. Keller, G. Courtine, A. Ijspeert, H. Vallery and J. v. Zitzewitz, "Wearable Sensor-Based Real-Time Gait Detection: A Systematic Review," *Sensors,* vol. 21, no. 8, pp. 2727-2755, 2021.

[170] J. Y. Goulermas, A. H. Findlow, C. J. Nester, P. Liatsis, X.-J. Zeng, L. P. Kenney, P. Tresadern, S. B. Thies and D. Howard, "An Instance-Based Algorithm With Auxiliary Similiarity Information for the Estimation of Gait Kinematics From Wearable Sensors," *IEEE Transactions on Neural Networks,* vol. 19, no. 9, pp. 1574-1582, 2008.

[171] Y. Sun, Y. Tang, J. Zheng, D. Dong, X. Chen and L. Bai, "From sensing to control of lower limb exoskeleton: a systematic review," *Annual Reviews in Control,* vol. 53, pp. 83-96, 2022.

[172] L. Rose, M. C. Bazzocchi and G. Nejat, "End-to-End Deep Reinforcement Learning for Exoskeleton Control," in *International Conference on Systems, Man, and Cybernetics (SMC)*, Toronto, 2020.

[173] Z. Guo, C. Wang and C. Song, "A Real-Time stable-control gait switching strategy for lower-limb rehabilitation," *Plos One,* vol. 15, no. 8, p. 19, 2020.

[174] M. Karakish, M. A. Fouz and A. Elsawaf, "Gait Trajectory Prediction on an Embedded Microcontroller Using Deep Learning," *Sensors,* vol. 22, no. 8441, p. 22, 2022.

[175] V. Varma, R. Y. Rao, P. Vundavilli, M. Pandit and P. Budarapu, "A Machine Learning-Based Approach for the Design of Lower Limb Exoskeleton," *International Journal of Computational Methods,* vol. 9, no. 18, p. 22, 2022.

[176] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation,* vol. 9, no. 8, pp. 1735-1780, 1997.

[177] A. Vaswani, N. Shazeer, N. Parmer, J. Uszkoreit, L. Jones, A. N. Gomez and L. Kaiser, "Attention is All You Need," in *31st Conference on Neural Information Processing Systems*, Long Beach, 2017.

[178] J. Ren, A. Wang, H. Li, X. Yue and L. Meng, "A Transformer-Based Neural Network for Gait Prediction in Lower Limb Exoskeleton Robots Using Plantar Force," *Sensors,* vol. 23, no. 6547, p. 17, 2023.

[179] S. Hosseini, N. N. Joojili and M. Ahmadi, "LLMT: A Transformer-Based Multi-Modal Lower Limb Human Motion Prediction Model for Assistive Robotics Applications," *IEEE Access,* vol. 12, p. 12, 2024.

[180] T. Lee, I. Kim and S.-H. Lee, "Estimation of the Continuous Walking Angle of Knee and Angle (Talocrural Joint, Subtalar Joint) of a Lower-Limb Exoskeleton Robot Using a Neural Network," *Sensors,* vol. 21, no. 8, p. 2807, 2021.

[181] Y. Huang, Z. He, Y. Liu, R. Yang, X. Zhang, G. Cheng, J. Yi, J. P. Ferreira and T. Liu, "Real-Time Intended Knee Joint Motion Prediction," *IEEE Sensors Journal,* vol. 19, no. 23, pp. 1558-1748, 2019.

[182] ST, "STM32F407G-DISC1," [Online]. Available: https://estore.st.com/en/stm32f407g-disc1-cpn.html. [Accessed 23 August 2023].

[183] Z.-Q. Ling, G.-Z. Cao, Y.-P. Zhang, H.-R. Cheng, B.-B. He and S.-B. Cao, "Real-time Knee Joint Angle Estimation Based on Surface Electromyograph and Back Propagation Neural Network," in *18th International Conference on Ubiquitous Robots (UR)*, Gangneung-si, Gangwon-do, Korea, 2021.

[184] T. T. Alemayoh, J. H. Lee and S. Okamoto, "Leg-Joint Angle Estimation from a Single Intertial Sensor Attached to Various Lower-Body Links during Walking Motion," *Applied Sciences,* vol. 13, no. 4794, pp. 1-17, 2023.

[185] L. Marchal-Crespo and D. J. Reinkensmeyer, "Review of Control Strategies for robotic movement training after neurologic injury," *Journal of NeuroEngineering and Rehabilitation,* vol. 6, no. 20, pp. 1-15, 2009.

[186] R. Baud, A. R. Manzoori, A. Ijspeert and M. Bouri, "Review of control strategies for lower-limb exoskeleons to assist gait," *Journal of Neuroengineering and Rehabilitation,* vol. 18, no. 119, pp. 1-34, 2021.

[187] H. Choi, Y. J. Park, K. Seo, J. Lee, S.-e. Lee and Y. Shim, "A Multifunctional Ankle Exoskeleton for Mobility Enhancement of Gait-Impared Individuals and Seniors," *IEEE Robotics and Automation Letters,* vol. 3, no. 1, pp. 411-418, 2018.

[188] Y. Heo, H.-J. Choi, S.-J. Hwang, J.-W. Lee, C.-Y. Kwon, H.-S. Cho and G.-S. Kim, "Development of a Knee Actuated Exoskeletal Gait Orthossis for Paraplegic Patients with Incomplete Spinal Cord Injury: A Single Case Study," *Applied Sciences,* vol. 11, no. 58, pp. 1-13, 2020.

[189] Z. F. Lerner, D. L. Damiano, H.-S. Park, A. J. Gravunder and T. C. Bulea, "A Robotic Exoskeleton for Treatment of Crouch Gait in Children with Cerebral Palsy: Design and Initial Application," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 25, no. 6, pp. 650-658, 2017.

[190] P. Tsangaridis, D. Obwegeser, S. Maggioni, R. Riener and L. Marchal-Crespo, "Visual and Haptic Error Modulating Controllers for Robotic Gait Training," in *7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*, Enshede, The Netherlands, 2018.

[191] J. L. Emken, R. Benitez and D. J. Reinkensmeyer, "Human-robot cooperative movement training: Learning a novel sensory motor transformation during walkin with robotic assistance-as-needed," *Journal of NeuroEngineering and Rehabilitation,* vol. 4, no. 8, pp. 1-16, 2007.

[192] C. Hollnagel, M. Brugger, H. Vallery, P. Wolf and V. Dietz, "Brain Activity during stepping: a novel MRI-compatible device," *Journal of Neuroscience Methods,* vol. 201, pp. 124-130, 2011.

[193] K. P. Michimizos and H. I. Krebs, "Assist-as-needed in Lower Extremity Robotic Therapy for Children with Cerebral Palsy," in *IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2012.

[194] H. Roberts, A. Shierk, N. J. Clegg, D. Baldwin, L. Smith, P. Yeatts and M. R. Delgado, "Constrain Induced Movement Therapy CAmp for Children with Hemiplegic Cerebral Palsy Augmented by Use of an Exoskeleton to Play Games in Virtual Reality," *Physical & Occupational Therapy in Pediatrics,* vol. 41, no. 2, pp. 150-165, 2021.

[195] J. F. Veneman, R. Kruidhof, E. E. G. Hekman, R. Ekkelenkamp, E. H. F. Van Asseldonk and H. van der Kooij, "Design and Evaluation of the LOPES Exoskeleton Robot for Interactive Gait Rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 15, no. 3, pp. 379-386, 2007.

[196] M. A. Alouane, W. Huo, H. Rifai, Y. Amirat and S. Mohammed, "Hybrid FES-Exoskeleton Controller to Assist Sit-To-Stand Movement," *IFAC-PapersOnLine,* vol. 51, no. 34, pp. 296-301, 2019.

[197] B. Jardim, A. A. Siqueira and L. M. Sampaio do Amaral, "Development of Series Elastic Actuators for Impedance Control of an Active Ankle Foot Orthosis," in *20th International Congress of Mechanical Engineering*, Gramado, Brazil, 2009.

[198] S. Glowinski and T. Krzyzynski, "An inverse kinematic algorithm for the human leg," *Journal of Theoretical and Applied Mechanics,* vol. 54, no. 1, pp. 53-61, 2016.

[199] Q. Chen, H. Cheng, C. Yue, R. Huang and H. Guo, "Dynamic Balance Gait for Walking Assistance Exoskeleton," *Applied Bionics and Biomechanics,* vol. 2018, pp. 1-10, 22 January 2018.

[200] T. Stockel, R. Jacksteit, M. Behrens, R. Skripitz, R. Bader and A. Mau-Moeller, "The Mental Representation of the human gait in young and older adults," *Frontiers in Psychology,* vol. 6, no. 943, pp. 1-11, 2015.

[201] M. W. Whittle, "Clinical Gait Analysis: A Review," *Human Movement Science,* vol. 15, no. 3, pp. 369-387, 1996.

[202] A. Azahari, W. Siswanto, M. Ngali, S. Salleh and E. M. Yusup, "Dynamic Simulation and Analysis of Human Walking Mechanism," in *International Conference on Materials Science and Engineering*, Johor, Malaysia, 2017.

[203] Stanford University, "Stanford Medicine 25 Gaits," 2014 March 17. [Online]. Available: https://stanfordmedicine25.stanford.edu/the25/gait.html#main_panel_builder_0_panel_0_panel_builder_1482875556_panel_0_accordion_content_1. [Accessed 2023 September 19].

[204] M. N. Kuperminc and R. D. Stevenson, "Growth and Nutrition Disorders in Children with Cerebral Palsy," *Dev Disabil Res,* vol. 14, no. 2, pp. 137-146, 2008.

[205] J. H. Ha, I. H. Choi, C. Y. Chung, T.-J. Cho, S. T. Jung, H.-S. Lee, S.-S. Park, H. Y. Lee, C.-W. Oh and I. O. Kim, "Distribution of Lengths of the normal femur and tibia in Korean children from three to sixteen years of age," *Journal of Korean Medical Science,* vol. 18, no. 5, pp. 715-721, 2003.

[206] M.O.T.I.O.N Interreg, "University of Kent - MOTION," University of Kent, 2019. [Online]. Available: https://www.motion-interreg.eu/university-of-kent.html. [Accessed 7 Feburary 2024].

[207] C. D. Fryar, M. D. Carroll, Q. Gu, J. Afful and C. L. Ogden, "Anthropometric reference data for children and adults: United States, 2003-2006," National Center for Health Statistics, Hyattsville, Maryland, 2008.

[208] R. M. Malina, P. V. Hamill and S. Lemeshow, "Selected Body Measurements of Children 6-11 Years," National Center for Health Statistics, Rockville, Md, 1973.

[209] A. M. Fredriks, S. v. Buuren, W. J. M. v. Heel, R. H. M. Dijkman-Neerincx, S. P. Verloove-Vanhorick and J. M. Wit, "Nationwide age references for sitting height, leg length, and sitting height/height ratio, and their diagnostic value for disproportionate growth disorders," *Arch Dis Child,* vol. 90, no. 8, pp. 807-812, 2005.

[210] RCPCH, "Boys UK Growth Chart 2-18 Years," RCPCH, 2012.

[211] M. Adolphe and J. Clerval, "Center of Mass of Human Body Segments," *Mechanics and Mechanical Engineering,* vol. 21, no. 3, 2017.

[212] J. P. Pollard, W. L. Porter and M. S. Redfern, "Forces and Moments on the Knee During Kneeling and Squatting," *Journal of Applied Biomechanics,* vol. 27, pp. 233-241, 2011.

[213] M. Sarajchi and K. Sirlantzis, "Design and Control of a Single-Leg Exoskeleton with Gravity Compensation for Children with Unilateral Cerebral Palsy," *Sensors,* vol. 23, no. 13, p. 6103, 2023.

[214] R. J. Schmitz, D. Harrison, H.-M. Wang and S. Shultz, “Sagittal-Plane Knee Moment During Gait and Knee Cartilage Thickness,” *Journal of Athletic Training,* vol. 52, no. 6, pp. 560-566, 2017.

[215] K. W. Hollander, T. G. Sugar and D. E. Herring, “Adjustable Robotic Tendon using a 'Jack Spring',” in *9th International Conference on Rehabilitation Robotics*, Chicago, IL, USA, 2005.

[216] K. Kamali, A. A. Akbari and A. Akbarzadeh, “Trajectory generation and control of a knee exoskeleton based on dynamic movement primitives for sit-to-stand assistance,” *Advanced Robotics,* vol. 30, no. 13, pp. 846-860, 2016.

[217] A. Abdullah, M. A. bin Mazelan, F. H. Ahmad, P. Krishnan and S. Yaacob, “Design and Physical Modelling Series Actuator for Ankle-Foot Orthosis,” *International Journal of Engineering and Advanced Technology,* vol. 9, no. 5, pp. 210-215, 2020.

[218] The Pi Hut, “42MM High Torque Hybrid Stepping Motor,” [Online]. Available: https://cdn.shopify.com/s/files/1/0176/3274/files/102970ds.jpg?v=1676910738. [Accessed 7 December 2023].

[219] RS, “RS PRO 250Ω Rotary Potentiometer 1-Gang Panel Mount,” [Online]. Available: https://uk.rs-online.com/web/p/potentiometers/8427080. [Accessed 17 November 2023].

[220] Adafruit, “Adafruit 9-DOF Accel/Mag/Gyro+Temp Breakout Board - LSM9DS1,” [Online]. Available: https://www.adafruit.com/product/3387. [Accessed 17 November 2023].

[221] RS, “I.E.E. Strain Gauge 15.25mm, >1MΩ -30°C +170°C,” [Online]. Available: https://www.rs-online.id/p/force-sensor-2/#:~:text=The%20gauge%20sticks%20to%20an,medical%20devices%20and%20automotive%20products.. [Accessed 17 November 2023].

[222] Hobby Components, “HOBBY COMPONENTS ARDUINO COMPATIBLE R3 MEGA,” [Online]. Available: https://hobbycomponents.com/development-boards/557-hobby-components-arduino-compatible-r3-mega. [Accessed 2023 November 17].

[223] Motech Motor Co., “MT-1704HS168A,” [Online]. Available: http://motechmotor.com/productDetail-0104-32.html. [Accessed 7 December 2023].

[224] Bourns, “EAW - Absolute Contacting Encoder (ACE™),” 31 March 2015. [Online]. Available: https://www.bourns.com/pdfs/ACE.pdf. [Accessed 7 December 2023].

[225] Robotkits, “RHINO MOTION CONTROLS RMCS-220X,” 2012. [Online]. Available: https://robokits.download/downloads/RMCS220x_DCServo_Driver.pdf. [Accessed 14 December 2023].

[226] Espressif, “FreeRTOS Overview,” 2023. [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos.html#overview. [Accessed 19 December 2023].

[227] RS, “Datasheet: RS Pro P25 Series Wire-Wound Potentiometer with a 6mm Dia. Shaft, 25Ω, ±10%, 1W, ±50ppm/°C, Panel Mount,” [Online]. Available: https://docs.rs-online.com/dfda/0900766b81584587.pdf. [Accessed 19 December 2023].

[228] Georgia Tech, “Epic Exoskeleton & Prosthetic Intelligent Controls,” 27 April 2022. [Online]. Available: https://www.epic.gatech.edu/opensource-biomechanics-camargo-et-al/. [Accessed 18 March 2024].

[229] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network,” *Physica D,* vol. 404, p. 28, 2020.

[230] I. L. Orasan, C. Seiculescu and C. D. Caleanu, “A Brief Review of Deep Neural Network Implementations for ARM Cortex-M Processor,” *Electronics,* vol. 11, no. 2545, p. 21, 2022.

[231] R. Kolaghassi, M. K. Al-Hares, G. Marcelli and K. Sirzlantzis, “Performance of Deep Learning Models in Forecasting Gait Trajectories of Children with Neurological Disorders,” *Sensors,* vol. 22, no. 8, p. 18, 2022.

[232] K. Hori, Y. Mao, Y. Ono, H. Ora, Y. Hirobe, K. Sawada, A. Inaba, S. Orimo and Y. Miyake, “Inertial Measurement Unit-Based Estimation of Foot Trajectory for Clinical Gait Analysis,” *Frontiers in Physiology,* vol. 10, no. 1530, p. 12, 2020.

[233] F. Fallahtafti, S. R. Wurdeman and J. M. Yentes, “Sampling Rate influences the regularity analysis of temporal domain measures of walking more than spatial domain measures,” *Gait Posture,* vol. 88, pp. 216-220, 2022.

[234] J. Sung, S. Han, H. Park, H.-M. Cho, S. Hwang, J. W. Park and I. Youn, “Prediction of Lower Extremity Multi-Joint Angles during Overground Walking by Using a Single IMU with a Low Frequency Based on an LSTM Recurrent Neural Network,” *Sensors,* vol. 22, no. 53, p. 14, 2021.

[235] J.-S. Tan, S. Tippaya, T. Binnie, P. Davey, K. Napier, J. Caneiro, P. Kent, A. Smith, P. O'Sullivan and A. Campbell, “Predicting Knee Joint Kinematics from Wearable Sensor Data in People with Knee Oseoarthritis and Clinical Considerations for Future Machine Learning Models,” *Sensors,* vol. 22, no. 446, p. 16, 2022.

[236] M. Mundt, W. Thomsen, T. Witter, A. Koeppe, S. David, F. Bamer, W. Potthast and B. Markert, “Prediction of lower limb joint angles and moments during gait using artificial neural networks,” *Medical & Biological Engineering & Computing,* vol. 58, pp. 211-225, 2020.

[237] TensorFlow, “tflite-micro,” 08 May 2023. [Online]. Available: https://github.com/tensorflow/tflite-micro. [Accessed 09 May 2023].

[238] Espressif, “tflite-micro-esp-examples,” 07 May 2023. [Online]. Available: https://github.com/espressif/tflite-micro-esp-examples. [Accessed 09 May 2023].

[239] TensorFlow, “tensorflow/tensorflow/lite/python/util.py,” [Online]. Available: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/python/util.py. [Accessed 4 January 2024].

[240] L. Roeder, “Netron,” [Online]. Available: https://netron.app/. [Accessed 4 January 2024].

[241] V. Dattu, T. Rezucha, I. Grothotkov, S. Tipnis, K. Sovani, B. T. Sebestik and L.-P. , “esp-tflite-micro,” [Online]. Available: https://github.com/espressif/esp-tflite-micro. [Accessed 5 January 2024].

[242] Asus, “ROG Zephyrus M16 (2022),” [Online]. Available: https://uk.store.asus.com/rog-zephyrus-m16-2022-203681799-90nr08r1-m000m0.html. [Accessed 10 August 2023].

[243] Amazon, “Teensy 4.1 (Without Pins),” [Online]. Available: https://www.amazon.co.uk/Teensy-4-1-Without-Pins/dp/B088D3FWR7. [Accessed 10 August 2023].

[244] T. Sandmann and M. S, “freertos-teensy,” [Online]. Available: https://github.com/tsandmann/freertos-teensy. [Accessed 10 Januray 2024].

[245] T. Rylaarsdamn, “TensorFlow Lite for Microcontrollers,” [Online]. Available: https://github.com/trylaarsdam/pio-tflite. [Accessed 10 January 2024].

[246] M. Kennard, H. Kadone, Y. Shimizu and K. Suzuki, “Passive Exoskeleton with Gait-Based Knee Joint Support for Individuals with Cerebral Palsy,” *Sensors,* vol. 22, no. 8935, pp. 1-17, 2022.

[247] M. A. H. Mohd Adib, S. Y. Han, P. R. Ramani, L. J. You, L. M. Yan, I. M. Sahat and N. H. M. Hasni, “Restoration of Kids Leg Function Using Exoskeleton Robotic Leg (ExRoLEG) Device,” in *Proceedings of the 10th National Technical Seminar on Underwater System Technology* , Singapore, 2019.

[248] A. C. Boynton, M. Mungiole and H. P. Crowell III, “Exoskeleton Power and Torque Requirements Based on Human Biomechanics,” Army Research Laboratory, Aberdeen, 2002.

[249] A. T. Asbeck, R. J. Dyer, A. F. Larusson and C. J. Walsh, “Biologically-inspired Soft Exosuit,” in *IEEE International Conference on Rehabilitation Robotics*, 2013.

[250] PowerTech, “Lithium-Ion Battery 12V – 7.5Ah – 96Wh – PowerBrick,” [Online]. Available: https://www.powertechsystems.eu/home/products/12v-lithium-battery-pack-powerbrick/7-5ah-12v-lithium-ion-battery-pack-powerbrick/. [Accessed 19 April 2024].

[251] R. Kolaghassi, “Deep learning for Gait Prediction: An Application to Exoskeletons for Children with Neurological Disorders,” University of Kent, Canterbury, 2023.

[252] L. J. Qingsong, W. Meng, Q. Liu and S. Q. Xie, “Individualized Gait Trajectory Prediction Based on Fusion LSTM Networks for Robotic Rehabilitation Training,” in *IEEE International Conference on Advanced Intelligent Mechatronics*, Delft, Netherlands, 2021.

[253] C. Zhu, Q. Liu, Q. Ai and S. Q. Xie, “An Attention-based CNN-LSTM Model with Limb Synergy for Gait Trajectory Prediction,” in *International Conference on Advanced Intelligent Mechatronics*, Delft, Netherlands, 2021.

[254] Z.-Q. Ling, G.-Z. Cao, Y.-P. Zhang, H.-R. Cheng, B.-B. He and S.-B. Cao, “Real-Time Knee Joint Angle Estimation Based on Surface Electromyograph and Back Propagation Neural Network,” in *18th International Conference on Ubiquitous Robots*, Gangwon-do, Korea, 2-21.

[255] Y. Huang, Z. He, Y. Liu, R. Yang, X. Zhang, G. Cheng, J. Yi, J. P. Ferreira and T. Liu, “Real-Time Intended Knee Joint Motion Prediction by Deep-Recurrent Neural Networks,” *IEEE Sensors Journal,* vol. 19, no. 23, pp. 1558-1748, 2019.

[256] J. Sotillo, “Staff and Student News,” 22 March 2022. [Online]. Available: https://blogs.kent.ac.uk/staff-student-news/2022/03/22/it-services-outage-update-and-next-steps/. [Accessed 13 October 2023].

[257] eloquentarduino, “...arm_convolve_1x1_HWC_q7_fast_nonsquare.c.d: No such file or directory,” 25 August 2022. [Online]. Available: https://github.com/eloquentarduino/EloquentTinyML/issues/46. [Accessed 7 Feburary 2024].

[258] Tensor-Micro, “Type INT32 (2) not supported for LSTM Models,” 14 March 2023. [Online]. Available: https://github.com/tensorflow/tflite-micro/issues/1825. [Accessed 7 Feburary 2024].

[259] tflite-micro, “Commit - Add MNIST LSTM model evaluation,” 8 September 2022. [Online]. Available: https://github.com/tensorflow/tflite-micro/commit/945d197610ac4e766956b55d2f11b9b195721000. [Accessed 7 Feburary 2024].

[260] S. Luo, G. Androwis, S. Adamovich, E. Nunez and X. Zhou, “Robust walking control of a lower limb rehabilitation exoskeleton coupled with a musculoskeletal model via deep reinforcement learning,” *Journal of Neuroengineering and Rehabilitation,* vol. 20, no. 34, pp. 1-19, 2023.

[261] T. Slucock, “A Low-Cost Semi-Active Orthosis and Neural Network Based Control System for Alleviating Gait Degradation: Addendums,” 28 July 2024. [Online]. Available: https://drive.google.com/drive/u/1/folders/1fppk0rHMnm63a2OW4uGdAwNXb-EJlXSm.

[262] Eloquent Arduino, “TinyMLgen for Python,” 2023. [Online]. Available: https://eloquentarduino.com/tinymlgen/. [Accessed 18 December 2023].

[263] Eloquent Arduino, “EloquentTinyML library for Arduino,” 2023. [Online]. Available: https://eloquentarduino.com/eloquent-tinyml/. [Accessed 18 December 2023].

Addendums

The Following citation links to an addendum of additional content produced as part of this thesis: [261]. (https://drive.google.com/drive/u/1/folders/1fppk0rHMnm63a2OW4uGdAwNXb-EJlXSm).

The content within consists of:

- Key Programs for generating the LSTM Prediction System, calculating Ideal Actuator position, and reading data from the Microcontroller, as well as relevant Sample data used by these programs.
- CREAG Application and Ethics Approval for Data Collection experiments.
- Additional Resources regarding Literature review component spreadsheet, Actuator Stickout calculation spreadsheets, and exoskeleton MATLAB Simulink Simulations.

Contents Breakdown

Ethics and Experiment Approval contains files relating to the approval process for gathering walking test data from 12 participants as part of acquiring training data for the final Recurrent Neural Network Prediction System implementation. It contains the Consent forms provided to participants, the Application, and Ethics Checklists, and Approval Confirmation.

Programs and Sample Data contains Teensy, Python, and Matlab Programs developed for the exoskeleton.

- The Teesny folder consists of a full VSCode Source Project used in the final implementation. By default the *src* file contains the Control System run upon the exoskeleton with necessary secondary files for the AI Model and optional Sample data. Additional programs within *programs* may be used to test the motor or run the exoskeleton purely to acquire data.
- The Python folder contains a number programs used in the development of the control system. Additionally, within *Content/WalkTest* is all data collected for the project by the exoskeleton, both used and unused. Experimental Training Data collected from the 12 Participants is located in *Content/WalkTest/newPeople*.
- The Matlab folder contains the Simulink Simulation Files to run the model seen in Figure 39.

Other Resources contains Datasheets for used components, other Academic Literature produced as part of this thesis, and several other files developed that provided utility in the Thesis. Namely *ExoskeletonListFinal.xlsx* which contains all collected data for the Literature Review, and *LegLengths.xlsx* and *StickoutComparison.xlsx* which are used to calculate force profiles of different exoskeleton stickouts.

For Additional Information and Questions, A Reader may contact the Author via *TomSlucock@gmail.com*.